Making Software Engineering Research Relevant

Sarah Beecham and Pádraig O'Leary, *Lero—the Irish Software Engineering Research Centre*

Sean Baker, *Incaplex*

Ita Richardson and John Noll, *Lero—the Irish Software Engineering Research Centre*

I t's probably no surprise that practitioners rarely look to academic literature for new and better ways to develop software. And why should they? Arguably, conducting such research isn't their job. Academic literature is vast and necessarily full of rigor. It often has a style that's hard to penetrate, even for audiences with years of postdoc research experience. As researchers working closely with industry, we want to try to meet industry halfway: we're confident that we have something to communicate to practitioners that, given the right platform, they could find beneficial and of practical use. Take for example the research area of global software engineering. A quick search of the IEEE Xplore digital library (http://ieeexplore.ieee.org) revealed more than 1,200 sources of information for the topic (see Table 1).

Because it's unlikely that practitioners would have the time to search for nuggets of information that fit their context by wading through pages and pages of material, we developed a decision support system (DSS) to act as a shortcut to project management support.[1] Our system aimed to capture recommended practices (on the basis of extensive practical research findings[2,3]) and provide users with a quick and easy way to find solutions to their specific development problems. As a starting point for delivering best practices via a DSS, we focused on the global software engineering (GSE) process. Having spent almost a decade working with distributed software development organizations, our team of researchers was well positioned to share lessons learned and recommended practices with the wider community.

Research and development is time-consuming and costly, so before investing resources in product development, we produced a prototype DSS[4] to test the market. Our initial goal was to explore practitioners' thoughts about GSE research and then gauge their interest in the idea of GSE-focused DSS. Our test group was an influential group of practitioners from organizations such as Google, KPMG, Microsoft, and Oracle, and we interviewed senior managers and project managers from these companies. We gained some interesting insight into what practitioners really want regarding support.

WHAT KIND OF SUPPORT ARE PRACTITIONERS LOOKING FOR?

Respondents' candid responses revealed that their organizations were struggling with the kind of GSE challenges that we've been actively researching over the past decade. Not surprisingly, they recounted problems in the following areas:

- culture—mismatched work ethics, languages, religions, and so on;
- communication overhead resulting from the need to communicate with more people;
- different time zones across sites;
- tool mismatch;
- vendor selection;
- sourcing skills; and
- task allocation.

Other recurring themes were members of the team being left out of relevant and timely conversations and the inability to roll out best practices across sites as intended. For those involved in outsourcing, concerns included dealing with vendor retention, the true cost of

outsourcing, and how to build a good supplier–vendor relationship.

So given that both researchers and practitioners recognize these problems and that research has solutions to match them, we asked practitioners in our study whether they were aware of such work. Participants indicated that though they perceive GSE research as potentially useful and that studying the subject would doubtlessly improve GSE performance, they didn't read articles on GSE. Perhaps a reason that practitioners don't read GSE domain–specific articles is that GSE is viewed as mainstream software engineering. This certainly seems the case here: many practitioners interviewed didn't view GSE as separate from general project management. Also, practitioners don't seem to want frameworks; they want patterns of context-specific help. Finally, although dissemination techniques need to be improved, that's not sufficient. Experience-based advice seems to trump all.

WHERE DO PRACTITIONERS REALLY GO FOR SUPPORT?

In addition to depending on their own experience to solve GSE problems, practitioners also consult books, blogs, colleagues, online video tutorials, forums, and short, one- to two-page experience reports. The main source of support comes from peers. Although participants acknowledged the academic literature and external consultancies, they never used them for GSE-related issues. Experience is a key factor to where practitioners go for help with their GSE issues, as noted by one participant: "We talk to other managers who run teams elsewhere in the world … that is where we get our advice." When practitioners can't speak to peers directly, they use resources such as blogs, wikis, and their corporate intranets. When we look at Table 1, which shows

Table 1. IEEE Xplore search results for publications on "global software engineering" published between January 1999 and February 2013.

Content type	Frequency
Conference and workshop proceedings	1,126
Journals and magazines	114
Books and e-books	6
Total	**1,246**

Table 2. Where practitioners go for global software engineering support.

Source	Example
Books	Publications on topics such as agile software development, GSE, outsourcing, and project management
Other practitioners (interactive networks)	LinkedIn, blogs, communities of practice, discussion forums, and peers
Web	Agile community Web and the World Wide Web
Non-GSE articles	Publications on topics such as project management
Vendor material	White papers
Intranet	Internal knowledge databases
White papers promoting book, service, or product	Book promotion article
Academic publications on GSE	None reported

where much of the GSE research is published, and compare it to where practitioners read about GSE solutions (summarized in Table 2), we see little overlap.

Robert Glass drew attention to this theory and practice divide in his 1996 inaugural Practical Programmer column in *Communications of the ACM*.[5] According to Glass, researchers simply didn't have the required experience to make their theories the solution of choice. At the time of writing, Glass didn't believe there was a convincing body of research in certain areas and that "theorists who fail to evaluate their ideas in a practical setting before advocating them are of particular concern." Since then, with more than 15 years of

research on GSE, Glass's point still seems to be valid; and although it would be rare to find a process solution or tool published in a top-tier conference or journal that hasn't gone through some form of validation, can we really expect practitioners to take a leap of faith and apply a theory that hasn't been proven in practice first?

MAKING RESEARCH ACCESSIBLE

Shari Lawrence Pfleeger clearly articulated the importance of relevance and contextualizing solutions when she noted that "practitioners, who are the audience for our evidence, must be able to understand our theories and findings in the context of their

The Why: Ensure research is relevant, reflects the needs of practice, and can confidently declare the reason for conducting it.

The What: Write shorter evidence-based papers using accessible, nonacademic language, where findings are validated to ensure their credibility. These studies need to include a detailed context (as companions to the theoretical, detailed, and academic work).

The Who: Researchers and practitioners should work more closely together, collaborating in both conducting and reporting research.

The Where: Researchers need to disseminate their work more widely—venture into the "gray" literature, and also use social networks, blogs, and wikis.

Figure 1. For researchers, a summary of considerations regarding potential application for practitioners.

work and values."[6] Ekrem Kocaguneli and his colleagues' first rule for researchers to communicate with practitioners is to report relevant results.[7] Researchers should also remember to recheck their findings and reflect on their significance. See Figure 1 for a summary of considerations we recommend.

Returning to GSE research as an example, we find that although researchers address relevant problems, that relevance is often buried in the details. Also, to address the question of whether the published solutions are reaching their intended audience, we can confidently say no. Why? Because research results often are inaccessible, lack credibility, and are irrelevant (see the sidebar "The Practice–Research Paradox").

Although our study sample is small and highly selective, these results are important because the participants are in project and senior management roles in a cross-section of organizations, and we suspect that they reflect the behavioral patterns of practitioners from other organizations. There is a caveat: we know there are many practitioners that do straddle the practice–research divide and play active roles in our community, chair our industry tracks, and give excellent presentations at our workshops and conferences, but

none of these practitioners formed part of our sample.

The message from our group of practitioners is clear. First, research studies need to include practitioners as authors as well as subjects, so they can speak with authority about the relevance of the results by virtue of having "skin in the game."

Next, results need to be presented in a way that is useful to practitioners. Research publications are necessary to establish the validity of methods and results, but practitioners need best practices, recipes, and patterns that address well-defined problems with tailorable solutions.

Finally, researchers need to engage practitioners in a dialogue beyond that of observer and subject or author and reader. Blogs and discussion forums provide opportunities for researchers and practitioners to engage in informal discussions. Creating such communities of practice should disabuse the false presumption that researchers are simply ivory tower eggheads. In fact, the idea of forming communities of practice came directly from one of our practitioner participants. Thus, there's definitely hope that together we can bridge the practice–research divide. **C**

References

1. S. Beecham, N. Carroll, and J. Noll, "A Decision Support System for Global Team Management: Expert Evaluation in REMIDI," *Proc. Int'l Workshop Tool Support Development and Management in Distributed Software Projects*, 2012, pp. 12–17.
2. S. Beecham et al., "Crafting a Global Teaming Model for Architectural Knowledge Management," *Proc. IEEE Int'l Conf. Global Software Eng.* (ICGSE 10), 2010, pp. 55–63.
3. I. Richardson et al., "A Process Framework for Global Software Engineering Teams," *Information and Software Technology*, vol. 54, no. 11, 2012, pp. 1175–1191.
4. S. Beecham et al., "Who Are We Doing Global Software Development Research For?" *Proc. Int'l Conf. Global Software Eng.* (ICGSE), 2013, pp. 41–50.
5. R. Glass, "The Relationship between Theory and Practice in Software Engineering," *Comm. ACM*, vol. 39, no. 11, 1996, pp. 11–13.
6. S.L. Pfleeger, "Albert Einstein and Empirical Software Engineering," *Computer*, vol. 32, no. 10, 1999, pp. 32–37.
7. E. Kocaguneli et al., "Distributed Development Considered Harmful? A Case Study of Reporting Research to Industry," *Proc. 35th Int'l Conf. Software Eng.* (ICSE 13), 2013, pp. 882–890.

Sarah Beecham is a research fellow at Lero—the Irish Software Engineering Research Centre. Contact her at sarah.beecham@lero.ie.

Pádraig O'Leary is a research fellow at ARCH—Applied Research for Connected Health at Lero—the Irish Software Engineering Research Centre. Contact him at padraig.oleary@lero.ie.

Sean Baker is a consultant at Incaplex Limited. Contact him at sean.patrick.baker@gmail.com.

Ita Richardson is a principal investigator at Lero—the Irish Software Engineering Research Centre. Contact her at ita.richardson@lero.ie.

John Noll is a research fellow at Lero—the Irish Software Engineering Research Centre. Contact him at john.noll@lero.ie.

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

THE PRACTICE-RESEARCH PARADOX

There are several arguments that illustrate the potential value that practitioners could receive from the research community:

- Problems mentioned by practitioners are well known and well studied by the research community.
- Many empirically based and validated solutions have been identified to address the problems raised by practitioners.
- Practitioners perceive that research is potentially valuable.

Despite these facts, none of the practitioners in our sample regularly look to academic literature for solutions. What are the reasons for this paradox?

The first is *accessibility*. Practitioners don't have time to read and digest academic publications to extract potentially relevant solutions to their specific problems. Also, practitioners must interpret these research results in the context of their own organizations.

The second is *credibility*. Robert Glass's assertion of the practice–research gap[1] is nicely reflected in practitioner preference for advice from people with "skin in the game."

Last is *relevance*. Academic publications are written to satisfy academic standards of scientific rigor and follow conventions appropriate for academic discourse. As such, even if the subject discussed is relevant, much of the content of a typical academic publication is not relevant to a manager seeking, for example, an introduction to global GSE issues or solutions to a specific problem.

Reference

1. R. Glass, "The Relationship between Theory and Practice in Software Engineering," *Comm. ACM*, vol. 39, no. 11, 1996, pp. 11–13.

Call for Articles

IEEE Software seeks practical, readable articles that will appeal to experts and nonexperts alike. The magazine aims to deliver reliable information to software developers and managers to help them stay on top of rapid technology change.

IEEE Software

Author guidelines:
www.computer.org/software/author.htm
Further details: software@computer.org
www.computer.org/software