



Belief and Evidence

How Software Engineers Form Their Opinions

Prem Devanbu, Thomas Zimmermann, and Christian Bird

Call for Submissions

Do you have a surprising result or industrial experience? Something that challenges decades of conventional thinking in software engineering? If so, email a one-paragraph synopsis to timm@ieee.org (use the subject line “REDIRECTIONS: Idea: your idea”). If that looks interesting, I’ll ask you to submit a 1,000- to 2,400-word article (in which each figure or table counts as 250 words) for review for the Redirections department. Please note: heresies are more than welcome (if supported by well-reasoned industrial experiences, case studies, or other empirical results).—*Tim Menzies*

AFTER PROGRAMMING FOR a combined 90 years, and having studied industrial and/or open source software development for more than 60 of those years, we’ve learned (and then sometimes unlearned) a great many things about our profession and colleagues. However, one observation has firmly endured: developers are stubborn, with passionately held beliefs. Most developers hold certain truths (about programming or software engineering) to be self-evident. However, “self-evident” here (unlike in the US Declaration of Independence) doesn’t refer to eternal truths: it refers to individual developers and their self-directed notion of what they “know” to be true.

You’ve surely heard many of these self-evident “truths.” For example, dynamic typing is inherently evil. Dynamic typing is the eternal salvation of humanity. Coding standards are soul-destroying and wasteful. Coding standards have prevented the downfall of Western (or Eastern) civilization. Assertions are for unit tests; injecting them elsewhere is mere superfluous sanctimony. Coding without prolific assertions is evidence of willful foolhardiness. You should hire the most experienced developers you can find. Experienced developers are rooted in old ways and lack creative energy. You probably hold some of these beliefs yourself ... and you know you’re right

and that those who think otherwise are just, well, misinformed.

Certainly, developers aren’t unique in this; it’s well known that changing people’s minds is difficult.¹ However, shouldn’t people expect more from developers? Aren’t we supposed to take a professional, empirical perspective on our practice? Shouldn’t we be observing what works and what doesn’t, and continually revising our opinions?

Sadly, over the many decades of our professional lives, we’ve rarely known this to be the case. Most engineers just seemed to know they were right; they pounced on data when it confirmed their biases and ignored it when it didn’t.

“Well, hang on,” you might say. “Isn’t this just your own bias speaking? Haven’t you just been remembering those few nasty developers who were stubborn, opinionated so-and-so’s and conveniently forgetting the virtuous majority of developers who are open-minded, always learning, and thoroughly evidence-based?”

Well, we always wondered that ourselves. So, we went meta and got all evidence-based regarding this, our own long-held, “self-evident” belief.

Gathering the Evidence

During the summer of 2015, Prem Devanbu visited Christian Bird and Thomas Zimmermann at Microsoft Research’s Empirical Software Engineering group. When discussing possible summer projects, Devanbu pitched this question as a subject worthy of investigation: Wouldn’t Microsoft want its developers’ practice to be evidence-based? If its developers’ beliefs weren’t evidence-based, then where did these beliefs come from? Did these beliefs arise from, and closely correspond with, actual data from the developers’ projects?

In the following, we summarize the results of investigating these questions. For more details, see the full paper.²

Early on, we had a series of discussions on how to explore these questions. We wanted to find out what people believed, why they believed it, and whether those beliefs corresponded with reality. We decided to pursue a triangulated³ experimental method, combining a survey with an observational study.

First, we designed the survey and selected the target audience. We aimed to compile a range of propositions—statements on which the respondents

could agree or disagree. In particular, we wanted propositions on which developers were likely to have informed opinions. That is, they were likely to have encountered pertinent evidence during their engineering practice. We also wanted to choose propositions for which we could actually gather evidence—that is, deploy an observational study based purely on archived project data.

Our survey included these types of propositions:

- Code quality (defect occurrence) depends on which programming language is used.
- Fixing defects is riskier (more likely to cause future defects) than adding new features.
- Geographically distributed teams produce code whose quality is just as good as that of teams that aren’t geographically distributed.
- When it comes to producing code with fewer defects, specific experience in the project matters more than overall programming experience.
- Stronger code ownership (fewer people owning a module or a file) leads to better code quality.
- Merge commits are buggier than other commits.
- Components with more unit tests have fewer customer-found defects.
- More defects are found in more complex code.
- Using assertions improves code quality.
- Using static-analysis tools improves quality.
- Coding standards help improve software quality.
- Code reviews improve software quality.

We were also interested where these opinions came from and how strongly the respondents held them. To that end, we asked the respondents to indicate the strength of their belief in each proposition, using a simple 5-point Likert scale (from strongly disagree to strongly agree). We then had the respondents select from six possible origins (Personal Experience, Peer Opinion, Mentor or Manager, Trade Journal, Research Paper, and Other) of their opinion and rank those origins by importance.

We also requested demographic data, including the respondents’ age, gender, number of years as a developer, number of years at Microsoft, title at Microsoft, geographic location, and highest level of schooling, and whether they had a supervisory role. To identify relevant repositories (for the corroborating observational studies), we also asked for the name of the project and the organizational (high-level) division. We maintained the respondents’ anonymity throughout. We sent the survey to approximately 2,500 Microsoft developers; we received 564 responses—a response rate of approximately 23 percent.

And the Survey Said ...

Unfortunately, there isn’t enough space here to reproduce the full story. One area we won’t go into much here is the level of disagreement between the respondents. The propositions listed earlier are arranged by the level of controversy. The first proposition aroused the most controversy; the last one virtually none. For more details, see the full paper.²

Here, we focus on two major issues that gave us concern: the opinion’s source and whether developers’

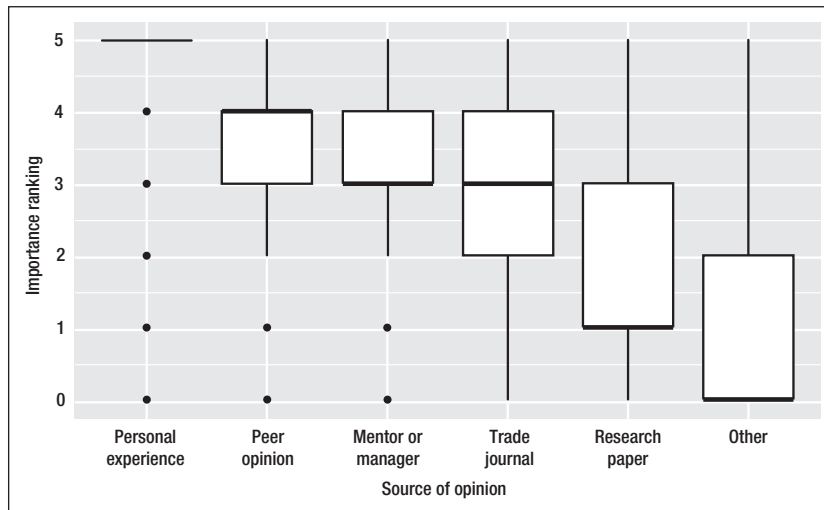


FIGURE 1. The survey respondents ranked the importance of their opinions' sources, quoted from our full paper.² We found no significant differences in these results across the demographic categories.

opinions were related to the observable phenomena in their projects.

The Opinion's Source

As we mentioned before, the developers chose the primary sources related to each proposition and ranked them. So, for each question–developer pair, we had a set of ranks, of cardinality 6, with values from 0 (lowest) to 5 (highest).

Figure 1 illustrates the results. Personal Experience ranked highest, then Peer Opinion, Mentor or Manager, Trade Journals, Research Papers, and Other. We found no significant differences in these results across the demographic categories.

These results weren't that surprising; earlier research in social science has found much the same thing: opinion influence preferentially flows along stronger social ties.⁴ But think on that a minute. Really? We, as empirical software engineering researchers, found this profoundly distressing. Our life's work, embodied in research papers, counted so little toward forming the

opinions of professional practitioners at one of the world's leading software companies?

Look at it another way. Suppose a doctor recommended some powerful opioid for a minor (but annoying) digestive problem. Then, when asked what the research was on the drug's effectiveness, he said, "Ah, I don't know about research! It just works well in my personal experience; trust me." That wouldn't be very reassuring.

However, these days, medicine offers an inspiring model. *Evidence-based medicine* (EBM) is relatively new,⁵ dating back to as recently as the early 1990s. Since then, EBM has caused a revolution in medical practice. Closer to home, Barbara Kitchenham and others have been advocating for evidence-based software engineering since 2004.⁶ Perhaps we will soon catch up to medicine.

The Relationship between Opinions and Evidence

For this investigation, we chose a highly controversial proposition: Geographically distributed teams

produce code whose quality (defect occurrence) is just as good as that of teams that are not geographically distributed.

We looked at two projects, which we'll call A and B. In the survey responses, developers working on A largely and strongly disagreed with this proposition, whereas developers working on B largely and strongly agreed with it. Surprisingly, both projects were widely distributed. Both had approximately 8,000 developers distributed across 100 buildings, and the repositories for both projects received commits from multiple buildings, campuses, and countries.

To quantify that distribution, we considered the percentage of files in each project that received at least 75 percent of their commits from one geographic area: building, city, region (for example, time zone), and country. The higher this percentage, the less distributed the development was. In project A, the percentages were 56 for building, 90 for city, 91 for region, and 92 for country. For B, the corresponding percentages were 76 for building, 80 for city, 83 for region, and 85 for country. Thus, B was slightly more distributed.

We then gathered defect data from both projects, based on the number of defect repairs in each file. We also gathered data per file on confounding predictive factors known to affect software quality, such as size, ownership, committer count, and commit count. Finally, we included a binary variable for each file's distribution, at each distribution level (building, city, and so on), indicating whether at least 75 percent of the commits to a file rolled up to that distribution level. After much careful modeling,² we found that the data supported some interesting interpretations.

First, in both projects, when we controlled for the confounding factors, the effect of distribution was statistically significant but very, very small. Prior research has found much the same situation.^{7,8} The effects were statistically significant only because the sample sizes were large (around half a million files). In most cases, the effect size was a small fraction of what's even considered a small effect. In both projects, it appeared to be very slightly better (for quality) to be in the same building; in project B, it appeared to be only slightly better to be in the same building. Notably, in all other cases, it appeared to be very slightly (but statistically) better to be in different regions or countries. Thus, it appeared that in Project B, developers' belief was consistent with the observable data, whereas in project A, it was the opposite.

We interpreted these findings as consistent with the survey. The developers' strong, prevalent opinions indeed appeared to be largely subjectively formed, rather than arising from the rigorous, evidence-based reasoning we'd like to see from professionals.

On one hand, these findings are distressing to researchers. We'd like to think our work is having an impact on development practice. Apparently, it isn't. On the other hand, we take these findings as inspiration. Empirical studies of software engineering are progressing in leaps and bounds, thanks to substantial investment in data gathering and analysis in some organizations, and thanks to great interest in academic circles due to the public availability of open source data. The rapidly growing body of people, ideas, and results suggests that much better times are ahead. Academics are introducing curricula that present the empirical

ABOUT THE AUTHORS



PREM DEVANBU is a professor in the Department of Computer Science at the University of California, Davis. Contact him at ptdevanbu@ucdavis.edu.



THOMAS ZIMMERMANN is a senior researcher at Microsoft Research. Contact him at tzimmer@microsoft.com.



CHRISTIAN BIRD is a senior researcher at Microsoft Research. Contact him at cbird@microsoft.com.

perspective early and often in software engineering courses. Practitioner magazines are increasingly paying attention. More and more organizations are getting savvy about data-based process improvement.

We hope that our research serves as a starting point, a rallying cry, to get software engineering practice more data-driven and evidence-based, and thus more effective in producing better software, cheaper and faster.

What do you think? Do you know of ways to increase the impact of empirical research in software engineering? Then join the conversation.

Write the next article for the Redirections department—after all, this is your community. 📧

References

1. C. Tan et al., "Winning Arguments: Interaction Dynamics and Persuasion Strategies in Good-Faith Online Discussions," *Proc. 25th Int'l Conf. World Wide Web (WWW 16)*, 2016, pp. 613–624; <https://dl.acm.org/citation.cfm?id=2883081>.
2. P. Devanbu, T. Zimmermann, and C. Bird, "Belief & Evidence in Empirical Software Engineering," *Proc. 38th Int'l Conf. Software Eng. (ICSE 16)*, 2016, pp. 108–119; <https://dl.acm.org/citation.cfm?id=2884812>.
3. F. Shull, J. Singer, and D.I.K. Sjøberg, *Guide to Advanced Empirical*

- Software Engineering*, Springer, 2008; <https://www.springer.com/us/book/9781848000438>.
4. J.J. Brown and P.H. Reingen, "Social Ties and Word-of-Mouth Referral Behavior," *J. Consumer Research*, vol. 14, no. 3, 1987, pp. 350–362; <https://www.jstor.org/stable/2489496>.
 5. "Evidence-Based Medicine: A New Approach to Teaching the Practice of Medicine," *J. Am. Medical Assoc.*, vol. 268, no. 17, 1992, pp. 2420–2425; <https://www.ncbi.nlm.nih.gov/pubmed/1404801>.
 6. B. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*, CRC Press, 2015; <https://www.crcpress.com/Evidence-Based-Software-Engineering-and-Systematic-Reviews/Kitchenham-Budgen-Brereton/p/book/9781482228656>.
 7. C. Bird et al., "Does Distributed Development Affect Software Quality? An Empirical Case Study of Windows Vista," *Comm. ACM*, vol. 52, no. 8, 2009, pp. 85–93; <https://cacm.acm.org/magazines/2009/8/34500>.
 - does-distributed-development -affect-software-quality/fulltext.
 8. E. Kocaguneli et al., "Distributed Development Considered Harmful?," *Proc. 35th Int'l Conf. Software Eng. (ICSE 13)*, 2013, pp. 882–890; <http://ieeexplore.ieee.org/document/6606637>.

myCS Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>

IEEE computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.

MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

OMBUDSMAN: Email ombudsman@computer.org.

COMPUTER SOCIETY WEBSITE: www.computer.org

EXECUTIVE COMMITTEE

President: Hironori Kasahara; **President-Elect:** Cecilia Metra; **Past President:** Jean-Luc Gaudiot; **First VP, Publications:** Greg T. Byrd; **Second VP, Secretary:** Dennis J. Frailey; **VP, Member & Geographic Activities:** Forrest Shull; **VP, Professional & Educational Activities:** Andy T. Chen; **VP, Standards Activities:** Jon Rosdahl; **VP, Technical & Conference Activities:** Hausi A. Müller; **2017-2018 IEEE Division VIII Director:** Dejan S. Milošević; **2018-2019 IEEE Division V Director:** John W. Walz; **2018 IEEE Director-Elect Division VIII Director-Elect:** Elizabeth L. Burd

BOARD OF GOVERNORS

Term Expiring 2018: Ann DeMarle, Sven Dietrich, Fred Douglass, Vladimir Getov, Bruce M. McMillin, Kunio Uchiyama, Stefano Zanero
Term Expiring 2019: Saurabh Bagchi, Leila De Florian, David S. Ebert, Jill I. Gostin, William Gropp, Sumi Helal, Avi Mendelson
Term Expiring 2020: Andy Chen, John Johnson, Sy-Ken Kuo, David Lomet, Dimitrios Serpanos, Forrest Shull, Hayato Yamana

EXECUTIVE STAFF

Executive Director: Melissa Russell; **Director, Governance & Associate Executive Director:** Anne Marie Kelly; **Director, Finance & Accounting:** Sunny Hwang; **Director, Information Technology & Services:** Sumit Kacker; **Director, Membership Development:** Eric Berkowitz

COMPUTER SOCIETY OFFICES

Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036-4924 **Phone:** +1 202 371 0101 • **Fax:** +1 202 728 9614 • **Email:** hq.ofc@computer.org
Alamitos: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720
Phone: +1 714 821 8380 • **Email:** help@computer.org

MEMBERSHIP & PUBLICATION ORDERS

Phone: +1 800 272 6657 • **Fax:** +1 714 821 4641 • **Email:** help@computer.org
Asia/Pacific: Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan • **Phone:** +81 3 3408 3118 • **Fax:** +81 3 3408 3553 • **Email:** tokyo.ofc@computer.org

IEEE BOARD OF DIRECTORS

President & CEO: James Jefferies; **President-Elect:** Jose M.F. Moura; **Past President:** Karen Bartleson; **Secretary:** William Walsh; **Treasurer:** Joseph V. Little; **Director & President, IEEE-USA:** Sandra "Candy" Robinson; **Director & President, Standards Association:** Forrest D. Wright; **Director & VP, Educational Activities:** Witold M. Kinsner; **Director & VP, Membership & Geographic Activities:** Martin Bastiaans; **Director & VP, Publication Services and Products:** Samir El-Ghazaly; **Director & VP, Technical Activities:** Susan "Kathy" Land; **Director & Delegate Division V:** John W. Walz; **Director & Delegate Division VIII:** Dejan S. Milošević

revised 03 July 2018

