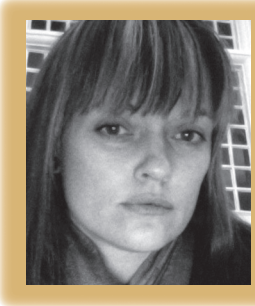
Features Editor: Dale Strok ■ dstrok@computer.org

The Top 10 Burning Research Questions from Practitioners

Sallyann Freudenberg and Helen Sharp

What research do software practitioners really want?

Software practitioners frequently complain that academic research doesn't meet their requirements or expectations—in short, that researchers are wasting their time. At the XP 2010 conference in Trondheim in June, this question was met



head-on in the context of agile software development in a panel entitled "Is Agile Research Dead in the Water?" The panel and audience identified a worrying disconnect between the research that is taking place and the practitioner community on which it's based and for which it's produced. Some of the issues raised included the somewhat difficult-to-digest way researchers sometimes write and present findings, the time lag between a question being of interest and the stringent research results being published, and the difficulty of gaining funding for relevant research in this area.

However, the main concern centered on the content of the research itself and whether the research community was finding answers to "the right questions"—particularly whether the concerns being addressed by the research community matched those of most interest and value to practitioners.

At the panel discussion, Sallyann suggested we apply agile techniques to the problem and use the conference audience to create a prioritized backlog (list) of research issues. So, during the plenary session the following morning, we asked the practitioners present to write down the most pressing question or issue that they'd like to be researched. We collected and displayed them on a notice board (see Figure 1) to let anyone interested vote on their favorites during the day. About 300 practitioners were at the conference, and we received about 60 different suggestions.

We then used the number of votes cast as a way to organize the issues into a prioritized list: the item with the most votes was at the top and those with no votes (although someone had obviously created them) at the bottom.

What Rose to the Top?

This backlog makes interesting reading. Some ideas were quite ambitious, and likely to have a complex and context-dependent response—for example, "using Kanban in globally distributed teams" and "how to adopt agile/lean." Other suggestions were specific to an issue practitioners face every day. Some were more readily researchable using traditional research methods, such as "what is the

 0740-7459/10/$26.00 © 2010 IEEE

chance of meeting team commitment based on your velocity of the last eight sprints?"

According to the number of votes practitioners cast for each research item (in parentheses), the top 10 were as follows ("*" indicates the item is abridged):

- Agile and large projects. (7)
- What factors can break self-organization? (6)
- Do teams really need to always be collocated to collaborate effectively?* (6)
- Architecture and agile—how much design is enough for different classes of problem?* (6)
- Hard facts on costs of distribution (in $, £, €, and so on). (5)
- The correlation between release length and success rate. (5)
- What metrics can we use with minimal side-effects? (5)
- Distributed agile and trust—what happens around 8–12 weeks?* (4)
- Statistics and data about how much money/time is saved by agile. (4)
- Sociological studies—what were the personalities in successful/failed agile teams?* (4)

There were also a number of common themes in the full list of suggestions: agile in distributed settings (note that three of the top 10 items focus on distributed teams); an economic or quality justification for adopting agile methods, for example, "what (if any) single practice in XP/agile has the highest ROI?"; scaling agile, for example, "how to scale small agile companies"; test-driven development, for example, "is TDDing a GUI viable or even desired?"; customers, for example, "how to get customers on board"; and gender, for example, "do women who pair program stay in the programming profession longer than women who code alone?"

From the top 10 topics on the list of prioritized research issues, it's clear that the agile practitioner community would like research to look at complex, multilayered questions. None of them are particularly "quick wins." If we are to continue using practices from the agile world to focus on agile research, then how can we use those same tools to get to the next step? Who will be our customers/product owners from the practitioner community, and how do we engage them and maintain that engagement? It is a common complaint from practice that academic research takes too long, so how can we refine this backlog of questions into readily researchable items with a quick feedback loop? What can academia realistically achieve alongside our practitioner partners? And most important, which sponsors will provide us with the vision and funding to proceed, particularly for those research areas that may not readily lend themselves to academic funding?

Figure 1. The notice board for people's votes. We ended up with about 60 burning research questions.

What Do You Think?

Do you agree with this Top 10? What would you have put up on the board, voted for, or challenged? And how do you propose we proceed from here? We invite you to share your comments at http://computingnow.computer.org/sw/ResearchQs. In addition, if you plan to attend Agile 2010 in Orlando, please come along to the research day (9 August) and contribute to the discussion.

Sallyann Freudenberg is an independent agile coach in the UK. She also holds a PhD in collaborative software development. Contact her at sallyann.freudenberg@googlemail.com.

Helen Sharp is professor of software engineering at the Open University, Milton Keynes, UK. Contact her at h.c.sharp@open.ac.uk.

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