



Cost Savings in Global Software Engineering

Where's the Evidence?

Darja Šmite, Fabio Calefato, and Claes Wohlin



THE POPULARITY OF offshoring software development to countries with lower labor costs keeps growing, fueled by the hope that it will cut expenses.¹ This remains the case despite occasional claims that offshoring is motivated by reasons other than reducing expenses, such as proximity to customers and markets, access to specific expertise, and the enabling of innovation and shared best practices.

A recent industry report claimed that a project that outsourced development from Germany to India resulted in high-quality software delivered on time, as well as cost savings of “several million euros per year”² due to lower salaries. But how accurate are such claims? Is actual evidence provided? Are the cost-calculating methods transparent?

Many corporate executives have moved and are continuing to move software development offshore, swayed by lower programmer salaries. However, a close look shows that it is unclear whether offshoring really yields overall economic benefits. In fact, experience demonstrates that the assumed savings should not be taken for granted.³

Several studies indicate that cost savings are not as great as pure salary comparisons suggest due to other factors such as the additional over-

head necessary to manage overseas work.^{2,3,4} Moreover, low-cost labor's benefits must be weighed against the risk of missed deadlines, lower-quality products, and dissatisfied customers.

To shed some light on this matter, we look at the evidence of whether offshoring actually yields cost savings. Evidence, most commonly employed in the legal process, is often used in decision making. We examine offshoring using the legal system as a metaphor, based on the evidence profile Claes Wohlin has proposed (see the “Evidence Profile” sidebar).⁵

Evidence of Cost Savings

Our review of more than 500 papers on global software engineering (GSE) found that only 14 presented evidence of achieving or failing to achieve cost savings (see Table 1), although most of the authors mentioned economic benefits in their articles.

Our evidence profile (see Figure 1) yielded inconclusive results because the numbers of studies demonstrating and not demonstrating cost savings were almost equal. We were also unable to determine patterns as to which factors influenced cost savings. For example, we found no connection between savings and sourcing models (outsourcing versus insourcing) or the number of sites to



EVIDENCE PROFILE

Countries' judicial systems differ, but most work with evidence and feature several key players: judge, jury, prosecutor, defense attorney, and defendant. Metaphorically speaking, these roles correspond to the following elements of our study.

The defendant is the study's object, which in our case is determining whether offshoring can reduce costs. The goal is to use the available evidence to determine whether this is the case. Prosecutors and defense attorneys are the authors of the papers we reviewed. Their roles are based on whether their papers say global software engineering does or does not yield cost savings. The judge or jury is a company's decision makers—such as senior executives or project managers—who must decide whether to offshore software development based on the available evidence.

The first decision to make is to judge what exactly constitutes evidence:

- If utilizing surveys, companies must determine whether an entire survey is one piece of evidence or each response is a separate piece of evidence.
- If using a collection of case studies, companies must decide whether the aggregation is one piece of evidence or each study is an individual piece of evidence.

We recommend using a study's objective to decide what constitutes a piece of evidence. For example, if the study reaches a single conclusion based on several case studies within a company, the aggregation is a single piece of evidence. Similarly, if it derives overall results from a survey, that also represents a single piece of evidence.

But if authors present multiple case studies and offer several conclusions based on each case, the study should be viewed as containing separate pieces of evidence. Each would be weaker than the evidence produced by a set of case studies leading to one conclusion.

Claes Wohlin discussed how to evaluate evidence in software engineering similarly to the way evidence is considered in legal proceedings.¹ He posited the following five levels of evidence.

Strong evidence is either a well-documented, controlled experiment with a representative sample from the intended population or a cross-company multicase study conducted by researchers with no vested interest in the study's results. The researchers should publish their work after peer review.

Evidence is a well-documented, controlled experiment with nonrepresentative subjects, a series of case studies from within a single company, or a well-documented study of a single case by a researcher with no vested interest in the results. The work should be peer-reviewed and presented at a conference or in a journal.

Circumstantial evidence is a well-documented, controlled experiment by anyone with a vested interest in the results, a study of a single case, or a well-conducted survey with a representative sample. The work should be peer-reviewed and presented at a conference or in a journal.

A *third-party claim* is an experience report, lessons learned, or a nonrepresentative survey by anyone with a vested interest. The work could be presented in any publication.

A *first- or second-party claim* is information published by the developer of, for example, the software tool being discussed or anyone with a vested interest in the results.

When the evidence is gathered and evaluated, the decision maker should take into account several aspects that might influence the final judgment. These include evidence strength, quality, and relevance; source reliability and impartiality; and the aging of evidence. This can be done in five steps:

1. identifying relevant sources of evidence,
2. extracting the data,
3. determining the number of data points from an evidence perspective,
4. judging the evidence's strength, and
5. evaluating for vested interests.

Reference

1. C. Wohlin, "An Evidence Profile for Software Engineering Research and Practice," *Perspectives on the Future of Software Engineering—Essays in Honor of Dieter Rombach*, Springer, 2013, pp. 145–157.

TABLE 1

Evidence sources

Article reference no.	Company	No. of sources	Type of work	Sourcing model	Sites	Result	Savings or losses	Basis for calculations
3	HP	1 collaboration	Remote customer support	Insourcing	US, Ireland, India	Achieved	10× savings compared to sending an engineer to the customer's site	N/A
		1 collaboration	Remote support work				Further 3× savings	
6	Nokia	3 projects	Basic-complexity projects	Outsourcing	Finland, US, Central and Eastern EU, India, China, others	Achieved	N/A	N/A
		6 projects	Moderate-complexity projects			Achieved		
		4 projects	Complex projects			Achieved		
		3 projects	Moderate-complexity projects			Not achieved		
		2 projects	Complex projects			Not achieved		
7	Various	Many	Telecom and automotive projects	N/A	N/A	Achieved	10–15% savings after 2–3 years	N/A
8	Schlumberger	2 projects	Complex and knowledge-intensive development tasks	Outsourcing	N/A	Not achieved	N/A	N/A
9	N/A	1 project	Development and maintenance of financial software	Outsourcing	2 sites in US, Ireland	Not achieved	N/A	N/A
10	Nokia	2 projects	Test automation	Outsourcing	Finland, India	Achieved	N/A	N/A
		1 project			Germany, China			
11	N/A	1 project	Improvement of a large legacy banking application	Insourcing	Finland, Eastern European country	Not achieved	N/A	N/A
12	N/A	1 collaboration	Complex system development	Outsourcing	Norway, India	Not achieved	N/A	N/A
13	N/A	19 projects	Agile projects	Various	Europe, Asia, the Americas	Achieved	>50% savings in 20% of projects, 25–50% savings in 60% of projects, and 10–25% savings in 20% of projects	N/A
		29 projects	Structured projects				>50% savings in 19% of projects, 25–50% savings in 31% of projects, 10–25% savings in 50% of projects	

TABLE 1 CONTINUED

Article reference no.	Company	No. of sources	Type of work	Sourcing model	Sites	Result	Savings or losses	Basis for calculations
14	Phillips	>200 projects	Consumer-electronics product software development	Various	Asia, Europe, India	Not achieved	2× to 3× costlier than collocated development	N/A
15	N/A	1 project	Continuation of payroll-management-system development	Outsourcing	Norway, Russia	Achieved	35–40% savings	N/A
16	N/A	1 project, phase 1	Application development and maintenance for a telecommunications carrier	Outsourcing	2 sites in US	Achieved	Dramatic savings	N/A
		1 project, phase 2			US, India	Not achieved	N/A	Salary comparisons
17	N/A	1 project	Embedded-software development	Outsourcing	US, India	Not achieved	N/A	N/A
		Several projects	Selected projects with well-defined deliverables		US, 2 sites in India, unclear location	Achieved	10–15% savings over onshore costs	
18	IBM	2 projects	Web application development with follow-the-sun approach	Insourcing	US, India	Not achieved	Significant decrease in gross profit	Net value and gross profit
		1 project					Significant net loss with positive gross profit	

which a project was outsourced (two versus three or more).

We also determined that none of the evidence provided in studies was reliable for determining whether GSE leads to cost savings. Although some studies looked at a large number of projects and collected data in different ways (surveys, interviews, and documentation analysis), the authors did not explicitly disclose actual cost savings. And none of them provided economic data that help reconstruct savings calculations, such as salary information, number of people employed, productivity data, and additional costs.

Achieving Cost Savings

Some researchers whose work we studied have associated big GSE cost

savings with relatively simple, basic projects⁶ and projects based on well-defined processes and deliverables that require little management.^{7,17}

Projects distributed among many sites demonstrated smaller cost benefits—perhaps 10 to 15 percent—achieved only after a two- to three-year developer learning curve.⁷ One study found that outsourcing individual processes—such as test automation—in a project decreased costs,¹⁰ although the authors admitted that such projects in their case faced minor schedule slippage, quality concerns, and hidden costs related to onshore support and the transition of development processes.

Unlike many other studies, our findings suggest that sometimes out-

sourcing complex development tasks can yield successful results and decrease costs.⁶

Not Achieving Cost Savings

Outsourcing complex projects—including those that require great expertise⁸ and those that are domain specific^{6,12} or highly technical—often doesn't save money. These types of projects include developing embedded software¹⁷ or evolving and maintaining legacy systems.^{9,11} As was the case with many other studies, our investigation found that the follow-the-sun development approach also didn't reduce costs due to a dramatic increase in overhead.¹⁸

Despite salary-based reductions on the surface, the failure to deliver working software turns the

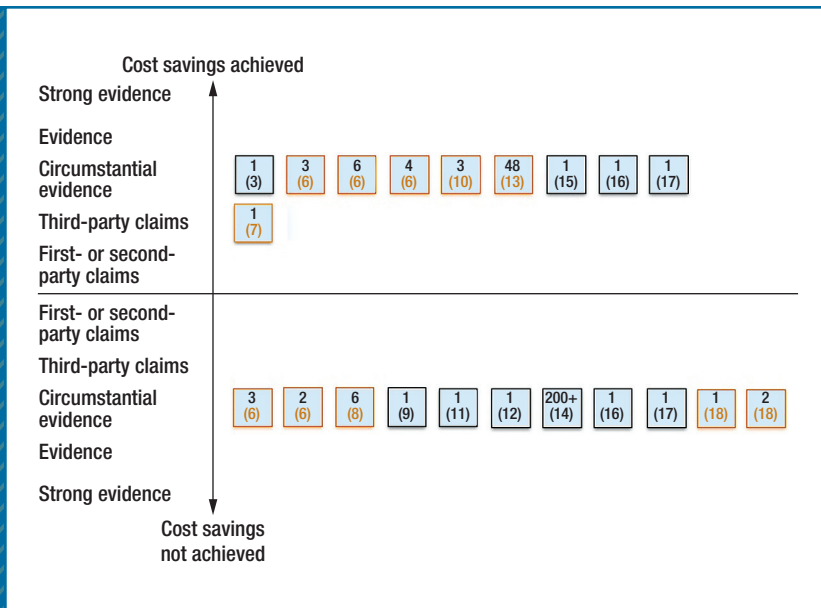


FIGURE 1. An evidence profile for cost savings in global software engineering. Each box indicates a study. The main number in each box is the number of data sources (projects, interviewees from different projects, or survey respondents). The number in parentheses is the number of the reference listed at the end of this article that refers to the project. The orange boxes indicate studies by authors who might have a vested interest in the results.

discussion of cost savings moot.^{6,16} And while certain initially troubled projects have recovered,⁸ high expectations of cost savings, especially on an immediate basis, have not always been met due to significant management overhead and longer-than-expected ramp-up.⁸

Recommendations

For companies considering offshoring, we recommend the following.

First, plan carefully. On top of typical project concerns, GSE adds its own risks, such as poor selection of outsourcing providers¹⁰ and high turnover at such businesses.¹²

Second, do not be carried away by salary levels alone. Many studies warn that GSE cost savings are less than expected.^{7,8} Strategies that allocate work based on outsourced

developers' availability and salary savings often result in imbalanced workloads, lower product quality, and the need to redo some work due to project complexity.¹⁹ GSE projects often involve hidden costs such as additional management and insufficient performance,⁹ project delivery failures,^{6,16} and lower-quality products requiring rework.^{12,19} Companies often must make significant investments in documentation, training, and onshore support to compensate for the initial gaps in overseas developers' specific knowledge and their lack of experience and understanding of the tasks.

Third, set clear goals. Offshore developers should be paid for results and not for the number of hours they work. One study reported that Malaysian engineers who were paid half

of what Irish engineers received per hour often lacked technical knowledge and extensive relevant experience, which significantly affected productivity.⁹ Similarly, a perceived upfront benefit of eightfold salary savings in using developers from India rather than the US was offset by additional project-coordination costs and lower productivity.³

Fourth, do not expect immediate cost savings. In GSE projects that result in savings, companies typically lose money first.¹ Overseas developers experience learning curves, and the projects often encounter problems before running smoothly. A company might not realize savings for two to four years.^{7,13} Thus, offshoring might not be appropriate for short-term projects or businesses that require immediate economic returns.

Finally, carefully calculate or, where necessary, estimate costs.⁴ Currently, however, no good models for this exist for offshore software development.³

The promise of cost savings via GSE is appealing. Many studies say outsourcing projects to countries with lower developer salaries will provide such benefits.⁹ However, our investigation has not found enough evidence to reach this conclusion.

In some cases, GSE doesn't work out. Success is not guaranteed, noted two leading industry managers who acted as jurors for the work presented here (see the "Voice of the Jury" sidebar).

Companies must perform critical risk assessment with due diligence to determine whether an offshoring project in progress is doomed to failure, no matter how much time and money have been



VOICE OF THE JURY

For the research described in the main article, two leading industry managers served as jurors judging global software engineering and its potential benefits. Their comments serve as a motivation for our verdict. Evidently, both experts agree that cost calculation is a complex context-dependent issue, which is of importance for industry.

Maria Larsson, head of software development and integration at Ericsson's Aachen R&D center:

My experience of offshoring to low-labor-cost countries and of whether cost savings are achieved is based on setup and time perspective. I have seen benefits from a cost perspective when a long-term partnership has been built up ... and areas of responsibility have been distributed to avoid coordination overhead and reduce dependencies. We have a tendency to underestimate cost that is not visible in the hourly rate such as attrition, increased need for travel, expertise, knowledge buildup, etc. On the other hand, values such as global representation and access to other markets are hard to put a price tag on.

Evaluating the evidence is complex. My conclusion from the evidence profile here is that the facts are hard to find, which supports my own experience.

Even if this study doesn't provide evidence of cost savings, I find this kind of research valuable for global companies like Ericsson to verify that we can benefit from our own and other companies' experience and to identify key areas to address when deciding whether to offshore to achieve cost savings.

Tormod Svensen, director of operations for QHSE (quality, health, safety, environment), integrity, and optimization at DNV GL Software:

Research on cost savings is important to enable businesses to make better decisions. One weakness with papers reporting on cost savings in this study is that they do not include any hard numbers and facts, which makes it difficult to learn from them.

Our business is about producing licenses that we sell worldwide. We also do bespoke development close to a single customer. For us to succeed, we need to be close to the market.

When we consider cost, we need to consider many factors, including the business model. We tried years ago to offshore pure coding. We spent so much time writing the specifications and verifying the resulting work that we really did not get any cost-efficiency gains.

From our first offshore attempt, we also realized that writing code is a small part of our total value chain. To understand cost calculation, we need to look at the total value chain, which is very complex.

Finally, salary is not the only consideration to be made. Coordination between sites is also a cost driver. The key things we have learned support the recommendations suggested in this paper, especially the recommendation of careful planning.

invested.¹² In that case, businesses might have to cut their losses and find a new partner or backsource their work for good. 🕒

Acknowledgments

The Swedish Knowledge Foundation funded this research under KK-Hög grant 20120200.

References

1. W. Burger, "Offshoring and Outsourcing to India," *Proc. 2nd IEEE Int'l Conf. Global Software Eng.* (ICGSE 07), 2007, pp. 173–176.
2. E. Carmel and P. Tjia, *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*, Cambridge Univ. Press, 2006, p. 282.
3. E. Ó Conchúir et al., "Exploring the Assumed Benefits of Global Software Development," *Proc. 1st Int'l Conf. Global Software Eng.* (ICGSE 06), 2006, pp. 159–168.
4. D. Šmite and R. van Solingen, "What's the True Hourly Cost of Offshoring?," to be published in *IEEE Software*; doi:10.1109/MS.2015.82.
5. C. Wohlin, "An Evidence Profile for Software Engineering Research and Practice," *Perspectives on the Future of Software Engineering—Essays in Honor of Dieter Rombach*, Springer, 2013, pp. 145–157.
6. T. Poikolainen and J. Paananen, "Perfor-

- mance Criteria in Inter-organizational Global Software Development Projects," *Proc. 2nd IEEE Int'l Conf. Global Software Eng. (ICGSE 07)*, 2007, pp. 60–70.
7. C. Ebert, "Optimizing Supplier Management in Global Software Engineering," *Proc. 2nd IEEE Int'l Conf. Global Software Eng. (ICGSE 07)*, 2007, pp. 177–185.
 8. M. Jensen et al., "Managing Offshore Outsourcing of Knowledge-Intensive Projects—a People-Centric Approach," *Proc. 2nd IEEE Int'l Conf. Global Software Eng. (ICGSE 07)*, 2007, pp. 186–196.
 9. V. Casey and I. Richardson, "The Impact of Fear on the Operation of Virtual Teams," *Proc. 3rd IEEE Int'l Conf. Global Software Eng. (ICGSE 08)*, 2008, pp. 163–172.
 10. I. Tervonen and T. Mustonen, "Offshoring Test Automation: Observations and Lessons Learned," *Proc. 4th IEEE Int'l Conf. Global Software Eng. (ICGSE 09)*, 2009, pp. 226–235.
 11. M. Paasivaara, "Coaching Global Software Development Projects," *Proc. 6th IEEE Int'l Conf. Global Software Eng. (ICGSE 11)*, 2011, pp. 84–93.
 12. N.B. Moe, D. Šmite, and G.K. Hanssen, "From Offshore Outsourcing to Offshore Insourcing: Three Stories," *Proc. 7th IEEE Int'l Conf. Global Software Eng. (ICGSE 12)*, 2012, pp. 1–10.
 13. H.-C. Estler et al., "Agile vs. Structured Distributed Software Development: A Case Study," *Proc. 7th IEEE Int'l Conf. Global Software Eng. (ICGSE 12)*, 2012, pp. 11–20.
 14. R. Kommeren and P. Parviainen, "Philips Experiences in Global Distributed Software Development," *Empirical Software Eng.*, vol. 12, no. 6, 2007, pp. 647–660.
 15. V. Imsland and S. Sahay, "Negotiating Knowledge: The Case of a Russian–Norwegian Software Outsourcing Project," *Scandinavian J. Information Systems*, vol. 17, no. 1, 2005, pp. 101–130.
 16. R. Mirani, "Procedural Coordination and Offshored Software Tasks: Lessons from Two Case Studies," *Information & Management*, vol. 44, no. 2, 2007, pp. 216–230.
 17. J.W. Rottman, "Successfully Outsourcing Embedded Software Development," *Computer*, vol. 39, no. 1, 2006, pp. 55–61.
 18. J.J. Treinen and S.L. Miller-Frost, "Following the Sun: Case Studies in Global Software Development," *IBM Systems J.*, vol. 45, no. 4, 2006, pp. 773–783.
 19. M. Cataldo and S. Nambiar, "Quality in Global Software Development Projects: A Closer Look at the Role of Distribution," *Proc. 4th IEEE Int'l Conf. Global Software Eng. (ICGSE 09)*, 2009, pp. 163–172.

DARJA ŠMITE is an associate professor of software engineering at the Blekinge Institute of Technology and a visiting professor at the University of Latvia. Contact her at darja.smite@bth.se.

FABIO CALEFATO is a senior research assistant in the University of Bari's Department of Informatics. Contact him at fabio.calefato@uniba.it.

CLAES WOHLIN is a professor of software engineering at the Blekinge Institute of Technology. Contact him at claes.wohlin@bth.se.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

38th International Conference on Software Engineering

ICSE 2016

Austin TX | May 14-22

<http://2016.icse-conferences.org>



The premier conference in software engineering is coming to the Texas state capital. Help shape ICSE 2016 by submitting your ideas and results to one or more of the many professional engagement tracks.

Submission Deadlines

Aug 28, 2015	Technical research papers
Oct 10, 2015	Workshop proposals
Oct 23, 2015	Software Engineering in Practice ♦ Software Engineering Education and Training ♦ Software Engineering in Society ♦ Visions of 2025 and Beyond
Nov 20, 2015	Doctoral Symposium ♦ ACM Student Research Competition ♦ Formal Demonstrations ♦ Technical briefing proposals
Jan 13, 2016	Posters
Jan 22, 2016	Workshop Papers

Join us in Austin to make the 38th ICSE one for the record books.

As the saying goes, "Everything is bigger in Texas!"

General Chair
Laura Dillon
Michigan State Univ., USA

Program Co-Chairs
Willem Visser
Stellenbosch Univ., South Africa

Laurie Williams
North Carolina State Univ., USA