Disseminating the Best Material to Practitioners

Phillip Laplante, Steve Counsell, and Giuliano Antoniol

AS WE MOVE forward through our second year of editing this department, we’d like to take this opportunity to restate its mission and scope of interest and invite appropriate articles. We’ll also look at some previous contributions.

The Mission
First, this department’s mission is to disseminate the best material that’s relevant to software practitioners. This mission is straightforward, but not always easily accomplished. Many other IEEE Software departments attract great, practitioner-oriented articles. We don’t aim to overlap with them and the regular focus and feature articles; rather, we seek niche, even emerging topics relevant to software engineering professionals—that is, practitioners. Most important, we aim to provide perspectives on topics we believe are starting to have (or will have) a real, lasting impact on how developers work and how projects are developed.

When we say “emerging topics,” it’s important to define what we mean. Many technologies might qualify, but we suggest looking at the IEEE CS 2022 Report as a guide.1 It was created by a panel of IEEE Computer Society technology leaders, led by 2014 president Dejan Milojčić. In a series of focused
meetings, they collaborated to identify industry advances that promised to change the world by 2022. They created a list of 23 technologies, many of which are of particular interest to software engineers. These included:

- big data and analytics,
- high-performance computing,
- cloud computing,
- the Internet of Things,
- software-defined networks,
- the open intellectual-property movement, and
- security cross-cutting issues.

Although the list was by no means exhaustive, the report has garnered tremendous attention, and its findings have had significant exposure in the mass media worldwide. Any of the 23 technologies would be an appropriate setting for Invited Content articles. And, although these are just emerging topics, we'd be interested in real-world experiences in those domains relevant to software practitioners—in particular, how technologies have been adopted, have been used, and have made an impact on practice.

We also note a change taking place in academic-research arenas. In many peer-reviewed venues, negative results have often been perceived as lacking in interest. However, there's increasing recognition that negative results have as much credibility and value as positive results. So, we solicit Invited Content articles that present “bad” (negative) experiences as well as “good” (positive) results; we feel that there’s much to learn from each type.

Whatever technology setting for the practitioner—emerging or established—we’re interested in the following types of articles:

- Articles that bridge the gap between research and practice. In the past, some people would have said that this gap has become a wide chasm. However, we believe a sea change is occurring as industry increasingly realizes the practical and economic value of academic applied research and as academia looks for closer ties with industry in the name of impact.
- Practice-focused roundtables of professionals and researchers. More initiatives are needed that bring academics and practitioners together in creative, productive ways. For example, one of us is running a project (funded by the UK Engineering and Physical Sciences Research Council Network) designed to do exactly that (visit www.farnet.org).
- Surveys and discussions of tools for the professional. Robust development tools, and especially the quality of their interfaces, are a frequent issue with developers. Academic research has pretty much failed to develop industry-strength tools or run trials on them, but industry itself doesn’t have the time to develop them.
- Articles that show how to improve the practice of software engineering.
- Articles that provide guidance on the direction of new and emerging topics.

**Some Accomplishments**

The following Invited Content articles illustrate what this department is about.

In the category of bridging the gap of research and practice, Christof Ebert and James Cain looked at the venerable Cyclomatic Complexity (CC) metric, highly criticized by academics but widely used by practitioners.2,3 (CC computes the number of linearly independent paths throughout a piece of code.) One goal was to understand why this was the case, not only for CC but also for other metrics, such as LOC. In particular, Cain posited that “industry looks for metrics that can be communicated between the different management levels and across the levels of the development stakeholders.” This article typifies what we’re looking for: articles that try to reconcile the differences between academic research and practice and promote discussion on the topic. Also, we’re not afraid of controversy!

Regarding practice-focused roundtables of professionals or researchers, in another article, nine experts at the Leaders of Tomorrow Symposium (at the 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering) discussed software engineering’s future.4 They described...
how they believed software engineering research would evolve and predicted “the rise of end-user programming, the monitoring of developers through neuroimaging and biometrics sensors, analysis of data from unstructured documents, the mining of mobile marketplaces, and changes to how we create and release software.” Other practice-based roundtables, perhaps focusing on the emerging topics discussed in the IEEE CS 2022 Report, would be highly desirable.

Regarding surveys and discussions of tools, Matt Gatrell described the state of a single-solution tool for recording and communicating project status. He discussed why a new approach was needed and introduced a new application-lifecycle-management tool. Clearly, industry needs to fully exploit the vast developer and user knowledge it has embedded in its repositories; Gatrell explained how to achieve this seamlessly. Of course, feeding that knowledge back into current processes (to provide a feedback loop) to improve them is an important feature of such tools.

Finally, in the category of improving the practice of software engineering, Bonita Sharif and her colleagues looked at how eye tracking could help software developers perform tasks in IDEs.

Of course, other possible topics exist. For example, with security, privacy, and safety being so timely, articles that explore the intersection of policy, politics, and legal issues from the software professional’s perspective would be most welcome.

Whatever your interest, if you think you can write an article that will interest our readers, please send us a query first. If we think your topic might be a good fit, we’ll guide you along the way. We’d like to close with a thank-you to all the contributors to this department to date.

References