Pursuing the Vision of CS for All: Views from the Front Lines

By Carol Tate, Julie Remold, and Marie Bienkowski,
SRI International

Headlines in the popular press like “How Silicon Valley Pushed Coding into American Classrooms” [8] and “Who Benefits from the Push to Teach Every Kid to Code? Tech Companies for One” [7] characterize computer science education initiatives as strictly top-down private-sector efforts to get public schools to prepare their future workforce. In our work, which includes field research with educators bringing computer science to K-12 settings, we are hearing a very different story about teachers advocating—sometimes doggedly—for computer science instruction. Their motivations have less to do with providing workers for the tech industry than with promoting equity and personal empowerment in the digital age. The voices of these teachers provide insight on current drivers and barriers to adoption of computer science in K-12.

INTRODUCTION

As consensus builds in the United States around the importance of providing computer science experiences for all students, various projects have launched with the goal of introducing computing to student populations who are underrepresented in the workforce. Funding has come from U.S. federal agencies (most notably the National Science Foundation), philanthropies, and technology companies. To learn about the landscape for K-12 computer science education, Google for Education and Gallup, Inc. have partnered to conduct surveys of teachers, parents, principals, and superintendents. Their results [2,4,5,6,9] reveal a diversity of viewpoints, even within these specific roles [9]. This national survey work identified supports and hindrances to expanded computer science opportunities in schools, but the picture is incomplete without in-depth, qualitative data to shed light on the context and the mechanisms behind the identified trends.

SRI Education (a Division of SRI International) has been researching K-12 computing interventions for many years in

various contexts. We are currently involved in several research and evaluation projects related to scaling access to computer science in middle and high schools in multifaceted ways. In this work, we have heard, first-hand, about the ambitions and frustrations of teachers and education leaders working to bring introductory computer science courses to schools where it has not been previously offered. Through teacher surveys, in-depth interviews with teachers and school leaders, and observations of teacher professional development, we have learned about the challenges these teachers encounter. Our data, while preliminary, does offer insight into some of the contextual factors that affect uptake of these introductory courses. More importantly, it refutes the common perception of ‘top down’ reform and brings into sharp relief the role teachers are playing as champions for computer science in their schools.

The details we report come from projects (listed below in Table 1) that take place in diverse settings, including urban, rural, and reservation lands, and include both public and charter schools. As part of our work on these projects, we have interviewed teachers throughout the school year, observed professional development, and interviewed administrators.

Table 1: Projects that provided data for our work.**Project Overviews*****Computer Science in Secondary Schools (CS3, NSF award number 1418149)***

CS3 investigates the impact of the introductory Exploring Computer Science course on student computational thinking outcomes. Our work on this project includes student assessments, teacher surveys, interviews, and classroom observations in participating classrooms during the 2016-17 and 2017-18 academic years.

Prioritizing & Expanding Access to Computer Science Instruction in High-Needs High Schools (evaluator for NSF award number 1418149)

This project leverages Teach for America's nationwide professional learning networks to expand the pool of high school teachers qualified to offer computer science courses at high-needs schools. As external evaluators for the project, we focus on the experiences of teachers and administrators promoting access to equitable computer science courses in schools.

oDREAMS: Promoting Computational Thinking through Game & Simulation Design

Led by researchers at the University of Colorado, this project trained teachers to implement curriculum modules that engage students in game design and simulation building using the AgentSheets and AgentCubes programming environments.

FITTING CS INTO THE CURRICULUM

As states and districts move to bring computer science learning to more students, many questions arise about where and how to place computing within the curriculum. Should it be an elective or required? A stand-alone course or integrated into other existing classes? What department (or track, such as career and technical education) should 'own' computer science, and what kinds of credit should it confer? What certification should be required for teachers? How should students be selected or recruited for computer science courses, especially to encourage all students to participate?

Strategies for promoting expansion of computer science have varied. The CS3 project and the Teach for America (TFA) initiative, which focus on the Exploring Computer Science (ECS) curriculum in high schools, reflect the belief that creating courses that fulfill key graduation requirements will expand participation among underrepresented groups. The oDREAMS project targets middle school students and has long employed a strategy of providing curriculum modules that can be combined to create a standalone computer science course or can be incorporated into existing courses in a variety of disciplines to spark students' interest and confidence in studying computer science.

TEACHERS AS CHAMPIONS

Teachers that we spoke with noted that although their administrations were mostly supportive of computer science, the adoption of introductory computer science was often the result of a lone teacher or an individual administrator's initiative. We heard concern from teachers that the program would not survive if that key person were to leave the school.

“Although we currently have supportive administration, the program is built around individuals and not around systems. I worry about the future of the program if a strong system isn't created to support it.”

Teacher from a Navajo school

The concerns we heard are consistent with Google-Gallup's finding of the low-priority status for computer science in schools; the status is seemingly unaffected by strong support among both principals and parents. Sixty-two percent of principals surveyed either agree or strongly agree that computer science should be a required course where it is offered [9] but few parents responding to the survey reported having approached school officials to specifically express support for computer science in the classroom. Furthermore, educators do not report that computer science is a priority at their school or district [9].

We found that in the case of some of the schools in the TFA project, course adoption rested on the initiative of a single individual. One teacher told us, “Most teachers at my school get the opportunity to pitch an elective that they are passionate about, any subject they want to introduce to kids. I use that as an opportunity to learn alongside my students. Computer science has been my elective for the past three years.”

Teachers sometimes characterize principals as reluctantly agreeing to “let them teach” computer science. Teachers may strike a bargain, agreeing to teach other courses if they can introduce computer science courses—“My principal asked me, ‘What's it going to take for you to stay?’ and I told her ‘I need personal growth, I want to do something the kids need,’ and she said ‘Okay, we're doing this.’” A teacher from an urban school relates, “The deal was, I could teach computer science if I taught four other self-contained classes, so I teach math, personal finance, science, and computer science.”

“My principal asked me, ‘What's it going to take for you to stay?’ and I told her ‘I need personal growth, I want to do something the kids need...’

Teacher from a rural school

BARRIERS TO IMPLEMENTATION

The Google-Gallup study reports that an oft-cited barrier to offering computer science in high schools is the lack of qualified teachers. Our observations accord with this—we have found that even when a course is in place, it is difficult to sustain when key teachers are in short supply. We learned of several administrators who had committed to computer science courses then ended up removing them from the schedule—not because they didn't have a trained CS teacher, but because they were strapped for qualified science and math teachers. In one rural district everything seemed to be in place: a TFA fellow was primed to teach ECS, the principal was supportive, and students had been recruited for the class. At the last minute, however, the class was scuttled so that the teacher could teach more (required) math classes.

Math is not the only specialty that pulls teachers from computer science. We listened to a teacher explain, “I teach special education, and I am the only resource teacher for over 60 kids, so I am not allowed to teach the [computer science] course.” Although math and science seem like natural departments from which to recruit CS teachers, teacher shortages in these tested subjects often mean high-demand teachers are unable to add introductory computer science to their teaching loads. This leads to the counter-intuitive conclusion that advocates for CS instruction would do well to focus their recruiting efforts on history, business, and language arts teachers.

CS FOR ALL?

The apparently high level of support for CS education that the Google-Gallup studies reveal is encouraging, but we are finding that adding computer science courses to the school schedule is no guarantee that underrepresented students will be offered, or will take, the opportunity to study computing. We learned from teachers striving to promote equity that when computer science is offered as an elective, circumstances can arise affecting access to the courses and thereby recruitment of diverse students. Logistical factors such as where a CS course appears in the schedule and when it is added can limit participation by the very students it is meant to target. For example, if ECS is scheduled at the same time slot as several required courses or popular electives, it becomes effectively unavailable to many students. One teacher described this administrative barrier as follows: “I talked to my principal and we got the class on the roster, but ... the way the master schedule worked out was not in my favor. ... There were students who wanted to take my class but couldn't. [A popular local course offering], football, and other elective classes and core classes for AP were scheduled in first period [in same slot as ECS]. My class dropped off the radar.”

In another case, where the teacher specifically made a conscious effort to recruit girls for participation in ECS, we learned that the course was scheduled to coincide with a dance elective that was very popular among girls in the school. Stories like this indicate that principal support and thoughtful consideration of broadening participation is key for building enrollment. One

frustrated teacher lamented, “Support for the course seems to be nonexistent with the current principal.”

Another important, but sometimes under-recognized, issue for achieving equity is the need for social support. The Google-Gallup survey found that girls were less likely than boys to be encouraged by teachers or parents to study computer science. The beliefs and attitudes that teachers and counselors communicate to students can also dissuade students from enrolling. We heard reports from TFA teachers of girls being actively discouraged by school personnel from taking ECS.

“They removed all the girls from my class. I can only tell you ‘he said, she said’ but one of the counselors told a girl that computer science was no place for a girl.”

Teacher from rural school

These beliefs and attitudes show why it will take more than new policies to effect change; in some schools, a culture of low expectations and biases regarding who can be successful in computer science is pervasive. Programs like the National Center for Women & Information Technology's Counselors for Computing program are examples of deliberate efforts to support new attitudes. Teachers see that getting more girls into computer science classes depends on making a personal connection with them. As one teacher said, “It's about talking to the students.” Fighting stereotypes and advocating for their subject becomes one more layer of responsibility on top of the usual challenges of teaching in a high-needs population. “That is the battle I am fighting,” one TFA teacher told us. Projects focused on addressing teacher beliefs as a way to fight stereotypes may have limited impact on access to the field unless they find ways to influence the expectations of other key school personnel.

Various policy issues, such as how the state decides to classify CS courses for credit (e.g., as math or career technical education), can also restrict access. Whether the class counts as a career/technical credit or a college preparation credit can exert a big influence on who enrolls, who can teach the course, and how it is taught. One teacher explained that, “We beefed up the algorithm unit because of it counting as math...with Scratch they added math about how they move around the screen.” Classifying introductory computer science as a career/technical class can boost enrollment among target populations but may disqualify potential teachers who do not hold the necessary certifications. It may also fuel debate over course content and goals. Several of the introductory computer science courses that are popular, such as ECS, have academic goals and pedagogy that conflict with the skills-based approach of most career/technical courses.

We learned of instances where introductory courses are positioned as lower-track classes or offered as a club which is un-

available to students who are required to spend elective time on remediation for core classes. Both of these scenarios can limit participation among target populations. We find that schools can vary widely in the availability of electives to students. Google-Gallup reports that many students who can't take computer science classes are able to learn about computing in school-sponsored clubs or activities. But our survey of TFA teachers indicates that in severely under-resourced schools, there are few out-of-school options, and a dedicated class is the only way to learn computer science. The oDREAMS approach to inserting computer science into middle school is to augment or supplant keyboarding and business software courses.

“I went to my superintendent and I said, ‘Hey, listen. I think this will be amazing. I think this is where we need to move with curriculum.’ Gone are the days where we’re trying to teach kids how to use Microsoft Word. That’s not what technology is anymore.”

An oDREAMS teacher

NO EXPERIENCE NECESSARY?

We are learning from our interviews with teachers that most of them had little to no background in computing prior to their summer professional development workshop, and that they believe that this does not impede their ability to teach the course. More than half of the teachers we surveyed for the TFA evaluation had no experience with CS as a student or teacher and had not even engaged in computer science-related hobbies. Their success indicates that deep knowledge of computer science is perhaps less important than administrators assume for teachers of introductory CS classes. Instead, a teacher's ability to facilitate student inquiry and to communicate expectations for success are perhaps more important than computer science knowledge for teaching introductory classes. This type of expertise supports the goals of courses like ECS and the game design lessons of oDREAMS in exposing students to the field of computing and encouraging their participation in a computing community.

In both the CS3 and oDREAMS studies, a greater proportion of the teachers have had some background in computing, but in many cases the nature of the new course is quite different from their previous experience, which often emphasized students' use of business software. A prominent theme that emerged in interviews with teachers new to teaching game design was enthusiasm for the potential of the lessons to teach skills beyond CS, including collaboration, metacognition, and resilience. We

heard this message from both computer and technology teachers and from those teaching disciplines such as world languages and English/language arts. This chips away at the assertion that the movement to broaden access to CS education is designed primarily to produce programmers for the tech industry.

One teacher of Spanish appreciated how the professional development experience in oDREAMS forced her to pay attention to her own learning process and felt it made her more empathetic to her students' struggles in her language classes. Several teachers mentioned the importance of computer science learning to their students as the motivation for their commitment to persevere through the training so that they could offer this opportunity. One very frustrated and overwhelmed newcomer was overheard in the hallway saying, through tears, "I am here for the kids. I know that they need this, and I want to be able to teach them." These teachers are learning a new style of pedagogy as well as new content, and, in the case of the game design curriculum, a new tool (AgentCubes), which can be extremely challenging. They are voluntarily putting themselves in the role of students so as to be better able to serve their own students.

THE WAY FORWARD

The ground-level view we have obtained from these three projects indicates that broadening participation in computer science is a far more complicated endeavor than it appears. It isn't simply a matter of scheduling classes and training teachers; many pieces need to be coordinated with sustained support for the movement to achieve its goals. Even in places where computer science is offered, access is often restricted for logistical reasons or because of beliefs—among teachers, counsellors, and administrators—about who can be successful. This points to the importance of understanding local conditions. For example, while much of the focus of Computer Science for All has been on providing access to computer science to underrepresented urban youth, much of the oDREAMS project's outreach

is taking place in rural and tribal schools, which can be especially vulnerable to teacher shortages.

We found that teachers acting as advocates for computer science instruction was a commonality across projects. The willingness of these teachers to engage in intensive professional development to learn a new curriculum with a steep learning curve belies the view of Computer Science for All as a top-down reform or the “new vocationalism.” [1,8] These teachers have taken on the mission to bring computer science to their students because they feel strongly that young people will need to understand computing to participate fully in the modern world. In some cases, teachers with little to no computer science background are willing to undergo extreme frustration to gain the skills needed to accomplish this mission.

Early results indicate that training teachers from arts and humanities disciplines may be a prudent strategy to prevent them being pulled away to teach another course in a subject that is included in high-stakes testing.

School change guru Michael Fullan has described educational change as a socio-political process, one in which it is essential to pay attention to both the big and the small pictures [3]. The Google-Gallup studies offer important insights into the big picture—the broader forces impacting the success of the CS for All movement. Our examples from the field offer a closer look at the details—the lived experiences of educators working to prepare their students for life and citizenship in a digital world. Listening to the voices of those who are charged with enacting sweeping reforms is essential for gaining an understanding of how curriculum policies play out in the classroom. An examination of the experiences of teachers and administrators in our three projects demonstrates that the story is always more nuanced than the broad slogans of reform would indicate. Monitoring the progress of the CS for All movement and keeping it on track toward its goals will require that we balance statistics on course adoption with these first-hand accounts from the field. ❖

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grants 1418149, 1542737, and 1312129. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would like to thank all of the participating teachers for their contributions to this work and Teach for America for the photographs.

References

1. Cuban, L. “Coding: The New Vocationalism (Part 2).” Larry Cuban on School Reform and Classroom Practice,” *Wordpress*, 13 July 2017; <https://larrycuban.wordpress.com/2017/07/14/coding-the-new-vocationalism-part-2/>. Accessed 2018 June 28.
2. Della Cava, M. “Should Students Learn Coding? Students, Schools Disagree, Poll Finds.” *USA Today*, August 20, 2015; <http://www.usatoday.com/story/tech/2015/08/20/google-gallup-poll-finds-parents-want-computer-science-education-but-administrators-arent-sure/31991889/>. Accessed 2018 June 28.
3. Fullan, M. *The new meaning of educational change*. (Routledge, 2007).
4. Gallup. “Searching for Computer Science: Access and Barriers in U.S. K-12 Education.” Gallup, August 20, 2015; https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf. Accessed 2018 July 3.
5. Google Gallup. “Trends in the State of Computer Science in U.S. K-12 School,” 2016; <http://goo.gl/j291EO>. Accessed 2018 July 3.
6. Google Inc. & Gallup Inc. “Diversity Gaps in Computer Science: Exploring the Underrepresentation of Girls, Blacks and Hispanics,” 2016; <http://goo.gl/PG34aH>. Accessed 2018 July 3.
7. Miltner, Kate M. “Who Benefits from Teaching All Kids to Code? Tech Companies for One.” *Slate*; http://www.slate.com/articles/technology/future_tense/2017/12/who_benefits_from_the_push_to_teach_all_kids_to_code.html. Accessed 2018 June 28.
8. Singer, N. “How Silicon Valley Pushed Coding into American Classrooms.” *The New York Times*, 27 June 2017; <https://www.nytimes.com/2017/06/27/technology/education-partovi-computer-science-coding-apple-microsoft.html>. Accessed 2018 June 28.
9. Wang, J., Hong, H., Ravitz, J. and Moghadam, S.H. Landscape of K-12 computer science education in the US: Perceptions, access, and barriers. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 2016, 645–650.

Carol Tate

SRI International
201 Washington Road
Princeton, NJ USA
carol.tate@sri.com

Julie Remold

SRI International
333 Ravenswood Ave
Menlo Park, CA USA
julie.remold@sri.com

Marie Bienkowski

SRI International
333 Ravenswood Ave
Menlo Park, CA USA
marie.bienkowski@sri.com

DOI: 10.1145/3230704

©2018 ACM 2153-2184/18/09 \$15.00