



The *Communications* website, <http://cacm.acm.org>, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.



Follow us on Twitter at <http://twitter.com/blogCACM>

DOI:10.1145/3453647

<http://cacm.acm.org/blogs/blog-cacm>

Teaching Other Teachers How to Teach CS Better

Mark Guzdial shares how he assesses the efforts of other computer science teachers.



Mark Guzdial
How I Evaluate a College Computer Science Teaching Record

<http://bit.ly/3kCmzJY>

February 12, 2021

If you spend enough years in academia, you get labelled “senior,” and you find yourself spending a significant amount of time reviewing other academics’ records. Between preparing letters for tenure and promotion cases, and serving on committees at my home institutions, I have read a lot of materials about teaching, research, and service. I evaluate teaching records differently than most of my colleagues. There is a method to what I am doing, and I am sharing it here.

An academic’s teaching recording typically has a personal teaching statement that describes their philosophy of teaching and how they teach their classes. It almost always includes student evaluations of teaching—rankings by students on statements like “This was a good instructor” where 1 might be “strongly disagree” and 5 might be

“strongly agree.” Sometimes the record includes letters from past students, or comments from a colleague who observed classes by the teacher being reviewed. There might also be descriptions of classes created, which is wonderful to do, but it is a different contribution and requires a different skillset than classroom teaching.

I don’t find much value in student evaluations of teaching. First, students do not know what helps them learn the most. From a 2013 study of electrical engineering students using problem-based learning (<https://bit.ly/2NVazYb>) to the 2019 Harvard study of active learning (<https://bit.ly/3uQnXh4>), evidence suggests that students do not know what is best for them. They prefer, and rate more highly, learning opportunities which are easier and less effective. Second, students have implicit bias (<http://bit.ly/3e3UzOe>), such that a white male computer science teacher will tend to get higher student evaluations than colleagues of different races or gender with comparable teaching quality. Students’ written comments in evalu-

ations give me some insights, in the sense of issues to highlight or to be concerned about. But in large classes, the same teaching features may get identified as both positives and negatives. It is hard to find a valuable signal in a large number of anonymous comments. I find student letters to be more valuable. Concerns raised in a signed letter are more likely to be significant.

Some of the common statements in a teaching evaluation have relatively little value for me in identifying teaching quality or skill. A letter writer or peer reviewer might say that the academic “cares about students.” I hope that’s a low bar. It should be the case that everyone teaching cares about students—it is a requisite for the job, not an indication of particular quality. I feel similarly about “has clear and understandable diction.”

Other common statements in teaching statements actually raise concern for me. “Use of active learning” is exactly the kind of statement I’m looking for, unless that’s equated with “is open to student questions” or “encourages class dialogue.” That’s not what I

mean by active learning, and that is not what is meant in the research on active learning methods. I worry when I read that a teacher “is available at any time for students” and “spends many hours meeting students one-on-one in office hours.” Frankly, that suggests to me that the teacher may have poor instructional design skills. Homework in a CS course should be able to be completed based on course content: lectures, discussion, lab, textbook materials, and so on. If the homework problem is so hard that it requires a visit to office hours for most students to complete, then it's just too hard and that's unfair to students. First-generation students are less likely to go to office hours (<http://wapo.st/3dYX8kJ>), so it puts them at a disadvantage. If a course is three credit-hours, but students have to go to office hours to finish the assignments, it's actually a four credit-hour course, and the teacher is demanding too much from the students.

My favorite paper on evaluating undergraduate teaching is Carl Wieman's article *A better way to evaluate undergraduate teaching* (<https://cwsei.ubc.ca/resources/tools/tpi>). Wieman (https://en.wikipedia.org/wiki/Carl_Wieman), a Nobel Prize laureate in Physics and former Associate Director of Science in The White House Office of Science and Technology Policy, is now an Education and Physics Professor at Stanford University. He argues against using student evaluations of teaching for similar reasons to me. He also argues against using peer evaluation of teachers. In general, teachers at the undergraduate level are not taught how to identify high-quality teaching, and likely don't know what they're looking for.

He recommends creating an inventory of teaching practices, and computing an “ETP score”—the Extent of use of research-based Teaching Practices. Teachers who use more research-based teaching methods are measurably more effective than teachers who do not. Use of research-based teaching methods makes teaching more effective (for example, the results on peer instruction in computer science classes (<http://peerinstruction4cs.com/>) are amazingly strong), and a teacher who seeks out research-based teaching methods likely has the characteristics of a good teacher—seeking out meth-

If the homework problem is so hard that it requires a visit to office hours for most students to complete, then it's just too hard and that's unfair to students.

ods that serve students' needs and is humble enough to recognize that their practice could be better.

I don't compute a score, but I read computer science teaching records actively looking for evidence of seeking and using research-based teaching practices.

- Any mention of peer instruction, POGIL (Process Oriented Guided Inquiry learning), or even “think-pair-share” counts as active learning for me. I'm looking for activities that engage *all* students (not just those who volunteer) in reflection and collaboration.

- Use of live coding in class is a good sign. Asking students to make predictions before executing code is a *great* sign.

- Talking about research-based measures of student success and retention, like self-efficacy, belonging, and intent-to-persist tells me that this is a researcher who is keeping themselves informed about what matters in CS education.

- I appreciate teachers who recognize how computing education can go wrong, and offer ways to avoid that. Computing classes often have a defensive climate (<https://bit.ly/3qizsdH>), and students get messages that they don't belong (<https://bit.ly/3sEBsOT>). I look for practices that aim to increase student's sense of belonging, such as strategies for preventing “show-off” questions in class.

- Sometimes, it's the student letters that say things like, “Teacher X asks us a question, and if we don't get it right, explains it over again.” Use of formative evaluation is a huge positive indica-

tor. (Formative evaluation is hot in the research literature, with a new paper on formative assessment in K–12 CS (<https://bit.ly/3sHAAJd>) at SIGCSE 2021 and a new paper in the *Computer Science Education* journal on effective design of formative feedback (<https://bit.ly/3qaF8Gz>).)

- Talking about building tools for students and *talking about evaluating them* (especially noting what *did not* work) is another huge positive mark for me. While that might not be using a research-based teaching method, that's treating teaching innovation as action research (<http://bit.ly/3efbyxl>). (Any *publications* in education conferences or journals is similarly a big sign for me in a positive direction.)

- Citing any papers from ACM SIGCSE conferences or from IEEE RE-SPECT is an automatic indicator to me of a teacher who looks for better practices and, particularly for RESPECT, is striving to broaden participation in computing.

- While it's very rare, I like to see teachers in undergraduate courses adapting practices drawn from K–12 CS. I learn a lot of tips and tricks from AP CS A teachers and from my daughter, a 9th grade science teacher. Adaptation from K–12 tells me that the teacher is seeking good practices, and is willing to innovate in their teaching.

In computer science, we often have a model of teaching that is heavily focused on content matter knowledge—if you know your CS, you'll likely be a good teacher. But all our evidence about quality teaching says that that's wrong. There's a lot of knowledge involved in teaching computer science well, and perhaps the most important is *pedagogical content knowledge* (see Chapter 7 in *How People Learn* at <https://www.nap.edu/read/9853/chapter/11>). Teaching is a skill that depends on learned methods. Those who use the best methods tend to teach the best. I value CS teaching that uses the best methods.

My thanks to Leo Porter, Beth Simon, and R. Benjamin Shapiro, who reviewed an early version of this post.

Mark Guzdial is professor of electrical engineering and computer science in the College of Engineering, and professor of information in the School of Information, of the University of Michigan.

© 2021 ACM 0001-0782/21/5 \$15.00