"The Spark Fizzles Out: Barriers and Strategies in Computer Science Teacher Preparation Program Design"

Caroline D. Hardin[a], Noah Dougen[a], Amy J. Ko[b], Jayne Everson[b], Peter A. Tucker[c] and Mark Oursland[d]

[a]Western Washington University, Bellingham, Washington, United States; [b]University of Washington, Seattle, Washington, United States; [c]Whitworth University, Spokane, Washington, United States; [d]Central Washington University, Ellensburg, Washington, United States

ABSTRACT

Background and Context: The United States is seeing an increasing demand for computer science education. However, many schools don't have and can't find CS teachers. One reason for this is the sputtering state of the computer science teacher preparation pipeline. Quite a few CS teacher preparation programs have been initiated over the last 15 years, however the vast majority have consistently struggled to certify more than a handful of teachers per year.

Objective: This study aimed to closely investigate how CS Education teacher preparation programs are initiated, designed, and launched to uncover insights regarding program sustainability.

Method: We used Kotter's framework of organizational change to structure semi-structured interviews with 23 program leaders, identify the barriers faced and the strategies used.

Findings: We found a number of substantial barriers, including ill-fitting state policies, institutional hurdles, and a lack of computer science teacher preparation resources. These barriers often compound each other and ultimately, programs that had trouble establishing the first four steps of Kotter's framework during the program development stage are highly likely to have trouble accomplishing the last four during the sustainment phase.

Implications: We recommended that program leaders solidify these four steps during the development phase to ensure success during the sustainment phase. We also encourage policy makers to take into consideration the barriers faced by these programs to ensure that the teacher pipeline remains robust.

KEYWORDS
Computer Science Teachers; Pre-Service; In-Service; K-12; Teacher Preparation; Education Policy

1. Introduction

Over the past 15 years the demand for computer science (CS) education has been growing worldwide; in the United States 30 states now require CS be taught in schools (of which eight have CS graduation requirements) (Vegas, Hansen, & Fowler, 2021; Weissman et al., 2024; Zarch, Boisvert, Bhatnagar, & Goldwasser, 2024). Code.org, a leading CS Education organization, succinctly describes a central challenge: "The capacity to offer computer science courses depends on the availability of qualified teach-

CONTACT C. Hardin. Email: Email:hardinc3@wwu.edu

ers." (Weissman et al., 2024, p. 10). The number of CS teachers to fill these positions is insufficient, and this lack of teachers has been identified as the greatest barrier to growing the number of people who have access to basic CS education (Guzdial, 2016). Increasing the number of CS teachers requires teacher preparation programs (TPP) — a mix of graduate degrees, undergraduate degrees, authorizations, licensures, certificates, badges, and endorsements (single, dual, and add-on) taken both by pre-service and in-service teachers in order to meet state standards for classroom teachers. TPPs typically teach subject-specific content, general teaching strategies, and pedagogical content knowledge (PCK), ensuring that future teachers have expertise in the most effective ways to teach their particular content areas (Shulman, 1986). Although much is known about teacher preparation program creation and design for better established disciplines, significant complexities arise when developing pathways for new content areas. Little prior work has examined the challenges (such as initial funding and interdisciplinary cooperation across computing and education) that might arise when creating CS TPPs or outlined strategies to help navigate them (Ko, Beitlers, Everson, Wortzman, & Gallagher, 2023a).

Given an existing shortage of in-service teachers, as well as the content depth limitations common to professional development programs, we chose to focus on pre-service programs which aim to prepare new CS educators. Not all states allow CS to be an initial license, and only 16 states have even one pre-service CS program (*Fulfilling Computer Science (CS) Preservice Teacher Education Programs: Policy Brief*, 2023). Currently, the number of teachers prepared through these pathways is low, presenting a major challenge to their continuation (Guzdial, 2011). While there is no single source for this number in the US, the Department of Education's Title II report and the National Center for Education Statistics provide some insight. From 2012 to 2024, Title II data on teachers awarded teacher education degrees in CS has consistently been below 115, and excepting a single outlier, declining over time (US Department of Education, 2025). These low numbers are similar to those published by the National Center for Education Statistics, which reports that only 71 bachelor's degrees and 123 master's degrees were awarded for "computer teacher education" in the US in 2018-19, declining to only 68 bachelor's degrees and 90 master's degrees awarded in 2020-21 (for Education Statistics, 2020). It is clear that these pathways are struggling to sustain graduation of more than a handful of students, and it is important to ask why - especially given the broad consensus on the importance of integrating CS education into the K-12 curriculum and the extensive teacher expertise shortages this creates (Zarch & McCune, 2025).

Ragonis, Hazzan, and Gal-Ezer (2010) describe how Israel's highly regarded CS TPP rests on the foundation of four components: well defined CS curriculum, mandatory formal licenses for CS teachers, research in CS education, and CS teacher training programs (Ragonis, Hazzan, & Gal-Ezer, 2010). In the United States, as in many other countries, these conditions are rarely all present in one region, creating fragmented (and sometimes contradictory) conditions (Lang et al., 2013; Vegas et al., 2021). For example, depending on what state they are in, teachers in the US can become certified in CS pathways as varied as initial certification, add-on license, or authorization (Kim et al., 2021; Ottenbreit-Leftwich et al., 2022). Code.org identified three different variations in the types of certifications which can be obtained: subject, course-specific, and determined by a local education agency (Code.org, 2018). There are also a variety of alternative, non-traditional, and adjacent program structures (such as micro-credentials) which further complicate CS TPP credentialing options (Weissman et al., 2024). Programs having drastically different shapes, and needing ongoing flexibility and innovation, has exacerbated a gap in the documented guidance and recorded institutional knowledge (Ericson et al., 2008). As a result, CS TPP leaders have few resources to consult for program design and development.

For new and existing CS education programs to meet the CS teacher shortage, it is imperative to examine the complexities of CS TPP program creation and administration, a lengthy process involving securing buy-in, developing a guiding coalition to lead a volunteer crew, and rooting the change in an organization's culture (Kotter, 1996). An understanding of the challenges program leaders face and the strategies they use to navigate these issues is enormously valuable in helping institutions resolve the variety of entangled issues between the policies and available resources both at the state and institutional levels.

Our study explores these challenges by answering the following research questions:

- RQ: What are the journeys that occur between the initiation and enactment of CS teacher education programs? Specifically:
  - What are the types of barriers which impede or halt progress on implementing these programs which other program leaders should be aware of?
  - What are the approaches and strategies which could be recommended to other program leaders?

To answer these questions, we conducted a set of 23 interviews with a sample of higher education CS TPP leaders across the United States, gathering their program creation stories, the difficulties they faced in creating and sustaining them, and how they attempted to overcome these difficulties.

2. Background

There are two potential approaches to bridge the gap between supply and demand for CS teachers: encourage in-service teachers to complete CS professional development (PD) or encourage in-service and pre-service teachers to acquire CS certification through a full CS TPP. Having in-service teachers 'add-on' a CS content area endorsement through PD typically has the advantage of faster completion (Menekse, 2015). This comes, however, with a complimentary disadvantage of covering less content or PCK, both of which are especially important for teaching quality CS courses (Ragonis et al., 2010; Yadav & Korb, 2012). With some notable exceptions (Goode, Margolis, & Chapman, 2014; Sentance & Csizmadia, 2017), in the US available CS PD often lack the methodology and practicum content long recognized as crucial for all subjects, but especially CS (Armoni, 2011; Ball, 1990; Gal-Ezer & Stephenson, 2010; Hazzan & Lapidot, 2004; Lapidot & Hazzan, 2003; Ragonis, Hazzan, & Gal-Ezer, 2011). A 2018 discussion among leaders within the CS education community highlighted "...the lack of available CS methodology classes..." as a major roadblock to creating more certified CS teachers, further noting that:

> ...teachers need at a minimum courses on CS content (such as, introductory programming and data structures) and pedagogical content knowledge. (Delyser, Goode, Guzdial, Kafai, & Yadav, 2018, p. 31).

Similarly, experts studying successful international models of CS teacher preparation conclude, "only teachers with formal CS background should teach CS in the high school" (Ragonis et al., 2010, p. 405). Many CS PD programs do not engage in program evaluation, making it challenging to assess or improve them (Friend, Twarek, Koontz, Bell, & Joseph, 2022). Existing multi-year studies on CS PD programs with low initial hours requirements show mixed results in terms of teacher confidence and persistence in teaching (Fancsali, Mark, & Lee, 2025; Franke et al., 2013). PD programs are also unlikely to scale in ways institutes of higher education can (Franke et al., 2013). Code.org concluded that:

> In-service training alone is insufficient to meet the demand for computer science courses. We must actively bring new teachers into the classroom through pre-service training programs. (Weissman et al., 2024, p. 18).

Finally, although state level funding is seen as essential for preparing current educators to teach CS (Weissman et al., 2024), teachers report insufficient access to PD resources (Yadav, Gretter, Hambrusch, & Sands, 2016).

Further limiting the potential for in-service programs to scale is an overall teacher shortage. Over the last few decades, the number of teacher vacancies and the number of under-qualified teachers in US schools has been increasing (Nguyen, Lam, & Bruno, 2022). Despite an overall increase in demand for teachers, enrollment in teacher preparation programs is only 70 percent of what it was 10 years ago (Saenz-Armstrong, 2023). US school districts have particular trouble finding teachers for CS classes (Guzdial, 2016; Mak & Torrejon Capurro, 2024). In 2016, 73 percent of superintendents stated that their district had no qualified teachers to teach a CS course (Wang, Hong, Ravitz, & Hejazi Moghadam, 2016). Other researchers found that 31 percent of public schools and 35 percent of private schools had a difficult time filling a CS teacher opening in 2020-21 (Irwin et al., 2023), an increase from 15 percent and 10 percent in 2011-12, respectively. These factors contribute to why only about 60 percent of schools in the U.S. offer a foundational Computer Science course (Stein, 2024).

While in some countries (such as Scotland, England, and Israel), CS teacher certification is clearly laid out, easy to understand, and has robust requirements for both CS content and CS pedagogy, the landscape in the United States is more complicated (Gal-Ezer & Stephenson, 2010; Sentance, 2024). State and national governments, industry representatives, local school districts, national CS organizations, higher education institutions, and other stakeholders often find themselves at odds over critical issues such as the specific courses that should be offered and the qualifications required for educators who teach these subjects (Wilson, Sudol, Stephenson, & Stehlik, 2010). Gal-Ezer and Stephenson (2010) describe how in the US, "teachers are profoundly confused about the certification requirements in their own states" (Gal-Ezer & Stephenson, 2010, p.62)

For example, while CS education experts recommend CS TPP leaders design programs with state and national standards in mind—and many do—various CS education organizations have written differing standards, and only 35 states have formally established state-specific standards (CSforAll, 2021; Delyser et al., 2018; Schafer & East, 2022; Weissman et al., 2024).

State mandates for CS education have resulted in the creation of CS teacher preparation programs, although a limited number, and sometimes housed within a different subject area's program (such as Math) (Gray, Odom-Bartel, Zelkowski, Hamner, & Rodgers-Farris, 2020; Odom-Bartel, Zelkowski, & Gray, 2021; Toolin, Dion, & Erickson, 2021). Despite these mandates, enrollment in CS TPP is a common issue. Some programs take the approach of targeting for education students while others try to convert CS students. However, converting CS students has been difficult due to interdisciplinary cultural differences and competitive CS industry salaries (Bell & Lambert, 2011a; Yeni, Aivaloglou, & Hermans, 2020). Students from other content areas are also cautious about committing to CS TPPs due to preconceived perceptions of CS; pre-service students who dropped out of CS TPPs reported having negative experiences in CS education, low CS self-efficacy, and a sense of not belonging or being respected in CS communities (Everson & Ko, 2022).

Some researchers and CS TPP leaders have relied on research on general TPP design (Armoni, 2011). The book 'Preparing Teachers for a Changing World: What Teachers Should Learn and Be Able to Do' for example, details the development of TPP, although it also emphasizes how teachers benefit from learning teaching methods related to the specific discipline they will be teaching on top of general teaching methods (Darling-Hammond & Bransford, 2007). This is especially important for computer science (Ericson et al., 2008; Hazzan, Gal-Ezer, & Ragonis, 2010).

How CS TPP leaders navigate missing or new standards and navigate the tension between state mandates and limited resources is not well researched, nor is much known about how CS TPP leaders make decisions. The CSTA organization offers an implementation guide with advice on how to navigate various potential policy contexts, tips on designing a plan of study, and some ideas for student recruitment, but the detailed discussion of program design considerations focuses on a single program Rosato et al. (2022). The importance for more guidance was laid out by Hazzan, Gal-Ezer, and Ragonis (Hazzan et al., 2010) and included a design for a prospective workshop for CS TPP leaders, and although it was expanded into a book (Hazzan, Lapidot, & Ragonis, 2020), the focus there and in the programs was only on the proposed structure of, and content for, CS teacher preparation programs. That work is similar to other research on CS TPP programs in only briefly addressing how a program is started and sustained (Ericson et al., 2008). Several papers trace how they took their CS TPP vision through to a recently completed program design, but many strategic process details and outcome information are not included (Fadjo, Brown, & DeLyser, 2013; Menekse, 2015; Whitehead, Ray, Khan, Summers, & Obando, 2011). Other work describes the design of a single course (Pellet, Parriaux, & Chevalier, 2022; Ragonis & Hazzan, 2008).

Ko et al. has written about the design process of their own programs (Ko, Beitlers, Everson, Wortzman, & Gallagher, 2023b). Another relevant paper identifies factors impacting program development, including funding, stakeholder buy-in, and state-level policies and standards - all of which could either enable progress or introduce barriers (Liao, Karlin, Margulieux, & Calandra, 2025). The importance of motivated champions was identified in that work as a primary factor in program development. Little is known, however, about why the number of graduates in CS TPP continues to sputter across the landscape, why many programs fail to secure momentum, and how these problems can be addressed in the program design process.

2.1. Positionality

It is from both our positionality and through research that we understand CS TPPs. Our research team is composed of professors of varying ranks, Ph.D. students, and a master's student, all of whom are experts or participants in the field of CS education. Many of us are part of the population we are studying, and have lived experience and connections in the domain of CS education. Our team includes former secondary CS teachers, creators and administrators of CS TPPs, leaders of secondary CS program development initiatives, and in-service CS PD instructors. Our positions also provide a network for connecting with participants and crafting nuanced questions based on real issues within the field.

Our work explores CS TPPs and CS education in general by using the lived experiences of the program leaders to understand what happens between the idea for a program and the program admitting students, and how that helps explain the modest number of students who subsequently graduate.

2.2. A Framework for Examining CS TPP

Creating and implementing a CS TPP is a substantial structural change to the department that proposed it. Although the process of adding a new TPP is highly institutionally dependent, universities tend to have well defined, sequential stages for new program approval. We therefore looked for a change framework similar to the 'waterfall' software development model, instead of one which was more iterative and agile. After examining numerous frameworks of change and comparing them to what we had learned doing background research on academic program creation and during our own program design processes, the authors selected Kotter's change framework. Although Kotter later expanded this "rigid, finite, sequential" (Kotter, 2012, 5) process to be more agile and iterative, we centered Kotter's original eight steps (Table 1) of institutional change (Kotter, 1996). The first step is to *Create a Sense of Urgency* by inspiring people to see the change as important. Program leaders then *Build a Guiding Coalition* to coordinate and communicate activities. Next, program leaders *Form a Strategic Vision* for the change and *Enlist a Volunteer Crew* larger than the guiding coalition to complete the work. Program leaders also *Enable Action by Removing Barriers* and *Generate Short-Term Wins*. This is done so they can *Sustain Acceleration* with each success. They ultimately *Institute Change* by connecting the change to the successes. We suggest that this framework is useful for capturing the various shapes that we observe in the development and sustainability of CS TPP.

In part, this is because the different shapes of programs emerge as a natural consequence of TPP development and sustainment. Developing a TPP at a U.S. university is a complex process unique to each university as it generally involves gathering and using data to make decisions within an institution's organizational context (Dee & Heineman, 2016). An institution's organizational context includes its structure, external environment, culture, and power dynamics (Dee & Heineman, 2016). These factors should also be taken into account when designing the content of the program. Darling-Hammond created an often cited guide to TPP content development. The guide recommends creating content which maintains a consistent and coherent vision, connecting the content to practice, and grounding the content in common standards (Darling-Hammond & Bransford, 2007). After designing a program and getting it approved, programs require resources to be sustained. Typically this means either securing sufficient enrollment to sustain the program or soliciting outside funding. Later, established programs aim to use routine evaluation of how well the academic program is meeting its goals. This process is also unique to each U.S. university, but typically involves asking evaluative questions that probe enabling strategies and outcomes (Cranton & Legge, 1978). They determine what strategies a program is taking to meet its goals and specific outcomes, the results of which are used to improve the programs or justify changes to them - up to and including closing under-performing programs. Therefore, in this context of development and sustainability that CS TPPs are created.

This led to our belief that mapping the steps of Kotter's framework to program

Kotter's Framework
1. Create a Sense of Urgency
2. Build a Guiding Coalition
3. Form a Strategic Vision
4. Enlist a Volunteer Crew
5. Enable Action by Removing Barriers
6. Generate Short-Term Wins
7. Sustain Acceleration
8. Institute Change

Table 1. The 8 steps of Kotter's framework.

development and sustainment may be insightful. We suggest that the first four steps can be related to program development, and the last four to program sustainment. A leader establishing a new program creates a sense of urgency, builds a guiding coalition and a strategic vision, and then enlists a volunteer crew to get the program up and running. Once the program is established, the crew enables action by removing barriers and generating short-term wins. This sustains acceleration, which ultimately institutes change by solidifying the new program as part of the organization. While these steps are not always cleanly split between program development and program sustainment, our work hopes to highlight how this is often the case, and that establishing the first four steps is crucial to succeeding in the last four and, therefore, succeeding in program sustainment.

3. Methodology

Our core approach to investigating program creation, barriers, and strategies was direct conversations with faculty involved in the creation and sustaining of CS TPPs. While a mixed methods approach involving other sources of data (e.g., enrollment data, surveys of a broader set of institutions) would have enriched this study, we found during pilot interviews that these conversations exposed sufficiently rich nuances which we could focus on, saving broader inquiry for future work.

3.1. Gathering Participants and Performing Interviews

While similar studies have utilized a list of people invited to a workshop on CS teacher preparation for participant recruitment (Liao et al., 2025), we recruited participants by establishing a list of CS TPP leaders across the U.S. As a CS education researcher, the first author used her positionality to network, gathering a comprehensive list of CS TPPs in the United States, comprising universities from 31 states. Over several years, the author made this list using online searches, snowballing, related literature searches, and querying of CS education experts. The information gathered for each CS TPP includes:

- A Program's Type (explained below)
- Authorization Level (Primary, Secondary, K-12)
- Inclusion Status of a CS Pedagogical Methods Course
- Contact Information of Possible Program Leaders
- Additional Links When Available (e.g. Program Description, Course Catalog)

Initially, the first author started the list looking for single-endorsed pre-service CS TPPs, However, very few were found that were actively enrolling and graduating students. Further complicating the picture, many programs combined a variety of pathways and authorization levels in a single program, and or started out as pre-service but had primarily moved to other models due to low enrollment. Therefore, the types of programs included in this study were based on three inclusion criteria: programs had to be for credit, to be at universities of higher education, and had to cover computer science topics specifically. No minimum threshold was set for the number of students enrolled or graduated. The program types included were a mix of graduate degrees, undergraduate degrees, authorizations, licensures, certificates, badges, and endorsements (single, dual, and add-on). An exhaustive search resulted in information on 77 programs that met the inclusion criteria. Compare this to the 2023-24 Title II report which says 593 'Teacher Education - Computer Science' programs were reported in 2024, although only 49 programs graduated at least 1 student (US Department of Education, 2025). It is possible institutions reported a number for the CS category even if they didn't have a CS education program.

We used semi-structured interviews for data collection in this study. Semi-structured interviews are a useful tool for analyzing CS TPPs and offered two advantages. For one, the different shape of the programs meant that a fully structured interview style would have been infeasible. The design status of these programs also ranged from newly created to fully embedded in the university, to approved but otherwise inactive. Additionally, semi-structured interviews allowed the interviewees to perform probing. As stated by Barriball and While, probing allows for more exploration of potentially sensitive issues (Barriball & While, 1994). The creation of these programs involves sensitive issues such as disagreements among faculty and administration, so probing facilitated participant discussion of issues they may have felt l sensitive to discuss.

After obtaining ethical approval from our Institutional Review Boards, our team contacted individuals involved in 62 programs to request an hour-long interview. The interviewees — whom we will call program "leaders" — were people who worked on these programs, typically as creators and directors, but occasionally as teachers of the program courses who were brought in after the program had launched. Cognizant of our positionality, these contacts were primarily made by the professors on our team; the interviews were done by the professors and PhD student. We recruited program leaders to participate up to three times via email . Occasionally contacts gave the name of a new, more relevant individual, others declined the interview as they felt their programs were not active enough, and for some programs we were unable to determine a current contact. Of the people we did reach, 21 potential program leaders replied. Each leader was asked to complete a consent form and a survey with preliminary questions about the program they created/direct. All questions except for the consent form were optional. Of the 21 leaders who replied, 19 completed the consent form and were ultimately interviewed by the research team. Due to the possibility of sensitive content being included in the interview, team members asked program leaders if there was any information they would like removed from the study before publishing. One leader did decide to withdraw, resulting in 18 interviews.

To prepare for these interviews and refine the question set, the first author and a graduate student studying CS education completed five pilot interviews with a set of pilot questions. These five interviews were 40 minutes to an hour-long with program leaders personally known to the researchers. Information on all participating participants can be found in Table 2. After reviewing the responses, the first author refined these pilot questions to align with a final version structured on Kotter's framework (listed in Table 3). The interview protocol was also updated for the remaining interviews, which were completed by the 4 professors and Ph.D. student. These interviewees were chosen so participants, who were mostly professors, could discuss their programs with fellow faculty members. These interviews were semi-structured and, when time permitted, extended into questions about program content design and program outcomes.

After we completed the interviews, our team decided to include the five pilot interviews in the study, as the programs met our inclusion criteria, and the updates to the interview questions added clarity and depth but otherwise covered the same topics analyzed in this paper.

3.2. Analysis

The interviews were transcribed automatically by Zoom. The graduate student manually cleaned transcripts for accuracy. Then a team of three professors and the graduate student inductively developed themes for the interviews in multiple stages. Our team was composed of diverse professional and academic backgrounds to ensure multiple perspectives on program development were present for analysis.

Inductive coding (Thomas, 2003) offered a useful framework for analyzing CS TPP. In inductive coding, authors prepare raw data files, such as interview transcripts, through data cleaning. Then authors take an initial close reading of the text. They create categories or codes from this reading, distinguishing between upper level and lower level codes. Unlike quantitative methods, text may be coded with more than one code or no code at all. Lastly, the categories are continually refined through combination, creation, or deletion into themes. Because CS TPPs are a relatively new field, it was unclear what common themes would be present. Inductive coding provided data-driven insights that do not rely on previous assumptions about the subject. Inductive coding is also flexible, which is useful when interpreting the diverse interview structures created by semi-structured interviews. Within this framework, reliability is assessed through consensus building and member checking, not quantitative measures of inter-rater reliability (Elliott, Fischer, & Rennie, 1999; McDonald, Schoenebeck, & Forte, 2019).

We applied this approach to induce codes from our interview data. Our process sought out patterns in barriers and negative case analysis to highlight approaches and strategies used to develop CS TPPs. We did not seek to analyze the data in a way that confirmed to Kotter's framework — it's role was primarily to structure the interviews, not structure our analysis. First the interviews were divided randomly into the two groups, then two team members separately created codes for half of the non-pilot interviews, and the remaining two team members did the same for the latter half. The team then discussed the codes together and combined them into a total of 11 themes. Then each team member was assigned 4-5 interviews randomly, comprising all the non-pilot interviews, and assigned themes to quotes within them. The team then decided to drop one of the 11 themes as it was too ambiguous. The final 10 themes are listed in Table 4. Each team member was finally assigned 1-2 pilot interviews randomly to label with the updated theme list. From over 1,200 coded transcript passages, the team discussed which quotes best represented each CS TPP program leaders perspectives. From these, the first author and graduate student completed the final analysis detailed in the Results section.

Finally, we prepared a short report on our findings. To member check, we sent

Participant ID	**Region**	**Location**	**Size**	**Self-Reported Summary of Pathway**
ID-01	Midwest	Urban	16,000	Supplemental Credential
ID-02	Northeast	Urban	13,000	CS Endorsement
ID-03	Southeast	Suburban	40,000	Masters, Graduate Certificate, BS, and Undergraduate Certificate
ID-04	Southwest	Urban	52,000	Preparation for CS Teaching Certification
ID-05	Midwest	Suburban	50,000	Graduate Certificate
ID-06	Southeast	Urban	32,000	CS Endorsement
ID-07	Midwest	Suburban	25,000	CS Education Graduate Certification
ID-08	Southwest	Urban	52,000	Secondary Endorsement in CS
ID-09	Northeast	Urban	30,000	Initial/Professional/Additional Certification
ID-10	West	Suburban	34,000	CS Supplementary Authorization
ID-11	Northeast	Rural	3,000	Secondary Education 7-12 Master of Education and Certification*
ID-12	Northeast	Urban	4,000	K-12 CS Teaching Certification
ID-13	Northwest	Rural	12,000	Residency Certification and Endorsement in Computer Science (K-12)
ID-14	West	Rural	10,000	Endorsement
ID-15	Southeast	Rural	1,300	Secondary CS Program
ID-16	Southeast	Suburban	61,000	In-Service Teacher CS Certificate*
ID-17	Midwest	Urban	9,000	5-12 CS Teaching Endorsement
ID-18	Midwest	Urban	12,000	Computer Information Science Licensure*
ID-P1	Northwest	Urban	50,000	Secondary CS Add-on Endorsement
ID-P2	Northwest	Rural	24,000	CS Education Add-on Endorsement
ID-P3	Northwest	Rural	40,000	Connect Teachers in Region with STEM Certification Programs*
ID-P4	Northwest	Urban	50,000	CS Endorsement
ID-P5	Northeast	Urban	23,000	MS/Advanced Certification

Table 2. Summary of participant data, including the 5 pilot interviews and 18 non-pilot interviews. *Combination of self-reported summary and program information obtained from program website.

Interview Questions

(1) Who instigated the program creation process?
(2) How was the timeline thought about at the start?
(3) Who were the core designers?
(4) How was your vision for the project established?
(5) How were milestones and strategy for this project established?
(6) What resources [were / are being] consulted to develop the vision and strategy?
(7) What things did the designers of the program disagree on?
(8) How [did you / will you know] the design process is done?
(9) What was the number of graduates which was desired?
(10) What were the ways in which you provided announcements and updates to people
(11) outside the core organizing team?
(12) How aware do you think the larger organization was of your efforts?
(13) How did you invite people outside the core organizing team to get involved?
(14) What institutional barriers did you encounter in the design?
(15) Were incentives for all stakeholders aligned with this initiative?
(16) What were the most important things which kept momentum going?
(17) Was there anything which presented a real risk to the initiative failing?
(18) What milestones were planned?
(19) What have been your strategies for recruitment?
(20) What barriers have you encountered in recruiting?
(21) Overall, would you say you met your goals in designing and implementing the program? If not, why?
(22) In what ways has the institution embraced ownership of this program?
(23) Do you have any concerns about the future of the program over the next 5 years?
(24) What do you wish you'd known before you started?
(25) What support would have made the process better?
(26) If you could go back and do the design over again, what would you do differently?
(27) What advice would you have for other pre-service program leaders?
(28) Who else would be valuable to talk to about this program design?
(29) Are there any other documents about the design or implementation of the program you can share with us?
(30) Any other resources or information would be helpful to you?

Table 3. An overview of the possible semi-structured interview questions, selected based on program leaders' program structure and knowledge of program history.

10 Identified Themes
1. Precipitating Events: Passionate Champions
2. Precipitating Events: Perceived School Demand
3. Precipitating Events: State Policy Change
4. Process: Key Resources
5. Process: Challenges
6. Design: Challenges
7. Design: Goals
8. Design: Successful Features
9. Sustain: Challenges
10. Sustain: Successful Strategies

Table 4. Final themes used for labeling and analysis.

this report to program leaders, along with the quotes from their interview, and asked them to verify for accuracy. Program leaders either approved our interpretation, made suggestions for changes, or did not respond. Several requested edits to their direct quotes for clarity, which we made. Citing the challenges in discussing systemic problems while negotiating relationships across bureaucracies, one program leader asked that a colloquially worded frustration be rephrased; this was also done.

4. Results

This section details the analysis of the coding process supported by direct quotes from the program leaders. As we noted, we did not structure our analysis to confirm to Kotter's framework, and the results largely did not reflect Kotter's framework. Rather, the salient themes that emerged were broadly indicators of challenges that program leaders faced in moving through the stages described by the framework. That said, the subsections are presented largely in chronological order in a way that somewhat reflect's Kotter's stages — from program creation motivation to program sustainment — however, the order varied and sometimes the stages overlapped, and so the sequence does not strictly conform to the stages. This section begins with a discussion of the design phase of program development, where the barriers and strategies of three sub-phases are discussed in relation to establishing the first four steps of Kotter's framework. This section then considers the barriers and strategies present in the sustainment phase. We discuss how, when the first four stages are poorly established, establishing the latter four stages becomes difficult for program leaders. We illustrate this analysis by quoting program leader voices; the speaker for each block quote is identified by ID numbers (ID-01 to ID-18), and quotes from the 5 pilot interviews are have ID numbers starting with a "P" (ID-P1 to ID-P5). Table 2 shows more information on each participant.

4.1. Establishing the first four steps is difficult

We found the design process for CS TPPs could be broken into three phases: finding or creating motivation to start the program, securing three key assets, and designing the program's content. Program leaders would typically try to establish the first four steps of Kotter's framework in these phases. However, each phase had a number of barriers that prevented most program leaders from solidifying these steps. We discuss the strategies program leaders used and the barriers they faced during these phases below.

Program leaders had to find or establish motivations for creating a new program. Most of the CS TPPs examined had their creation sparked by recent developments in the field of CS education. In particular, perceived school demand paired with changes to state policy necessitated the production of more CS teachers. As a result, passionate champions often rose to the challenge of producing these teachers. However, despite building a guiding coalition, much of the work was done in small groups of one or two. This made it easy to be nimble, but difficult for them to move beyond their own sense of urgency to getting the larger institution to buy into a strategic vision for the program and invest resources into reducing barriers. This stage therefore foreshadowed the challenges they would later face in moving the program from an enthusiastic idea to a robust reality.

4.1.1. Program leaders were motivated by perceived school demand and state policy changes

One of the main drivers of program creation was program leaders perceiving a current or imminent demand for CS teachers in K12 schools. This could serve as a tool for creating a sense of urgency. For example, two program leaders explained it as:

> I mean, I think the big impetus is that, you know, this was a need of some schools, too. That we're trying to do more around coding and programming, and we need some teachers with this background, and could we look at folks that got their computer science degree and have a pathway for them to get a teaching license to do this. (ID-18)

> If we reach the point where every high school student in [state] is required to complete a year of computer science before graduating, there's going to be a significant need. We did some quick napkin math to estimate the need and [state] will require approximately 2,000 certified high school computer science teachers. (ID-12)

They also perceived a demand from in-service teachers for more CS professional development opportunities.

> I feel like teachers still need it. And there's an interest from teachers to do it, right? They're already signing up for a master's in educational technology. So why don't we develop on 3 courses for 9 credits that they can use towards their master's program and get a graduate certificate in K12 computer science education, right? (ID-05)

However, demand for CS teachers did not affect all schools in a state uniformly. For example, many rural schools simply did not have the resources to hire new teachers with CS expertise or give current teachers sufficient professional development.

> Like, sure, like the wealthy school district outside of [City], where [University] is at, offers CS and will continue to offer CS, but there's a lot of districts that aren't gonna. That experience more scarcity in their resources and aren't gonna offer it unless they feel they need to. (ID-09)

State policy change was another common motivation for these programs to be created; for some programs it was the primary motivator. The policy changes instigating CS TPPs included new or expanded CS endorsement options, a requirement for high schools to offer a CS course, or a CS graduation requirement.

> [Program leader] was on this as soon as the opportunity became available from [state]. In fact, we had been hearing that the state Board of Regents was planning to move in this direction, and [Passionate Champion] had been preparing things in advance. So when

> we could finally send in an application, hers was ready and it went in about as early as possible. As a result, we were the first program they approved. (ID-12)

As excited as program leaders were to have state policy motivating an increase in CS Education, they sometimes found the timelines the policy created to be challenging.

> Yeah, so we... it was like a mad dash, actually. We had, I think it was like 2 weeks once the Professional Standards Commission of [state] said that this new rule was in place to when, like, the curriculum proposals for the University were required.(ID-06)

4.1.2. Passionate champions lead the way

The term "passionate champion" can best be described by a quote from the interview with program leader ID-01:

> ...there's a name for the person, but like there's the person who's like passionate. This is who's gonna kind of push it, right? (ID-01)

While changes to state policy and perceived demand in schools for CS teachers could be the impetus for program creation, programs were created because someone believed it was important to have more highly qualified CS teachers. These people played the role of passionate champions. Whether one person or a small group of people, these passionate champions were able to leverage the growth in demand for CS education and updates to state policy to create a sense of urgency. Typically, they got involved not because they were assigned or appointed, but because they stepped up into a leadership role to launch a program proactively.

> The idea of starting the program and its design came from us, not from higher up in the college. Everyone who was working on the program design was a volunteer... we assigned ourselves. (ID-12)

These passionate champions were one of the most important drivers of a program's successful design and initiation. They came from a wide range of Computer Science and Education backgrounds. Typically university professors, they were also higher education administrators, education nonprofit STEM directors or former high school CS teachers who had come to work at a university's TPP.

> But I so passionately believe in what I'm doing, that that's why I have not just thrown the towel and left. Because sometimes it's hard, and sometimes when you feel that you're not even appreciated, and when you hear people talk about your program being just another thing that can be on the chopping board after all the hours, they don't realize, the hours it spends for me to be making every course I wrote. (ID-02)

These champions led with a sense of urgency and determined who was in the guiding coalition. Collaborators recruited by passionate champions included faculty for program design, industry and school district connections, people in administrative positions, and cross-disciplinary staff.

> So I would call myself an intermediary. As somebody who is working in the region, my role was as a director of strategic partnerships for [education service district] during this point. And that is literally somebody who's building strategic partnerships across K-12 industry community-based organizations to get to a certain goal or outcome. (ID-P3)

Passionate champions also typically established the vision. They often took great ownership over the entire project, assuming many roles and moving quickly. However, their tendency to be creative outside-the-box thinkers who moved quickly left some parts of Kotter's framework underdeveloped. For example, often the vision was informal at best, or written down but poorly communicated. These under-realized Kotter's framework stages would later create problems:

> The vision was to have an approved program by [state], and then to try to get students to complete it. (ID-12)

> I felt like it was gonna fall apart the entire time. Like, is this course even gonna happen? Yes. Lack of communication, lack of clarity, lack of vision, or lack of vision being communicated clearly and regularly to everyone who needs to be involved, lack of staff time. Yeah. It was like, there were so many hurdles, I'm surprised that it got off the ground, honestly. (ID-04)

Once a motivation was established, program leaders needed resources for the program. While the actual processes and milestones varied, leaders reported it was essential to establish three key resources: administrative/institutional support, funding, and a team with diverse skills. These resources could be established in any order. However, there were often barriers to establishing these resources. Actors within a program's institution, such as administration or other departments, could inhibit progress. Funding for various aspects of the program were also difficult to secure. Finally, teams suffered when they didn't have diverse talent pools for navigating a litany of program creation challenges.

4.1.3. Program development was easier with administrative and institutional support

Acquiring institutional approval and administrative support was rarely a straightforward process. For this process to work, the administration had to be, at a minimum, cooperative.

> So one of the reasons that I really like working at [University] is we don't have a lot of institutional barriers. Like you can be like, 'Hey, this thing came down and I think it'd be a really good idea for us to jump on it. But I need a lot of help, and maybe some deadline extensions.' And they're like, 'Yeah, totally.' (ID-06)

Sometimes getting through the approval process was easy, such as when administration trusted the program leaders or were excited about the project.

> The department trusts us. (ID-12)

> ...my department chair is a huge fan of the program. (ID-09)

However, a lack of support — or even indifference — from administration could exacerbate barriers. Program leaders sometimes felt discouraged by the amount of bureaucracy.

> I really wish we would have had more participation from some of the University administration. We have been fighting battles with the registrar and the financial office... we have to pay [the students'] tuition bill, but at [University] it's like paying taxes. They won't tell you how much it is, so you send them a check, and if it's wrong, they'll tell you it's wrong. (ID-07)

The interdisciplinary nature of CS TPPs usually required cross-department coordination, adding additional complications and barriers.

> ID-P4: "We made an explicit strategic choice to not share anything with anybody in the College of Education at the time. And still I think the College of Ed is pretty anti-big tech and anti-computer science really and that was the vibe that we got when we were originally talking about it with the faculty, too.... I didn't tell computer science either, other than like "You promised us one course load. Don't forget." Interviewer: "Do you

think they were aware of it?" ID-P4: "They're aware of it now. And I think part of this strategy was a little bit of FOMO. We wanted the college of Ed and computer science and engineering to feel like they were just kind of missing out..."

Things kind of fell apart with CIS, which is our computer science department. They just had complications and it didn't work out. So we ended up taking full possession of the product. (ID-16)

Issues of ownership also emerged:

And then, when computing education sort of became a thing in [state], CS was like, 'Oh. we want to own this.' I'm like, 'Uh-uh-uh. You signed off on it. And I have emails to prove that.' (ID-05)

By nature of our inclusion criteria, we did not talk to anybody from programs which were unable to overcome the approval process, although we are certain there are many. However, in the end, even though the programs were approved, the process could be so exhausting and frustrating that some passionate champions left the program, leading to fragile longevity.

...one of the people who really wanted this program to happen, [Professor], she left because of also not having that support. (ID-02)

4.1.4. Securing funding sources was critical

Every program we talked to was concerned about funding. This funding could be internal to a department, such as faculty time being freed for program design work, or low enrollment in courses being tolerated, or could come from wider university administration. However, external funding was often necessary as well - sometimes campus administration required it before offering any internal administrative support. Leaders tapped a variety of external sources, including federal or state grants, corporate sponsorships, and philanthropy.

The [University] Office of the President has had a big funding pot for developing fully online courses that can be offered. (ID-10)

And then this [computer science education] initiative was started in 2015 with a big NSF grant, the big CSforALL push, you know? (ID-08)

Seeking funding from industry wasn't always easy:

We actually started with philanthropic fundraising efforts and failed at all of those. We got pretty close on some of them. That was a couple of years of effort and failure. (ID-P4)

Many programs relied on national and state grants to complete essential stages, or to get students in the door.

Initially we had grant funding that covered the costs such as the tuition for the teachers who were taking our courses. (ID-12)

These grants could also double as a commitment device for program leaders:

You're committing, maybe for 3 years to work together to spend the money and make progress on something. So it's kind of like a commitment device for everybody to say, we're going to work together for multiple years and get this across the finish line. If there's not money, there's no commitment. Momentum fizzles. (ID-P4)

4.1.5. Building a program took individuals with specific skills and roles

As discussed above, program leadership was critical. Program leaders typically built a guiding coalition opportunistically from people they knew were interested in the mission. However, a small guiding coalition alone could lead to issues like lopsided role delegation where one person was taking on too much responsibility.

> And then, when you're having your contract, have it in such a way that you understand: don't be the creator, the educator, and the recruiter. (ID-02)

One of the most helpful strategies for getting a new CS TPP started was enlisting a broader volunteer crew with the right mix of skills. Critical skills included expertise in program development, CS content, teacher education content, and of particular value, specialized expertise for designing and teaching the computer science education courses. Finally, collaborators from administration, industry, and school districts were also highly valued.

> ...you've got to have somebody who knows the CS, because probably nobody in your teacher ed department knows CS. And you need somebody at an administrative level who can help you get the courses in the program approved because the faculty member who's just excited about it... like you need administration. You need somebody in administration. (ID-01)

Lacking any of these elements, such as someone with program creation knowledge, presented a hurdle in the process. A poorly chosen team member could even inhibit progress.

> All that I created was not approvable because there's a system, in a way, but I did not know that there was a system in that to create courses. So I did learn that. And then I had to redo, so I lost one year in that. (ID-02)

> She's kind of one of those faculty members that commits to things but doesn't actually mean it. And so, she just never really showed up and helped. (ID-P4)

On the other hand, when teams were large, people could get confused by who was responsible for what. This was especially true if program ownership was unclear.

> And I think it suffered a little bit from, like, lack of ownership. Well, it wasn't totally clear who was supposed to be doing it. (ID-04)

Beyond the resources required, a program also had to be designed. Our data showed that this was a challenging process because there were few CS TPPs to reference when creating a new one, and little in terms of research to support many parts of the program design. Although the guiding coalition may have started with a big-picture vision for an 'ideal' CS TPP, in reality the details of the program design had to address many unanticipated barriers.

CS TPPs have unique challenges which needed to be considered by the guiding coalition when they designed the content of their program. Darling-Hammond created a commonly referenced guide for TPP design in Chapter 11 of Preparing Teachers for a Changing World: What Teachers Should Learn and Be Able to Do (Darling-Hammond & Bransford, 2007). In this section we compare and contrast the program design decisions our program leaders described against of more typical TPPs.

4.1.6. Maintaining a consistent vision was elusive

Darling-Hammond emphasizes that creating a coherent program requires a consistent vision of teaching (Darling-Hammond & Bransford, 2007). The program leaders we talked to had innovative and powerful visions for CS TPPs. One leader, for example, designed a course in the program with specific learning theory at its heart:

> ...it was basically exploring the premise of like, 'what if we sort of took Constructionism seriously again, in computer science?' (ID-09)

Many program leaders had explicit equity goals, seeing CS TPPs as an opportunity to fill in a gap:

> There weren't any justice focused CS programs in the country at that point. (ID-P1)

However, program leaders sometimes had to compromise their vision to meet requirements from the state for teacher certification. They also had to compromise the vision to accommodate the realities of the students who would enroll in their program, a concern which was particularly acute for programs which served in-service teachers. Sufficient time and bandwidth of in-service teachers to deeply learn content knowledge had to be considered:

> ...because we want teachers to be more experts in the things that they're teaching and not just have a 1 week workshop, and now, all of a sudden they know enough to teach computer science. And honestly, I think that's where the real challenge is for teachers, is just learning computer science. You're teaching your brain a different way of thinking. (ID-07)

> ...everybody's getting these masters that are shorter and shorter. They're like 30 credits now, instead of like 36 or 39. We can't make it so expensive that they can't do it either. So we had to really look at it. And we said, 'Okay, the methods, though, is really important. And also their field classes and their student teaching that has to be in certification area.' (ID-11)

Maintaining a coherent program was especially difficult in CS. Leaders noted that the field of CS is rapidly changing, causing CS TPP creators to need to constantly adjust the course content to keep it relevant:

> Any teaching impacts of technology, the course is gonna have to iterate. It's gonna, there's, you know, generative AI changes the game. Like that's 2 entire weeks now of this new course. (ID-10)

4.1.7. Connecting courses to practice was valued but difficult

In addition, Darling-Hammond recommend that courses should be connected to practice and placed within the context of the discipline (Darling-Hammond & Bransford, 2007). Many of the program leaders interviewed expressed a desire to do this:

> Yeah, I would say that if I were attributing some of our success to anything I would say it was our laser focused attention on structures, power, resources, and really understanding what was possible within those. That's what made this possible. Had we just focused on designing courses and teacher learning in those courses, like on the ground kinds of things, the beginning, and not paid attention to any of those structural things, I don't know that we would have been able to build it. (ID-P4)

Program leaders sometimes accomplished this in methods classes. These classes focus on CS pedagogy, providing students the opportunity to explore how they want to teach CS:

> Special Methods class... for about an hour they get this kind of general education approach, or whatever, and then the separate subjects all break out into different rooms. Each of them has a subject expert... And we would have specifically talked about, you know, what does assessment look like like? They've already covered assessment. How does assessment look different in CS education? What does discipline look like in a CS class, resources needed. (ID-15)

They also connected content to practice by building students' prior knowledge. This was especially useful for in-service teachers whose teaching experience could be used to help meet their preference for shorter programs:

> Another strategy is, because they are involved in teaching through another program, or as in service teachers, we really try to leverage like what they know about teaching from their field. (ID-06)

CS content knowledge was also a core component of the practice. Programs often required students to take cross-discipline courses to obtain this knowledge, such as introductory CS classes. Some tried to teach both content knowledge and pedagogy in the same class, but this required heavy interdisciplinary cooperation, which could be a challenge:

> We had sympathy and support from the computer science and engineering department, but we were not able to find practical ways to collaborate on this. Their priorities, as I see them, are not well aligned, that they're not interested in teaching courses to graduate students that are introductory CS courses. They have not prioritized CS courses that are attending explicitly and intentionally to teaching pedagogy. (ID-09)

This was especially difficult because program leaders recognized that their programs could not fit everything that should be taught in the number of courses they had:

> And then the tricky part is, how do you balance developing... the need to develop teacher's CS content knowledge as well as their pedagogical content knowledge, right? (ID-05)

> Recognizing you cannot cram a whole computer science undergraduate education into those 4 classes. (ID-10)

In some cases, this led to content inclusion disagreements among team members:

> ...maybe not like strong disagreement, but a lot of discussion and debate around what courses would be required in the pathway. Which CS courses and how many of them was a big question..." (ID-04)

Another consideration unique to CS education was Career and Technical Education (CTE) certification. Teachers in some states could teach CS with a CTE certification, and this shaped the expectations schools have for incoming CS teachers. However, CS did not directly translate to CTE:

> I think that CTE at the state is actually a bigger issue... Computer science is just one example of it. Currently, there is not a straightforward pathway for an undergraduate student to become a CTE certified computer science teacher. A student that completes majors in education and computer science can earn the state endorsement in computer science, but since that is not a CTE endorsement, they will have to graduate, start working, and then go back to school for CTE to be certified to teach most computer science courses that are offered in schools. A student that graduates with an endorsement in the CTE fields of technology education or business and marketing education can teach some computer science-related CTE courses, but their training is likely to be limited as, business and marketing education, for example, is a very broad field that also requires training in accounting, marketing, personal financial, management and other areas. (ID-

13)

4.1.8. Grounding programs in common standards was more challenging for CS

Finally, Darling-Hammond suggest programs should be grounded in common learning standards (Darling-Hammond & Bransford, 2007). In CS in particular, standards are "crucial for distinguishing between computer science and digital literacy" (Weissman et al., 2024), however, as discussed earlier, not all states have formally adopted CS standards. Program leaders commonly consulted a wide breadth of available standards while typically considering state specific standards to be preferred.

> We looked at standards we looked at like, for instance, we looked at the praxis. We looked at CSTA. We looked at ISTE. We looked at [state standards]. We looked at... we looked at any standards that were out there and tried to inform what we were doing. (ID-16)

> Yeah, you know, what would be really awesome would be the [state] having some learning standards for K-12. Because then we could have some teaching standards or methods, classes that could map to what we want our students to learn. (ID-P2)

However, program design could be heavily guided or restricted by standards and requirements. Having someone on the team with teacher certification skills was very helpful at this stage.

> ...so much of the design of it was dictated by like the standards and sort of the constraints on the program. (ID-06)

Whether programs had restrictive or helpful standards, deciding what content to include was made more difficult by a lack of existing resources. Leaders described CS education as a nascent field, meaning they had to create many of the resources they used from scratch:

> So, I didn't have any other programs to mirror on. (ID-02)

> We prepared all our documentation, generated syllabi, coursework, and then everything. (ID-03)

Some program leaders even took it upon themselves to create planning documents for the process:

> I've basically written up a handbook, which no one told me I had to do, but I felt like it was a good idea... (ID-09)

Due to all of the above factors, many programs had not properly established the first four steps of Kotter's framework by factors by the conclusion of the design phase. This created many challenges when transitioning to the sustainment phase.

4.2. Program sustainability was a central concern

Many CS TPPs became centered around program sustainability. Leaders had to consider both factors outside their university and factors specific to their university and departments. They also gave considerable thought to having a wide potential audience, both in terms of student backgrounds and their desired pathways. The recruitment strategies they discussed were, likewise, creative and extensive. Most of the programs we talked to felt their future was very uncertain, and were deploying a wide variety of creative program designs to boost enrollment, with mixed results. The composition of the design and implementation team often had a major impact on these efforts.

In this section, we discuss the external factors which affect these programs in ways which support or threaten their existence, administrative context and factors which impact the programs' future, who these programs are recruiting and how they are getting the word out, and the role of the core organizing team in the programs' long-term sustainability.

4.2.1. Teams suffered in practice due to composition issues

In contrast to the approval process (which often had clearly defined stages to 'check off'), the passionate champions often struggled to maintain a guiding coalition for the program implementation. Teams were often passionate and nimble, but it was difficult to find people with the requisite combination of skills:

> There was a new CS faculty member that was interested in taking over the CS seminar course. Then I think she went on sabbatical. So nothing happened with that. We had a big meeting with myself, Dr. [M], and some CS faculty; and had this conversation about them, CS, taking over that CS seminar course. I guess there's nobody in [our teacher training program] that has a computer science background to teach the CS seminar course. (ID-08)

This challenges in creating the guiding coalition created a lot of concerns about how brittle the programs were in regards to key personnel:

> Unfortunately, since I left [university], that program has sort of, you know, phased out, and [core team member] retired. I left, and I don't think the methods course has been taught, I've been gone 10 years now, since I left, so... (ID-05)

> I mean, honestly, it was me and that high school director. But then he left. He left the high school to take a job in... kind of education industry a few years ago. So yeah, I'm not really sure that they're would be anyone. And the professor who, I think, got it on the books like a couple years before I got there. She's also left. She was the one we tried to hire to replace. So yeah, I think I'm the remaining remnant of this initiative at our school. (ID-15)

These small teams were expected to be able to navigate academic bureaucracy, design and teach new interdisciplinary courses, and recruit for the program. This last part in particular was a challenge for many teams who were unaware that there was insufficient resources to help them with recruitment:

> So I find that very strange that I'm expected to be the person going to recruit. I would expect the University to do that part. I've done all the other work. So I'm not very happy about that part. And the reason why they are not all about it is because nobody really sat down to understand the importance of computer science for all. (ID-02)

> And how do we recruit efficiently cause everybody's stretched really thin. And like, especially for this program, I mean, it's this kind of a tiny program that is this subset of anybody's work. So like the guy who's the chair of the graduate program committee and I are probably our main recruiting people. And we're both like, 'yeah, that is one of 800 priorities.' (ID-01)

Some teams were fortunate to have robust, experienced compositions:

> We're networked with and different committees we're on and do the sharing out both strategically. But again, also relationally, because I think it matters that we're not new to doing teacher education. We're not new to creating courses or structures. And we're not new to even CS as as a group. Maybe individuals - we don't all have all of that knowledge. But together, I think we bring a lot of strengths in, and we'll use those strengths to do

that sharing out. So it's really authentic and not, 'here's the flyer'. (ID-P2)

The ratio of work required to available time was often a major barrier for the program design and implementation:

> If I knew what a quagmire computer teacher preparation was I don't know that I would have done it.... that's one of my like... why I keep trying to figure out if we need to offer the endorsement, or if we can just focus on this sort of integrated computing aspect is because, like, we're only serving, you know, on average 10 teachers a year. And the amount of things I have to do as the program coordinator are just ridiculous. For what we're getting out of it. (ID-06)

Likewise, the programs had to consider what faculty would be available to teach before requiring any specialty courses be added to the program:

> (Interviewer) : "You've described a lot of courses. I can't imagine you're teaching every single one of them. Maybe you are?" (ID-09) "I am right now, but I can't keep doing it, right? Like, I can't. I am right now, because we've only had to have one or 2 per term. So there's a couple of strategies that I think can reduce that. I mean, you've identified a key issue. This keeps me up."

In regards to teams members from outside the university, program teams varied in whether or not they included (or consulted with) members of the K12 community during the design process:

> You also need to think about the community partnerships that will help to sustain the learning and environment, and the administrative support or 'buy-in' that is needed be able to sustain it. You have to figure out how we're going to engage parents and the community. So it's a design challenge for the entire ecosystem. It's not just as simple as we design the perfect curriculum for this learning journey. We shouldn't do systems thinking work in isolation. (ID-P3)

4.2.2. Sustainment is heavily influenced by external factors

Computer Science teaching is also impacted by the decades long trend of flagging interest in teaching careers (Kraft & Lyon, 2024), especially since many CS Education programs require most of the same classes a CS major does, but with stark differences in salary upon graduation:

> "You have to be a CS major to do the program, and our CS majors have a lot of other opportunities available to them, high paying opportunities" (ID-12)

> The barrier is that, as a computer science person, you can go make a decent amount of money without having to have a master's degree in education. Why would I? I mean, the student even said to me, he's like, 'I've got a $70,000 job that I can go do. Like, I kind of like this idea of being a teacher. But... I'm gonna get a master's degree and then go work for $35,000 at a school district?' (ID-18)

> "There were a few other students who were open to the idea, but they weren't real excited about the salary. So they were comparing industry salary versus education salary. And a lot of these were actually international students. We have a lot from Ethiopia and Nepal. And so, I think they're, they had obligations to their home culture to send money back, and kind of, they were on a mission to get maybe an industry job, even though their personality would have been great as a teacher." (ID-15)

Even in areas where teachers are paid better, students are sometimes reluctant to enter the education field due to the way culture and politics have impacted public perception of and support for teachers:

> “The big one, you know, the elephant in the room is basically the national you know, if they stop that national war against public school teachers that would probably help.” (ID-P5)

Efforts to expand the CS Education profession face additional hurdles: despite perceived demand for CS teachers being a primary instigating event for the creation of teacher preparation programs, the complex interplay of state policy and on-the-ground logistics of schools often made hiring dedicated CS teachers trained through programs at traditional institutes of higher education untenable. One aspect of this is a school’s ability to meet recommendations or requirements to add additional courses in student schedules:

> “If we teach CS what are we going to take away. And we already like, you know school’s already hard enough like, we need to stop just adding.” (ID-P1)

Where CS classes are K-12 electives, schools had the same challenges as they do with other elective areas: being able to hire a dedicated teacher when student demand is uncertain:

> “I think one of the big risks, or maybe the most vulnerable place is that nothing is mandated by the [state]. So even though we have a wonderful CS implementation plan, and it suggests teacher education programs offer something, and it suggests a requirement of 2 courses offered at each high school which will require teachers. It’s not law, it’s not legislated, it’s not [policy]. Yet. It’s a recommendations report that is going to the Board of Education and the Governor and everybody seeing it but the the reality is, while there is both a pressure to have CS teachers in the schools, and everybody’s talking about the need. Education is always running on scarcity and and moral panic and other political backlash.” (ID-P2)

Even where CS classes are required by states, the enforcement mechanisms either are not there or not used - possibly in recognition of there not being sufficient pipeline of teachers to meet the mandate:

> We have these standards without testing or enforcement. (ID-09)

Insufficient or non-existent enforcement joined several other complaints from our CS teacher training program leaders regarding state policies. Often, the policies were not designed with realistic timelines or infrastructure to create and support the roll out of new CS teacher pipelines. Therefore, the state level policies created loopholes around the ‘catch-22’ of CS education in ways which undermined efforts at a systemic solution: you cannot mandate that schools teach CS if there are no CS teachers, but there are no CS teachers because there are no CS teacher training programs, and there are no CS Teacher training programs because schools aren’t required to teach CS. Meanwhile, institutes of higher education that attempted to create traditional teacher preparation programs for CS find they are unable to attract sufficient student enrollments because schools are not hiring CS teachers because they are not required to. Loopholes and work-arounds were therefore created, most of which allowed teachers of other subjects to ‘add on’ CS in ways which either require more time and money than was palatable to most candidates, or were so paltry in content that most program leaders do not feel prepared to successfully teach CS, and thus only did so for a short time. This situation made it difficult to advocate for the program to University administration:

> It’s sort of a chicken and egg kind of issue. You need to have an audience before you can start selling a program to your administration, but until you start talking about a program, you don’t have an audience. (ID-17)

Sometimes states attempted to resolve these tensions by encouraging CS integration into other subjects. Some of the CS TPP leaders we talked to thought that add-on endorsements were the most realistic way to address the CS teacher shortage and designed for it. But some were skeptical:

> "'Oh, just make it interdisciplinary. Weave it into your courses.' Like, sure, that's going to just end up watering it down to the point where, whatever you happen to be doing, you just check the box and you're done." (ID-17)

As part of describing the need for policy that is more aware of the larger context of CS Education teacher training ecosystem, our program leaders often described state-level financial support that was insufficient or unreliable. Funding was seen as being necessary for the programs to be designed and administered for student scholarships and for schools to be able to hire the graduates. Funding which was available was often intermittent, short term, not aligned with K12 or University fiscal calendars, or subject to shifting political priorities. Program leaders reported that this frequently resulted in only schools in wealthy districts being able to expand their CS Education program.

Another barrier program leaders reported was the complicated and often changing process for program approval and modification. Navigation of these processes typically had to be done with little assistance despite program leaders often having no prior experience.

Other external factors mentioned by program leaders as being challenging included how differences between state policies and requirements limit the ability to create resources which could be shared, the frequent changes and cost increases of proprietary software and websites used to teach the CS Education curriculum, and shifting goals for CS education in regard to content coverage:

> "This present enthusiasm for AI, a lot of people I've talked to have this feeling of like, 'Hold on. We haven't really gotten the CSed thing figured out. Maybe we don't need to jump on the next thing yet.' (ID-17)

These external factors made it difficult for CS TPP leaders to generate short-term wins which would sustain the program's acceleration.

4.2.3. Internal university factors also impact sustainment

The factors outside of a university became entwined with internal factors in determining how much an institution changed to embrace their new CS TPP. This played out primarily in the form of budgets constraining administration's ability to allow programs and classes which have lower enrollment numbers. While the passionate champions believed that the need for CS teachers would ultimately result in their programs growing to a respectable size, they also knew that it would take some time to resolve the various barriers and build up a recruitment pipeline:

> I haven't had to get any like, dispensations yet to run a course with 5 students in it. But will I? I, that I don't know. We'll see. We'll see what our numbers look like coming up. [my department chair and I] haven't talked through it yet, but like, what's the runway like? Is there? Is there an explicit like, 'Here's how long you have to like, get the numbers up to a sustainable number, or like we can't, we can't sustain the program.' (ID-09)

For many programs, the resources available for these CS TPPs to be approved, designed, and launched was subsequently found to be insufficient to also tackle the external factors which hamstrung recruitment efforts. A double bind could arise, where departments or universities were reluctant to allocate resources to a program with low enrollment, but the biggest limitation in increasing enrollment was a lack of resources:

> The problem that we're facing is, when you're only able to get 4 students to move to the next course, you have to ask 'will the university cancel it from underneath me?' And they have done that about 50% of the time. So that's been a big problem. If you're going to start this kind of program, it's going to start slowly more than likely. And I keep telling my Dean and my Provost 'Look, you canceled this course, and we delay offerings to students. The word gets around that this is a shaky program and we'll never build it.' (ID-17)

One way programs mitigated this issue was limiting the number of classes which were unique to the major (such as CS teaching methods courses) and instead relying on a combination of traditional CS undergraduate classes and methods courses which weren't subject specific. Some programs were structured in forms such as a graduate certification which generated extra revenue. Other program leaders said they were able to get permission to run lower enrolled classes by also running classes with very high enrollment in other content areas, or by offering the CS Ed classes in an asynchronous, online format that allowed them to draw in hundreds of in-service teachers across a wide geographic area. One program leader avoided the issue by being very careful to obfuscate the enrollment numbers:

> Interviewer: "How has it survived from being small?" ID-14: "They don't see it." Interviewer: "You you hide it." ID-14: "Yeah, so that computer science class - the methods class is only offered when I have a student. It doesn't exist. I taught it as an overload. yeah. So they're mixed in with all these other people. And we put it as a, instead of a degree, it is an endorsement. So what that means is, it's tracked at the graduate level. And, and you have to just sign up for this endorsement. Say, yes, I had this class, if so, that the they can check it. But you have another degree that you're coming out with. It's hidden. And that's scary to have to say that."

For a number of programs, external funding was essential to help get things off the ground:

> The other thing I would say is, at this stage, you're gonna have to find some philanthropic or district level money or something to get you going. Because most most colleges of education aren't in a position to be like, 'Oh, yeah, we're gonna do a high risk program right now.' That's just not happening right now in colleges of Ed. (ID-01)

CS TPP leaders also faced internal challenges beyond enrollment numbers. Their small, passionate teams could at first operate with independence and nimbleness. This was appealing compared to methodically moving stakeholders through stages of organizational change. The risk in that strategy, however, was apparent when programs later couldn't successfully build momentum towards true institutional change due to bureaucratic snags such as required courses housed in different departments being scheduled at overlapping times. These nascent programs sometimes found they simply didn't have enough buy-in from the various administrative teams to negotiate the required flexibility.

> "So I create the program, I teach in the program, and then I'm the one recruiting for the program. I mean, that is a little much for one person, you know?" (ID-02)

Finally, sometimes the program leaders found that their innovative efforts simply didn't fit neatly into existing incentive structures. One leader received negative feedback on an early career award application because the program was still small. Another found that their program outcomes were being assessed at a 'check the box' level without any meaningful feedback and little interest in updating them to better meet student needs. Programs which were cross-departmental sometimes found neither department wanted ownership and its attached obligations. And in some places, creative attempts to create opportunities for in-service teachers and other subject teacher candidates were discouraged because the number of required classes was too small to reach the threshold of a full endorsement or certificate.

> "I mean, I should do it this year, but I also should like, make more papers to finish with tenure this year, so I probably won't do it this year." (ID-09)

4.2.4. Program leaders utilized enrollment focused features

> Factors affecting program viability: Generally, programs that have survived and maintained their enrollments have done so by remaining flexible in how they seek funding, in how they recruit, and how they package their programs (Darling-Hammond et al., 1989).

For many of program leaders, the aspect of the program design they became centrally concerned with was not aligning student learning outcomes to the demands of the job, or ensuring equitable teaching practices were covered in multiple classes. It was designing the program to accomplish the required enrollment levels to keep the program alive. A wide variety of strategies were used in an attempt to reduce barriers that might prevent potential students from enrolling, and in these we see many of the same tensions discussed earlier. While approaches varied, they can generally grouped into three categories: designing a program which can serve a wide variety of potential students, choosing program design elements which reduce barriers for enrollment, and using extensive and creative recruitment strategies.

4.2.5. Recruiting methods were broad

Whatever the original audience had in mind, leaders of CS TPPs often realized that the traditional audience of undergraduate students who know they want to be CS subject area teachers was too small to sustain a traditional teacher preparation program. Therefore, they designed programs to serve multiple pathways.

At the undergraduate level, CS teacher preparation programs tended to target either students who were already candidates for a different TPP content area, or CS undergraduate programs. For the former, some programs created an "add-on" endorsement, while others looked towards integration of CS topics into other content areas. One benefit of the "add-on" approach was candidates job prospects would be improved with multiple endorsements, and that "add on endorsements" were an already well known option. For both "add-on" and integration, however, the challenge described earlier of 'how much CS' was a difficult one to resolve. Typically, the education students did not have any prior experience with CS. Neither did the state policies typically require any university CS credits; instead, passing a content area exam was often sufficient to earn an endorsement. As a result, some candidates had little motivation to go through a formal program when they could, for much less work and expense, self-study enough pass the exam at the time of their choosing. Leaders of CS TPP, however, often considered this option as being insufficient for candidates to gain either the CS expertise or the CS PCK to make an excellent CS teacher capable of teaching broad set of CS classes. It was this insufficiency, in fact, which was one of the motivations leaders had for designing their programs.

> I'm not sure how effective a strategy it is, because, like, yes, like they bring lots of

pedagogy from their own disciplines... but also CS has its own pedagogical strategies... it is an ad hoc solution. (ID-03)

At the other extreme, program leaders considered it to be unrealistic to ask a student who already has the TPP requirements for a different content area to also take a full CS undergraduate degree. Many programs therefore specifically targeted math education majors with the hope that they would have had more technology background already under their belt.

Recruiting CS undergraduates into a CS TPP, on the other hand, had other challenges. The pay disparity between CS and Education was a major sticking point for many students. Several program leaders found the CS student body had been encouraged to have a laser focus on preparing for the workforce for so long that they were not particularly receptive to appeals to idealism as a motivation to switch to teaching. Another issues program leaders articulated was that many students were already quite late in their degree program by the time they realized that they might be interested in education (or that it was even an option) at which point switching to the education course sequence would substantially add on to their time to degree.

Recruitment efforts targeting CS undergraduates wasn't limited to those currently enrolled. The CS TPP leaders we interviewed discussed how they also targeted high school students or recently enrolled university students who intended to major in CS, or even those who had just graduated and would be open to a post-baccalaureate program. Interestingly, while Darling-Hammond's 1989 text on recruiting candidates to STEM teaching careers highlighted that retirees were a often targeted demographic, none of the programs we talked to mentioned this group (Darling-Hammond et al., 1989). A number of the leaders we talked with, however, did want to enable pathways for mid- career CS professionals who were looking for a more fulfilling career or junior developers who had struggled to get a toehold in the current challenging industry hiring market.

As for graduate students, a strategy which appeared to be less common (and less successful) than recruiting directly to CS post-baccalaureate, Masters in Teaching or Master of Arts in Teaching programs was to target students already enrolled in other graduate programs (either CS or Education) with an appeal to switch to a CS Education masters program.

In-service teachers were the other main population these CS Education programs attempted to recruit. Program leaders reported that in-service teacher motivations varied. Some were looking for increased pay or career advancement, others were looking for personal fulfillment and some were CTE teachers looking to add one more content area expertise. Some had already taken a CS focused professional development class with the leaders.

A common strategy for managing a low population saturation of potential cs-teachers was to design a program which could serve students from a wide geographic location, even across state lines.

In general, CS TPP leaders either identified a target audience whose pathway into CS education was supported by their state's context or cobbled together several marginal audiences, regardless; most were unable to recruit more than two to five candidates at a time.

4.2.6. Designing a program for enrollment became essential

Given the recruitment challenges, the program design elements that were chosen to reduce barriers for enrollment could be generally grouped into three categories: flexibility for a variety of pathways, funding, and features valued by students.

Flexible program design was frequently utilized to meet the needs of the various candidate students described above. The most commonly used tactic was designing a program with a reasonable 'time-to-degree':

> Requirements for any teacher prep program is just insane. So like, you're basically triple majoring in like whatever area you're working in and education, and you have to do all of your gen ed requirements. And then you basically have to do like a year long field like internship... (ID-06)

'Time-to-degree' could be improved by requiring a very limited number of credits, or allowing students to select from a number of elective options. Programs which mostly recruited education students, for example, may require only a handful of CS classes - some only required a single specialized CS Education class. These classes or workshops were specifically designed to get them up-to-speed with the CS content as quickly as possible. Compared to utilizing a standard undergraduate CS Intro course, leaders appreciated having more control over the content of these classes and the ability to set a more realistic learning curve. Where leaders wished to keep the number of classes very low in ways that conflicted with how many classes their university required for a program or minor, one strategy was to call it a "badge" or "certificate." Limiting the number of classes, of course, limited what could be covered:

> Not a super pleasant experience because it's very, like, test prep only, and ....what was missing in that course [was] a more holistic approach to understanding what works and why in teaching CS. (ID-04)

> You can't overload them during an 8 week summer workshop. They're not going to remember what you said anyway by the time it comes to teaching. (ID-08)

A few program leaders approached the challenges of recruiting CS undergraduates by offering a low credit 'teaser' course on CS Education. This would be marketed to CS majors by having a focus on technology communication and fun outreach opportunities with the hope that the class could spark an excitement about teaching to a wider CS undergraduate audience earlier on in their academic careers (a description of such class offered in New Zealand can be found at (Bell & Lambert, 2011b)). These classes, however, often struggled with fitting into the schedules and credit requirements of CS students.

Coordinating with the schedules of the computer science and education programs was a general challenge for many programs - something as mundane as the day of the week the classes were allowed to be scheduled could make or break enrollment goals. Similarly, if the CS teaching program required a CS class which wasn't also required for CS majors, the program director would have to advocate that the class was offered at the right cadence for students to be able to graduate on time.

Having required courses offered on a convenient schedule was seen as essential for many of the program leaders. This was particularly relevant for programs serving in-service teachers who could only take classes during evenings, weekends, summer terms, or asynchronously. Of these, the last was most popular among in-service teachers:

> I'd say the attrition rate is about 50%. 50% of the teachers year one, they're like, 'I'm good. I'm not coming back. It's too much to give a week in the summer, and then have all these commitments during the year.' (ID-05)

Similarly, having classes offered (even partially) online was seen as essential for recruiting candidates who were geographically distant or working as in-service teachers:

> Our program runs in the in the evening so people who are either subbing or teaching can come in the evenings and take classes remotely Monday, Tuesday, Wednesday and then in person on Thursdays. (ID-P1)

Where the programs would still increase student time to degrees, some programs made it more palatable by structuring it as accelerated Master's programs or a '5th quarter' added onto an existing Master's in Teaching program.

Beyond flexible program design, funding was the next most often mentioned strategy for increasing enrollment. Student scholarships were seen as especially critical for in-service teachers, but funding was also sought to 'sweeten' the deal' for CS students.

> I think just funding is another point of contention like, it's really hard to ask people to give up money and time to become certified to teach CS. (ID-P1)

One program leader mentioned that even with state level funding, because the the state fiscal calendar did not match up with the academic calendar, they often were left with awkward options for scholarships which made it difficult to utilize them well.

Funding was also sought for faculty to be able to teach the lower enrolled classes, as well as for the time it took for advising and recruitment:

> I mean, the scholarships were huge. That helped. I felt we had to...do a lot more repeat outreach than I would have liked. You know, like, once I got interest from someone by mailing them over and over and over again. You know, application deadline is in 2 days. That type of thing. And I think a lot of that was there were a lot of teachers that are just so overwhelmed during the school year, that that even just to get them in the mindset of, I'm home, you know. And I had a strategically do emails, too, you know, like, let me email this at 7:30. So it's after they get home and had dinner. But before, you know. I got responses from people. It's like, Oh, I'm going. I just got your email. I'm doing it right now. But that was, I think, that you know, being very active in in communicating with them as opposed to just, here it is. Apply. Don't apply. I think if we didn't do that there are a bunch of people who just would have fallen through the cracks, because also for a lot of these, even with the scholarships involved... (ID-P5)

A few even used funding to support recruitment through paid advertising.

As for other features that supported student enrollment, one was, ironically, having large enough enrollment for a cohort model:

> Even when I can get the Dean to to let me run a class with five or six, there's not enough variety of opinions and discussion in the room. (ID-17)

Where online programs were created for flexibility, they sometimes subsequently struggled with their students feeling a sense of community.

> I still worry about creating that sense of community. We're working at it, but I don't think we're there yet. I don't think the teachers have gelled as much as I'd really love to see. (ID-07)

Likewise, while many students initially preferred programs with few additional course requirements, those with more robust requirements were sometimes favored by potential students as being higher quality and preparing them better for classroom CS teaching.

4.2.7. Recruitment strategy implementation and success varied

Most of the people we talked to did not have any prior experience or expertise in recruitment - and few, if any other exemplar CS Education recruitment drives to model.

Therefore, while their efforts towards recruitment were enthusiastic and creative, they were also sometimes inconsistent, ineffectual, or even naive. In this section we'll discuss the program leaders strategies, techniques (both targeted and general), and ways of categorizing the big-picture efforts.

When discussing their strategy for recruitment, program leaders described most of their efforts as going into making a program that was easy to recruit for (as described in the previous section). Beyond that, many divided their recruitment time between increasing the pool of interested CS students, and offering personalized advising to interested students. For both of these, the content of the advertising was focused on awareness of the program, emphasizing convenient features of the program, and selling a vision of life as a highly-valued, inspirational CS teacher. In some states they included information about teacher salary and benefits (such as summers off). When talking to CS students, a few CS TPP leaders we talked with said they thought it would be helpful to their efforts if CS students had more cross-disciplinary experiences, or even just more diversity in the CS study body:

> ...some of the research I've done or read suggests that certain students are more attuned to the the needs of the community and relevance of whatever they study. They're motivated more by, 'Is it helping other people? Is it a purpose filled job? Is it...' And so, as we move into populations that might care more about those things, I'm also hopeful that maybe, maybe we'll start to get that cohort that I'm looking for. (ID-15)

The variety of channels they utilized for recruitment was extensive. For general get-the-word-out recruitment, existing university and community channels were commonly used: university websites; personal websites; fliers, posters, articles in local press; public talk series; and newsletters. One program even rented a billboard, another paid for online advertising. On campus, major fairs and academic events were often used, as was telling campus advisors and other professors about the program. As mentioned above, a few programs had a 'teaser' course about CS communication or CS Education which was open to students not yet in the major. These courses were designed around enjoyable micro-teaching or outreach activities and helped the CS TPP faculty identify students who might be interested in the full major. One program found their students and graduates contributed to recruitment with positive word-of-mouth.

Strategies were often selected based on which students were targeted for enrollment. For the target audience of students in CS classes, the most commonly used strategy was visiting the CS classes to give a short promotional talk and possibly hand out fliers. A couple programs had current high school CS teachers do these visits to help sell the vision of a CS teaching career. For faculty of CS TPPs who taught CS classes, they often reported they recruited pitching it to specific students who seemed promising or potentially interested. Finally, sending advertising materials to faculty of other CS classes and talking about the program with the faculty was seen as helpful.

Students in other content area education programs were also targeted, but mostly for add-on programs and less often for switching programs entirely. In-service teachers were most often recruited through existing CS Professional Development programs, but newsletters and professional trade organizations were also sometimes used.

A number of the program leaders discussed how they realized that to build the pipeline, they had to start in high school. Some had partner CS secondary teachers promote the program to their students, but more often the CS TPP faculty would visit high school classes and technology clubs (both in and out of school). Some invited high school students to come to the campus and experience a tour to hear more about the program.

A few CS TPP leaders would work with K12 administrators to encourage them to offer more CS classes, to send their teachers to a in-service focused program, and to hire the graduates of pre-service programs.

Several CS TPP leaders discussed recruitment from a big-picture perspective, discussing how the best recruitment tool would be for the teaching profession to be supported better:

> It's beyond our program, I think it's a national issue, and it's very similar to the what we discussed in the beginning. Making teaching profession, and and again, like regain it's respect. And asking teachers, collaborating with teachers, asking their needs and fulfilling them. It can be in terms of workload and assisting them with their needs, and also like in giving them a like livable salary styles right? So they can have a better life. So they can... When the teachers are happy they can serve their students happy. And I think it's like the life cycle. So if we can fix it in United States, I think enrollment issue might be solved. And especially like... I might be biased on those, but especially for computer science teachers. When the salaries is so high. And for the computer science teachers, I think we should offer them a comparable package similar to industry. So they can... we can retain those teachers in the classroom. (ID-03)

Supporting current CS teachers so that students have better CS experiences was also key to students being interested in the profession:

> "We will attract a nice group of students so they can become great teachers in the future and inspire more students." (ID-03)

All these efforts were very time consuming. Posters had to be created and put up, websites had to be updated, visits to individual classes had to be scheduled. Many CS TPP leaders discussed how important it was for them to have administrative support in terms of either funding for materials or time off of other obligations for this work. When funds were available, leaders found much benefit in getting help from student workers. One suggested the effort was substantial enough that it would justify hiring the help of a professional recruiter. For some programs, support from state officials was used to help with these efforts.

5. Discussion

Our research aimed to describe the journeys that occur between the initiation and enactment of CS teacher education programs. We set out to identify the barriers that impede or halt progress, and the strategies program leaders took to navigate these challenges. By comparing the journeys with Kotter's 8-step framework for leading change as a guiding framework, we hoped to observe why many of these programs struggle to get off the ground and grow.

Prior work on CS education has identified that there is demand for CS teachers and schools, but these positions are hard to fill. Enrollment in teacher preparation is down across all topic areas in the U.S., not just CS. While non-traditional PD programs do support in-service teachers in gaining CS training, it is difficult for their format to cover the breadth and depth of both a new content topics and it's PCK. On the other hand, it is also complicated to scale traditional teacher training programs at institutes of higher education as the academic program creation process is unique to each university, and program developers must consider their organizational context. (Dee & Heineman, 2016). Our study shows this holds true for CS TPP for both the internal context (such as administrative support and budgets) and external context (such as state CS policies and the needs of school districts).

Prior research on CS TPPs has identified factors important to the CS TPP creation process and examined how they enable or restrict program development (Liao et al., 2025). Our work presents more of these factors, such as the importance of a diverse guiding coalition and the influence of perceived school demand on program creation motivation. We demonstrate how these factors affect different stages of program development and ultimately identify a crucial 'Catch-22' in the teacher pipeline.

We found that program creation was motivated by passionate champions and perceived demand for CS teachers, and that this demand often stemmed from state policy changes and local school districts. Passionate champions drove the creation of CS TPP, believing that producing more CS teachers was necessary and/or an important cause. They, along with a guiding coalition, acted to complete three goals: create a sense of urgency, form a strategic vision, and enlist a volunteer (sometimes 'voluntold') crew.

However, we found that guiding coalitions sometimes struggled to fully complete these goals due to factors both external and internal to their institution. Becoming a program meant acquiring administrative and institutional support, securing funding for program development, and creating a team with specific skills and proper role delegation. The first task could be hampered by complicated bureaucracy and interdisciplinary conflict. The second task was crucial for getting students in the door. Finally, because CS TPPs were cross-disciplinary, the third task required the creation of balanced teams with a wide array of knowledge such as recruitment expertise and the ability to navigate institutional structures.

When designing the program, program leaders had to compromise on aspects important to TPP design (Darling-Hammond & Bransford, 2007). In the literature, best practices for the development of TPP outline the importance of developing coherent TPP with a consistent vision of teaching, grounding the programs in common learning standards , and connecting courses to practice and within the context of their discipline (Darling-Hammond & Bransford, 2007). However, CS TPP are nascent and have unique conditions that can make it difficult to accomplish these recommendations. Our research found that the coherency of a program often had to stray from the visions of program leaders' to meet the requirements of states, institutions, and students. Connecting courses to practice and placing them in the context of the discipline could also be difficult due to interdisciplinary conflict and expectations created by existing CTE CS teacher pathways. While programs were often grounded in common standards, these standards did not always exist at a local level due to the nascent nature of CS education.

Overall, these challenges made it difficult for program leaders to firmly establish the first four steps of Kotter's framework after which the program evaluation — and the program's future — were likely to falter. Some of the challenges faced in sustaining a program include maintaining a team, recruiting students, scheduling classes, and routine evaluation (Cranton & Legge, 1978). Programs which still accomplished 'Enable action by removing barriers' and 'Generate short-term wins' found that given the challenging ecosystem for CS Education, 'Sustain acceleration' and 'Institute change' seemed elusive.

Creating acceleration was difficult because sustainability decisions were focused mostly on survival. Programs had to consider internal factors, such as budget constraints, and external factors, such as a flagging interest in teaching careers. Team composition was important for navigating these decisions, and small teams could become brittle if they relied too heavily on an a single individual. Increasing enrollment became the top priority for many programs, which often made it intractable for programs to require more than a narrowly focused set of intro CS and CS PCK topics. This also influenced how and where program leaders focused their recruitment. Instituting change under these conditions proved difficult, and many programs either fizzled out or sputtered along with a just a handful of students.

A central concern of most program leaders during the sustainment phase was awareness that the program was vulnerable to closure if funding or enrollment levels were not sufficient. Our participants talked about how student interest in the programs was difficult to establish, partially because candidates can't aspire to a career they don't know exists, and until recently, 'CS Teacher' didn't exist as a career. This situation is exacerbated by the common practice of states initializing their efforts to expand CS educational opportunities by having teachers of other content areas 'add-on' a CS teaching endorsement. For these teachers, they often remained primarily Math teachers or Business Ed teachers first, with CS classes being taught only occasionally as an elective.

As states set policies to broaden the number, variety and sophistication of CS classes offered in K12 schools, the need becomes more acute for pathways which better prepare future teachers with a breadth of expertise in CS comparable to what we expect of our other content areas (such as Math or Spanish). The existing patchwork of pathways for CS teacher certification is currently producing threadbare outcomes. CSTA recommends offering four models of CS TPP given the distinctly different educational and professional backgrounds of candidate future CS teachers (Ericson et al., 2008; Gal-Ezer & Stephenson, 2010). Building momentum in a *single* program for such a heterogeneous set of students and pathways becomes very complicated, as the variety of barriers to address can quickly exceed available resources. Program leaders reported this complexity made it very difficult to accomplish short-term wins.

Recommendations we can offer to TPP program leaders (and aspiring CS TPP program leaders) centers on understanding what pitfalls other TPP leaders have faced, mining the strategies we have listed in this work, and being deliberate regarding each phase of Kotter's framework. New program leaders most strategic move is to solidify the first 4 steps of Kotter's framework for their programs: a guiding coalition committed to a strategic vision can convey the urgency of producing qualified CS teachers to a volunteer crew. This crew should have the necessary skills to navigate the complex process of running and sustaining a TPP for a nascent, interdisciplinary field. However, given the challenges identified in this paper, most CS TPP are likely to find it difficult to achieve sustainability.

To transform how and how much CS is taught in K12 in ways which scale and persist requires a systemic approach. Primarily, state policies which mandates a demand for CS teachers need to be paired with increases in the supply of CS teachers. These policy level efforts towards increasing K12 CS access needs to begin with funding to support the development of new CS TPP programs and scholarships to support the establishment of stable cohorts of students. At the same time, K12 schools need guidance, funding, and reasonable timelines to support their hiring teachers from these new pipelines as they work to establish reliable CS programs.

Passionate champions have been essential in sparking high quality CS teacher preparation programs, but (as discussed in the introduction) despite their efforts the number of CS teacher degrees awarded in the United States per year has declined over the last 12 years. We hope this work describing the complex nature of the problem will help TPP leaders, policy makers, and school administrators be strategic in navigating the challenges we describe while also guiding advocacy to improve the systemic picture.

6. Limitations and Future Work

This study provides a glimpse into the current world of CS TPPs through the lens of our teams' positionality. Despite interest in CS education increasing, many CS TPPs still struggle to get off the ground. Future work is necessary to solve the problems of CS teacher shortages. Our study utilized a deep but narrow scope that does not necessarily represent the entire population of CS TPP and is centered on programs in the United States. It is likely that the systemic interactions between teacher licensing policies, curricular mandates, student tuition costs, and teacher career trajectories play out differently in other countries. It also potentially contained participation bias as people who successfully got a program off the ground were more likely to respond than those who did not. Future research should focus on broadening the scope. For example, a survey could reach more programs and be used to develop a database of CS TPP statistics such as enrollment, graduation rate, and job placement. This would provide a shallower but more representative glimpse of CS TPPs compared to the current study. The data would also be more structured than that collected from semi-structured interviews.

Future work should also focus on the impact of state policy changes as they are become more common. A study of documents and materials pertaining to the creation of CS TPPs, such as design documents, recruitment materials, and case studies, could explain how different state policies impact the development of these programs and their content. An in-depth study comparing these materials to relevant state policies would highlight the specific aspects of these policies that affect CS TPP.

The computer science education landscape has recently been shifting as generative AI increases it's presence in education, industry, and society. Specifically, the demand for K12 schools to offer computer science classes is impacted by the demands of the job market, where the number and nature of technology jobs has been rapidly evolving. Related, the support for K12 CS teachers provided by industry philanthropy projects has seen many programs be discontinued or shifted to focus on generative AI literacy or AI driven CS educational initiatives. How these developments impact the sustainability of current CS TPP and the growth of new CS TPP will be an important aspect for future research.

Finally, future work should closely examine programs which have achieved stable success in preparing CS teachers to compare and contrast with struggling and closed programs in order to suggest solutions and strategies which could impact policy or be adopted by other programs.

7. Conclusion

We examined the development and challenges of Computer Science Teacher Preparation Programs (CS TPP) by analyzing their journeys through the lens of Kotter's 8-step change model. The research identified that while these programs often begin with strong motivation from passionate advocates and policy-driven demand, they quickly encounter structural and institutional obstacles. These include bureaucratic delays, funding challenges, and the need for interdisciplinary cooperation. Program leaders frequently had to compromise on their original visions due to conflicting requirements from institutions, states, and existing teaching pathways. As a result, many programs struggled to fully establish the foundational steps of Kotter's framework, often leading unsustainable, patchwork implementations.

Sustaining these programs proved to be a major hurdle, as leaders had to focus on survival amid decreasing enrollment and limited awareness of CS TPPs. While leaders could occasionally enable action by removing barriers and generate short-term wins, they struggled to sustain acceleration and institute change under these conditions. A key paradox emerged: programs needed more students to grow, but the lack of CS teachers in schools meant few people even knew such programs existed. This 'Catch-22', alongside internal budget constraints and external systemic issues, kept many programs from advancing beyond their first small successes. We call for new program leaders to prioritize the early steps of Kotter's framework, especially building diverse, committed teams. We also recommend future research to better understand how state policies and institutional contexts shape program outcomes, suggesting broader data collection efforts to support the growth and sustainability of CS teacher education.

8. Acknowledgments

The authors thank Dr. Jonah Firestone of Washington State University Tri-Cities for assistance with planning, conceptual framing, and the design of the study. This work was supported by the NSF under Grant No. 2318259, 1539179, 1703304, 1836813, 2031265, 2100296, 2122950, 2137834, 2137312, 2023338658 and unrestricted gifts from Microsoft, Adobe, and Google.

9. Disclosure Statement

No potential conflict of interest was reported by the author(s). The authors report generative AI was not used in their research or preparation of this manuscript

References

Armoni, M. (2011, November). Looking at Secondary Teacher Preparation Through the Lens of Computer Science. *ACM Transactions on Computing Education*, *11*(4), 1–38. Retrieved 2026-01-05, from `https://dl.acm.org/doi/10.1145/2048931.2048934`

Ball, D. L. (1990). Breaking with experience in learning to teach mathematics: The role of a preservice methods course. *For the learning of mathematics*, *10*(2), 10–16.

Barriball, K. L., & While, A. (1994, February). Collecting data using a semi-structured interview: a discussion paper. *Journal of Advanced Nursing*, *19*(2), 328–335. Retrieved from `https://doi.org/10.1111/j.1365-2648.1994.tb01088.x`

Bell, T., & Lambert, L. (2011a). Teaching computer science majors about teaching computer science. In *Proceedings of the 42nd acm technical symposium on computer science education* (pp. 541–546).

Bell, T., & Lambert, L. (2011b, March). Teaching computer science majors about teaching computer science. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 541–546). Dallas TX USA: ACM. Retrieved 2026-01-03, from `https://dl.acm.org/doi/10.1145/1953163.1953317`

Code.org. (2018). *Everyone and no one can teach cs: Certifications eligible to teach cs (by state).* Retrieved from `https://docs.google.com/document/d/1XA1zIjuR222BtRWTljNatCD6TYFAZan8eDWqZneG6I8/edit?tab=t.0#heading=h.7wzydv7y78vb`

Cranton, P., & Legge, L. (1978). Program evaluation in higher education. *The Journal of Higher Education*, 464–471.

CSforAll. (2021). *Investing in teacher preparation for the classrooms of today.* Retrieved from https://csforall.medium.com/investing-in-teacher-preparation-for-the-classrooms-of-today-c7dd7c41ab68

Darling-Hammond, L., & Bransford, J. (2007). *Preparing teachers for a changing world: What teachers should learn and be able to do.* John Wiley & Sons.

Darling-Hammond, L., et al. (1989). *Redesigning teacher education. opening the door for new recruits to science and mathematics teaching.* ERIC.

Dee, J. R., & Heineman, W. A. (2016). Understanding the organizational context of academic program development. *New Directions for Institutional Research*, *2015*(168), 9–35.

Delyser, L. A., Goode, J., Guzdial, M., Kafai, Y., & Yadav, A. (2018). Priming the computer science teacher pump: Integrating computer science education into schools of education. *CSforAll, New York, NY*.

Elliott, R., Fischer, C. T., & Rennie, D. L. (1999). Evolving guidelines for publication of qualitative research studies in psychology and related fields. *British journal of clinical psychology*, *38*(3), 215–229.

Ericson, B., Armoni, M., Gal-Ezer, J., Seehorn, D., Stephenson, C., & Trees, F. (2008, August). *Ensuring Exemplary Teaching in an Essential Discipline: Addressing the Crisis in Computer Science Teacher Certification - Final Report of the CSTA Teacher Certification Task Force* (Tech. Rep.). CSTA.

Everson, J., & Ko, A. J. (2022). I would be afraid to be a bad cs teacher: Factors influencing participation in pre-service secondary cs teacher education. In *Proceedings of the 2022 acm conference on international computing education research-volume 1* (pp. 237–246).

Fadjo, C. L., Brown, T., & DeLyser, L. A. (2013). A Curriculum Model for Preparing K-12 Computer Science Teachers. In *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS).* The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Fancsali, C., Mark, J., & Lee, J. (2025). *Building teacher capacity for equitable cs: Insights from nyc's cs4all initiative* (Tech. Rep.). New York University Steinhardt School of Culture, Education, and Human Development.

for Education Statistics, N. C. (2020). *Table 318.30. bachelor's, master's, and doctor's degrees conferred by postsecondary institutions, by sex of student and field of study: 2018-19.* Retrieved from https://nces.ed.gov/programs/digest/d20/tables/dt20_318.30.asp

Franke, B., Century, J., Lach, M., Wilson, C., Guzdial, M., Chapman, G., & Astrachan, O. (2013, March). Expanding access to K-12 computer science education: research on the landscape of computer science professional development. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 541–542). New York, NY, USA: Association for Computing Machinery. Retrieved 2026-01-10, from https://dl.acm.org/doi/10.1145/2445196.2445358

Friend, M., Twarek, B., Koontz, J., Bell, A., & Joseph, A. (2022). Trends in cs teacher professional development: A report from the csta pd committee. In *Proceedings of the 53rd acm technical symposium on computer science education-volume 1* (pp. 390–396).

Fulfilling Computer Science (CS) Preservice Teacher Education Programs: Policy Brief (Tech. Rep.). (2023, December). Expanding Computing Education Pathways (ECEP). Retrieved 2026-05-23, from https://ecepalliance.org/resources/toolkits-guides/policy-briefs/

Gal-Ezer, J., & Stephenson, C. (2010, March). Computer science teacher preparation is critical. *ACM Inroads*, *1*(1), 61–66. Retrieved 2026-01-03, from https://dl.acm.org/doi/10.1145/1721933.1721953

Goode, J., Margolis, J., & Chapman, G. (2014, March). Curriculum is not enough: the educational theory and research foundation of the exploring computer science professional development model. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 493–498). New York, NY, USA: Association for Computing Machinery. Retrieved 2026-01-10, from https://dl.acm.org/doi/10.1145/2538862.2538948

Gray, J., Odom-Bartel, R., Zelkowski, J., Hamner, K., & Rodgers-Farris, S. (2020). A pre-service pathway for preparing future ap cs principles teachers. In *Proceedings of the 51st acm technical symposium on computer science education* (pp. 1127–1132).

Guzdial, M. (2011, February). *What's the argument for becoming a computer science teacher?* [Blog]. Retrieved 2026-01-10, from `https://computinged.wordpress.com/2011/02/07/whats-the-argument-for-becoming-a-computer-science-teacher/`

Guzdial, M. (2016). Bringing computer science to us schools, state by state. *Communications of the ACM*, *59*(5), 24–25.

Hazzan, O., Gal-Ezer, J., & Ragonis, N. (2010, March). How to establish a computer science teacher preparation program at your university?: the ECSTPP workshop. *ACM Inroads*, *1*(1), 35–39. Retrieved 2026-01-03, from `https://dl.acm.org/doi/10.1145/1721933.1721949`

Hazzan, O., & Lapidot, T. (2004). The practicum in computer science education: bridging gaps between theoretical knowledge and actual performance. *ACM SIGCSE Bulletin*, *36*(4), 47–51.

Hazzan, O., Lapidot, T., & Ragonis, N. (2020). *Guide to teaching computer science.* Springer.

Irwin, V., Wang, K., Tezil, T., Zhang, J., Filbey, A., Jung, J., ... Parker, S. (2023). *Report on the condition of education 2023. nces 2023-144rev.* (Tech. Rep.). National Center for Education Statistics.

Kim, J., Childs, J., Leftwich, A., Edwards, K. M., Fletcher, C. L., & Hendrickson, K. (2021). Landscape of computer science teacher qualification pathway. In *Proceedings of the 52nd acm technical symposium on computer science education* (pp. 1299–1299).

Ko, A. J., Beitlers, A., Everson, J., Wortzman, B., & Gallagher, D. (2023a). Proposing, planning, and teaching an equity-and justice-centered secondary pre-service cs teacher education program. In *Proceedings of the 54th acm technical symposium on computer science education v. 1* (pp. 583–589).

Ko, A. J., Beitlers, A., Everson, J., Wortzman, B., & Gallagher, D. (2023b, March). Proposing, Planning, and Teaching an Equity- and Justice-Centered Secondary Pre-Service CS Teacher Education Program. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (pp. 583–589). New York, NY, USA: Association for Computing Machinery. Retrieved 2026-01-10, from `https://dl.acm.org/doi/10.1145/3545945.3569735`

Kotter, J. P. (1996). *Leading change.* Boston, Mass.: Harvard Business School Press. (OCLC: 34658795)

Kotter, J. P. (2012, 11). Accelerate! *Harvard business review*, *90*, 44-52, 54.

Kraft, M. A., & Lyon, M. A. (2024). The rise and fall of the teaching profession: Prestige, interest, preparation, and satisfaction over the last half century. *American Educational Research Journal*, *61*(6), 1192–1236.

Lang, K., Galanos, R., Goode, J., Seehorn, D., Trees, F., Phillips, P., & Stephenson, C. (2013). *Bugs in the System: Computer Science Teacher Certification in the U.S.* (Tech. Rep.). The Computer Science Teachers Association and The Association for Computing Machinery.

Lapidot, T., & Hazzan, O. (2003). Methods of teaching a computer science course for prospective teachers. *ACM SIGCSE Bulletin*, *35*(4), 29–34.

Liao, Y.-C., Karlin, M., Margulieux, L., & Calandra, B. (2025). Building bridges, breaking barriers: Exploring the dynamics and key factors shaping computer science education pathways in teacher preparation. *Journal of Technology and Teacher Education*, *33*(1), 5–35.

Mak, J., & Torrejon Capurro, C. (2024). State-level perspectives of high school computer science education policy: A complex systems analysis. In *Proceedings of the 2024 on respect annual conference* (pp. 245–253).

McDonald, N., Schoenebeck, S., & Forte, A. (2019, 11). Reliability and inter-rater reliability in qualitative research: Norms and guidelines for cscw and hci practice. *Proceedings of the ACM on Human-Computer Interaction*, *3*, 1-23.

Menekse, M. (2015, December). Computer science teacher professional development in the

United States: a review of studies published between 2004 and 2014. *Computer Science Education*, 1–26.

Nguyen, T. D., Lam, C. B., & Bruno, P. (2022). Is there a national teacher shortage? a systematic examination of reports of teacher shortages in the united states. edworkingpaper no. 22-631. *Annenberg Institute for School Reform at Brown University*.

Odom-Bartel, B., Zelkowski, J., & Gray, J. (2021). Preparing secondary education mathematics teacher candidates for ap computer science principles: A two-course design model. *Preparing pre-service teachers to teach Computer Science: Models, practices, and policies*, 153–172.

Ottenbreit-Leftwich, A. T., Childs, J., Fletcher, C., Kim, J., Edwards, K. M., & Hendrickson, K. (2022). Defining computer science teacher qualification pathways. *Policy Futures in Education*, 1–14.

Pellet, J.-P., Parriaux, G., & Chevalier, M. (2022). Design and Analysis of a Disciplinary Computer Science Course for Pre-service Primary Teachers. In A. Bollin & G. Futschek (Eds.), *Informatics in Schools. A Step Beyond Digital Education* (Vol. 13488, pp. 125–137). Cham: Springer International Publishing. Retrieved 2026-01-11, from https://link.springer.com/10.1007/978-3-031-15851-3_11 (Series Title: Lecture Notes in Computer Science)

Ragonis, N., & Hazzan, O. (2008). Disciplinary-Pedagogical Teacher Preparation for Pre-service Computer Science Teachers: Rational and Implementation. In R. T. Mittermeir & M. M. Sysło (Eds.), *Informatics Education - Supporting Computational Thinking* (pp. 253–264). Berlin, Heidelberg: Springer.

Ragonis, N., Hazzan, O., & Gal-Ezer, J. (2010, March). A survey of computer science teacher preparation programs in Israel tells us: computer science deserves a designated high school teacher preparation! In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 401–405). Milwaukee Wisconsin USA: ACM. Retrieved 2026-01-03, from https://dl.acm.org/doi/10.1145/1734263.1734402

Ragonis, N., Hazzan, O., & Gal-Ezer, J. (2011, March). A study on attitudes and emphases in computer science teacher preparation. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 559–564). Dallas TX USA: ACM. Retrieved 2026-01-03, from https://dl.acm.org/doi/10.1145/1953163.1953320

Rosato, J., Friend, M., Abell, O., Margulieux, L., Moallem, M., Nadelson, L., & Leftwich, A. (2022). *Guidance for Schools of Education.* Retrieved 2026-05-22, from https://csteachers.org/schoolsofed/

Saenz-Armstrong, P. (2023). Data brief: How do trends in teacher preparation enrollment and completion vary by state. *National Council on Teacher Quality*.

Schafer, J. B., & East, J. P. (2022). Creating a high quality, high impact cs teacher prep program. In *Proceedings of the 53rd acm technical symposium on computer science education-volume 1* (pp. 599–605).

Sentance, S. (2024, September). Building a National Center for K-12 Computing Education. *ACM Inroads*, *15*(3), 29–41. Retrieved 2026-01-03, from https://dl.acm.org/doi/10.1145/3679023

Sentance, S., & Csizmadia, A. (2017, March). Professional Recognition Matters: Certification for In-service Computer Science Teachers. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 537–542). Seattle Washington USA: ACM. Retrieved 2026-01-03, from https://dl.acm.org/doi/10.1145/3017680.3017752

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational researcher*, *15*(2), 4–14.

Stein, R. (2024). *Three ways to meet the growing demand for k-12 computer science educators.* Retrieved from https://www.air.org/resource/blog-post/three-ways-meet-growing-demand-k-12-computer-science-educators#:~:text=Sixty%20percent%20of%20U.S.%20high,way%20to%20increase%20that%20number.

Thomas, D. R. (2003). A general inductive approach for qualitative data analysis.

Toolin, R., Dion, L., & Erickson, R. (2021). The computer science education collaborative: Promoting computer science teacher education programs for preservice and in-service teachers. *Journal of Computational Science Education*.

US Department of Education, O. o. P. E. (2025). *Title ii.* Washington, DC. Retrieved from https://title2.ed.gov/Public/Home.aspx

Vegas, E., Hansen, M., & Fowler, B. (2021). *Building skills for life. How to expand and improve computer science education around the world* (Tech. Rep.). The Brookings Institution. Retrieved 2025-12-13, from https://www.brookings.edu/wp-content/uploads/2021/10/Building_skills_for_life.pdf

Wang, J., Hong, H., Ravitz, J., & Hejazi Moghadam, S. (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* (pp. 645–650).

Weissman, H., Glennon, M., Twarek, B., Tabor, C., Dunton, S., Childs, J., & Martin, N. (2024). *2024 state of computer science education* (Tech. Rep.). Code.org, CSTA, and ECEP Alliance.

Whitehead, C., Ray, L., Khan, S., Summers, W., & Obando, R. (2011, March). Implementing a computer science endorsement program for secondary school teachers. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 547–552). Dallas TX USA: ACM. Retrieved 2026-01-05, from https://dl.acm.org/doi/10.1145/1953163.1953318

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010, October). *Running on Empty: the Failure to Teach K–12 Computer Science in the Digital Age* (Tech. Rep.). New York, NY, USA: ACM. Retrieved 2026-01-10, from https://dl.acm.org/doi/book/10.1145/3414583

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer science education*, *26*(4), 235–254.

Yadav, A., & Korb, J. T. (2012). Learning to teach computer science: The need for a methods course. *Communications of the ACM*, *55*(11), 31–33.

Yeni, S., Aivaloglou, E., & Hermans, F. (2020). To be or not to be a teacher? exploring cs students' perceptions of a teaching career. In *Proceedings of the 20th koli calling international conference on computing education research* (pp. 1–11).

Zarch, R., Boisvert, D., Bhatnagar, S., & Goldwasser, T. (2024). *Computer science as a graduation requirement: A landscape analysis report and webinar.* (Tech. Rep.). Sagefox Consulting Group.

Zarch, R., & McCune, J. (2025). Minding the gap: Methods for understanding teacher capacity when developing high school computer science graduation requirements. In *2025 asee annual conference & exposition.*