Editor: **Tore Dybå**
SINTEF
tore.dyba@sintef.no

Editor: **Helen Sharp**
The Open University, London
h.c.sharp@open.ac.uk

A Whisper of Evidence in Global Software Engineering

Darja Šmite and Claes Wohlin

THERE'S HARDLY ANY large company that is not involved in globalization, and services to help smaller businesses capitalize on global resources are also emerging (for example, see "Outsourcing for Small Business" at www.freelancer.com). Global software engineering (GSE) has become a "normal" way of doing business.[1] Anecdotal claims about it abound, ranging from stories of tremendous success to those of total failure. In any case, the popularity of global collaboration, especially offshore development, continues to grow.[2]

In one study of global software development,[3] researchers found that companies expect it to reduce both development costs through lower salaries and development durations through "follow-the-sun" workflow scheduling. Companies also see new opportunities emerging from cross-site modularization of development work, access to a larger pool of skilled developers, shared best practices, and proximity to markets and customers. However, when the researchers took a closer look at the experience of three international software organizations, they found these benefits to be neither clear-cut nor guaranteed.[3] The results showed that global collaborations are risky, and the benefits only partly realized, if at all.

Nevertheless, the forces driving globalization are significant. Today, it's not just about cheaper and faster development but also about satisfying investment requirements imposed by governments in foreign markets.[2] These driving forces aren't expected to diminish in the near future.

Given the popularity of global collaboration and the growing interest in improving its outcome, we expected to see a large amount of research evidence to help practitioners understand the keys to success and reasons for failure. This motivated us to conduct a systematic review of the empirical GSE literature from 2000 to 2007.[4] After a thorough screening process, we identified 59 studies of acceptable rigor, credibility, and relevance to our investigation. These included 37 industry studies, 16 experiments performed with students, and 6 unclassified studies. (The list of studies is available as a Web addendum to this article at http://doi.ieeecomputersociety.org/10.1109/MS.2011.70.)

What Do We Know about GSE?

Literature reviews are expected to help reveal stories of both success and failure. To our disappointment, we could identify only a handful of clear success stories (10 percent) and even fewer clear

0740-7459/11/$26.00 © 2011 IEEE

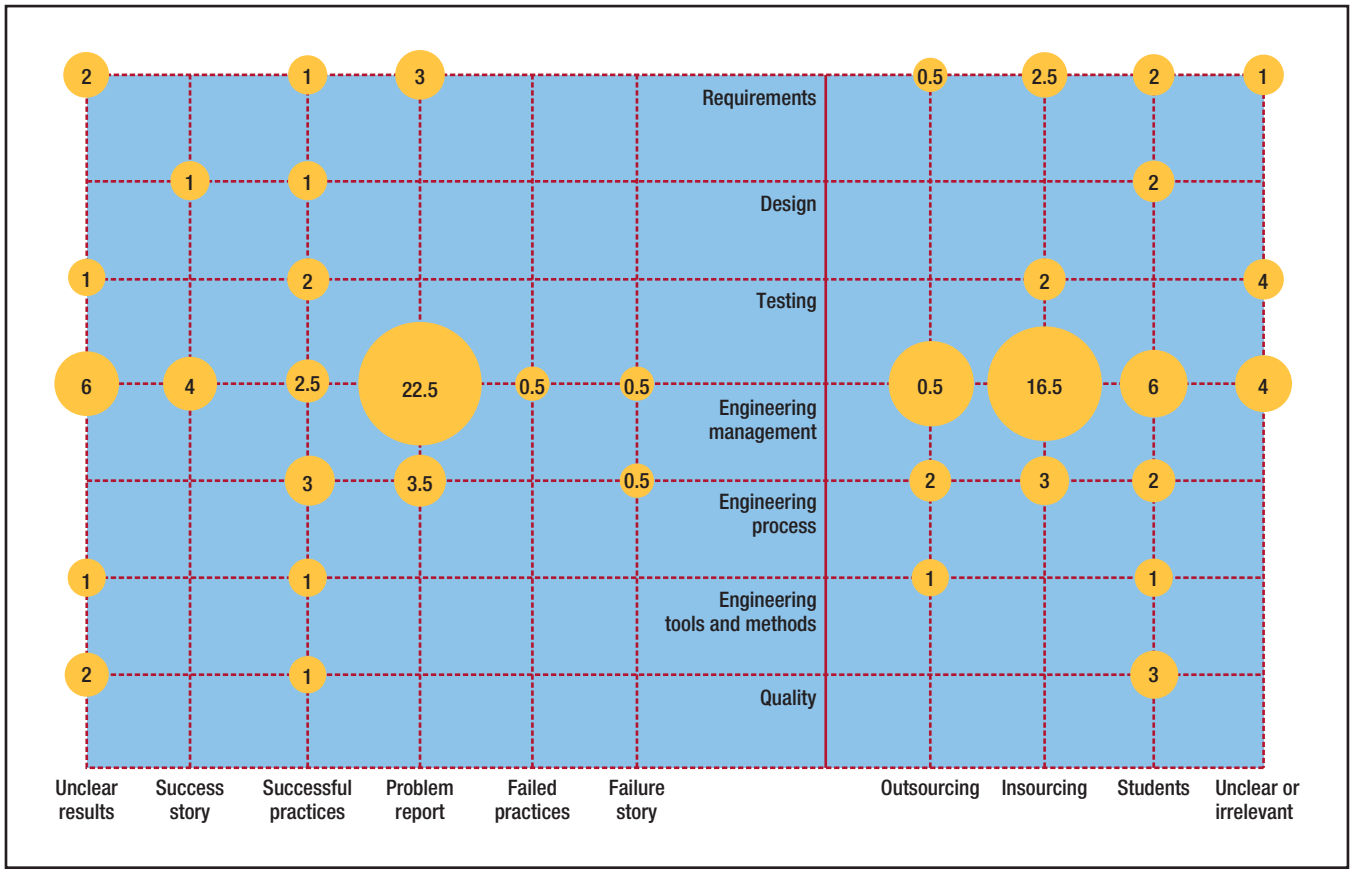

FIGURE 1. Bubble-plot overview of what we know about global software engineering (GSE). Results are based on a systematic review of the GSE literature available from 2000 to 2007.[4] The left side classifies the 59 relevant studies thematically in terms of success or failure, and the right side classifies them according to globalization type.

failures (3 percent) from the 59 studies we reviewed. One study referred to something that went wrong, and 25 percent referred to things that worked well.

We classified the majority of studies (54 percent) as problem reports. Most of them focused on general challenges of cross-border collaboration, not on problems related directly to a particular practice. The absence of success stories and proven solutions indicates that GSE has not yet matured, as does the repeated emphasis on the general challenges of global projects.

Because of the high percentage of studies with unclear results, we couldn't reach general conclusions regarding the links between success or failure and different project characteristics, such as the reasons a company decides to go global, the development methodologies selected, and the ways work is divided among collaborating locations. The research offered only a whisper of general evidence to help global endeavors. The experiences and lessons learned must be interpreted in context—that is, each global endeavor must judge the transferability of evidence to its context. Decisions to offshore/onshore or insource/outsource software development require careful evaluation of the business case (see the "Types of Global Software Engineering" sidebar).

The most popular focus among the studies was found to be managerial (20 studies) because they addressed different aspects of managing distributed collaborations. The majority of studies explored the challenges of working together, although some reported experiences solely from the supplier or contractor perspective. Few studies focused on a particular topic, practice, or development phase. From these studies, the most popular topics were requirements engineering, coordination and communication, and the application of agile processes.

Figure 1 presents an overview of the GSE empirical research. The bubble-plot diagram shows the number of studies devoted to a particular topic according to two classifications: success or failure and globalization type. Each classification space shows a distribu-

tion of 59 total studies. If we classified a study into two categories, we split the score in half.

Is It All about Costs?

Research has shown that, despite company claims to the contrary, reducing costs is the main driving force for offshoring.[5] Our investigation confirmed that costs were the main reason companies decided to start global collaboration (explicitly stated in nine studies). Next was a need for extra knowledge (mentioned in three studies) and extra people (mentioned in three studies). In 28 cases, the study mentioned no reason for starting a global project. In 19 cases, the question was irrelevant because the studies were noncommercial or student projects.

Because cost savings was a primary GSE rationale, we looked further for evidence of it. However, our analysis revealed a scarcity of evaluations of cost savings, investments, or returns on investments. Two detailed studies did show interesting interdependencies:

- An empirical project postmortem of duration, effort, staff, rework cycles, and number of reports and meetings suggested no significant cycle-time or cost differences between distributed and collocated work.[6] The context was a single massive maintenance project that involved both single-site and distributed work on loosely coupled components. The need to manage common knowledge on the project was minimal.
- An empirical investigation of detailed data from 42 completed projects in a large CMM Level 5 software service company suggested that work dispersion—even in high-maturity environments—has a significant effect on productivity and a harder-to-capture secondary effect on quality.[7]

The evidence from these two studies seems contradictory—one of success and one of failure with distributed work. We suggest that distributed collaboration results depend on the nature of the work—in this case, maintenance and development—and other factors, such as decoupled versus integrated tasks. This signifies the importance of distinguishing GSE project types.

In other more qualitative observations and summary results, the expected benefits were offset by factors such as a dramatic overload on local teams and consequently idle remote teams,[8] coordination and managerial overhead,[3,8] and productivity losses.[3] However, concrete figures that could help indicate the range of additional costs or losses weren't evident.

Global Projects Have Different Flavors

The studies showed geographic, temporal, and cultural "distances" complicating GSE[5,9] (see the sidebar on GSE terminology). Engineering culture or style also appears to differ significantly around the world.[10] It seems fair to assume that collaborations between different countries have unique flavors.

Companies initiating global collaborations must decide not only between insourcing versus outsourcing and nearshoring versus farshoring; they must also consider the scope of global projects. For example, there is some evidence that a follow-the-sun approach is inappropriate for complex development activities.[3] Furthermore, outsourced projects require special attention to mutual incentives when they involve collaboration between entities that are competitors in other contexts.[10]

We also found that the most frequently discussed experiences in the studies we reviewed related to insourcing between two sites of the same company. This could mean that outsourcing's early popularity in GSE has started trending toward insourcing and partnerships.[9]

Recommendations

Our literature review yielded five key recommendations:[4]

- Invest in face-to-face meetings, temporal collocation, and exchange visits.
- Invest in reliable infrastructure,

TYPES OF GLOBAL SOFTWARE ENGINEERING

Sourcing refers to collaboration forms; in general, there are two types:

- *Insourcing* involves company-internal collaboration.
- *Outsourcing* involves external third-party collaboration.

Shoring refers to distance or location of a collaboration site:

- *Onshoring* occurs in the same country.
- *Offshoring* occurs in a different country.
- *Nearshoring* occurs in a neighboring country.
- *Farshoring* occurs in a distant country.

Combinations of the terms are also widely used, such as offshore insourcing and nearshore outsourcing.

including a centralized repository, common configuration management tools, and rich communication media.
- Enable effective, frequent communication through synchronous interaction.
- Keep task dependencies across sites low by implementing decoupled architectural solutions.
- Implement short incremental development cycles for timely feedback loops.

Our review results show that suitable software projects, temporal proximity, and additional investments are required to make global projects work. Distance between sites clearly matters: geographic distances increase traveling costs, time differences lead to inconvenient working hours and the associated overhead, organizational differences lead to higher costs for achieving compatibility. Thus, "cheap and far" aren't always a good combination, and offshoring seldom brings immediate cost-saving benefits.

Our review of GSE research revealed more questions than answers, but we were able to formulate specific questions for practitioners to consider before starting their GSE endeavors. The calculation of true GSE cost savings is more complex than a simple comparison. Some losses are inevitable, and additional investments are required, although we weren't able to evaluate the range of these investments. Realistic cost-estimation models to support practitioners haven't yet emerged. Although we identified success stories focusing on practices in managing global software development, these stories answer questions of how to survive rather than how to succeed.

Our recommendations suggest steps practitioners can take to compensate for various aspects of distance within tight project budgets. Most importantly, we found that global environments are diverse, and the strategies for companies that start global collaborations neither are nor can be the same.

References

1. D. Damian and D. Moitra, "Guest Editors' Introduction: Global Software Development—How Far Have We Come?" *IEEE Software*, vol. 23, no. 5, 2006, pp. 17–19.
2. J.D. Herbsleb, "Global Software Engineering: The Future of Socio-technical Coordination," *Future of Software Engineering* (FOSE 07), IEEE CS Press, 2007, pp. 188–198.
3. E.Ó. Conchúir et al., "Global Software Development: Where Are the Benefits?" *Comm. ACM*, vol. 52, no. 8, 2009, pp. 127–131.
4. D. Šmite et al., "Empirical Evidence in Global Software Engineering: A Systematic Review," *Empirical Software Eng.*, vol. 15, no. 1, 2010, pp. 91–118.
5. E. Carmel and P. Tjia,*Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*, Cambridge Univ. Press, 2005.
6. A. Bianchi et al., "An Empirical Study of Distributed Software Maintenance," *Proc. Int'l Conf. Software Maintenance* (ICSM 02), IEEE CS Press, 2002, p. 103.
7. N. Ramasubbu and R.K. Balan, "Globally Distributed Software Development Project Performance: An Empirical Analysis," *Proc. 1st India Software Eng. Conf.* (ISEC 08), ACM Press, 2008, pp. 17–18.
8. 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.
9. S. Sahay, B. Nicholson, and S. Krishna, *Global IT Outsourcing: Software Development across Borders*, Cambridge Univ. Press, 2003.
10. J. Herbsleb, D.J. Paulish, and M. Bass, "Global Software Development at Siemens: Experience from Nine Projects," *Proc. Int'l Conf. Software Eng.* (ICSE 05), ACM Press, 2005, pp. 524–533.

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

CLAES WOHLIN is a professor of software engineering at Blekinge Institute of Technology and a professorial visiting fellow at the University of New South Wales.