Experimental Evaluation in Computer Science: A Quantitative Study

Paul Lukowicz, Ernst A. Heinz, Lutz Prechelt, Walter F. Tichy
email: (lukowicz | heinze | prechelt | tichy) @ira.uka.de
Department of Informatics
University of Karlsruhe, Germany

Technical Report 17/94
To appear in "Journal of Systems and Software", January(?) 1995

August 25, 1994

"The fundamental principle of science, the definition almost, is this: the sole test of the validity of any idea is experiment." — Richard P. Feynman

"Beware of bugs in the above code; I have only proved it correct, not tried it." — Donald E. Knuth

Abstract

A survey of over 400 recent research articles suggests that computer scientists publish relatively few papers with experimentally validated results.

The survey includes complete volumes of several refereed computer science journals, a conference, and 50 titles drawn at random from all articles published by ACM in 1993. The journals Optical Engineering (OE) and Neural Computation (NC) were used for comparison.

Of the papers in the random sample that would require experimental validation, 40% have none at all. In journals related to software engineering, this fraction is over 50%. In comparison, the fraction of papers lacking quantitative evaluation in OE and NC is only 15% and 12%, respectively.

Conversely, the fraction of papers that devote one fifth or more of their space to experimental validation is almost 70% for OE and NC, while it is a mere 30% for the CS random sample and 20% for software engineering.

The low ratio of validated results appears to be a serious weakness in computer science research. This weakness should be rectified for the long-term health of the field.

Contents

1 Introduction

A large part of computer science research consists of proposing new designs — systems, algorithms, and models. Such designs must be judged by whether they increase our knowledge about what are useful and cost-effective problem solutions. In most cases, objective judgement can only be achieved on the basis of reproducible experiments.

This study was motivated by our subjective impression that experimental evaluation is often neglected in CS research. We feared that the quality of CS research might be inferior to other disciplines, in particular the natural sciences, the engineering sciences, and applied mathematics. To test whether this impression was merely scientific pessimism, we performed an empirical study involving both CS and non-CS publications. This article presents the design and the results of this study.

We classified research articles in peer-reviewed journals and conferences. The classification divides the set of articles into theoretical work, design and modeling work, empirical work, hypothesis testing, and other (for details see Section 3). Ideally, theoretical work should be well-balanced with empirical work, and design and modeling work should contain experimental evaluation. Assessing the quantity and quality of such evaluations is the main purpose of this study. We used the fraction of space each article devotes to evaluation as an indicator of quality. Section 3.3 explains the rationale for this approach.

We sampled a broad set of recent CS publications: the complete volumes 9–11 (1991-93) of *ACM Transactions on Computer Systems (TOCS)*, the complete volumes 14–15 (1992-93) and numbers 1 and 2 of volume 16 (1994) of *ACM Transactions on Programming Languages and Systems (TOPLAS)*, the complete volume 19 (1993) of *IEEE Transactions on Software Engineering (TSE)*, and all papers from the *Proceedings of the 1993 SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. Moreover, we drew a random sample of 74 titles from the set of all works published by ACM in 1993, using the INSPEC database [9]. From this sample, we excluded 24 articles that were either inappropriate (because they were not peer-reviewed research papers) or not available in our library. See Appendix A for details. The resulting set contains 50 papers, of which 30 are refereed conference contributions. This sample represents a fair cross-section of peer-reviewed research in CS.

For comparison, we reviewed publications from two other fields: volume 5 (1993) of *Neural Computation (NC)*, and numbers 1 and 3 of volume 33 (1994) of *Optical Engineering (OE)*. *Neural Computation*, published by MIT Press, is an interdisciplinary journal in the field of neuro-science. It contains articles about artificial neural networks, neural modeling, and the theory of neural computation; the contributors come from many disciplines, e.g. biology, computer science, mathematics, medicine, physics, and psychology. We chose NC because it might share characteristics with CS due to its youth and partial overlap with CS. *Optical Engineering*, published by the International Society for Optical Engineering, is a journal devoted to applied optics, opto-mechanics, opto-electronics, image processing research, and related fields. Most contributors come from physics, electrical engineering, optics, astronomy, space science, and mechanical engineering. We chose OE because optics, similar to CS, has many immediate applications, but in contrast has a longer history.

The remaining sections review related work, introduce the methodology of our study, present the observations, and discuss accuracy.

2 Related Work

The literature contains only a few articles about the nature of experimental CS, and we are not aware of any systematic attempt to assess research in this area.

Early surveys [6, 12] published in 1979 describe the state of experimental CS with respect to the poor support it received. Today, the situation is perceived as largely unchanged: In 1994, the authors of reference [13] conclude that experimental CS is still underfunded, and that researchers in the area often face difficult career paths at universities.

In 1980, Denning [3] gives a definition of what experimental CS is: *"Measuring an apparatus in order to test a hypothesis."* Denning notes that standards in the natural sciences describe how to carry out such work properly, but that CS is rarely performing well by these standards. He concludes that *"If we do not live up to the traditional standards of science, there will come a time when no one takes us seriously"*. In later articles, Denning cites the field of performance evaluation as a positive example of experimental CS research [4, 5].

Several articles describe the role of experimental research in branches of CS, e.g. machine learning [11], algorithms [8], or software engineering [7].

The latter article is quite critical of software engineering research and states: *"There are far too few examples of moderately effective research."* Baldwin and Koomen [2] discuss practicing experimental computer science during CS education.

In 1990, Iyer [10] writes that *"experimental CS is a relatively new, yet fast developing area"* and finds: *"It is indeed encouraging to see that there is substantial research going on in this important area."* Four years later, however, Hooker [8] notes that experimental research is dramatically underdeveloped in algorithms research. He states that most experimental efforts *"fall short of science on several levels"* and continues *"It is symptomatic of the situation in OR and computer science one cannot publish reports that an algorithm does* not *perform well in computational tests."* In a similar spirit, Bailey [1] presents a list of common experimental flaws in the field of computer performance evaluation, suggesting that many of these errors are committed intentionally — to *"fool the masses"*.

Obviously, the views on the quality of experimental CS are quite contradictory; yet we could not find any attempt in the literature to objectively assess the quantity or quality of experimental work in CS.

3 Methodology

The initial step was to define reasonable classification criteria. Each author then performed his classification tasks independently of the others, in order to minimize possible distortions caused by direct or indirect mutual influence.

All four authors classified the ACM papers drawn at random; groups of two did so for PLDI, TOPLAS, and TSE, whereas only single persons handled NC, OE, and TOCS. A degree of uniformity was achieved by having the same person classify nearly all samples (except NC). The following table shows who actually classified which sample.

	Ernst	Lutz	Paul	Walter
NC		X		
OE			X	
TOCS			X	
Random	X	X	X	X
PLDI	X		X	
TOPLAS	X		X	
TSE			X	X

3.1 Classification

Our classification scheme distinguishes five major categories: formal theory, design & modeling, empirical study, hypothesis testing, and other. These categories suffice for our purpose. Moreover, they appear general enough to be applicable to other disciplines as well.

Publications of the design & modeling category require reproducible experiments for validation of claims. Without validation, they fail to establish useful and credible results. Our classification captures the importance of experimental validation by further subdividing the design & modeling category according to the space devoted to the description of experimental evaluation.

3.2 Major Categories

In order to achieve an acceptable degree of objectivity, we applied the classification criteria to the main claims and contributions of the surveyed papers. Main claims and contributions are usually clearly stated in the abstracts, introductions, or conclusions of articles. The classification criteria are as follow.

Formal Theory. This category consists of articles whose main contributions are formally tractable propositions, e.g. lemmata and theorems, and their proofs.

Design & Modeling. The main contributions of articles in this category are systems, techniques, or models, whose claimed properties cannot be proven formally. Examples include software tools, performance prediction models, and complex hard- and software systems of all kinds.

Empirical Work. Articles in this category collect, analyze, and interpret observations about known designs, systems, or models, or about abstract theories or subjects (as this paper does). The emphasis is on evaluation, not on new designs or models.

Hypothesis Testing. Articles in this category define hypotheses and describe experiments to test them.

Other. This category includes articles that do not fit any of the four categories above, e.g. surveys.

3.3 Subclasses of Design & Modeling

Work in design & modeling is further classified according to the experimental evaluation that appears in it. We used a simple and objectively quantifiable criterion, namely the physical space devoted to describing experimental setups, presenting observations, and interpreting results. We partitioned the papers into five subclasses of 0%, (0% – 10%], (10% – 20%], (20% – 50%], and over 50% of space per article devoted to such material.

Although space is a purely quantitative measure, we believe that it is also indicative of quality, based on the following two assumptions.

1. The amount of space devoted to the description of experimental evaluation and the importance attached to it by authors and reviewers are closely correlated.

2. The importance attached to, and the quality of experimental evaluation are closely correlated.

Both assumptions are plausible, but we have not validated them. Together they suggest a correlation between the quality of experimental evaluation and the amount of space devoted thereto.

Although these assumptions need not always apply, our collective impressions during the study support a positive correlation. Where we felt confident to judge quality, we rarely found mismatches. Intuitively, it is difficult to write something meaningful about a difficult experimental setup and the interpretation of results in, say, three pages of a twenty-page paper. Conversely, a long description of an uninteresting experiment is likely to be rejected by reviewers. We attribute the positive correlation between quality and space to a functioning process of peer review.

In any event, one of our main observations concerns papers that have no experimental validation at all. For an absent evaluation, a correlation between quality and space is moot.

3.4 Assessing Experimental Evaluation

Recall that design & modeling papers state claims that cannot be proven by logical reasoning, but require experimental evaluation. Hence, we looked for designs, systems, algorithms, etc. executed, techniques and methods applied, models validated. This material is generally easy to spot: It manifests itself in tables, graphs, or section headings and is often summarized in abstracts.

We did not attempt to assess quality of experimental work in any way. But we did try to include only what appeared to be true experimental work. The nature of true experiments is characterized by testing claims in an objective and repeatable manner. For example, benchmark measurements are acceptable, because they are repeatable and their outcomes are not completely determined in advance. The only subjective part is the composition of the benchmark.

We excluded demonstrations of systems, because in essence they are predetermined functionality shows, not objective measurements. Their outcomes are completely determined in advance and often not measurable. Examples that appear in papers, even if extensive, are also excluded, because they merely illustrate concepts.

It proved difficult to assess whether simulation setups constituted acceptable experimental work. After some initial experience, we formulated the following guideline: Simulation is regarded as true experimentation if and only if

1. it is employed to generate input data for other true experiments, or

2. it uses data traces of real world events as inputs and is conducted in realistic setups, e.g. in generally accepted simulation environments.

4 Observations

This section summarizes the overall results of our study according to the two-level classification in the previous section. The complete classification data can be found in Appendix B.

Rather than averaging the class sizes, we only took one persons (Paul's) classification data to compile the results in this section. The other classifiers' data is used for bounding the error in Section 5. This approach has the advantage that the classification criteria are applied uniformly to all samples. The only exception is NC, which was classified by a different person.

	NC	OE	TOCS	Random	PLDI	TOPLAS	TSE
Theory	14	6	3	6	2	19	18
Design	49	46	31	35	25	26	47
Empirical	8	12	3	1	2	4	15
Hypothesis	0	3	0	1	0	0	0
Emp. + Hyp.	8	15	3	2	2	4	15
Other	1	8	1	7	0	3	7
Total	72	75	38	50	29	52	87

Figure 1: The absolute cardinalities of major categories (total number of all classified articles = 403).

4.1 Major Categories

Figure 1 presents class sizes for the major categories per sample, whereas Figure 2 depicts the classes as percentages of the total number of articles in each sample. Three important observations directly follow from this data.

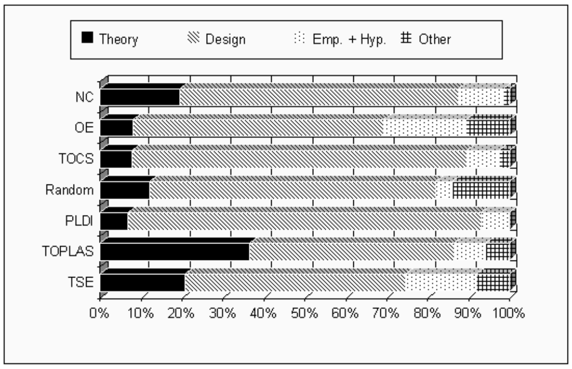

Figure 2: The relative cardinalities of major categories (sum of all articles per sample = 100%).

1. *The majority of articles in all samples consists of design & modeling work.*
2. *With the exception of TSE, the CS samples have a significantly lower percentage of empirical work than OE and NC.*
3. *Hypothesis testing is extremely rare in all samples (4 articles out of a total of 403).*

Because hypothesis testing is so rare, we combined it with empirical work in Figure 2.

4.2 Subclasses of Design & Modeling

The subclass cardinalities for experimental evaluation in design & modeling work appear in Figure 3; percentages relative to the total number of design & modeling articles are shown in Figure 4.

	NC	OE	TOCS	Random	PLDI	TOPLAS	TSE
0%	6	7	12	15	9	12	26
(0%..10%]	3	6	2	3	6	7	9
(10%..20%]	6	2	10	6	0	3	2
(20%..50%]	28	28	7	11	8	4	7
> 50%	6	3	0	0	2	0	3
Total	**49**	**46**	**31**	**35**	**25**	**26**	**47**
>20% / Total	**69%**	**67%**	**23%**	**31%**	**40%**	**15%**	**21%**
0% / Total	**12%**	**15%**	**39%**	**43%**	**36%**	**46%**	**55%**

Figure 3: The absolute cardinalities of all design and modeling subclasses plus the relative cardinality of the subclass 0% and the cumulative relative cardinality of the subclasses >20%.

The following observations are obvious:

1. *There is a disproportionally high percentage of design & modeling work without any experimental evaluation in the CS samples as compared to NC and OE (43% vs. 14%).*

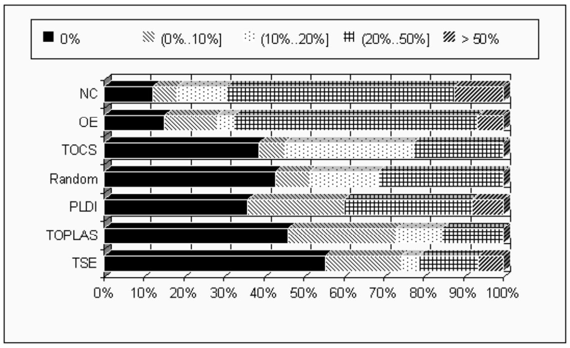

Figure 4: The relative cardinalities of the design and modeling subclasses (sum of d&m articles per sample = 100%).

2. *In NC and OE there are significantly more design & modeling articles devoting over 20% of their space to experimental evaluation than in the CS samples. (67% in OE vs. 31% in the random sample).*
3. *Samples related to software engineering (TSE and TOPLAS) are worse than the random CS sample.*

To underscore these observations, Figure 5 shows the fraction of design & modeling articles that have no experimental evaluation at all, and Figure 6 shows the fraction of design articles with more than 20% of their space devoted to experimental evaluation.

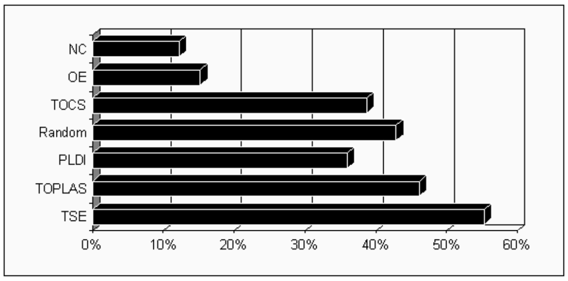

Figure 5: The percentage of design & modeling articles without any experimental evaluation.

5 Accuracy of Study

Any experiment dealing with humans involves a considerable amount of ambiguity. Unlike physically measurable quantities, human judgement is often subjective. In a strict empirical study, statistical techniques should be used to determine and minimize the margin of human error. This implies large numbers of independent trials with different

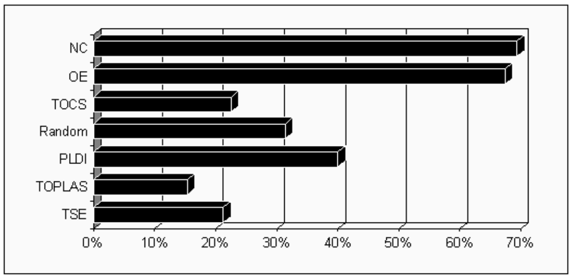

Figure 6: The percentage of design & modeling articles with more than 20% of space devoted to experimental evaluation.

individuals. Unfortunately, this kind of analysis is beyond our resources. However, the trends exposed by our study are so clear-cut, and our conclusions are so modest that they remain valid even for large margins of error. Therefore, we restrict our error analysis to a discussion of the sources of inaccuracies and present plausible arguments for our error estimates. Furthermore, Appendix B makes our classification data publicly available for analysis by anyone. If enough additional people classify the same papers, we might be able to derive a statistically sound error estimate. Since the margin of error analysis is based on a rough estimate instead of a rigorous analysis, this study can only present evidence but cannot supply a conclusive scientific proof.

5.1 Systematic Error

The main sources of systematic error are misclassification and publication selection bias.

5.1.1 Classification Error

Systematic classification errors consist of classification ambiguities and inaccuracies in determining the amount of space devoted to experimental evaluation. To get an idea of the impact of systematic classification errors, consider the classification deviations when the same sample is evaluated by different individuals. Since the random sample was classified by four people, and PLDI, TOPLAS, and TSE where each classified by two individuals, we obtain 468 article classification pairs. Of these, 93, i.e., 20%, show differences. The absolute and relative number of deviations between all class pairs is detailed in Figure 7.

Classification Ambiguity. Classification ambiguities result from subjectivity in interpreting and applying the criteria described in Section 3.1. A close look reveals that the vast majority of classification differences arose between

	Theory	Empirical	Hypothesis	Other	Design				
					0%	(0%..10%]	(10%..20%]	(20%..50%]	> 50%
Theory		0	0	2	**24**	2	0	1	0
Empirical	0%		1	0	2	1	2	4	0
Hypothesis	0%	1%		0	0	0	2	1	0
Other	2%	0%	0%		**9**	1	2	1	1
0%	**26%**	2%	0%	**10%**		**6**	**4**	**9**	**0**
(0%..10%]	2%	1%	0%	1%	**6%**		**5**	4	1
(10%..20%]	0%	2%	2%	2%	**4%**	**5%**		**7**	0
(20%..50%]	1%	4%	1%	1%	**10%**	4%	**8%**		**1**
> 50%	0%	0%	0%	1%	**0%**	1%	0%	**1%**	

Figure 7: The absolute and relative numbers of classification discrepancies observed between different individuals (shown for each pair of classes). The highlighted numbers are discussed in the text.

1. formal theory and design & modeling subclass 0%, due to disagreement on their exact distinction (26% of all discrepancies),
2. design & modeling subclass 0% and category "other", as it was difficult to apply our criteria to some unorthodox work (10% of all discrepancies),
3. design & modeling subclass 0% and the remaining design & modeling subclasses due to different views on how to classify simulations ($6\% + 4\% + 10\% + 0\% = 20\%$ of all discrepancies).

Counting Inaccuracy. About $5\% + 8\% + 1\% = 14\%$ of all discrepancies are between neighboring design & modeling subclasses, due to inaccuracies in determining the exact amount of space devoted to the evaluation, in particular for articles not containing a separate section for experimental evaluation.

To judge the effect of the classification error on the observations in Section 4, the deviations in class cardinalities should be established. Unfortunately, an exact estimation using statistical techniques is not possible given the small number of classifications we have for each sample. Instead, we can make an educated guess by looking directly at the class cardinality deviations in the samples classified by different individuals (Figure 8). For large classes the deviation is less than 20%. For small ones it ranges between 30% and 60% ($\approx$ 2 items).

5.1.2 Publication Selection Bias

The second source of systematic error is that the selection of articles we reviewed could be biased towards a particular style or quality.

	Random	PLDI	TOPLAS	TSE	Average
Theory	1.83	0.00	4.00	1.00	15%
Design	2.00	0.00	1.00	4.00	5%
Emp. + Hyp.	1.00	0.00	3.00	2.00	26%
Other	1.17	0.00	2.00	3.00	36%

	Random	PLDI	TOPLAS	TSE	Average
0%	2.83	1.00	0.00	7.00	17%
(0%..10%]	1.17	0.00	2.00	0.00	13%
(10%..20%]	2.00	0.00	3.00	0.00	45%
(20%..50%]	2.17	0.00	0.00	1.00	11%
> 50%.	0.00	1.00	0.00	2.00	60%

Figure 8: The deviations in cardinalities of major categories and subclasses as classified by different individuals. The deviations are given as absolute mean values for each sample classified by more than one person. The rightmost column shows the corresponding relative mean value over all samples.

Selection of Journals. The CS journals were selected to be representative of different areas of CS. We have concentrated on renowned journals that are widely recognized as leading in their respective fields. Furthermore, we were careful not to consider journals with an editorial policy explicitly encouraging specific kinds of contributions. It is unlikely that the character of the actual research going on in those fields significantly differs from what is published in these journals. It is possible however, that our results do not generalize to other fields within CS. The 1993 PLDI proceedings cannot be considered to be more than a case study of conference contributions.

Random Sample Quality. We claim that the random sample provides a fairly representative cross-section of all areas of CS. This claim is valid if neither the set of publications contained in the INSPEC database nor our inability to get hold of some articles is correlated with the objectives of this study. These seem to be reasonable assumptions.

5.2 Statistical Error

There are two kinds of statistical error in our study. The first one, random classification mistakes, is neglected, because the classification deviation data shown in Figure 7 suggests that it is much smaller than the systematic classification error. The second one concerns questions about how well our samples represent the underlying populations in a statistical sense.

Journal Sample Quality. For the journals NC, TOCS, TOPLAS, and TSE, at least one year was under consideration. Within the considered time span, all articles of a journal were included in the sample, resulting in zero statistical error. We do not claim a particular error bound for generalizations to other time spans. Due to the large number of articles (about 40 per issue) only two issues of OE were studied. Again, within these issues, the statistical error is zero since all articles were included in the sample. We assume the deviations between these issues and others of the same volume to be negligible.

Random Sample Quality. Since the sample of 50 ACM publications was taken at random (from a population of over 800), confidence intervals for the random deviations between observed and true class frequencies can be calculated. Due to the small sample size, the intervals become relatively large if a high confidence niveau is chosen. In Figure 9, confidence intervals (at a 0.7 confidence niveau) for the true class sizes are shown, given observed class sizes of n in a sample of 50 items.

n	interval	n	interval	n	interval
1	0.4...2.6	6	4.0...8.8	11	8.7...13.6
2	1.0...4.0	7	4.8...9.9	12	9.7...14.7
3	1.7...5.2	8	5.7...11.0	13	10.6...15.8
4	2.4...6.4	9	6.9...11.5	14	11.5...16.8
5	3.2...7.6	10	7.8...12.6	15	12.4...17.8

Figure 9: The confidence intervals for different observed class cardinalities n of a sample with 50 items at a 70% confidence niveau.

5.3 Overall Accuracy

The overall error is dominated by class cardinality deviations caused by the systematic classification error and the statistical inaccuracy of the random sample. Based on the discussions in Section 5.1.1 and Section 5.2, we make a worst case analysis to underscore the plausibility of our claims. For the class cardinality deviation the average values presented in Figure 8 are used. The statistical error in the random sample is approximated by the confidence intervals shown in Figure 9. In Figures 10–12, the above error estimates have been applied to the data from Figures 2–6, respectively, in such a way that the maximum possible distortion of class cardinalities in Section 4 is achieved. For NC and OE the sizes of the categories theory, other, design & modeling subclass 0%, and design & modeling subclass (0%..10%] were increased, whereas those of the empirical category, design & modeling subclass (20%..50%], and design & modeling subclass >50% were decreased. For the CS samples, the opposite was done. The trends become weakened but still remain quite visible as we see in the resulting figures.

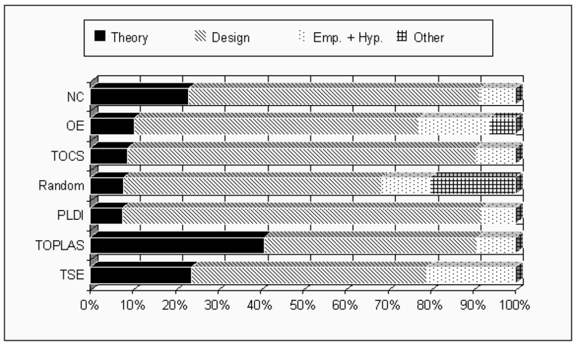

Figure 10: The relative cardinalities of major categories after applying the error estimates.

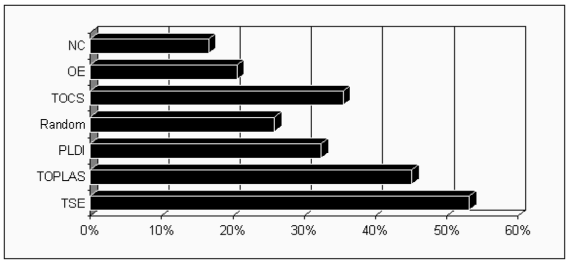

Figure 11: The percentage of design & modeling articles without any experimental evaluation after applying the error estimates.

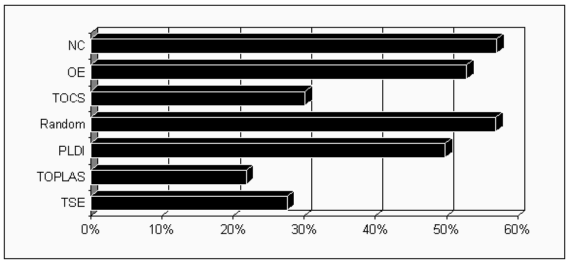

Figure 12: The percentage of design & modeling articles with more than 20% of space devoted to experimental evaluation after applying the error estimates.

6 Conclusions

We presented an empirical study of the amount of experimental evaluation in refereed CS publications. In a random sample, over 40% of articles about new designs and models completely lack such experimentation. For samples related to software engineering, this fraction is higher; it is over 50% for TSE. When considering papers with at least one fifth of their space devoted to evaluation, we find that only 30% of CS papers satisfy this (rather mild) criterion, and only 20% for TSE and 15% for TOPLAS. Even when allowing for the worst possible error in this study, the fraction of unvalidated papers seems high. There is no significant number of articles with purely empirical work that could compensate for this deficiency.

Over half of the CS random sample consists of refereed conference contributions. One might suspect that this is the reason for the high number of articles lacking validation. However, when conferences are excluded, both the ratio of unvalidated work and the ratio for papers with acceptable evaluation change insignificantly (by only two percentage points). Note that these numbers are quite unreliable, because they are based on only 13 papers in the design & modeling subclass. However, two of the three selected journals are worse than the random sample *including* conferences; PLDI, a conference, turns out to be better.

On the whole, we consider this situation as unacceptable, even alarming. The results suggest that large parts of CS may not meet standards long established in the natural and engineering sciences. Among other things, such standards hold that only validated claims are published in journals.

Computer scientists that we have contacted informally with our results (admittedly a biased selection!) are not surprised by our numbers, but are quick with explanations. The youth of computer science is often advanced as a reason for low standards. However, when comparing to *Neural Computing*, this explanation becomes doubtful. NC is only six years old and thus younger than all the CS journals surveyed. Furthermore, computational approaches to an area can hardly be older than computer science. Yet the scientific standards applied in NC appear far better than in computer science in general, and are nearly indistinguishable from an established field as represented by the *Optical Engineering*. We think that youth alone is not a sufficient explanation for poor standards. The most damaging observation one might make is that computer scientists are a minority among the contributors to NC and OE!

Other explanations point to the difficulty of conducting experiments in CS, especially when humans are involved. There may be some truth in that, especially when looking at the software area. However, psychologists have evolved techniques to deal with humans in experimental settings and perhaps CS has simply not embraced those techniques. Furthermore, the experiments that physicists and other scientists conduct are far more complicated and costly than what computer scientists have ever attempted. A more plausible explanation for low standards is that computer scientists, on the whole, have neglected to develop adequate measuring techniques. CS labs seem poorly equipped for evaluating their own progress. Workers who wish to base their claims on solid evidence face a tremendous effort in building up measuring equipment and expertise. Naturally, they are quickly discouraged, and why bother if experimental work is not rewarded and papers are accepted without it?

We also have the impression that while many computer scientists agree that standards should be raised, as individuals they are afraid to take the first step. This is an understandable fear, because investing in experimentation may damage or slow careers. This fear can only be counterbalanced by concerted, open, and positive action. We suggest the following steps:

- Editors, reviewers, and tenure committees must all set higher standards for what constitutes acceptable design papers. Reasonable evaluation of design ideas must be included in almost all papers.
- We must recognize that empirical work is first-class science. Purely empirical work that makes no design contribution of its own should be sought-after material by journals and conferences.
- Wherever appropriate, publicly accessible sets of benchmark problems must be established to be used in experimental evaluations.
- In many areas within CS, rules for how to conduct repeatable experiments still have to be discovered. Workshops, laboratories, and prizes should be organized to help with this process.
- Tenure committees and funding agencies must recognize that high-quality experimental CS needs time and money to produce validated results; but these results will be more valuable than the ones we usually get today.
- Finally, and most effectively, computer scientists have to begin with themselves, in their own laboratories, with their own colleagues and students, to produce results that are grounded in evidence.

We do not expect the situation to change over night. Nor do we require that all design work stop and every computer scientist do nothing but measure. Quite the contrary — new ideas are needed more than ever. But computer scientists must find out how good these ideas are and use experimentation to guide them to the profitable ones.

We submit that computer science, after having been in existence for about half a century (we assume modern CS started with the first digital, electronic computer) is no longer a "young" science whose standards are by necessity weaker than that of established sciences. With the shrinking amount of research funding, CS will face stiff competition from other fields, young and old. "Business as usual" may become extremely damaging for CS. The time has come to act so everyone can take CS seriously once more.

Acknowledgements

We thank Michael Philippsen for an inspiring review of a late version of this report.

FTP address

This technical report is available for anonymous FTP on the Internet from the following source:
machine: ftp.ira.uka.de
directory:
/pub/uni-karlsruhe/papers/techreports/1994
file: 1994-17.ps.Z

References

[1] David H. Bailey. Twelve ways to fool the masses when giving performance results on parallel computers. *Supercomputing Review*, pages 54–55, August 1991. Also in: Supercomputer, September 1991, 4-7.

[2] Doug Baldwin and Johannes Koomen. Using scientific experiments in early computer science laboratories. *ACM SIGCSE Bulletin*, 24(1):102–106, 1992.

[3] Peter J. Denning. What is experimental computer science? *Communications of the ACM*, 23(10):543–544, October 1980.

[4] Peter J. Denning. Performance analysis: Experimental computer science at its best. *Communications of the ACM*, 24(11):725–727, November 1981.

[5] Peter J. Denning. Performance evaluation: experimental computer science at its best. *ACM Performance Evaluation Review (SIGMETRICS)*, 10(3):106–109, 1981.

[6] Jerome A. Feldman and William R. Sutherland. Rejuvenating experimental computer science — A report to the national science foundation and others. *Communications of the ACM*, 22(9):497–502, September 1979.

[7] Norman Fenton, Shari Lawrence Pfleeger, and Robert L. Glass. Science and substance: A challenge to software engineers. *IEEE Software*, pages 86–95, July 1994.

[8] John N. Hooker. Needed: An empirical science of algorithms. *Operations Research*, 42(2):201–212, March 1994.

[9] STN International. INSPEC: Information service for physics and engineering communities. c/o Chemical Abstracts Service, P.O. Box 3012, Columbus, Ohio, 1994. Made by IEE, Station House, 70 Nightingale Road, Hitchin, Herts SG5 1RJ, Great Britain.

[10] Ravi K. Iyer. Experimental computer science. *IEEE Transactions on Software Engineering*, 16(2):109–110, February 1990.

[11] Pat Langley. Machine learning as an experimental science. *Machine Learning*, 3(1):5–8, August 1988.

[12] Daniel D. McCracken, Peter J. Denning, and David H. Brandin. An ACM executive committee position on the crisis in experimental computer science. *Communications of the ACM*, 22(9):503–504, September 1979.

[13] Computer Science and Telecommunications Board. Academic careers for experimental computer scientists and engineers. *Communications of the ACM*, 37(4):87–90, April 1994.

A Titles Drawn at Random

The original random sample contained 74 titles. For the final classification, we excluded 7 articles from the *Communications of the ACM* (CACM), 2 articles from the *History of Programming Languages* conference (HOPL-II), 3 non-CS articles that had gotten into the sample accidentally, 1 title that was a complete workshop proceedings volume, 1 title that was a complete journal issue, and 10 articles not available in our library.

The CACM and HOPL-II articles were excluded because we felt that these publications were reviews or historical accounts that did not claim to advance the science per se.

0 = CMMCS'93, ACM Performance Evaluation Review, 21(1):135–45	
1 = CMMCS'93, ACM Performance Evaluation Review, 21(1):158–70	
2 = OOPSLA'93, ACM SIGPLAN Notices, 28(10):137–43	
3 = OOPSLA'93, ACM SIGPLAN Notices, 28(10):16–28	
4 = PLDI'93, ACM SIGPLAN Notices, 28(6):248–57	
5 = PLDI'93, ACM SIGPLAN Notices, 28(6):268–77	
6 = HOPL-II'93, ACM SIGPLAN Notices, 28(3):351–52	(*)
7 = ACM Computing Surveys, 25(4):415–36	
8 = ACM Letters on Programming Lang. and Systems, 1(4):303–22	
9 = ACM Transactions on Computer Systems, 11(3):226–52	
10 = ACM Transactions on Graphics, 12(4):305–26	
11 = ACM Transactions on Information Systems, 11(2):133–42	
12 = ACM Transactions on Information Systems, 11(3):287–317	
13 = ACM Transactions on Information Systems, 11(4):376–400	
14 = ACM Transactions on Mathematical Software, 19(1):33–43	
15 = ACM Transactions on Mathematical Software, 19(3):419–41	
16 = ACM Trans. on Programming Lang. and Systems, 15(1):182–205	
17 = ACM Trans. on Programming Lang. and Systems, 15(2):337–56	
18 = ACM Trans. on Softw. Engineering and Methodology, 2(2):109–27	
19 = ACM Trans. on Softw. Engineering and Methodology, 2(3):270–85	
20 = PDD'93, ACM SIGPLAN Notices, 28(12):32–42	
21 = POPL'93, ACM Conference Proceedings, pages 16–28	
22 = POPL'93, ACM Conference Proceedings, pages 220–31	
23 = POPL'93, ACM Conference Proceedings, pages 246–59	
24 = POPL'93, ACM Conference Proceedings, pages 325–33	
25 = POPL'93, ACM Conference Proceedings, pages 419–28	
26 = POPL'93, ACM Conference Proceedings, pages 493–501	
27 = POPL'93, ACM Conference Proceedings, pages 99–112	
28 = PPOPP'93, ACM SIGPLAN Notices, 28(7):249–59	
29 = IEEE/ACM Transactions on Networking, 1(1):130–41	
30 = IEEE/ACM Transactions on Networking, 1(2):246–60	
31 = IEEE/ACM Transactions on Networking, 1(3):314–28	
32 = IEEE/ACM Transactions on Networking, 1(3):358–71	
33 = IEEE/ACM Transactions on Networking, 1(4):397–413	
34 = IEEE/ACM Transactions on Networking, 1(5):522–33	
35 = IEEE/ACM Transactions on Networking, 1(6):709–17	
36 = ASEC'93, ACM Conference Proceedings, pages 149–55	(***)
37 = ASEC'93, ACM Conference Proceedings, pages 267–74	(***)
38 = ASEC'93, ACM Conference Proceedings, pages 323–26	(***)
39 = ASEC'93, ACM Conference Proceedings, pages 359–62	(***)
40 = ASEC'93, ACM Conference Proceedings, pages 395–98	(***)
41 = ASEC'93, ACM Conference Proceedings, pages 415–18	(***)
42 = ASEC'93, ACM Conference Proceedings, pages 75–80	(***)
43 = ICMOD'93, ACM SIGMOD Record, 22(2):129–38	
44 = ICMOD'93, ACM SIGMOD Record, 22(2):197–206	
45 = ICMOD'93, ACM SIGMOD Record, 22(2):207–16	
46 = ICMOD'93, ACM SIGMOD Record, 22(2):32–41	
47 = ICMOD'93, ACM SIGMOD Record, 22(2):403–07	
48 = ICMOD'93, ACM SIGMOD Record, 22(2):422–25	
49 = ICMOD'93, ACM SIGMOD Record, 22(2):426–29	
50 = ICMOD'93, ACM SIGMOD Record, 22(2):453–55	
51 = ICMOD'93, ACM SIGMOD Record, 22(2):542–43	
52 = FSE'93, ACM SIGSOFT Software Engineering Notes, 18(5):71–78	
53 = FSE'93, ACM SIGSOFT Software Engineering Notes, 18(5):99–106	
54 = RDIR'93, ACM SIGIR Forum (special issue), pages 281–90	
55 = RDIR'93, ACM SIGIR Forum (special issue), pages 291–97	
56 = CSE'93, ACM SIGCSE Bulletin, 25(1)	(**)
57 = CSE'93, ACM SIGCSE Bulletin, 25(1):213–17	
58 = CSE'93, ACM SIGCSE Bulletin, 25(1):48–52	
59 = VCIP'93, Proc. of the SPIE, 2094(3):1094-102	(*)
60 = PDW'93, UCLA Workshop Proceedings, pages 213–14	(***)
61 = PDW'93, UCLA Workshop Proceedings, pages 52–62	(***)
62 = PDW'93, UCLA Workshop Proceedings, pages 93–104	(***)
63 = PDW'93, UCLA Workshop Proceedings	(**)
64 = Communications of the ACM, 36(4):82–99	(*)
65 = Communications of the ACM, 36(5):54–56	(*)
66 = Communications of the ACM, 36(5):66–69	(*)
67 = Communications of the ACM, 36(6):94–101	(*)
68 = Communications of the ACM, 36(8):78–89	(*)
69 = Communications of the ACM, 36(9):117–26	(*)
70 = Communications of the ACM, 36(11):41–49	(*)
71 = HOPL-II'93, ACM SIGPLAN Notices, 28(3):349–50	(*)
72 = Nucl. Instr. Meth. in Phys. Research, B79(1-4):785–87	(*)
73 = Diamond and Related Materials, 3(1+2):61–65	(*)

(*) non-CS, CACM, and HOPL-II articles were intentionally left out
(**) whole journals and conference records could not be considered
(***) articles not available in our library were not considered

B Classification Data

Except for the ACM random sample, all listed values represent the number of the first page of an article in the respective publication. For the ACM random sample, values correspond to the numbering introduced in Appendix A.

Ernst: ACM 1993 Random from INSPEC

Theory	Empirical	Design	Hypo.	Other
11, 24, 25, 27, 32, 34, 35, 46	43	(see below)	—	2, 7, 13, 26, 47, 58
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
3, 4, 8, 9, 10, 12, 16, 17, 20, 21, 48, 49, 50	22, 23, 29, 30, 53	14, 19, 45, 57	0, 1, 5, 15, 18, 28, 31, 33, 44, 51, 52, 54, 55	—

Lutz: ACM 1993 Random from INSPEC

Theory	Empirical	Design	Hypo.	Other
16, 20, 24, 25, 27	43	(see below)	—	2, 7, 26, 47, 58
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
3, 4, 9, 10, 21, 32, 34, 46, 48, 49, 50	11, 12, 22, 29, 30	0, 1, 8, 13, 14, 19, 45, 52	5, 15, 17, 18, 23, 28, 31, 33, 35, 44, 51, 53, 54, 55, 57	—

Paul: ACM 1993 Random from INSPEC

Theory	Empirical	Design	Hypo.	Other
11, 24, 25, 32, 34, 35	43	(see below)	52	2, 7, 12, 13, 26, 47 58
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
3, 4, 8, 9, 10, 16, 17, 20, 21, 27, 31, 46, 48, 49, 50	22, 29, 30	14, 19, 23, 45, 55, 57	0, 1, 5, 15, 18, 28, 33, 44, 51, 53, 54	—

Walter: ACM 1993 Random from INSPEC

Theory	Empirical	Design	Hypo.	Other
3, 10, 16, 24, 25, 27, 32, 46	—	(see below)	—	2, 26, 47, 57, 58
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
4, 7, 8, 9, 12, 13, 17, 20, 21, 34, 35, 48, 49, 50, 51, 53	11, 22, 23, 29	5, 14, 19, 30, 45, 52	0, 1, 15, 18, 28, 31, 33, 43, 44, 54, 55	—

Ernst: PLDI 1993

Theory	Empirical	Design	Hypothesis	Other
78, 237	177, 187	(see below)	—	—
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
26, 68, 139, 147, 156, 166, 207, 227, 248, 290	36, 46, 100, 112, 126, 197	—	1, 13, 56, 90, 217, 258, 268, 278	300

Paul: PLDI 1993

Theory	Empirical	Design	Hypothesis	Other
78, 237	90, 177	(see below)	—	—
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
26, 68, 139, 147, 156, 166, 207, 227, 248	36, 46, 100, 112, 126, 290	—	1, 13, 56, 187, 197, 217, 258, 268	278, 300

Ernst: TOPLAS 1992

Theory	Empirical	Design	Hypo.	Other
107, 127, 147, 396, 521, 589	1	(see below)	—	462
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
28, 173, 201, 417, 471	339, 574	490	54, 265, 299	—

Paul: TOPLAS 1992

Theory	Empirical	Design	Hypo.	Other
107, 127, 147, 201, 396, 521, 589	1, 265	(see below)	—	462
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
28, 173, 417, 471	339, 574	490	54, 299	—

Ernst: TOPLAS 1993

Theory	Emp.	Design	Hyp.	Other
1, 73, 133, 211, 253, 290, 463, 563, 575, 632, 659, 681, 706, 771, 795, 876	—	(see below)	—	—
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
182, 312, 337, 367, 494	36, 535	400, 826	745	—

Paul: TOPLAS 1993

Theory	Empir.	Design	Hypo.	Other
1, 73, 133, 211, 290, 463, 575, 659, 681, 706, 771	—	(see below)	—	494, 795
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
182, 253, 312, 337, 367, 563, 632, 876	535, 826	400	36, 745	—

Ernst: TOPLAS 1994 No. 1+2

Theory	Empirical	Design	Hypothesis	Other
259	—	(see below)	—	—
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
102, 205	151	3, 35, 175	—	—

Paul: TOPLAS 1994 No. 1+2

Theory	Empirical	Design	Hypothesis	Other
259	35, 102	(see below)	—	—
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
—	151, 175, 205	3	—	—

Paul: TSE 1993

Theory	Empirical	Design	Hypo.	Other
3, 41, 89, 202, 268, 366, 410, 453, 554, 687, 698, 742, 845, 856, 886, 902, 920, 962, 1119	120, 379, 390, 425, 529, 603, 661, 707, 774, 912, 941, 1087, 1095, 1157	(below)	—	13, 70, 307, 503, 950, 1128, 1187
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
56, 108, 139, 214, 231, 165, 187, 313, 344, 436, 478, 533, 571, 584, 594, 641, 749, 826, 935, 976, 982, 997, 1015, 1045, 1071, 1171	253, 277, 297, 486, 613, 672, 765, 863, 1145	181, 813	155, 625, 788, 804, 835, 1028, 1105	24, 720, 1055

Walter: TSE 1993

Theory	Empirical	Design	Hypo.	Other
3, 13, 41, 89, 366, 453, 554, 687, 698, 742, 826, 845, 856, 886, 902, 920, 962, 976, 1015	120, 379, 425, 529, 603, 661, 774, 912, 941, 1045, 1087, 1157	(below)	390	24, 503, 625, 1187
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
56, 70, 108, 139, 165, 187, 202, 214, 231, 268, 307, 313, 344, 410, 436, 478, 533, 571, 584, 594, 613, 749, 813, 835, 863, 935, 950, 997, 1071, 1119, 1128, 1145, 1171	253, 277, 297, 486, 672, 707, 720, 765, 1028	181, 1095	155, 641, 788, 804, 982, 1105	1055

Lutz: NC 1993

Theory	Empirical	Design	Hypo.	Other
132, 140, 165, 205, 278, 305, 371, 392, 550, 767, 783, 812, 893, 910	260, 483, 492, 570, 587, 665, 681, 856	(see below)	—	505
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
18, 154, 359, 597, 613, 823	1, 89, 443	75, 200, 210, 402, 430, 625	21, 32, 45, 61, 105, 115, 213, 228, 267, 289, 317, 331, 341, 363, 367, 374, 419, 456, 463, 649, 719, 736, 750, 795, 869, 885, 928, 954	242, 473, 636, 695, 843, 939

Paul: OE 1994 No. 1+3

Theory	Empirical	Design	Hypothesis	Other
230, 237, 246, 737, 820, 865	85, 102, 167, 198, 204, 213, 303, 692, 721, 881, 951, 967	(see below)	97, 278, 746	681, 685, 751, 762, 776, 785, 889, 957
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
242, 267, 273, 675, 809, 835, 915	37, 64, 79, 160, 697, 908	180, 219	26, 54, 72, 116, 134, 150, 175, 180, 194, 209, 251, 285, 294, 704, 725, 771, 791, 801, 830, 850, 856, 875, 897, 902, 924, 939, 946, 975, 981	776, 730, 845, 932

Paul: TOCS 1991

Theory	Empirical	Design	Hypothesis	Other
364	319	(see below)	—	—
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
1, 101, 201, 242, 374, 399	143	125, 175, 222, 272	21, 66	—

Paul: TOCS 1992

Theory	Empirical	Design	Hypothesis	Other
226, 265	81	(see below)	—	33
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
144, 167, 360	—	53, 110	3, 26, 190, 311, 338	—

Paul: TOCS 1993

Theory	Empirical	Design	Hypothesis	Other
—	253	(see below)	—	—
0%	≤ 10%	≤ 20%	≤ 50%	> 50%
205, 226, 319	73	1, 300, 353, 376	—	—