Proposing, Preparing, and Teaching an Equity- and Justice-Centered Secondary Pre-Service CS Teacher Education Program

Amy J. Ko
ajko@uw.edu
University of Washington
Seattle, WA, USA

Anne Beitlers
annes@uw.edu
University of Washington
Seattle, WA, USA

Jayne Everson
everjay@uw.edu
University of Washington
Seattle, WA, USA

Brett Wortzman
brettwo@uw.edu
University of Washington
Seattle, WA, USA

Dan Gallagher
dan.gallagher@shorelineschools.org
Shoreline Schools
Shoreline, WA, USA

ABSTRACT

Teachers are essential to equitably broadening participation in computing in schools, but the creation of CS teacher education pathways faces many challenges. In this experience report, we share the many political, administrative, institutional, and sustainability barriers our institution faced in creating a secondary CS pre-service pathway. Throughout, we discuss the particular design choices we made in order to center equity and justice, both in the content of the program, but also in its structure, policies, and resources, which were often in tension with state standards and policies. We also describe our experience teaching and supporting the inaugural cohort of graduates as well as the graduates' experiences, which revealed tension between utopian and dystopian futures of computing and their role in helping students navigate them. We end with a reflection on key factors that we believe led to its successful first year launch, including leadership, interdisciplinarity, capacity, timing, and funding, and on sustainability concerns, including tuition subsidy and instructional capacity.

KEYWORDS

pre-service, equity, justice

ACM Reference Format:
Amy J. Ko, Anne Beitlers, Jayne Everson, Brett Wortzman, and Dan Gallagher. 2023. Proposing, Preparing, and Teaching an Equity- and Justice-Centered Secondary Pre-Service CS Teacher Education Program. In *Proceedings of the 54th ACM Technical Symposium on Computing Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569735

1 INTRODUCTION

Calls for computer science for all in primary and secondary education have many (sometimes conflicting) motives but they all have one thing in common: a need for well-prepared teachers [4]. Without teachers, there are no CS classes, there is no CS integration, there is no learning about whatever we deem important to know about computing. Without teachers, whatever change we hope to see through CS education, whether technological, social, or political, will not happen. Teachers are best positioned to augment young people's understanding of technology and to prepare youth to create a more just computing world.

To have CS teachers in primary and secondary schools, we need pathways that enable teachers to learn about CS and CS teaching. Across the world, these emerging pathways are taking many forms [14]. Some are short- or long-term term professional development, offering depth or breadth of CS content knowledge and perhaps some guidance on teaching methods and educational technologies [5, 16]. Other pathways focus mostly on certification, licensing, and credentials [13]; for example, in the United States, many former software engineers pivot to secondary education, earning teaching endorsements to complement their existing CS content knowledge [6, 15]. Other pathways are informal, including teachers reading books about CS teaching, learning CS skills independently, advocating for CS teaching assignments to make space in their schools for the topic, or connecting with networks of other CS teachers doing similar learning [18]. All of these pathways have matured over a decade or longer, and research about them has revealed numerous insights about their strengths and weaknesses, such as the importance of sustained learning about CS over extended periods [12], the sometimes unique stereotype threats posed by cultures of CS [3], school leaders' perceptions of the lack of student demand and qualified teachers for CS [19], and the need for dedicated learning about CS teaching methods [20].

In subjects other than CS, pre-service teacher education pathways exist to help prepare teachers. These pathways, unlike most professional development, certification, and self-learning pathways, have the advantage of sustained focus on learning to teach. Some are structured as post-secondary undergraduate degrees that complement a content area – for example, UTeach programs in the United States engage students in one subject area while earning their teaching license, often over the course of 2-3 years [1]. Masters in Teaching programs are often more concentrated, with 1-2 years

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9431-4/23/03.
https://doi.org/10.1145/3545945.3569735

of intense learning about teaching [2]. These programs most commonly prepare teachers for the broad canon of learning in primary schools and content-specific subject areas in secondary schools.

CS, however, is rarely part of pre-service teacher education. There have been sustained calls for this change and careful analysis of its challenges. Most notably, DeLyser et al. in the report *Priming the CS Teacher Pump*[1] note that faculty in colleges of education and computer science need to connect; there need to be new or revised courses in teacher education programs to integrate CS into preparation; they may need to hire faculty with expertise in CS education, just like they already have faculty with expertise in math, science, and literacy education; and there need to be pathways to train CS education doctoral students who will become these faculty.

There is also a long-standing need to address inequities and injustices in CS education at all levels. This can come in two forms: addressing inequities in CS education systems, systems, and policies (e.g., [11]), but also addressing the ways that computing itself can create and amplify inequities in society (e.g, [8, 10, 17]). Pathways need to prepare teachers to understand both of these structural forms of inequity, and to help youth understand them.

While these needs capture the core challenges, they elide the concrete experience of such systems change, such as the specific leadership, politics, and motivations of a particular institution or region. These issues can often be just as significant a deterrent to change as the structural factors, especially when engaging issues of equity and justice. In this experience report, we share one case of these concrete contextual challenges, but also of tentative success. In the rest of this paper, we will describe the context for the case – the University of Washington, in Seattle, USA – and the path we took to successfully launch an equity and justice-centered pre-service CS teacher education program and graduate our first cohort. Throughout, we highlight the specific contextual challenges we faced and the local tactics necessary for adressing them. We hope that by sharing this particular case, we can inspire other faculty to persist their their own local challenges and create programs that expand CS teacher education pathways.

2 CONTEXT

Our efforts, beginning in 2017, were situated in Seattle, Washington, in the United States. The city has a progressive culture and a growing economy that includes many software companies, and many companies in other industries that heavily leverage software. The broader state's economy is rural, agricultural, and generally conservative, fiscally and socially. At the time, public schools in the state were generally underfunded; the state's Supreme Court had even judged the state legislature as being out of constitutional compliance in providing sufficient funding.

Our efforts to explore the possibility of pre-service CS teacher education were situated at the University of Washington, one the state's flagship research universities. The university has a large and research-active college of education and several large and research-active units concerned with computing, information, and design. The broader campus faculty and leadership center diversity, equity, and inclusion. The campus also has a robust culture of interdisciplinarity, with many long-lived and sustainable collaborations across research, teaching, and administration spanning multiple academic units, including many interdisciplinary academic programs concerned with computing (e.g., data science, entrepreneurship, and design).

The university's College of Education is departmentalized. One department of the college focuses on teacher preparation, administering multiple one-year, cohort-based graduate programs in teacher education, including three for primary education, one for secondary education, and one for special education. The programs are relatively small, matriculating between 25-70 teacher candidates per year, with small classes of 10-30 students. For the secondary masters in teaching program, candidates are expected to have already met content knowledge requirements for their subject area of choice; the coursework focuses on history, equity, justice, teaching methods, assessment, and extensive field placements. Students are also required to participate in extensive identity caucusing, a strategy for building anti-racist collectives that help surface implicit bias, develop awareness of positionality, and build solidarity amongst candidates with marginalized, oppressed identities. The college's teacher education programs have had consistent demand and enrollment, despite national declines.

None of the education faculty at the time claimed expertise in CS or computing education; none of the teacher education coursework involved discussions of computing and none of the certification pathways required teachers to obtain any computing content knowledge. Some teaching-track faculty in the university's CS department had CS education research experience, and some were interested in and/or had experience with teacher preparation. One tenure-track faculty in a non-CS department (the first author) had expertise in CS education, but not in teacher preparation. Some of the college of education's faculty held hostile views toward technology companies, in contrast to the CS department's close collaborations and funding relationships with technology companies of all sizes.

At the time, the state had made some inroads to state support for K-12 CS education, including some funding allocation for in-service teacher professional development, an exam-based CS endorsement pathway, and some community-based, federal grant-funded activity for catalyzing STEM and CS education. The majority of schools in the state did not teach CS and those that did generally taught AP CS A, AP CS Principles, or introductory programming. No colleges or universities were offering pathways to earning the CS endorsement, and nearly all endorsed CS teachers in the state were teaching under Career and Technical Education (CTE) certifications, which did not require any CS content knowledge or CS education knowledge. Based on state data, capacity to teach CS in schools was limited, access to CS education was primarily found in well-funded districts near technology companies, and participation followed the familiar trends in CS, primarily engaging white and Asian boys. Overall, the state politics, policies, laws, and agencies viewed equity narrowly as opt-in opportunities to access to courses.

3 PROPOSING

It was in this context that the first author began to explore the possibility of creating CS teacher education pathways on campus. She had been participating in a broader, federally-funded state effort to expand STEM teacher education pathways, including CS, and

[1] http://www.computingteacher.org

thought that a way to contribute would be to identify opportunities and barriers in program creation. Without any prior collaborations with the university's college of education, they reached out to a handful of faculty to begin to identify who would be the decision maker about creating such programs.

After some networking, the 1st author was invited to talk for 5 minutes at an academics planning meeting with all of the directors of the teacher education pathways. Building on their prior administrative experiences, she focused her message on the pain points and problems that administrators likely faced (namely resources) but also appealing to their values of equity, justice, and prestige. The director of the primary pathways did not see a way to fit it in to their already packed curricula, but the director of secondary pathways wanted to learn more. This led to a meeting with her and the college's Assistant Dean of teacher education, where the assistant dean made clear that they viewed the opportunity as a way to lead, and that if we could find resources to sustain it, it would be of interest. The director of secondary saw the opportunity as way to reach beyond the college to connect with others on campus. While fitting CS as another subject within the one year program would likely not work for many reasons, she saw an opportunity to develop an opt-in supplementary certification.

This began a multi-year fundraising effort, led by the first author, which included conversations with state senators, philanthropic foundations, individual philanthropists, as well as the National Science Foundation. The first author had some experience with non-research fundraising from prior entrepreneurial experiences; much of the experience felt similar, as she talked to different groups, pitching the opportunity, and providing evidence, data, and arguments to try to persuade decision makers with resources. Ultimately, none of these fundraising efforts yielded funding, and the Assistant Dean that supported the effort took a position at another university.

Since state and philanthropic pathways seemed fruitless, the first and second author decided to pursue federal research funding. Building upon the first author's efforts to connect local CS for All advocates, they recruited two local CTE directors passionate about CS education in their districts and an advisory board of local researchers, teachers, and CS specialists, and wrote a funded proposal that brought enough resources to start a program (but not sustain it). Shortly after funding, the team expanded to include a new doctoral student who was a former high school CS teacher (the 3rd author), and a teaching-track faculty member in CS with prior experience in CS teaching and CS teacher professional development (the 4th author).

4 PREPARING

The COVID-19 pandemic began just after the funding arrived. Luckily, the team had structured the first year of the project as a preparatory year, getting state approvals for the new certification pathway, university approvals for the new courses, college of education approval for the new program, teaching load and commitments from the three participating units on campus (education, computer science, and information science), curriculum design, and recruiting. The project also planned to write an equity and justice-centered textbook for secondary teacher candidates to address a gap that the team saw in learning materials for CS. Most of this work was possible without partnering with teachers, schools, and districts.

But the pandemic was not without impact. Our team, like many, faced fatigue from entirely online work lives, from racial unrest, and from frequent administrative crises unrelated to the project. The preservice candidates we intended to recruit from the masters program to stay for the supplementary certification were also exhausted. It was amidst all of this that we tried to create the new program and its administrative supports.

One of the first challenges was seeking state approval. Ours was the second program to seek approval to offer a CS certification in the state, and the state office responsible for evaluating proposals did not have CS subject matter expertise. They did, however, have detailed requirements for the program proposal, many of which were in tension with our equity and justice goals. This was particularly true of the state's CS endorsement competency requirements, which were largely derived from the K-12 CS standards (*k12cs.org*) and defined equity as access to marketable programming skills. This left little room for broader goals of teacher and youth critical consciousness about computing and society. It took multiple rounds of iteration with the state office, and testimony at a public hearing, to get the program approved. Our perception was that many of the barriers imposed by the state were an exercise in both regulatory compliance and trust building with state officials.

University level approvals were straightforward. The college of education's staff was skilled at wrangling the university course approvals process, as were both the first and second authors, as both had directed academic programs previously. We worked with our newly recruited instructional team (the first four authors of the paper) to sketch the course learning objectives.

Cross-department approvals were an exercise in campus politics. The college of education was taking much of the risk in offering the program and sought resources from the first author's home unit (an information school) and the CS department to support the effort. Specifically, we wanted to secure 5 years of one teaching load from each unit to staff the program and a commitment for classroom space in the CS department's relatively ample space. The 1st author leveraged their relationships with both unit's leaders to secure this agreement, using their status in their home unit to secure release and leveraging the CS department's new building and teaching faculty retention concerns to secure space and teaching release.

Textbook writing was comparatively simple. The first author took the lead, collaboratively shaping a book outline with the team as well as the many doctoral students in their research lab, and then used much of the quiet time of lockdown to write about computing, gender, race, class, equity, justice, and teaching. Throughout, they sought feedback and co-authors for several chapters, balancing the need for a consistent voice with the need for diverse expertise. To achieve this, the first and second author met weekly one summer to discuss lesson and unit planning for a secondary setting, capitalizing on the second author's secondary classroom teaching experience. By the end of the first year, the first draft of the book was complete, with twenty chapters covering history, pedagogy, assessment, and the foundations of computing and computing education through an equity and justice lens (now available at *criticallyconsciouscomputing.org*, [9]).

Throughout all of these preparatory activities, we also built recruiting practices. This included information sessions with current students in the Masters in Teaching program as well as separate information sessions with computing and information undergraduates with a potential interest in secondary teaching. Recruiting challenges amongst the masters students primarily concerned funding for tuition support and the opportunity cost of staying for the program instead of doing long-term substitute teaching in order to resume making income after a year of graduate school. For undergraduates, it included many of the expected difficulties in the United States – questions about work/life balance, teacher salaries, and teacher respect. Throughout, though, many indicated that if these barriers were not present, many might prefer teaching CS over other subject areas, or over positions in industry.

Throughout all of these first year efforts, our team met every two weeks to sustain progress, refine program and course designs, develop tactics for securing resources, and shape our recruiting plans. We used many tools and resources for sustaining researcher-practitioner partnerships [7] to ensure that we had capacity to contribute, shared goals and values, and organizational alignment. The 5th author was particularly helpful in ensuring we were thoughtful about joint decisions, but also *how* we made decisions together.

5 TEACHING

After a year of preparation, Spring 2022 arrived, and we had successfully recruited and funded 9 teacher candidates: 5 students already enrolled in the masters in teaching program and 4 in-service teachers. All of the pre-service teachers had some prior exposure to programming, but none had computer science degrees; four had earned credentials in math or science education and one in social science. The in-service teachers all had prior experience teaching CS, but sought to expand their perspectives to include equity and justice topics. All received tuition support, either from our federal or state grants or from their districts. Three identified as men and six as women; five as Asian, two as white, and one as middle eastern.

Our program design was as follows. First, students were required to demonstrate exposure to introductory programming concepts, as a course or equivalent, in any language. We intentionally kept content requirements minimal, to reduce barriers to enrollment, to reduce the demotivating effects of many introductory programming courses, and to signal to candidates that programming is part of CS teaching, but not the only important thing. Our assessment of programming knowledge was flexible and informal: a course on a transcript, verbal report of having completing an online tutorial, or pointers to side projects were all accepted.

Building upon this content knowledge, the program required enrollment in four interconnected courses, all taught through a lens of critical consciousness about computing, schools, and society:

- *Equity and Justice in CS.* This had two connected goals: bolster students CS content knowledge to a level sufficient to pass the state's certification exam through required readings, but do it in a way that helped students see the inherent relationships between CS, equity, and justice in society. The course required candidates to complete a programming project mirroring the *AP CS Principles* portfolio. The course used the team's book, *Critically Conscious Computing*, as its core text, involved substantial practice in teaching about equity and justice topics in CS and reflection on pedagogy.
- *CS Teaching Methods.* This focused on student-centered pedagogy and pedagogical content knowledge and included required reading about CS pedagogy, teaching demonstrations, and reflections on teaching in the candidates' field experiences or in-service teaching. Major topics included how to balance technical and sociopolitical content.
- *CS Assessment.* This course focused on assessment, examining assessment practices from a critical perspective, and its connections to student agency, identity, self-efficacy, and learning. It challenged teachers to develop a student-centered lens on assessment, focusing it on how to facilitate and empower students.
- *CS Field Placement.* This involved pre-service students working in CS classrooms with experienced CS teachers in the region. Although it occurred during the COVID-19 pandemic, all teachers worked in-person with students (and experienced all of the challenges of student illness, hybrid classrooms, and staffing shortages).

Each of the first three courses above met for two hours online in the late afternoon, Monday through Wednesday. We intentionally sequenced the three classes in this order, starting each week with CS content, then talking about the challenges of opportunities in teaching it, then discussing how, when, and whether to weave assessment into that teaching. These three classes met in a custom online classroom that facilitated discussion, small group work, and presentations, along with the ability for candidates to move and resize their videos, giving a sense of personalization. Crucially, all three classes used the same virtual classroom, and so there was a single space for the whole program. To complement the field placement, and integrate perspectives from all four courses together, the cohort met on Thursdays in-person in a "studio" environment, with a co-constructed mixture of CS warm up activities, project work and presentations, teaching demonstrations, networking with the local CSTA chapter, and reflections on candidates' field placements and teaching. These end of week studio meetings generally had a celebratory, affirming, and community-based vibe.

Before each studio day, the teaching team huddled for 30 minutes to collaboratively structure the studio day, to coordinate the next week's instruction, and to share information about students. These sessions were also a way of building a sense of interdisciplinary teamwork amongst the instructors.

6 REFLECTION

Here we offer our reflections on teaching and administration, and our candidate's reflection on their learning.

6.1 Teaching Experiences

The morning before the huddle, the instructors reflected in a shared document on five topics: candidates' content knowledge, pedagogical content knowledge, self efficacy, and field placement experiences; and the efficacy of our pedagogy. Several themes emerged.

First, there was clear value in having both pre-service and in-service teachers in the program. The in-service teachers helped build the self-efficacy of the pre-service students and the pre-service

students helped the in-service teachers see the broader possibilities in content and pedagogy beyond their current practices.

Second, candidates quickly built confidence about their own grasp of critical perspectives on equity and justice, but underneath these was a consistent fear of not knowing enough about technical concepts in CS or their ability to raise critical perspectives with students. Candidates also often conflated CS knowledge with practical knowledge about languages and APIs; we had to continually reassure candidates that *few* in computing feel like they know enough, especially with respect to programming languages and APIs, and that authentic practice means always learning.

Third, candidates brought immense creativity to their pedagogy and often found, after some encouragement, that their skills in teaching other subject areas transferred relatively easily. Many on our instructional team were teaching teachers for the first time and struggled with their own teaching self-efficacy. They shared these struggles with the candidates as a way of creating spaces for candidates to reflect as well. But most candidates struggled with how to reconcile what they wanted to teach with the chaotic schedules, low resources, and limited time in their classrooms. The instructors did not feel they had easy answers to these tensions.

Finally, engaging candidates in programming projects, especially in studio time, was crucial to growing their CS self-efficacy, as everyone observed others' failing, persisting, and finding success, including the candidates with significantly more programming experience. This seemed to normalize failure in a constructive way.

6.2 Learning Experiences

To complement the instructor reflections, we sought feedback weekly in studio, as well as at the end of the quarter in a program level exit-survey. Each instructor also sought feedback in unique ways; for example, the equity and justice instructor gathered weekly reading reflections, revealing content knowledge gaps, and the assessment instructor sought end of quarter reflections about assessments. These reflections revealed several themes about the candidates' learning and experiences.

CS was more interesting than they thought it would be. In reading the textbook prepared for the program, many expected to find CS technical content dry. But candidates noted in their reflections how fascinated they were about the surprising connections between abstract ideas in computation and the broader world. For example, many were fascinated by the power and limitations of Boolean logic in mimicking human decision making in the world:

> *I really enjoyed learning a bit more about Boole and the origin of Boolean values. I can definitely see myself including this kind of information in my geometry classes when teaching about logic and truth value, and maybe even then having a discussion about why these are used and how they can be both valuable and limiting.*

> *It is interesting to think about how human decision making can and can't be represented by logic. I think a hallmark of human-ness is our ability to use logic and to solve problems using tools. Yet one of our major tools right now (computers) takes the human-ness out of decisions because it over logics.*

> *Throughout this course, I think the major idea was illustrated in this chapter with "edge cases". There are so many things we can logic through, decisions to make, choices to customize, data to non-boolean, but there will always be some edge cases that deserve consideration as well. I'm not sure if is fascinating or frustrating to consider these, I suppose it depends on your role.*

Others resonated with the ways that abstractions can simultaneously simplify but also hide reasoning:

> *Something fascinating about abstractions is how it's used to make programming easier, but also hides information from users. It's easy to take advantage of abstractions and not think about the consequences, and I appreciate the different ways this chapter presents to inform students more about this.*

> *This statement was discouraging from a consumer viewpoint: "theoretically impossible to find all possible defects". As noted in the text some very critical decisions are made using possibly defective programs.*

And others still resonated with the externalities of computing technology on sustainability:

> *What I found to be the most fascinating was the section about computer waste, mainly because I never really thought about it before. It's interesting to me how computer waste are often searched through for rare earth metals and otherwise burned producing toxic waste into the soils and air of the earth. It makes me wonder how we can recycle and reuse more of the computers rather than just the rare metals.*

Community grew self-efficacy. Candidates explicitly reported that despite still feeling like there was an endless amount to learn, they felt more capable of learning it and more secure in the knowledge they did have, largely because of the relationships they formed. For example, some mentioned the importance of having a cohort:

> *I really am glad to have joined this cohort and community of people aiming to teach computer science with this critical lens. I am not expecting to do it particularly well at first, but I feel better knowing I have a group of people to come to for advice, commiseration, and celebration.*

> *Resources (including colleagues and experts) to help teach CS better - I am coming away with lots of resources and some practice using them, and a helpful group of colleagues to share ideas with.*

The textbook anchored learning. Candidates consistently reported that having a textbook for the program, and having that textbook be the basis for the discussions of all of the courses, was critical to their learning. For example, one candidate noted how they viewed it as a key resource after the program ended:

> *I will definitely be coming back to this book regularly as I use the unit sketches or ideas or resources to refresh my learning and learn more. I have definitely ordered many of the resources for my*

library and will look for ways to use the justice centered standards for my own direction.

Some candidates left unsettled but motivated. While much of the coursework was celebratory and joyful, the equity and justice themes in the program led many students to leave the course feeling uncertain about how to proceed as a teacher, advocate, and consumer. For example, one candidate noted in a reflection:

I literally had to take a break in the middle of reading the conclusion. Then I read that paragraph about it being a heavy burden. I really appreciate how you scaffolded "how to be a critical CS teacher" into bite-sized pieces. The unit sketches and these participation suggestions seem doable! Though I still feel overwhelmed, I am excited about putting them into practice.

Others found the mix of optimism and pessimism in the course ultimately optimistic:

I'm totally on board with the idea that solving all of the world's ills cannot be done with technology alone. I disagree a bit the with the initial claim of the conclusion that being critical of the computer science field is pessimistic and not fun. Rather, I fall pretty firmly into the belief that knowledge is power (suppose it makes sense I'm a teacher, eh?) While I did not know or had not considered before many of the social aspects of computing that this text introduced, I feel this introduction has been enormously useful and indeed empowering.

But some found it too much:

I'm feeling like the weight of the world is on my shoulders at the moment. I'm overwhelmed with the content of this book and current events with the gravity of the lives lost in Uvalde and Buffalo and all these other places across the country. I'm feeling a little defeated.

And some, while resolute in the commitment to examine CS teaching critically, centered students, referring to a statement by the 3rd author:

Something I will be taking with me from this course is the quote from you: "I want to burn the system down, but I don't want to burn it down with the kids still inside." I think about this every day.

These reflections illustrate that while candidates were generally positive about the relationships they formed, the knowledge they learned, and their ability to teach CS in the future, they did not all leave with the same sentiments about CS and society: some were inspired to act and some were overwhelmed.

6.3 Administrative Experiences

Reflecting on our experiences, there are several aspects of our program's initial success that seemed like essential ingredients:

- **Leadership**. Our effort had not one, but two faculty in administrative leadership positions who were highly motivated to make this work and one brought extensive entrepreneurial skills. This partnership felt essential to navigating the complexities of resources and cross-unit collaboration.
- **Interdisciplinarity**. Our team had expertise on teacher education, pedagogy, computer science, computing education, and K-12 teaching. All of our members brought essential expertise, and had any of it been missing, the results would have lacked disciplinary authenticity.
- **Capacity**. The 1st author's tenure-track position offered entrepreneurial flexibility.
- **Timing**. Much of the work was about communicating narratives about CS to decision makers; politics were aligned with these during the particular period in which we worked, which led to funding, interest, and engagement.
- **Money**. Initial investment was essential. This came from many sources, including internal gifts of time, external grants from our state and federal government, and the opportunity costs that all of our candidates paid to participate.

Looking ahead, the team also sees several looming sustainability challenges. First is tuition subsidy; recruiting made clear that candidates could not justify the expense of the extra quarter after a year away from work paying graduate tuition. The grants we secured for tuition subsidy were temporary, and the prospect of an endless cycle of fundraising is overwhelming. This has led the team to focus on seeking philanthropic gifts in the form of endowed scholarships that are sufficiently large to subsidize each cohort annually.

A second sustainability question is teaching load. One instructor is tenured, but their unit's release commitment is not perpetual. Two of the instructors are teaching-track faculty on three year contracts and in some cases their load is even more tightly constrained. Finally, the 4th instructor is a doctoral and when they graduate, there may not be a suitable doctoral student to take their place. And only one of these four instructors is in our university's college of education, raising questions about the College's long-term support.

While these issues are acute, they are not unique to CS or to preservice or to academic programs in general. What likely is unique is the open question about how sustainable demand for secondary CS educators will be. Will schools hire for CS at scale in our region? Will public education in the U.S. embrace CS as part of it's canon? Will schools have the resources to teach CS and the demand to hire teachers with preparation to teach it? All of these questions will drive the our particular program's strategic planning and future.

7 NEXT STEPS

Our program's story is just one of many. The numerous efforts across the world to prepare CS educators have their own idiosyncratic complexities of timing, luck, personality, politics, funding, and communication. Thus, for anyone considering starting or contributing to a CS pre-service effort, one lesson from our story might therefore be to examine the many roles and factors surfaced in our story and find a way to address them in your context. For some faculty and doctoral students in education, that might mean leading a program administratively or teaching in one. For faculty and doctoral students in CS, it might mean finding partners in education and sharing your expertise and resources. For primary and secondary teachers, it might mean offering to be a teacher mentor or welcoming pre-service teachers into your community. For students, it might mean actively pursuing pre-service programs and joining the CS teaching workforce. And for funding agencies,

it might mean strategically investing not only in the creation of these programs, but their long term sustainability. Through these many roles and efforts, and a persistent focus on equity and justice in computing, we might succeed at creating a more just future of computing not just in schools, but also society at large.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1539179, 1703304, 1836813, 2031265, 2100296, 2122950, 2137834, 2137312, and unrestricted gifts from Google. Thank you to cohort one for an excellent Spring!

REFERENCES

[1] Ben Backes, Dan Goldhaber, Whitney Cade, Kate Sullivan, and Melissa Dodson. 2018. Can UTeach? Assessing the relative effectiveness of STEM teachers. *Economics of Education Review* 64 (2018), 184–198.

[2] Diana Burton and Ruth Goodman. 2011. The masters in teaching and learning: a revolution in teacher education or a bright light quickly extinguished? *Journal of education for teaching* 37, 1 (2011), 51–61.

[3] Mehmet Celepkolu, Erin O'Halloran, and Kristy Elizabeth Boyer. 2020. Upper Elementary and Middle Grade Teachers' Perceptions, Concerns, and Goals for Integrating CS into Classrooms. In *Proceedings of the 51st ACM technical symposium on computer science education*. 965–970.

[4] Joanna Goode. 2007. If you build teachers, will students come? The role of teachers in broadening computer science learning for urban youth. *Journal of Educational Computing Research* 36, 1 (2007), 65–88.

[5] Joanna Goode and Jane Margolis. 2011. Exploring computer science: A case study of school reform. *ACM Transactions on Computing Education (TOCE)* 11, 2 (2011), 1–16.

[6] Nathaniel Granor, Leigh Ann DeLyser, and Kevin Wang. 2016. Teals: Teacher professional development using industry volunteers. In *Proceedings of the 47th acm technical symposium on computing science education*. 60–65.

[7] Erin C Henrick, Paul Cobb, William R Penuel, Kara Jackson, and Tiffany Clark. 2017. Assessing Research-Practice Partnerships: Five Dimensions of Effectiveness. *William T. Grant Foundation* (2017).

[8] Gayithri Jayathirtha, Joseph Chipps, and Luis Morales-Navarro. 2021. Redesigning an Electronic Textiles Computer Science Activity to Promote Critical Engagement. In *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE, 1–2.

[9] Amy J. Ko, Anne Beitlers, Brett Wortzman, Matt Davidson, Alannah Oleson, Mara Kirdani-Ryan, Stefania Druga, and Jayne Everson. 2020. *Critically Conscious Computing: Methods for Secondary Education*. Online. https://criticallyconsciouscomputing.org/

[10] Amy J. Ko, Alannah Oleson, Mara Kirdani-Ryan, Yim Register, Benjamin Xie, Mina Tari, Matthew Davidson, Stefania Druga, and Dastyni Loksa. 2020. It is time for more critical CS education. *Commun. ACM* 63, 11 (2020), 31–33.

[11] Jane Margolis. 2017. *Stuck in the Shallow End, updated edition: Education, Race, and Computing*. MIT press.

[12] Stacie L Mason and Peter J. Rich. 2019. Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education* 19, 4 (2019), 790–824.

[13] Muhsin Menekse. 2015. Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014. *Computer Science Education* 25, 4 (2015), 325–350.

[14] Lijun Ni, Gillian Bausch, and Rebecca Benjamin. 2021. Computer science teacher professional development and professional learning communities: a review of the research literature. *Computer Science Education* (2021), 1–32.

[15] Anthony Papini, Leigh Ann DeLyser, Nathaniel Granor, and Kevin Wang. 2017. Preparing and supporting industry professionals as volunteer high school computer science co-instructors. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*. 441–446.

[16] Michele Roberts, Kiki Prottsman, and Jeff Gray. 2018. Priming the pump: reflections on training K-5 teachers in computer science. In *Proceedings of the 49th ACM technical symposium on computer science education*. 723–728.

[17] Wendy Roldan, Kung Jin Lee, Kevin Nguyen, Lia Berhe, and Jason Yip. 2022. Disrupting Computing Education: Teen-Led Participatory Design in Libraries. *ACM Transactions on Computing Education* (2022).

[18] Sue Sentance, Simon Humphreys, and Mark Dorling. 2014. The network of teaching excellence in computer science and master teachers. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*. 80–88.

[19] Jennifer Wang, Hai Hong, Jason Ravitz, and Sepehr Hejazi Moghadam. 2016. 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*. 645–650.

[20] Aman Yadav and John T Korb. 2012. Learning to teach computer science: The need for a methods course. *Commun. ACM* 55, 11 (2012), 31–33.