Editor: **Philippe Kruchten**
University of British Columbia
pbk@ece.ubc.ca

Experience Report: It Will Never Work in Theory

Greg Wilson, Jorge Aranda, Michael Hoye, and Brittany Johnson

From the Editor

I am taking over the reins of the "Sounding Board" column after two decades of serving *IEEE Software* in areas ranging from the return on investment in the software business to automotive software. In "Sounding Board," we welcome eclectic provocative articles on subjects that are important to raise but may not get a hearing in other columns. But provocative doesn't mean frivolous or ungrounded. This issue's article is a great example of serious fact-based reporting undergirding an important message to the community. The authors report on a yearslong initiative, complete with numbers, trying to help bridge the gap between research and practice in software engineering. The article conveys their sense of frustration (and humor), accompanied by a set of recommendations and a call to action for our readership.—*John Favaro*

IN 2011, JORGE Aranda (then a graduate student at the University of Toronto) and Greg Wilson (who had recently left academia for the third time) began posting short reviews of software engineering research papers to It Will Never Work in Theory.[1] Both were frustrated by the gulf between what researchers studied and what practitioners in industry cared about and, conversely, by how little attention working programmers paid to results that were relevant and useful.

History

Aranda, Wilson, and a handful of other contributors posted almost 90 reviews over the next three years (Table 1). They selected papers that they believed practitioners were most likely to find interesting and/or actionable, that were not hidden behind the "Great Paywall of Academia," and that had a strong bias toward empirical studies (both quantitative and qualitative).

A handful of posts attracted half a dozen comments or a passing mention on social media, but neither

Table 1. The posts per year.	
Year	Posts
2011	43
2012	30
2013	16
2014	4
2015	3
2016	33
2021	96
2022	74
2023	82

of the communities the site hoped to reach engaged in any significant way. Posting continued sporadically in 2014–2015, and there was another burst of activity in 2016, but the project officially went on hiatus in December of that year. As the announcement at the time said, "In the wake of recent political events, our community's energy and attention should be focused on more important things."

Wilson revived the project in 2021, again focusing on open access papers that practitioners were most likely to find actionable. These reviews were shorter—sometimes only a paragraph—and led up to the first of three live events (Table 2). In collaboration with Michael Hoye (then at Mozilla) and Prof. Brittany Johnson (George Mason University), It Will Never Work in Theory hosted two sets of online lightning talks and one set co-located with the Strange Loop conference.[2] Speakers were given some coaching on the differences between speaking to academic and nonacademic audiences and then had 10 min to explain a research finding of interest to practitioners. Recordings of their talks were posted online, with transcripts in English and Spanish (the latter created by Yanina Bellini Saibene and her colleagues).

Readership on the site hovered around 50–100 visits per post (compared to 700–1,000 visits per day for another of the first author's projects that hasn't been updated in over a decade[3]). Each of the lightning talk videos was viewed by 150–1,000 people in the first 30 days after publication, but only a handful of those viewers ever reached out to the presenters.

Retrospective
This attempt to get researchers and practitioners to talk to each other has been personally rewarding but

has had no impact on software engineering's two solitudes.

Most software developers in industry have never heard of any findings more recent than *The Mythical Man-Month: Essays on Software Engineering*[4] (which few of them ever actually read) and routinely dismiss studies as "not statistically significant," even when those studies are carefully done and directly relevant to their work. When awareness of

research does seep into developers' conversations, it is usually as ammunition to support preconceived notions rather than any sincere attempt to improve their knowledge or practice of the art. For example, people who prefer strongly typed languages will broadcast the fact that some recent paper has proven they're better, without examining what the paper's authors actually mean by "proven" and "better."

Likewise, those researchers whose papers we reviewed and who presented at our lightning talks have

been no more likely to attend nonacademic conferences than they were before. Research has had some impact on developer tooling—for example, all of today's integrated developer environments draw on work in static analysis—but uptake has primarily been by individuals and in their individual work rather than at the larger scale of standardized or commonly understood practices that other fields would call "engineering."

> When awareness of research does seep into developers' conversations, it is usually as ammunition to support preconceived notions.

Recommendations
Twelve years after It Will Never Work in Theory launched, the real challenge in software engineering research is not what to do about ChatGPT or whatever else Silicon Valley is gushing about at the moment. Rather, it is how to get researchers to focus on problems that practitioners care about and practitioners to pay attention to what researchers discover. This was true when we started, it was true 10 years ago,[5] and it remains true today.

| Table 2. The speakers per event.		
Date	Format	Speakers
April 2022	Online	22
September 2022	In person	8
April 2023	Online	22

ABOUT THE AUTHORS

GREG WILSON is a senior software engineering manager at Deep Genomics, Toronto, ON M4L 2T9, Canada. Contact him at gvwilson@third-bit.com.

JORGE ARANDA is a principal software engineer at Workday, Victoria, BC V8T 3H2, Canada. Contact him at jorge.aranda@cuevano.ca.

MICHAEL HOYE is the director of Developer Engagement at Pluralsight, Toronto, ON M4C 3T7, Canada. Contact him at mike-hoye@pluralsight.com.
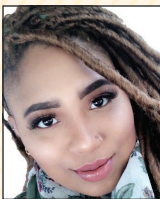
BRITTANY JOHNSON is an assistant professor in the Department of Computer Science, George Mason University, Fairfax, VA 20109 USA, where she directs the Interdisciplinary Software Practice Improvement Research and Development (INSPIRED) Lab. Contact her at johnsonb@gmu.edu.

as well). Crucially, students would be more likely to understand and value researchers' findings, having done a little research of their own.

The comedian W. C. Fields once said, "If at first you don't succeed, try, try again. Then quit. There's no point in being a damn fool about it." Thirteen years after our first post, it is clear that our attempts to bridge the gulf between research and practice haven't worked. We look forward to hearing what actionable plans others have that will find real support from both communities. 𝕾

Acknowledgment

References

1. It Will Never Work in Theory. Accessed: Jan. 2024. [Online]. Available: ttps://neverworkintheory.org/
2. Strange Loop. Accessed: Jan. 2024. [Online]. Available: https://thestrangeloop.com/
3. The Architecture of Open Source Applications. Accessed: Jan. 2024. [Online]. Available: https://aosabook.org/
4. F. P. Brooks Jr., *The Mythical Man-Month: Essays on Software Engineering.* Reading, MA, USA: Addison-Wesley, 1975.
5. A. Begel and T. Zimmermann, "Analyze this! 145 questions for data scientists in software engineering," in *Proc. 36th Int. Conf. Softw. Eng. (ICSE)*, 2014, pp. 12–23, doi: 10.1145/2568225.2568233.

We believe the best time and place to bridge this divide is when we have the attention of future researchers and practitioners, i.e., in undergraduate programs. After all, if students leave academia without having been exposed to both research methods and useful discoveries, why would those who leave look to researchers later for help or answers?

Software engineering faculty could, if they wanted, replace the team programming project course that most students do in their third or fourth year with one in which students design a small study or experiment, collect data, analyze them, and figure out what (if anything) they've proven. Such a class would not disrupt other curricula, would give students a chance to learn some practical data science, and would help prepare them for graduate school (which is in professors' own interests