Large-scale software engineering questions – expert opinion or empirical evidence?

B. Kitchenham, D. Budgen, P. Brereton, M. Turner, S. Charters and S. Linkman

Abstract: A recent report on the state of the UK information technology (IT) industry based most of its findings and recommendations on expert opinion. It is surprising that the report was unable to incorporate more empirical evidence. This paper aims to assess whether it is necessary to base IT industry and academic policy on expert opinion rather than on empirical evidence. Current evidence related to the rate of project failure is identified and the methods used to accumulate that evidence discussed. This shows that the report failed to identify relevant evidence and most evidence related to project failure is based on convenience samples. The status of empirical research in the computing disciplines is reviewed showing that empirical evidence covers a restricted range of subjects and seldom addresses the 'Society' level of analysis. Other more robust designs that would address large-scale IT questions are discussed. We recommend adopting a more systematic approach to accumulating and reporting evidence. In addition, we propose using quasi-experimental designs developed and used in the social sciences to improve the methodology used for undertaking large-scale empirical studies in software engineering.

1 Introduction

In April 2004, the British Royal Academy of Engineering published a report entitled 'The Challenges of Complex Information Technology (IT) Projects' [1]. This document (which we refer to as the RAE report) reported findings and made recommendations about:

- The low level of professionalism in software engineering (SE).
- The poor standard of education in UK universities and Management schools.
- Lack of understanding of the importance of project management.
- Lack of appreciation of the need for Risk Management.
- Lack of appreciation of the critical role software architects play in IT projects.
- The urgent need to promote best practice among IT practitioners.
- The need for basic research into complexity and associated issues.

The RAE report aimed to influence the UK IT Industry, the UK Government Department of Trade and Industry (DTI), and the Engineering and Physical Sciences Research Council (EPSRC) which funds academic research in the UK.

doi:10.1049/iet-sen:20060052

Paper first received 2nd October 2006 and in revised form 27th June 2007

B. Kitchenham, P. Brereton, M. Turner and S. Linkman are with the School of Computing and Mathematics, Keele University, Staffordshire, UK

D. Budgen is with the Department of Computer Science, Durham University, Durham, UK

S. Charters is with Applied Computing Group, Lincoln University, PO Box 84, Lincoln 7647, New Zealand

E-mail: ap_kitchenham@onetel.net.uk

It might be expected that a report aiming to influence decision makers in Industry, Academia and Government would have adopted the most rigorous scientific approach possible to assembling its findings. In fact, the report was based on 'evidence gathered both orally and in the form of written submissions' from 70 individuals (None of the authors of this paper were asked to submit evidence, so it is possible that we may be accused of sour grapes!). Where the report does reference research evidence, there is no suggestion that authors of the report made any attempt to gather the evidence systematically or to critically appraise it as evidence-based SE would recommend [2].

The main rationale for the RAE report was a belief that 'the success rates of software and IT systems are disappointingly low'. This was one of the few areas where the report presented empirical evidence to support their case. In Section 2 of this paper, we review the evidence referenced in the RAE report and evidence available from other sources. In particular, we consider the strength of evidence in terms of the validity of the methodology used to obtain the evidence. We observe that generally the methodology used in the various studies is weak but that the RAE report also failed to consider the most reliable evidence. This failure to review and assess all relevant empirical evidence coupled with an over-reliance on expert opinion may lead to incorrect conclusions about the nature of the problems facing the software industry.

This raises the question of why the RAE report preferred expert opinion with only occasional references to empirical evidence. Is empirical evidence lacking? Is empirical evidence particularly weak? The main purpose of this paper is to look at this issue in more detail. In Section 3, we review the status of empirical evidence in SE research. We observe some weaknesses that might explain the reluctance of software engineers in the IT industry to trust empirical evidence. In Section 4, we consider the empirical methods that would be most appropriate for addressing questions related to the status of SE in the IT industry, rather than individual projects or specific development

methods. We present our conclusion in Section 5. (We consider software engineering the means by which large-scale IT projects are developed. We consider the terms IT industry and software industry as interchangeable.)

2 Rate of failure of software projects

In this section, we present the available evidence concerning the rate of failure of software projects as derived from three sources: first, we present the evidence provided by the RAE report, secondly we present the evidence provided by a systematic review and finally the evidence available from a recent survey of the Norwegian IT industry. Although the last two sources of evidence are reported in the same journal paper [3], we present them separately because they offer two different forms of evidence, that is, a systematic review of past studies (see Section 2.2) and a recent large-scale industrial study (see Section 2.3). The aim of this section is to demonstrate the problem of relying on informal evidence that may be influenced by personal opinion. In addition, the Norwegian study, discussed in Section 2.3, provides an example of good methodology for undertaking an industry survey. The issue of appropriate methodology for industry-level research questions is addressed in more detail in Section 4.

2.1 Evidence reported by the RAE complex IT projects report

When discussing IT project problems, the RAE report references five sources of evidence:

1. The State of IT Management in the UK, Templeton College, Oxford [4].
2. IT projects Sink or Swim, British Computer Society Review [5].
3. Latest Standish Group CHAOS report shows project success rates have improved by 50%, Press release Standish Group, March 2003.
4. Avoiding Information systems (IS)/IT Implementation Failure, TASM 15(4) [6].
5. Courts Libra System 'is one of the worst IT projects ever seen', Computer Weekly, 20 January 2003.

Superficially, we can identify only one of these articles (item 4, Dalcher and Genus [6]) as being a peer-reviewed document. However, on closer examination, this document is in fact an editorial not a journal paper.

It seems fair to ask how strong this evidence is and whether any other relevant evidence has been omitted. Strength of evidence is related to the nature of the evidence and rigour of the methodology used to gather the evidence. The nature and strength of evidence presented in these articles is itemised in Table 1. It is clear that the methods

Table 1: Evidence presented in RAE Report

Report	Claim	Methodology	Methodology strengths and weaknesses
Sauer and Cuthbertson [4]	16% of projects were produced on time, on budget and delivered the required functionality 9% were abandoned Mean overrun on budget = 18% Mean overrun on schedule is 23% Mean underachievement on scope = 7% Cluster analysis indicated that 55% of projects were less than 5% behind time, just over 5% under specification and less than 4% over budget[a]	questionnaires available to anyone looking at Computer Weekly website completed by 1457 IT managers information available on 421 projects	Relatively large sample Convenience sample, so results are difficult to generalise because individuals who complete questionnaires may not be 'typical'
Taylor [5]	12.7% of projects successful. Successful projects (130) were mainly data conversion (79.7%) and only 2.3% development although 50% of all projects were development projects	interviews with 38 project managers, who reported on 1027 projects	The report suggests that IT managers were able to report accurately on about 30 projects each. However, there is no indication of how interviewees were identified or interviewed
Chaos report	US software project success rate is 34%, 50% improvement over the 16% found in first CHAOS report	no information provided	Lack of any defined methodology is of concern
Dalcher and Genus [6]	cost of IS failures is $140 billion in Western Europe and $150 billion in USA	figure reported with no reference	Lack of any reliable reference is of concern
Computer Weekly	a specific project is 134 million pounds over budget	figures obtained from the House of Commons Public Accounts Committee	Information about a single project does not provide any evidence about the basic failure rate of projects

[a]This point was not reported in the RAE report

used to gather evidence are fairly weak. The most rigorous methodology was a large-scale convenience sample (Sauer and Cuthbertson [4]).

2.2 Systematic literature review of peer-reviewed surveys

Recently, Moløkken-Østvold *et al.* undertook a survey to investigate software estimation in the Norwegian IT Industry [3]. Their report on the survey included a systematic literature review of all previous survey studies that discussed the failure rate of software projects. They restricted themselves to peer-reviewed studies and for that reason explicitly omitted the Standish 1994 CHAOS report, other commercial reports and press stories.

Moløkken-Østvold *et al.* noted that the Standish 1994 CHAOS report is frequently referenced, and, indeed, it is explicitly referenced by Sauer and Cuthbertson [4], Taylor [5] and Dalcher and Genus [6]. However, Jørgensen and Moløkken-Østvold have also published a paper expressing strong reservations about the reliability of the 1994 CHAOS report [7]. They point out that the results presented in the CHAOS report are out of step with other evidence, it presents no details of the methodology it used to obtain the reported data, and furthermore the CHAOS report itself implies that it explicitly requested reports of projects having problems (which would bias any results).

The evidence presented by Moløkken-Østvold *et al.* is shown in Table 2. The concentration on peer-reviewed studies means that many of the papers are somewhat dated but they are similar in time frame to the 1994 CHAOS report, which is often assumed to provide the baseline for assessing the rate of project overruns. Table 2 makes it clear that there is more evidence available than was quoted in the RAE report and that generally the problem appears to be less severe than it appears to be from the RAE report. However, Table 2 suggests that the methodology used in the studies is rather weak and much of the evidence is somewhat dated. Three of the four articles that reported their methodology used convenience samples where information is obtained on past projects (and is, therefore, based on recall). When researchers have attempted to use random sampling the response rate has not been very high.

2.3 Survey of Norwegian industry

Moløkken-Østvold *et al.* also report their own survey of the Norwegian software industry in [3]. They used stratified random sampling to identify 18 Norwegian organisations representative of the Norwegian software industry and interviewed project managers at each of the organizations. They interviewed managers of 51 projects from the organisations, where each organisation submitted 1–4 projects. Organisations were asked to submit their most recently completed projects (for projects of at least 100 man-hours).

Full data were available on 44 of the 51 projects they discussed. They reported:

- 32 projects (76%) had effort overruns, with a mean cost overrun of 41% and a median of 21%.
- 26 projects (62%) had schedule overruns, with a mean schedule overrun of 25% and a median overrun of 9%.

In striking contrast to all other researchers, Moløkken-Østvold *et al.* reported their raw data, and gave a detailed account of their survey methodology.

In another paper, Moløkken and Jørgensen use the same data set to report on differences between the estimation accuracy of public and privately funded projects [8]. In this paper, they reported that the average overrun for public projects was 67% and for private projects was 21%.

Overall Moløkken-Østvold *et al.*'s study [3] is extremely rigorous. The research design they used was well-suited to address the issue of estimation accuracy. However, it has some weaknesses, particularly with respect to the analysis of private and public sector projects:

- The researchers used a cluster design (project within organisation) but did not adjust for the cluster effect in their analysis. In addition, the number of projects was not the same for each organisation, so the clusters were not balanced.
- The randomisation was performed at the organisation level but not at the project level. The projects were those that happened to finish when the organisations were approached.
- There was missing data on seven of the projects which could bias the results.
- The private and public sector projects were not a random sample of such projects. This means that the results of the comparison may be due to unreported confounding effects.

In addition, Jørgensen expressed the view that the main methodological limitation of the survey was the use of a questionnaire which required participants to report retrospectively on their projects [9]. He also noted two general problems:

- Estimation terminology is not standardised which makes it difficult to know precisely what a 25% overrun really measures [10].
- Current studies combine budget overrun from contexts where 25% inaccuracy is amazingly accurate with budget overruns where we should expect high accuracy.

With respect to software project overruns, we agree with Jørgensen that it is difficult to know what an overrun really means. Most studies reported the mean overrun which varied from 18% to 50%; one study [11] reported only the median overrun (34%). However, for a skewed variable, such as project overrun, the mean is likely to be misleading. This is confirmed by Moløkken-Østvold *et al.*'s research where the mean cost overrun was 41% but the median was 21%.

With respect to accuracy for different types of project, Capers Jones suggests that overruns are likely to be larger for larger projects [12], so simple percentages (whether mean or median) may be misleading without qualifying the range of project size. However, Capers Jones does not provide any description of the methodology he used to obtain the project data on which he based his conclusions.

2.4 IT complexity problem

So is there a problem with IT projects? This is difficult to determine. The empirical evidence suggests that most IT projects overrun but that the majority do not overrun by very much. In addition, if we discount the Standish CHAOS reports, the magnitude of overruns does not appear to have changed over the last 15–20 years. Given that companies will trim estimated costs to win bids, then the magnitude of overruns (i.e. an 18–50% mean overrun with, probably, a much lower median overrun) may be acceptable for a competitive industry.

What about projects that exhibit abnormally large overruns, for example, 100% or more? It is clear that some IT projects exhibit severe overruns. There is also anecdotal evidence that such large IT projects are more likely to overrun than small projects – although verifiable evidence is limited. These are clearly a problem for SE, but are not unknown in conventional engineering projects too: the Channel Tunnel, Concorde, and the (Wobbling) Millennium Bridge being cases in point. Furthermore, there is no evidence concerning the relative failure rate of large projects in different engineering disciplines.

The RAE report states that:

> '. . . there is a perception that IT projects have a lower success rate than those in more established branches of engineering. Irrespective of the accuracy of this presumption, it is worthwhile exploring the distinctive qualities of IT projects in comparison to other engineering projects . . .' (page 13 Paragraph 2)

Thus, although conceding that there is no actual evidence, the report goes on to discuss the differences between software intensive projects and other forms of engineering project as if there is a specific problem with IT projects, in particular a complexity problem. However, given that other large engineering projects also exhibit both cost

Table 2: Evidence presented by Moløkken-Østvold *et al.*

Report	Year	Claim	Methodology	Methodology strengths and weaknesses
Jenkins [11]	1984	34% cost overrun (median) 22% schedule overrun (average[a]) 61% of projects over budget 65% over schedule	72 projects in 23 major US corporations in three geographical areas with a maximum of 6 projects per organisation The organisations were intended to be broadly representative of large US organisations	probably a convenience sample In many cases initial cost and effort values were estimated retrospectively No adjustment for the clustering
Phan [14] reported in Phan *et al.* [47]	1988	33% mean cost overrun	no information available[b]	
Heemstra *et al.* [15] reported in Heemstra [48]	1989	80%[c] of projects have overruns of budget and duration Mean overruns of effort and duration are 50%	survey of 598 Dutch organisations. No details available	
Lederer and Prasad [13]	1991	63% of large projects significantly overrun their estimates 14% of large projects under run	questionnaire sent to a random sample of 400 members of a 'large, nation wide association of information systems managers and analysts'. 112 viable replies out of 116 replies	no details about the number of projects, or whether accuracy was measured or estimated retrospectively
Bergeron *et al.* [49]	1992	feasibility estimate with mean accuracy 36% (33 projects) Preliminary study estimate with mean accuracy 32% (28 projects) Functional study with mean accuracy 26% (28 projects)	identified 152 organisations using government publications, professional association and contacts. Distributed 374 copies of questionnaire, 67 organisations returned 110 questionnaires of which 89 were usable	convenience sample of organisations data obtained retrospectively
Sauer and Cuthbertson [4]	2003	mean over run on budget = 18% 59% projects used more than estimated effort Mean schedule overrun = 23% 35% projects overran schedule[d]	see Table 1	see Table 1

[a]Moløkken-Østvold *et al.* [3] say the overrun was the median but the paper says 'the average'; [b]We were unable to find either of the original references, we report the information given in [3]; [c]Moløkken-Østvold *et al.* [3] report 70%; [d]Moløkken-Østvold *et al.* [3] differ in emphasis from the RAE report [1]

and schedule overruns and operational problems, it is not clear whether it is the complexity of software or the complexity of the project. If the problem is project complexity, or some other issue such as project novelty not software complexity, the RAE report risks making the wrong recommendations to address the problem.

2.5 Implications for evaluating the IT industry

This discussion has raised a number of issues concerned with the problem of evaluating the state of the IT industry. The comparison of the evaluation of software overruns in the RAE report (Section 2.1) and the Norwegian systematic literature review (Section 2.2) suggests that if we are to make decisions aided by evidence, we need to

- Improve our methods for gathering evidence about the IT industry, using survey method such as those adopted in the Norwegian survey (Section 2.3). This issue is addressed further in Section 4.1.
- Ensure that all the relevant evidence is obtained, not just the most widely reported evidence.
- Evaluate the quality of available evidence.
- Aggregate best-quality evidence as objectively as possible.

We return to the last three issues in Section 4.2. We also need to address the issue of why SE researchers are reluctant to adopt evidence-based SE. This issue is addressed in Section 3.

3 Empirical evidence in SE

In this section, we consider why an important and influential report preferred to rely on expert opinion rather than empirical evidence. There are several possibilities. It may be that reliable evidence is less dramatic than expert opinion, or less persuasive than evidence that supports our own viewpoints. Certainly, the Standish CHAOS reports are frequently cited both to indicate the level of the 'SE crisis' in the early 1990's and to confirm how much projects have improved since then [7].

It may be that SE and IT do not have a tradition of accumulating evidence in an unbiased manner. For example, Lederer and Prasad [13] did not reference Jenkins *et al.* [11], Phan [14] or Heemstra *et al.* [15]. In addition, Jørgensen and Shepperd [16] report a systematic review of software cost estimation based on 304 articles. They found that although software cost-estimation papers were published in 76 journals, in a random selection of 10% of the 304 articles, most only referenced papers from three journals and did not reference seemingly relevant papers.

It may be that there is only a little empirical evidence to call upon, that it is of poor methodological quality or that it is irrelevant. We investigate these three issues in the following subsections.

3.1 How extensive is empirical evidence in SE?

Several researchers have undertaken large-scale reviews of the use of empirical research in SE research (see Table 3). The first two studies compared research in SE with research in other disciplines:

- Tichy *et al.* [17] found 10% of papers were primarily concerned with empirical studies in Computer Science (CS) and SE journals compared with 15.6% in Neural Computation (NC) and Optical Engineering (OE). In SE and CS journals 64% addressed design and modelling, a similar proportion (65%) was found in NC and OE. However, Tichy *et al.* assume that design and modelling papers should include some evaluation, so they assessed the amount of evaluation in each design paper. They found that 43% of CS and SE design papers contained no evaluation compared with 14% of NC and OE papers. Overall about 45% of SE and CS papers included evaluation (i.e. articles classified as 'Empirical', 'Hypothesis testing' or 'Design' papers containing some evaluation).
- Zelkowitz and Wallace [18, 19] found 30% of SE papers included no empirical evaluation compared with 20% for a selection of six journals from other disciplines. However, 34% of SE evaluations were based on 'Assertion' (a weak form of evaluation) compared with 5% in other disciplines.

Both Tichy *et al.* and Zelkowitz and Wallace concluded that SE and CS did not undertake sufficient empirical evaluation. However, Tichy *et al.*'s results appear rather worse than Zelkowitz and Wallace's. Furthermore, Zelkowitz and Wallace reported an improvement in the percentage of articles with no empirical evaluation in 1995 (19%) compared with 1985 (26%).

Subsequently, Glass and his colleagues undertook studies of SE [20] and CS journals [21] and reported a comparison of research methods in CS, SE and IS [22]. They found 11% of CS and 14% of SE papers were evaluative compared with 67% of IS papers. These figures seem incompatible with the previous studies, but probably reflect Glass *et al.*'s decision to classify each paper only according to its main content. Thus, their results may be more comparable to Tichy *et al.*'s 10% of 'Empirical' and 'Hypothesis testing' studies.

In a recent study [23], Zelkowitz updated his research to cover the years 2000 and 2005. He concluded that the amount and rigour of empirical work was improving:

- Except for 2000, articles without evaluation dropped from 27% in 1985 to 11% in 2005. The 2000 results were explained by the unusually large number of ICSE papers and the relatively large number of 'No experimentation' papers in those proceedings.
- Evaluation by 'Assertion' (i.e. a preliminary test by a technology developer) dropped from 35% in 1985 to 21% in 2005.
- Evaluation by dynamic analysis increased dramatically from less than 2% in 1990 (none in 1985) to 20% in 2005. This was explained as being a result of the availability of open-source software.

Other recent studies reported rather contradictory results:

- Segal *et al.* [24] found 53% of Empirical Software Engineering Journal (EMSE) papers were 'Evaluative' (using Glass *et al.*'s classification).
- Zannier *et al.* [25] found that 44 of a sample of 63 ICSE papers (i.e. 70%) contained some form of evaluation. However, they took a very inclusive view of what constituted 'evaluation', including 'Discussion' (5 articles) and 'Example Application' (6 articles). They also note that a significantly higher proportion of papers including evaluation were published in the period 1991–2005 compared with in the period 1975–1990.

Table 3: Studies of empirical research in the computing disciplines

Authors	Date	Source material	Number of articles reviewed
Tichy, *et al.* [17]	1995	ACM TOCS volumes 9-11 (1991–1993)	TOCS 38
		ACM TOPLAS, volume 14, 15, 16 (nos 1, 2), 1992–1994	TOPLAS 52
		IEEE TSE volumes volume 19, 1994 ACM SIGPLAN Conference	TSE 87 Random 50
		PLDI, 1993	NC 72
		Random sample of 75 ACM titles, 1993	OE 75
		NC volume 5, 1993	
		OE volume 33 (nos 1 and 3) 1994	
Zelkowitz and Wallace [18]	1997	ICSE 1985, 1990, 1995	612 SE papers
		IEEE Software 1985, 1990, 1995	137 non-computing papers
		IEEE TSE 1985, 1990, 1995	
		6 non-computing journals	
Zelkowitz and Wallace [19]	1998	as above (but excluded analysis of non-computing papers)	612 SE papers
Glass [20] *et al.*	2002	Information and Software Technology (IST)	369
		Journal of Systems and Software (JSS)	
		Software Practice and Experience	
		IEEE Software	
		ACM TOSEM	
		IEEE TSE 1995–1999	
		Every 5th paper	
Ramesh *et al.* [21]	2004	IEEE Transactions on Computers	IEEE papers 309
		Journal of ACM	ACM papers 286
		IEEE Transactions on Knowledge and Data Engineering	Joint papers 33
		IEEE Transactions on Pattern Analysis and Distributed Systems	
		ACM Transactions on Human-Computer Interaction	
		ACM Transactions on Database Systems	
		ACM Transactions on Graphics	
		ACM Transactions on Modelling and Computer Simulation	
		IEEE/ACM Transactions on Networking	
		ACM Transactions on Programming Languages and Systems	
		IEEE Transactions on Visualisation and Computer Graphics	
		1995–1999.	
		1 in 10 from IEEE journals	
		1 in 3 from ACM journals	
Glass *et al.* [22]	2004	As above plus IS papers	
Sjøberg, *et al.* [27]	2004	As per Glass *et al.* (2002) plus EASE, EMSE, ICSE, IEEE Computer, ISESE, JSME, METRICS	103 (software experiments)
Segal *et al.* [24]	2005	EMSE 1997–2003	119
Zannier *et al.* [25]	2006	ICSE	66 in sample 44 including empirical evaluation
		1975–2005	
		5% sample	
Zelkowitz [23]	2007	As previously but including Years, 2000 and 2005	187 (year 2000)
			174 (year 2005)
Höfer and Tichy [28]	2007	EMSE years 1996-June 2006	133
Jørgensen and Shepperd [16]	2007	76 Journals	304 (cost estimation related)
		All issues up to April 2004	

3.2 Nature of empirical studies in SE research

As noted above, all the different researchers used different categorisation schemes, so it is difficult to compare their results. This is made worse because the classification labels they used are not always intuitive. For example:

- Zelkowitz and Wallace use the term 'Replicated Experiment' to mean 'an experiment in which several projects (e.g. subjects) are staffed to perform a task using alternative treatments.' in other words a controlled experiment. The term 'replication' thus refers to the multiple projects not to a replication of a previously undertaken experiment.

• Glass and his coworkers [26] use the term 'Profession' in the context of classifying the unit/level of analysis to refer to teaching or research in the academic community, not the IT industry.

Variations in each classification scheme may explain the differences in the rates of using specific experiment types shown in Table 4. Compared with Table 3, Table 4 excludes papers that reported the same results. Studies are also sorted according to whether the source material came from general SE sources or specialist SE sources. Overall, there appears to be an increase in empirical methods of all types over time, but field studies (and surveys which are one form of field study) are seldom considered in SE although they are frequently used in IS research [22].

Although experiments seem to be being used more frequently than in the past, they have drawbacks. Experiments often use student subjects and/or small scale tasks:

• Sjøberg *et al.* [27] reported 73% of experiments used student subjects. About two-thirds of the 41 papers that reported task duration used tasks of less than 2 h.
• Høfer and Tichy [28] report 62% of experiments reported in EMSE journal had student subjects.

Experiments span a limited range of topics:

• Sjøberg *et al.* [27] found that the distribution of SE controlled experiments and general SE research did not match very well. There were much larger percentages of experiments related to software lifecycle engineering and methods/techniques than there was SE research. They also found that 37% of SE controlled experiments related to inspections.
• Høfer and Tichy [28] reported 15 out of 17 articles studying inspection papers reported experiments (the largest proportion of any of the topics). They also comment that, irrespective of experimental method, 'the range of software topics studied empirically is rather narrow'.

In addition, there are problems replicating experimental results. Overall, not many replications take place. Zannier *et al.* [25] observed only one replication in their study. Høfer and Tichy [28] reported 14 of the 53 papers (26%) describing experiments were replications. Sjøberg *et al.* [27] observed 20 replication experiments (19% of the 103 papers) that addressed 14 topics (i.e. some topics attracted more than one replication). They also observed that replications by the same researchers usually confirmed the original results whereas independent replications usually did not.

Case-study research appears to be undertaken about as frequently as experiments and has the advantage that it usually involves practitioners and not students [28]. There have been no detailed studies of case studies comparable to Sjøberg *et al.*'s study of experiments so little is known in detail about the type of case studies being undertaken. However, Høfer and Tichy [28] reported that 32% of the 22 Metrics/Measurement papers were case studies. Thus, it appears that certain topics attract certain types of research method. For example 55% of cost-estimation studies were based on analysis of historical data [16].

Finally, only Glass *et al.* [20–22] consider the 'level of analysis' of a study which assesses the object on which the research study focused. If we want to understand issues relevant to the software industry, we need to address the 'Society' level. However, for SE the proportion of papers addressing this level is 0.27% and for CS no papers addressed this level. This compares with 3.1% for IS research (also a relatively small percentage).

3.3 How sound is empirical evidence in SE?

There have been several criticisms of empirical standards in SE. Kitchenham *et al.* [30] criticised current standards of performing and reporting empirical studies and proposed guidelines for improving empirical studies. Other researchers have made more specific criticisms that a lack of reporting standards is causing problems when researchers attempt to aggregate empirical evidence because important

Table 4: Frequency of different types of empirical study

Authors	Source	Date	Frequency of experiments (%)	Frequency of case studies (%)	Frequency of surveys (%)	Frequency of field studies and experiments (%)
Tichy *et al.* [17]	General SE	1995	<1			
Zelkowitz and Wallace [18]	General SE	1997	3.1	10.3		2.3
Ramesh *et al.* [21]	General SE	2002	4	2		1
Sjøberg *et al.* [27]	General SE	2004	1.9			
Zannier *et al.* [25]	General SE (single source)	2006	17.5 (all 'pseudo' experiments, no controlled experiments)	17.5 (6 of the 11 studies were 'exploratory' case studies)	0	
Zelkowitz [23]	General	2007	5.5	16		5
Segal *et al.* [24]	Specialist (Empirical SE)	2005	36	13		
Höfer and Tichy [28]	Specialist (Empirical SE)	2007	37.6	29	6	
Jørgensen and Sheppped [16]	Specialist (Cost Estimation articles)	2007	6	3	9	

information is not reported or is reported in an inconsistent fashion (e.g. Pickard *et al.* [31], Wohlin *et al.* [32]).

Other results also suggest problems with the quality of SE research:

- Zannier *et al.* [25] were unable to find any papers that met their criteria for controlled experiments and only 5 of 11 studies that met their criteria for rigorous case studies.
- Mendes reviewed 173 papers on web engineering and concluded that only 5% could be considered methodologically sound [29].
- Dyba *et al.* [33] reviewed the 103 articles on formal experiments selected from 5453 articles identified by Sjøberg *et al.* [27]. They found seven experiments where they were unable to track which test addressed which research question. In addition, they concluded that the statistical power of software SE experiments 'falls substantially below accepted norms as well as levels found in the related discipline of information systems research'.
- Jørgensen and Shepperd [16] observed that cost-estimation researchers based their assessment of current research on a very limited number of journals; fail to research the cost-estimation methods used in industry; and concentrate too much on analysing historical data sets which are no longer of relevance to the software industry.

3.4 Can we trust SE evidence?

The evidence suggests that a majority of SE papers do include some form of evaluation (with estimates varying from 50% to 90%). However, only a relatively small number have evaluation as their main concern (~10%). Nonetheless, there is evidence that the number of empirical studies is increasing. However, there are problems with the quality of empirical studies which is not surprising given that empirical methods are not usually taught at undergraduate level on CS or SE, although 'Empirical Approaches' does now appear in the new 2007 subject benchmark from the Quality Assurance Agency (QAA), which is the organisation responsible for standards in UK higher education.

Additional problems arise because the range of empirical studies is relatively narrow, so important issues are not being addressed. There is also a problem if we are concerned about industry-level issues since very few studies address this level of analysis.

Thus, we can only place limited trust in empirical studies. If we are concerned with lifecycle issues (particularly inspection methods) and cost estimation, there is a growing body of empirical evidence to consult. If we want empirical evidence about factors that impact the IT industry rather than the project lifecycle, there is little empirical evidence. To address issues of this sort, we need not only to perform more empirical studies but also to improve their quality. We also need to focus more on the 'Society' level of analysis. This means that we need to perform more high-quality studies in an industrial context. The most appropriate forms of empirical study for this type of investigation are case studies, surveys and industry-scale quasi-experiments. We discuss these forms of empirical study in the next section.

4 Improving empirical research at the society level

In Section 2, we provided an example to demonstrate that expert opinion and informally collected evidence might result in invalid conclusions. Furthermore, we noted that with the exception of the Norwegian study [3], the empirical studies we discussed used rather poor research methodology. In Section 3, we confirmed that empirical SE is relatively sparse, concentrated on particular topic areas (e.g. inspections and cost estimation), and on lifecycle tasks rather than industry-level research questions. In this section, we discuss what methodologies might be appropriate for industry-level empirical studies.

The social sciences have developed a large number of quasi-experimental designs for field experiments [34], procedures for case studies (e.g. [35, 36] and procedures for surveys [37]. In Section 3, we discussed the need for more rigour in surveys, particularly the need for random sampling rather than convenience sampling. In Section 2.3, we described a survey that used such a methodology. However, we recognise that not all empirical studies can adopt randomisation. In some cases, neither random sampling nor random allocation to treatment are possible, and replication is financially or practically infeasible. These constraints imply that empirical methods other than experiments and surveys are needed to obtain evidence about SE methods and the IT industry. In this section, we consider the value of quasi-experimental design and some related case-study methods to cope with lack of randomisation and replication. We recognise that the suggestions in this section are fairly speculative, but we have provided what evidence we could find in SE of the value of the suggested methods.

4.1 Quasi-experiments and case studies

We suggest the use of quasi-experiments because they are suitable for large-scale studies but have not been used to any great extent in SE research. Quasi-experimental designs are no longer restricted to simple, methodologically weak, 'before and after' studies. In fact, some quasi-experimental designs are almost as rigorous as controlled experimental, for example:

- Interrupted time-series designs where a series of measurements are taken before a change, and a series of measurements are taken after a change, are capable of assessing the impact of large-scale industry-wide process changes. For example, McGarry *et al.* [38] plotted project outcomes before and after the introduction of CMM level 2. The graphs showed that improvements in productivity and defect rates were not due to the introduction of CMM. This was because the improvement rate seen after the introduction of CM level 2 was a continuation of the improvement rate that had been observed prior to the introduction of CMM. The plots spanned a 14-year period and were based on 89 projects.
- Regression-discontinuity designs where experimental units are assigned to an experimental condition based on a cut-off point of a measurable attribute (e.g. project budget, manager experience). If the experimental condition has an impact on an outcome variable (e.g. project cost overrun), it should be visible when the attribute used to decide the cut-off point is plotted against the outcome variable. To our knowledge, this type of design has not been used in SE but could be used, for example, to assess the impact of a new management regime by using the new method for projects over a budget cut-off point then plotting the difference between budget and actual against the original budget.

Even 'before and after' studies can be made more rigorous by the inclusion of control groups and/or repeated

before and after measurements. For example, Dion [39] recorded cost of quality and productivity data for 18 projects, undertaken during a 5-year process improvement activity. The first two projects were started before the process changes were introduced whereas the subsequent projects were started as a series of process changes were introduced. Simple plots of the results show an ongoing improvement over time consistent with a continual process improvement strategy. However, the provision of data on projects started prior to the process changes gives additional confidence that the effect was due to the process change rather than other factors.

We recognise that there is no 'magic bullet' for empirical studies, and that all empirical methodologies have weaknesses. However, the weaknesses of large-scale quasi-experiments are at least well understood and fully documented [34] and there is substantial evidence from other disciplines that these techniques are appropriate for addressing questions at the Society level.

Using quasi-experimental design principles, we can also make some general improvements to our empirical procedures:

- We should consider prospective studies rather than retrospective studies. That is, we should plan to collect data on the next project undertaken, not the last project undertaken. It is much more difficult, but much more reliable, to collect data as it occurs rather than via retrospective recall.
- To compare different groups of projects we should consider case-control studies. For example if we want to compare private and public projects, we should take a random sample of organisations that do both types of project, identify a case (public project) and a control (a private project) matched on some basic criteria: size, complexity, staff capability. If we are unable to find matches between public and private projects, this itself is an important step forward in our understanding of the IT industry. This design could also be used to investigate the reasons for project success and failure and would be an improvement on the current standard of post hoc surveys based on convenience samples.

At a more detailed level, we can address some of the specific limitations we observed in Moløkken and Jørgensen's [8] study:

- Moløkken and Jørgensen [8] used a cluster-type design for their cross-company surveys. However, to use this form of design appropriately, it is necessary to randomise at both levels of abstraction: the organisation level and the project level. In addition, the data needs to be analysed using appropriate methods. (An alternative to randomising at the organisation level would be a census of all projects completed in the previous 12 months.)
- We should agree protocols for handling missing values and dropouts. Ignoring dropouts or using imputation methods is only appropriate if you can confirm that there is no systematic reason for missing data.

4.2 Aggregating evidence

In Section 4.1, we discussed the ways of improving the methods to obtain evidence. However, even if we improve our methods of obtaining empirical evidence about the IT industry, we will not be able to use that evidence effectively unless we can integrate it. This requires secondary research methods such as systematic literature reviews (see e.g. [40, 41]. Systematic literature reviews differ from conventional descriptive reviews because they employ a well-defined methodology intended to ensure that evidence is assembled fairly, rigorously and transparently. Some of the features that differentiate a systematic review from a conventional expert literature review are:

- They address a well-defined research question.
- They are based on a defined search strategy that aims to detect as much of the relevant literature as possible.
- They document their search strategy so that readers can assess its rigour and completeness and the process as a degree of repeatability (bearing in mind that searches of digital libraries are almost impossible to replicate).
- They require explicit inclusion and exclusion criteria to assess each potential primary study.
- They specify the information to be obtained from each primary study including quality criteria by which to evaluate each primary study.

Recently, systematic literature reviews have been used to good effect to address cost-estimation research questions. Apart from the studies already discussed (i.e. [7, 16]), there have been five systematic literature reviews since 2004 (up to February 2007) addressing the following research questions:

- Are mathematical estimating models more accurate than expert opinion-based estimates?
  - No. [42]
- Are regression-based estimation models more accurate than analogy-based models?
  - No. [43]
- Do researchers use cost-estimation terms consistently and appropriately?
  - No they confuse prices, estimates, and budgets. [10]
- Should you use a benchmarking database to construct an estimating model for a particular company if you have no data of your own?
  - Not if you work for a small company doing niche applications [44].
- When should you use expert opinion estimates?
  - When you do not have a calibrated model, or important contextual information is not incorporated into your model [45].

These studies illustrate how data can be aggregated either to answer questions that could not be addressed directly by the primary studies on which they were based, or to provide explanations for differing results found in different primary studies.

5 Conclusions and recommendations

Our aim in this paper is not simply to criticise the RAE report but to raise the issue of the extent to which SE relies on expert opinion rather than empirical evidence to influence public policy. We believe that public policy is best served by more rigorous empirical studies aimed at understanding the extent, nature and causes of IT industry problems and by adopting better methods of aggregating evidence from such studies.

It is ironic that we appear to be better-informed about the limitations of our evidence than about issues of major importance to the software industry and academia. Nonetheless, the preceding discussion indicates that if we are to make informed decisions about issues affecting the IT industry as a whole, we need reliable empirical evidence. In order to obtain reliable evidence, more researchers need to adopt the approach taken by Moløkken-Østvold *et al.* [3]:

- Report previous research in a fair and transparent fashion.
- Use more reliable research methods for large-scale research questions.

To address the first point we recommend researchers use the methodology for performing systematic literature reviews (e.g. [40, 41]. To address the second point we recommend researchers adopt more rigorous quasi-experimental designs [34] including more rigorous case studies, and surveys based on random sampling rather than convenience sampling.

However, we believe improvements in the rigour of empirical studies and the value we place on empirical evidence will not occur without changes in the way we educate software engineers. Software engineers are not encouraged to expect decisions to be informed by evidence. Neither our text books nor our international standards are evidence-based. Software engineers are not trained to expect management decisions and industry analysis to be based on evidence. We believe that training in systematic literature review and critical assessment is essential. Furthermore, experience suggests that students themselves appreciate the need for such skills [46].

6 Acknowledgments

We thank the anonymous reviewers for their encouragement to revise and extend our original paper. Our thanks to Kjetil Moløkken-Østvold and Magne Jørgensen for discussing their survey methodology with us. Our research was supported by the EPSRC EBSE project (EP/C51839X/1). A previous version of this paper was reported at the Workshop on Empirical Software Engineering, Amsterdam, June 2006.

7 References

1 RAE Report. The Challenges of Complex IT Projects. The Report of a working group from the Royal Academy of Engineering and The British Computer Society, The Royal Academy of Engineering, April 2004
2 Kitchenham, B., Dybå, T., and Jørgensen, M.: 'Evidence-based Software Engineering'. Proc. 26th Int. Conf. on Software Engineering, (ICSE '04), IEEE Computer Society, Washington, DC, USA, 2004, pp. 273–281
3 Moløkken-Østvold, K., Jørgensen, M., Tanilkan, S.S., Gallis, H., Lien, A., and Hove, S.: 'A Survey on Software Estimation in the Norwegian Industry'. Proc. 10th Int. Symp. on Software metrics. Metrics 2004, IEEE Computer Society, 2004, pp. 208–219
4 Sauer, C., and Cuthbertson, C.: 'The State of IT Management in the UK' (Templeton College, Oxford, November 2003)
5 Taylor, A.: 'IT projects sink or swim', *Brit. Comput. Soc. Rev.*, 2001
6 Dalcher, D., and Genus, A.: 'Avoiding IS/IT implementation failure', *TASM*, 2003, **15**, (4), pp. 404–407
7 Jørgensen, M., and Moløkken-Østvold, K.: 'How large are software cost overruns? A review of the 1994 CHAOS report', *Inf. Softw. Technol.*, 2006, **48**, pp. 297–301
8 Moløkken, K., and Jørgensen, M.: 'Project Estimation in the Norwegian Software Industry – a summary' (Simula Research Laboratory, 3 March 2004)
9 Jørgensen, M.: 'Personal communication', 19 March 2006
10 Grimstad, S., Jørgensen, M., and Moløkken-Østvold, K.: 'Software effort estimation terminology: the tower of Babel', *Inf. Softw. Technol.*, 2006, **48**, (4), pp. 302–310
11 Jenkins, A.M., Naumann, J.D., and Wetherbe, J.C.: 'Empirical investigation of systems development practice and results', *Inf. Manag.*, 1984, **7**, pp. 73–82
12 Jones, T.C.: 'Project management tools and software failures and successes', Crosstalk, July 1998 available at: http://www.stsc.hill.af.mil/crosstalk/1998/07/tools.asp
13 Lederer, A.L., and Prasad, J.: 'Causes of Inaccurate Software Development Cost Estimates', *J. Syst. Softw.*, 1995, **31**, pp. 125–134
14 Phan, D.: 'Information Systems project management: an Integrated Resource Planning Perspective Model', Department of management and Information Systems, Arizona, Tuscon, 1990
15 Heemstra, F.J., Kusters, R., and van Genuchten, M.: 'Selections of software cost estimation models'. Report TUE/BDK University of Technology, Eindhoven, 1989
16 Jørgensen,, M., and Shepperd, M.: 'A systematic review of software development cost estimation studies', *IEEE Trans. Softw. Eng.*, 2007, **33**, (1), pp. 33–53
17 Tichy, W.F., Lukowicz, P., Prechelt, L., and Heinz, E.A.: 'Experimental evaluation in computer science: A quantitative study', *J. Syst. Softw.*, 1995, **28**, (1), pp. 9–18
18 Zelkowitz, M., and Wallace, D.: 'Experimental validation in software engineering', *Inf. Softw. Technol.*, 1997, **39**, pp. 735–743
19 Zelkowitz, M., and Wallace, D.: 'Experimental models for validating computer technology', *IEEE Comput.*, 1998, **31**, (5), pp. 23–31
20 Glass, R.L., Vessey, I., and Ramesh, V.: 'Research in software engineering: an analysis of the literature', *Inf. Softw. Techol.*, 2002, **44**, (8), pp. 491–506
21 Ramesh, V., Glass, R.L., and Vessey, I.: 'Research in computer science: an empirical study', *JSS*, 2004, **70**, pp. 165–176
22 Glass, R.L., Vessey, I., and Ramesh, V.: 'An analysis of Research in Computing Disciplines', *Commun. ACM*, 2004, **47**, (6), pp. 89–94
23 Zelkowitz, M.V.: 'Techniques for Empirical validation' in Basili, V. *et al.* (eds.): 'Empirical Software Engineering Issues', LNCS 4336, (Springer-Verlag, Berlin Heidelberg, 2007)
24 Segal, J., Grinyer, A., and Sharp, H.: 'The type of evidence produced by empirical software engineers'. Proceedings of REBSE'05, ICSE, 2005
25 Zannier, C., Melnik, G., and Maurer, F.: 'On the Success of Empirical Studies in the Int. Conf. on Software Engineering'. ICSE06, 2006, pp. 341–350
26 Vessey, I., Ramesh, V., and Glass, R.L.: 'A united classification system for research in the computing disciplines', TR 107-1, 2002, available at: www.indiana.edu/~isdept/research/workingpapers.html
27 Sjøberg, D.I.K., Hannay, J.E., Hansen, O., Kampenes, V.B., Karahasanovic, A., Liborg, N.K., and Rekdal, A.C.: 'A survey of controlled experiments in software engineering', *IEEE Trans. SE*, 2005, **31**, (9), pp. 733–753
28 Høfer, A., and Tichy, W.F.: 'Status of Empirical Research in Software Engineering' in Basili, V. *et al.* (Eds.): 'Empirical Software Engineering Issues', LNCS 4336, (Springer-Verlag, Berlin Heidelberg, 2007)
29 Mendes, E.: 'A systematic review of Web engineering research', Int. Symp. on Empirical Software Engineering, 2005
30 Kitchenham, B., Pfleeger, S.L., Pickard, L.M., Jones, P., Hoaglin, D., El Emam, K., and Rosenberg, J.: 'Preliminary Guidelines for Empirical Research in Software Engineering', *IEEE Trans. Softw. Eng.*, 2002, **28**, (8), pp. 721–734
31 Pickard, L.M., Kitchenham, B.A., and Jones, P.: 'Combining Empirical Results in Software Engineering', *Inf. Softw. Technol.*, 1998, **40**, (14), pp. 811–821
32 Wohlin, C., Petersson, H., and Aurum, A.: 'Combining data from reading experiments in Software Inspections' in Juristo, N., and Moreno, A. (eds.): 'Lecture Notes on Empirical Software Engineering', (World Scientific Publishing, 2003)
33 Dyba, Te, Kampenes, V.B., and Sjoberg, D.I.K.: 'A systematic review of statistical power in software engineering experiments', *Inf. Softw. Technol.*, 2006, **48**, (8), pp. 745–755
34 Shaddish, W.R., Cook, T.D., and Campbell, D.T.: 'Experimental and Quasi-Experimental Designs for Generalized Causal Inference' (Houghton Mifflin Company, 2002)
35 Yin, R.K.: 'Case Study Research Design and Methods' (Sage Publications Inc., 2003, 3rd edn.)
36 Stake, R.E.: 'The Art of Case Study Research' (Sage Publications Inc., 1995)
37 Fowler, F.J. Jr.: 'Survey Research Methods' (Sage Publications, 2002, 3d edn.)
38 McGarry, F., Burke, S., and Decker, B.: 'Measuring the impacts individual process maturity attributes have on software Products'. Proc. Fifth Int. Software Metrics Symp., IEEE Computer Society, 1998, pp. 52–60

39 Dion, R.: 'Process Improvement and the Corporate Balance Sheet', *IEEE Softw.*, **10**, (4), pp. 28–35
40 Petticrew, M., and Roberts, H.: 'Systematic Reviews in the Social Sciences. A Practical Guide' (Blackwell Publishing, 2006.)
41 Kitchenham, B.: 'Procedures for Performing Systematic Reviews', Joint Technical Report, Keele University TR/SE-0401 and NICTA 0400011T.1, July 2004
42 Jørgensen, M.: 'A review of studies on expert estimation of software development effort', *J. Syst. Softw.*, 2004, **70**, (1–2), pp. 37–60
43 Mair, C., and Shepperd, M.: 'The consistency of empirical comparisons of regression and analogy-based software project cost prediction', Int Symp. on Empirical Software Engineering, ISESE05, 2005
44 Kitchenham, B., Mendes, E., and Travassos, G.H.: 'A systematic review of cross vs. within-company cost estimation studies', *IEEE Trans. SE*, 2007, **33**, (5), pp. 316–329 (Short version published in proceedings of EASE06)
45 Jørgensen, M.: 'Estimation of software development work effort: evidence on expert judgement and formal models', *Int. J. Forecast.*, 2007, **23**, (3), pp. 449–462
46 Jørgensen, M., Dybå, T., and Kitchenham, B.: 'Teaching Evidence-Based Software Engineering to University Students'. 11th IEEE Int. Software Metrics Symp. (METRICS'05), 2005, p. 24
47 Phan, D., Vogel, D., and Nunamaker, J.F.: 'The Search for Perfect project management', *Computerworld*, 1988, pp. 95–100
48 Heemstra, F.J.: 'Software Cost Estimation', *Inf. Softw. Technol.*, 1992, **34**, (10), pp. 627–639
49 Bergeron, F., and St-Arnaud, J.-Y.: 'Estimation of information systems Development efforts: a pilot study', *Inf. Manag.*, 1992, **22**, pp. 238–254