DOI:10.1145/2076450.2076452

Credit Non-Anonymous Reviewers with a Name

I AGREE WITH Bertrand Meyer's blog "Fixing the Process of Computer Science Refereeing" (Nov. 2011) and "Why I Sign My Reviews" (http://se.ethz.ch/~meyer/publications/online/whysign/) in favor of open reviewing but suggest we go further with the quality of refereeing by rewarding reviewers and encouraging their contribution. Reviewing papers and grant proposals is part of academic life but receives no reward in the publish-or-perish culture, yet writing a good review requires thought and time. Perhaps Meyer's open reviewing would mean fewer reviews, as referees could be reluctant to take on reviewing work for fear of (inadvertently) writing a low-quality review.

A simple response is to credit attributed reviewers. Though their contribution is relatively limited, it is vital, and acknowledging it publicly by attaching their names to a published work is a way to acknowledge a referee's place in the scientific community.

Indeed, Meyer said "Even honest people will produce bad-quality reviews out of negligence, laziness or lack of time because they know they will not be challenged." Giving reviewers credit would be a carrot rather than a stick, assuming, of course, academic management recognizes the need for referees and rewards their effort.

Moreover, inexperienced academics must learn to review just as they learn other research practices. Editors and program committee chairs play a vital role in the process of challenging and guiding referees to produce better reviews. More important, a better quality of scientific debate would likely prevail.

Most, if not all, academics have received reviews that were constructive and helpful, though they cannot easily contact anonymous reviewers to continue the discussion. Open reviews enable that discussion, leading to more valuable work in the future.

Phil Brooke, Middlesbrough, U.K.

Bertrand Meyer's blog (Nov. 2011) argued passionately for non-anonymous reviews, an idea that may sound revolutionary to computer scientists, proposing to change the very way science is done, but in the context of science in general is not radical at all. Computer scientists with experience in interdisciplinary collaboration know that, in many areas of science, non-anonymous reviews are the norm. For example, among geologists, it is up to reviewers to disclose their names to authors, and about half the time, they do. In spite of this non-anonymity, many reviews are still harsh, and, at least in good-quality journals, the quality of accepted papers is equally high.

Vladik Kreinovich, El Paso, TX

What Liability for Faulty Software?

It was great to read advocacy of software liability laws, as in Poul-Henning Kamp's article "The Software Industry Is the Problem" (Nov. 2011) but a pity that Kamp's arguments were so frivolous and unrealistic. Whether one creates the code oneself is irrelevant; programmers frequently find bugs in their own code. Gödel's theorem is also irrelevant. The right to disable unwanted code could be enjoyed by only a tiny percentage of consumers and doesn't meet anybody's needs. Consumers don't need code to be disabled; they need it fixed.

It is a disgrace that someone buying a software product gets only a warranty for the media but nothing for the software itself and no remedy even if the software fails to launch. It is a disgrace that a software product can crash while reading its own preference files because they were corrupted by a previous crash. It is even a disgrace when installers cannot set file permissions correctly (one of my personal bugbears). Software companies have become lazy because their customers have no legal rights, and, in many cases, their products have no significant competition. Please let's have a serious, substantive proposal for warranties and liability laws covering software products.

Lawrence C Paulson,
Cambridge, England

Author's Response:

I would support such a proposal, but it would totally pull out the economic rug, so, in addition to the lobbyists from the software industry, all economists would be against it. Good luck with that. My proposal leaves the economy intact, provides transparency and remedies for users, and creates a market for software-audit consulting that economists might even call job creation. Not ideal, but at least not impossible.

Poul-Henning Kamp,
Stagelse, Denmark

The Jobs Factor

Thank you to Michael A. Cusumano for his Viewpoint "The Legacy of Steve Jobs" (Dec. 2011). Before exploring that legacy, I'd like to express another view of why Microsoft DOS and later Windows became the dominant "platform" despite Apple's superior Macintosh "product." Microsoft platform dominance was a legacy of the "IBM factor" that said: "Nobody ever got fired for buying IBM."

Anyone who worked for a non-IBM vendor, 1960s–1980s, was continually thwarted by it, particularly mainframe vendor Burroughs, with a far superior mainframe, the B5000, guided by software concepts that are still with us today in Apple (via Alan Kay, a student of the B5000 designer, Bob Barton), including virtual machines (such as JVM), virtual memory, and combined software and hardware design leading to systems software written exclusively in high-level languages (such as ALGOL). The IBM factor was far stronger than even the "dominant platform" effect, and inherited by Microsoft from the lumbering IBM. Burroughs was extremely open, distributing the source of its software; following the theory of openness, Burroughs should have won.

Jobs stood against the resistance of those who were too ready to compromise, using "engineering" to justify obscurity while speaking in terms of megacycles and megabits. To those stuck in this technical rut, Jobs declared "Think Different," changing computing's focus to: "Yes, but what can computers do for me?," meaning the customer, rather than technologist or IT manager. The power-breaking effect explains the disdain for Apple by many people and earlier failure to accept Barton's B5000 concepts, which would have changed the focus to designing hardware to support software, but that battle was lost to the IBM factor, as well as being too far ahead of its time.

Jobs elevated design above technology, reversing the constraints of engineering compromise that puts technical specifications before design. Burroughs followed this ethic, and any serious student of computing should explore the resulting machines, as well as their direct and indirect descendants. Being far ahead in the computer industry usually does not pay off. Jobs and Apple understood this but were unwilling to compromise the principle of design over technical specification. Specifications are important only for enabling the possibilities of design. Specifications are not an end themselves, and changing this perspective is, perhaps, Jobs's greatest legacy. He also had the right no-nonsense, acerbic personality to see it through.

Jobs broke the power of IT managers, putting users and customers first, which should indeed be the foremost management paradigm of the 21st-century corporation: "Manage without management."[1]

The IBM factor could not last, as indeed it did not for IBM and is now breaking down for Microsoft. While others embraced such a vision before Jobs, the Jobs legacy is the breakdown of false power and longevity of good design. However, if Apple ever falls into making mediocre products (like IBM, with its 360, and Microsoft, with DOS and Windows), depending solely on reputation, I hope the day never comes when one could be "fired for not buying Apple."

Ian Joyner, Sydney, Australia

Reference

1. Koch, R. and Godden, I. *Managing Without Management: A Post-Management Manifesto for Business Simplicity.* Nicholas Brealey Publishing, 1999.

Give Me Competent Communication

Moshe Y. Vardi's Editor's Letter "Are You Talking to Me?" (Sept. 2011) really spoke to me. I have been frustrated by conference presentations that promise so much in their titles and abstracts but get lost in the presenter's delivery. If it is "too intrusive" for conference organizers to require video drafts of presentations, then possibly suggesting preference will be given to presenters who first document their presentation skills training, through, say, a course on public speaking, Toastmasters International membership, or other supporting details. Toastmasters can lead to an initial certification as a "competent communicator." One would hope competent communication is a basic goal of every presenter at every technical conference.

August Schau, Lewiston, ME

I could not agree more with Moshe Y. Vardi's Editor's Letter (Sept. 2011). Those who stand before a technical conference, especially a prestigious one, should appreciate they are taking part in a theatrical performance.

The ACM SIGPLAN conference on the History of Programming Languages held in Cambridge, MA, in 1993 was one such event. My written paper was long, with copious examples and all sources meticulously referenced. My spoken presentation was an entirely separate project. I edited over and over, presenting it to my tape recorder many times, then checking the timing. My script was annotated with the elapsed time (to within 10 seconds) where I should be at each stage. I indicated where I intended to give particular emphasis to some point and inserted stage directions to guide my presentation of each slide.

Those slides were brief, except where I deliberately intended to exemplify the stupidity of some complicated verbiage or indicate the style of some text rather than its content. The aide operating the projector had a copy of my script, so if the slides got out of synch (which they nearly did at one point) he could easily get us back on track.

At one point, after describing a particularly important issue, I put up the slide with the result of the vote, then stood back and said nothing, letting the audience absorb the consequences of the entirely unexpected result, which they duly did.

My stopwatch sat before me so I could check my progress. Thus when the increasingly agitated chairman passed me warning notes "5 mins left," "3 minutes," "1 MINUTE!" I happily ignored him and carried on. I reached my denouement with fully 10 seconds to spare. The response from the audience clearly showed my message had indeed got across.

I was followed on the podium by Niklaus Wirth describing his language PASCAL. He had a fine paper to present but no separate presentation so just read the paper as submitted, with the difference obvious to all.

During the post-presentation Q&A, Doug Ross of SofTech Inc. stood up and complained about my omission of his stance on some particular issue, but I was able later to show him in the main paper (which he had not read) the corresponding text describing that very matter, in slightly more detail, including reference to his dissenting view.

This was all an effort to put into a presentation but a matter of passionate concern to me, so I was happy to do it. I didn't do all my presentations this way but always viewed them as theatrical performance, aimed at helping people understand my material.

Charles H. Lindsey, Cheadle, U.K.

Communications welcomes your opinion. To submit a Letter to the Editor, please limit yourself to 500 words or less, and send to letters@cacm.acm.org.