Hiding in plain sight: Unmasking computer science heroes in K-6 classrooms

by

Kiera Shuler Elledge

B.S., Florida State University, 1995
M.S., University of North Texas, 2003

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF EDUCATION

Department of Curriculum and Instruction
College of Education

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2024

Abstract

This research project employs a case study design to examine how K-6 non-computer science (CS) trained teachers can be effectively prepared to teach CS, metaphorically unmasking hidden superheroes in the education system. Grounded in the Technological Pedagogical Content Knowledge (TPACK) framework, the study also integrates Bandura’s Theory of Self-Efficacy, the Crawl-Walk-Run-Fly Model (CWRF), and Bruner's Spiral Curriculum. Through detailed case studies, the research investigates teachers' professional development experiences, their evolving self-efficacy, and their practical application of CS concepts in the classroom. Data collected from surveys, interviews, and the researcher’s journal are analyzed to identify best practices and challenges. This program is designed not only to inform strategies for teacher training programs that will enhance the integration of CS education in diverse educational settings, but also to foster a mindset of growth, bravery, and resilience. Through a series of workshops, hands-on activities, and ongoing support, teachers will acquire the skills they need and, more importantly, the courage to venture into uncharted territory.

The findings reveal that with the right tools and support systems in place, K-6 teachers can indeed become proficient in teaching computer science. More significantly, they develop a newfound confidence that permeates their teaching practice, inspiring them to continue to be life-long learners who boldly guide their students to embrace the challenges of the digital age.

Hiding in plain sight: Unmasking computer science heroes in K-6 classrooms

by

Kiera Shuler Elledge

B.S., Florida State University, 1995
M.S., University of North Texas, 2003

A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF EDUCATION

Department of Curriculum and Instruction
College of Education

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2024

Approved by:

Major Professor
Dr. Suzanne Porath

Copyright

Abstract

This research project employs a case study design to examine how K-6 non-computer science (CS) trained teachers can be effectively prepared to teach CS, metaphorically unmasking hidden superheroes in the education system. Grounded in the Technological Pedagogical Content Knowledge (TPACK) framework, the study also integrates Bandura’s Theory of Self-Efficacy, the Crawl-Walk-Run-Fly Model (CWRF), and Bruner's Spiral Curriculum. Through detailed case studies, the research investigates teachers' professional development experiences, their evolving self-efficacy, and their practical application of CS concepts in the classroom. Data collected from surveys, interviews, and the researcher’s journal are analyzed to identify best practices and challenges. This program is designed not only to inform strategies for teacher training programs that will enhance the integration of CS education in diverse educational settings, but also to foster a mindset of growth, bravery, and resilience. Through a series of workshops, hands-on activities, and ongoing support, teachers will acquire the skills they need and, more importantly, the courage to venture into uncharted territory.

The findings reveal that with the right tools and support systems in place, K-6 teachers can indeed become proficient in teaching computer science. More significantly, they develop a newfound confidence that permeates their teaching practice, inspiring them to continue to be life-long learners who boldly guide their students to embrace the challenges of the digital age.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Completing this doctorate degree has been a profound journey of personal growth and discovery, and it would not have been possible without the support, guidance, and encouragement of numerous individuals.

First, to all the Ed.D.'s on my team! Your unwavering support, insightful feedback, and constant encouragement were invaluable throughout this process.

Second, to all of the incredible teachers who participated in this study. Your willingness to share your experiences and insights made this research possible and your dedication to teaching and your passion for learning are truly inspiring.

Third, to all of my family and friends. Your belief in my ability to complete this degree was often stronger than my own, and your expressions of reassurance were a tremendous source of strength.

I can do hard things.

Kiera Shuler Elledge

July 2024

Dedication

To Dad and Mom who taught me I could do anything I set my mind to, to Rhett who challenged me to never accept second place, and to George who always cheered for me the loudest!

Preface

The field of computer science (CS) is rapidly evolving and becoming an integral part of our educational landscape. However, many educators, particularly those without a formal background in CS, face significant challenges in teaching this subject effectively. This doctoral dissertation is dedicated to addressing these challenges and exploring ways to empower non-CS trained teachers to become proficient and confident educators in the realm of computer science.

The metaphor of unmasking hidden superheroes is central to this work. In many classrooms, there are teachers with untapped potential, poised to make a significant impact on their students' understanding of computer science. This research aims to uncover and nurture that potential, providing teachers with the tools and knowledge they need to succeed.

The inspiration for this study came from my own experiences in education and my interactions with dedicated teachers who, despite their lack of formal training in CS, were determined to bring this essential knowledge to their students. Witnessing their commitment and the obstacles they faced motivated me to delve deeper into this area of research.

Ultimately, this work aims to contribute to the broader goal of integrating computer science education into diverse educational contexts, ensuring equitable access to CS literacy for all students. By unmasking the hidden superheroes in our education system, we can unlock a future where every student has the opportunity to thrive in an increasingly digital world.

Kiera Shuler Elledge

July 2024

Chapter 1 - Introduction to the Problem

Overview

Teaching computer science in elementary schools continues to be crucial in today's digital world as it empowers young students with the fundamental skills and knowledge needed to successfully navigate and thrive in a technology-driven society. It has also been considered a long-term investment in the future of our nation's workforce, bridging the skills gap between the demands of the labor market and the availability of qualified people to fill the positions. With the Bureau of Labor and Statistics predicting the number of STEM jobs in the United States to exceed one million by the end of 2022 and over half not able to be filled due to the lack of skilled workers, all eyes seemed to turn towards K-12 to help mitigate the shortfall of capable Americans who had the necessary training to do these in-demand jobs (BLC, 2021). Compound this shortage with the continuous shifting and changing of job requirements and skill sets due to rising technological advances, one might have felt as though they were trying to hit a moving target, and this is where K-12 educators have found themselves. In addition to teaching reading, writing, and arithmetic, many of these teachers, particularly those at the elementary level who did not have a background in computer science and its related skills, have suddenly found themselves being tasked with preparing their students for jobs that did not yet exist. How could they teach what they did not know? Removing the barriers to the greatest extent possible and equipping in-service teachers with the knowledge and skills necessary to accomplish this seemingly insurmountable task was the focus of this research.

Historical Context

In 2015, international experts – policy makers, practioners, and researchers from both IT and education – gathered in Bangkok, Thailand at EDUSummIT 2015 and announced three compelling arguments for the inclusion of computer science education in K-12 focused on economic, social and cultural impacts. For economics, it was cited as the need to sustain a competitive edge in a world driven by technology and the need for computer science-skilled professionals in all industries to be able to support innovation and development. Socially, it was identified as the need to cultivate a diverse society of active creators (and not just) consumers of technology. Then, the cultural impact would enable a society to drive the technology changes in the world, rather than have them imposed on them (Webb et al., 2015). In January of that same year, U.S. President Barack Obama stated:

> America thrived in the 20th century because we made high school free, sent a generation of GIs to college, and trained the best workforce in the world. We were ahead of the curve…and now in a 21st century economy that rewards knowledge like never before, we need to up our game. We need to do more. (2015)

President Obama believed that a strong and sustained approach to STEM (Science, Technology, Engineering and Mathematics) education in this country would be critical to enable the United States to continue to lead the world in innovation and development of new technologies. He called heads of major US corporations, industry giants, philanthropies and other nongovernment agencies to the table and was able amass over $1 billion dollars to commit to K-12 STEM education, with goals such as recruiting underrepresented groups more access to STEM education and career opportunities and furthering teacher professional learning and retention. His successor, President Donald Trump, issued a Presidential Memo early in his administration that included an additional focus on teaching Computer Science and devoted at least $200 million per year in grant funds to sustain the initiative, in an effort to develop more STEM-skilled workers for jobs in U.S. industries (Trump, 2017).   And in June 2021, the Biden administration submitted a $6 trillion budget for public education, with a focus on assisting low-income students and students of color. The proposed budget would expand funding for increasing participation for racial and ethnic groups underrepresented in STEM fields in K-12 and would provide $5 billion earmarked for colleges that traditionally serve these students (Dennon, 2021). This was a strong start and on-going show of support from our nation's leaders, but there was more to be done, starting with training non-computer science background teachers to not only learn the computer science skills, but also to have the self-efficacy to turn their learning around and teach it to their students.

Current State

On-going advancements in technology, coupled with the growing number of professions that rely on computer science skills and the current shortage of qualified workers in the U.S. to fill those gaps, make it crucial for all students to have the opportunity to learn about computer science and its related fields from an early age. Yet schools continue to argue that they simply do not have the time to devote to computer science classes and that they have no budget for computer science teachers (Google & Gallup, 2015). Further, with legislative agendas and school district leadership priorities focused on high-stakes testing in core subjects (of which computer science is not one) and state accountability ratings, finding time in an already overcrowded school day to include computer science is a tough ask. The pressure to perform well on these high-stakes tests tends to keep staff and students hyper-focused on those subjects so that their performances reflects positively on their school and their district, as these scores are often also tied to teacher recruitment and retention, merit-based pay models and school funding.

This is an acutely valid argument at the elementary level, where nation-wide, since the introduction of No Child Left Behind (2002), mathematics and literacy instruction have taken center stage, leaving science, social studies, the arts and other subjects clamoring for a piece of the remainder of the instructional time. But research has consistently shown that computer science instruction in the elementary grades is critical in preparing students for a competitive job market and that early childhood experiences allowed students to be better prepared for specialized courses in high school, which supported the need for a pathway across grade levels (Howard, 2018). Further, meaningful integration in elementary classrooms emphasized that quality instruction must provide students with real-world (authentic) opportunities to engage in creativity and problem-solving while interacting with computing technologies. However, as countless studies have enumerated, adding additional content into the K-6 curriculum framework in the United States would be a complex task (Angeli et al., 2016; Gallup Inc., 2020; Howard, 2018).

Still, the value of computer science skills at all levels of K-12 education was not lost on educational leaders, as research had shown the use of these skills benefitting students across a number of disciplines. Benefits included increased engagement, motivation, confidence, problem-solving, critical-thinking and communication (Kim et al., 2015; Schanzer, Fisler, & Krishnamurti, 2018), just to name a few. This in turn has national agencies to develop policy requiring state education agencies to mandate computer science courses in K-12 education. Movement towards states adopting standards was slow at first, with only seven states adopting computer science policies from 2014 – 2017 (Stanton et al., 2017); but as our nation's sense of urgency toward addressing this issue, this has increased, as of 2023, 37 of our 50 states had either adopted policies requiring computer science education in K-12 or were in the process of doing so (Code.org, 2023).

In order to deliver high-quality computer science education, teachers must be well prepared, have an established support system, and be professionally connected to one another. However, due to the aforementioned lack of common standards and clear certification pathways, school districts have continued to report struggles in hiring qualified computer science teachers (Google & Gallup, 2016; Yadav et al., 2016). The data was showing that the in-service teachers who are becoming computer science teachers were first having to be certified in another content area before being allowed to teach computer science (Code.org, CSTA' & ECEP Alliance, 2020)! Further, these teachers who ultimately received approval to teach computer science reported facing a continuous lack of accessible and high-quality professional development opportunities, and resources to assist them in improving their computer science knowledge base and being able to better assess student understanding (Menekse, 2015; Yadav et al. 2016).

What had been routinely observed was that computer science teacher professional development that did exist was often short in duration (usually lasting about a week in the summer) and only sometimes had an ongoing engagement component during the school year (Darling-Hammond et al., 2017; Google & Gallup, 2015; Qian, Hambrusch, Yadav, & Gretter, 2018). These one-and-done professional development offerings sometimes lacked the necessary consistent follow-up opportunities needed for teachers to participate in order to feel that they could successfully learn new instructional techniques, and may left them without the capacity to improve, often contributing to a sense of professional isolation and loneliness (Menekse, 2015; Franke et al., 2013; Yadav et al., 2016). This sense of isolation may have inhibited teachers' ability to take part in and share knowledge within a professional community in the same way as teachers in core subjects and could be compounded by a low sense of computer science teachers' self-efficacy—or the strength of teachers' beliefs in their abilities to teach CS. Many computer science teachers have reported low levels of self-efficacy (Yadav et al., 2016), which is related to teachers’ institutional practices, occupational satisfaction, and job commitment.

Purpose of Research and Research Questions

Purpose

As the demand for computer science instruction has increased, there continues to be a shortage of K-12 teachers who are trained to teach the subject, particularly at the elementary level. In the United States, there are a limited number of teacher preparation programs that offer computer science teacher certification that leads to licensing, and currently, none of them focus on K-6, as they are strictly secondary-based programs (Code.org, 2013; CSTA, 2015). On top of that, computer science has been integrated with other career and technical education areas such as Technology Education, Educational Technology, Industrial or Instructional Technology, Information Management Systems, or even the use of computers to support learning in other subject areas (Google, 2015; CSTA & ECEP Alliance, 2020) which has caused additional confusion. This has led to inconsistent standards for teaching certification throughout the nation and among the states, which in turn has left school districts and universities without a clear path to the coursework needed for licensure at the secondary level and appropriate training needed at the elementary level.

While there was no disputing that preparing pre-service teachers to teach computer science is vital to building a sustainable computer science teaching model, this research project focused on in-service teachers who had already begun their educational journey and would be able to affect change in the classroom immediately. This study is referred to them as superheroes hiding in plain sight – educators who are willing to take a chance, step out of their content comfort zones and learn new knowledge that not only empowers them, but can positively impact their students as well. From Google & Gallup's report referenced previously, approximately 83% of teachers, 79% of principals, and 81% of superintendents surveyed in the study agreed that teachers would need extensive training to teach computer science successfully (2017). But in a PricewaterhouseCoopers (PwC) report produced in 2018, only an average of 10% of K–12 teachers felt confident incorporating higher-level technology into student learning. This data held true across grade level, school affluence, and teacher experience level (PwC, 2018) and lead to a host of training, teacher capacity, self-efficacy and other questions needing to be addressed for it to be successful. When would this critical training take place and how would it be facilitated? Would the teachers be capable of learning and understanding the material at a level that they are then able to turn it around and successfully teach it to students? Could they convince others of the importance of computer science and engage them in the effort? With technology changing at such a rapid pace in today's world, would more mature teachers be able to, or even want to, keep up with training necessary to be an effective computer science teacher?

Providing the technology and pedagogical content knowledge in elementary would be a key component however, there was little consensus about the best approach to prepare teachers regarding the need for these vital changes that occur when implementing computer science or the way computer science should be included in K-5 instruction (Howard, 2019). Researchers contended that the effective integration of computer science occurred as a distinct discipline and a school subject in the K-12 curriculum (Angeli et al., 2016; Fluck et al., 2016). However, teacher resistance to such pedagogical changes, taken together with known barriers to computer science integration with current K-5 curriculum, suggested that the implementation of computer science programs may prove intimidating for teachers who have not undergone extensive training or who have a background in a related field. Despite the potential for resistance and owing to the fact that the United States still lags behind in producing significant number of teachers prepared to teach computing classes at any level within K-12 due to the aforementioned challenges, for these skills to be appropriately addressed, the teacher was the must-have first piece (CSTA, 2013; Sentance & Csizmadia, 2016).

K-12 administrators often encounter difficulties when searching for a qualified (certified) teacher for computer science especially at the elementary level (Howard, 2018). Based on a stand-alone Gallup poll, 74% of superintendents and 63% of elementary school principals cite certification status as the number one cause for failing to find acceptable teachers (Gallup, 2015). In a situation where a teacher may or may not be able to be found, districts can find themselves in the midst of having to "grow your own" (GYO) so they have someone in place to teach. Many states have a GYO program to recruit and train teachers from within communities to bring highly-needed or specialized skill sets, such as bilingualism or industry-based programs into schools. These programs often come out of a collaboration between schools or districts, community organization and higher-education or alternative-education teacher preparation programs (Motamedia et al., 2017). This research will be loosely based on a GYO model, but with some minor adjustments to accommodate for current teaching credentials and classroom experience.

Research Questions

Research questions in a dissertation serve to guide the study by clearly defining its focus and scope, ensuring that the research remains aligned with its objectives (Creswell, 2014). This research was guided by two questions that provided a systematic way to investigate a known problem of practice in K-6 elementary education that included development, deployment and assessment of a potential solution.

1. What are the effective components of a professional learning model that includes the necessary academic and social supports for teachers to successfully learn and learn to teach computer science?

The first research question focused on the effectiveness of a professional learning model for teachers that included the necessary academic and social supports to enable teachers to learn and learn to teach computer science. During the initial phase, district administrators identified needs, defined program goals and developed a plan to recruit teachers to participate. In this model, teachers volunteered to be a part of the program, and the learning expectations, academic and social supports, and evaluation pieces were set forth from the outset and group norms established. Determining a clear path to the coursework was also a part of this research question. Areas that were considered included best practices for preparing elementary teachers, the minimal skills needed for an elementary teachers to be successful teaching computer science and also how these aligned with the grade-level skills that the students need.

Without question, teachers recognized the benefits that technology can have in the classroom – whether it be to deliver content, add additional features to simplify work, or provide additional tools to produce authentic work. But too many often view technology as a silver bullet to the challenges they face and it is sometimes assumed, consciously or not, that digital tools alone can improve education. The primary objective of this question is to determine what supports are necessary for a non-computer science trained teacher to be able to learn requisite computer science skills and how to use current technology tools, as well as how they can be used in teaching the same subject matter.

2. In what ways, if any, are the participants of this cohort able to take their new knowledge and implement it in their classroom or school as a way to increase awareness of the need for all students to have access to computer science education?

This question focused on the teachers being able to take what they had learned and then implement it in their classrooms and schools as they became advocates for computer science education for all students. With the recent pushes to include more computer science and computational thinking skills in K-12 education - consider CSforAll, Expanding Computing Education Pathways (ECEP), Exploring Computer Science (ECS) and other broadening participation alliances funded by the National Science Foundation – it has become paramount to establish criteria, expectations and on-going support needs for primary teachers to be able to teach computer science and computational thinking skills to their students. In order to prepare elementary school teachers who typically do not hold degrees in computer science or any other related field, school districts must develop their own plans to provide training for these teachers, preparing them to integrate these critical skills into their own content areas and ensuring sustainability through an ongoing support model (Howard, 2018; Motamori, 2017).

Research Frameworks

Research frameworks are structured plans that guide the collection and analysis of data in a study while providing a clear outline of the concepts, variables, and their relationships, which helps ensure that the research is systematically conducted and grounded in theory (Creswell & Creswell, 2018). By offering a structured approach, research frameworks help ensure the direction and focus of the study, allowing researchers to clearly define their questions, methods, and interpretations of the resulting data. They are also crucial for validating the research process, ensuring consistency and producing documentation that will allow the research to be replicated by others.

This research utilized both theoretical and conceptual frameworks in order to fully address the research questions. Researchers Grant and Osanloo (2018) employed the analogy of building a home when selecting a research framework: "…the theoretical framework would entail the elevation blueprints… whereas the conceptual framework would involve the floor plan blueprint" (p. 17), suggesting that a house is not complete without both sets and neither is the dissertation complete without both a theoretical and a conceptual framework.

Theoretical Frameworks

A theoretical framework provides a foundation for understanding phenomena, guiding research, and informing practice, and is based on established theories that explain the relationships between different concepts and variables (Creswell, 2009). The two utilized in this research project were Mishra and Koehler's TPACK Framework (2006) and Bandura's Theory of Self-Efficacy (1977).   Both will be discussed in greater detail in subsequent sections.

The TPACK was developed to understand and describe the types of knowledge teachers need to effectively integrate technology into their teaching and was an extension of Shulman's concept of Pedagogical Content Knowledge (PCK), which described the blending of pedagogy and content knowledge as essential for effective teaching (Mishra & Koehler, 2006). It provided a structured way to understand and analyze the complex relationships between technology, pedagogy, and content knowledge, as well as identify challenges and gaps in current teacher knowledge and develop strategies to increase technology integration into the classroom.

Albert Bandura's Theory of Self-Efficacy (1977) is a subset of his (1986) *Social Cognitive Theory*, and the second theoretical framework utilized in this research. Self-efficacy, according to Bandura, is a "personal judgement of how well one can execute courses of action required to deal with prospective situations" and that individuals develop self-efficacy from four main sources of influence – mastery experience (previous personal successes), vicarious experiences (seeing others successes), social persuasion (positive verbal feedback) and emotional state (well-being) (Bandura, 1977, 1986, 1997). In simpler terms, it is a person's belief in their own ability to be successful in a particular situation or task. Further, it reflects confidence in one's ability to control personal behavior and motivation. In educational settings, Bandura's theory helps explain why students and teachers with high self-efficacy are more likely to embrace challenging tasks, persist longer, and achieve better outcomes.

Conceptual Frameworks

The conceptual framework provides the visual layout of all of the ideas and experiences within the study and the implications of each. Ravitch and Riggan (2017) defined it as: "an argument about why the topic one wishes to study matters, and why the means proposed to study it are appropriate and rigorous." In other words, it is why the work is worth doing and how it should be done. The two conceptual frameworks aligned in this research project were the Crawl-Walk-Run-Fly model and Jerome Bruner's Spiral Curriculum Approach (1966). They will be introduced briefly here and discussed in greater detail at a later time.

The origins of the crawl-walk-run-fly (CWRF) model have not been tied to a specific individual or source but rather represent a combination of existing frameworks within multiple fields. In each instance, it was formed out of a desire to continuously improvement and make existing work better by going one step further pursuing excellence. The model is used to describe the progression of learning in education, starting with the basics and moving upward from there.

Bruner's spiral curriculum approach, developed by educational psychologist Jerome Bruner (1966), is a framework that emphasizes revisiting key concepts and skills at increasing levels of complexity over time. The spiral approach is based on the idea that learning is an iterative process, where students continuously build upon and deepen their understanding of foundational ideas through repeated exposure and engagement. It focuses on the ability to revisit topics regularly to gain a greater and deeper understanding of, as well as opportunities for hands-on, project-based learning.

Combining the crawl-walk-run-fly model with Bruner's spiral approach creates a learning environment that is both structured and adaptive. This method provides support through a logical progression of skills while continually reinforcing and expanding understanding. It has the potential to foster deeper learning of, in this case, computer science concepts and the pedagogical skills to teach it. This integrated approach can prepare learners for the dynamic and evolving landscape of technology, equipping them with the skills and confidence needed to excel and innovate.

Summary

Computer science is broader than the study of computers and computing systems and extends beyond their hardware, software and networking capabilities. It is more than knowing how to code a robot to perform a set of commands, how to create visually stunning presentations and animations or how to build a network to keep digital information safe. These are all important benefits that we can reap by learning to be producers and not just consumers of technology, but equally important is knowing the why and how these computing technologies work and impact the global connectivity that our world revolves around. Other skills acquired in the effort to learn computer science, such as creativity, a growth mindset, problem-solving skills and critical-thinking aptitudes, are essential in every job that exists and are often as high on a list of desired traits in a candidates as technical expertise. Thus, the principles, concepts and practices acquired through studying computer science can benefit students not in both college and career readiness, but are essential life skills as well.

For students to have the opportunity to learn these skills, they must have access to high-quality instruction. This will require major shifts in learning for teachers whose backgrounds are not in computer science, but who see the potential it has to be a positive force in the lives of their students and are willing to commit to learning it in order to be able to turn around and teach it. The teachers in this research recognized the importance of this knowledge for their students to be better able to compete on a global stage, and were willing to put themselves in the position of a student to not only learn the material themselves, but learn how to teach it too.

This qualitative case study was designed to document the experiences of those teachers and use the results from the research to gain a deeper understanding of the teacher's needs to be successful. By better understanding these experiences, schools and districts can provide supports teachers need to successfully learn and teach computer science skills to elementary students.

Chapter 2 - Review of the Literature

Overview

This literature review highlights the urgent need for a strong K-12 computer science program in US public schools, and has a specific focus on elementary in-service programs. It begins with acknowledging on-going labor shortages in the US (BLS, 2021) and proposes a grow-your-own model to prepare teachers to be a part of the solution. It also establishes the need for a professional learning model for non-computer science trained K-6 teachers that includes a targeted needs assessment. Theoretical (TPACK, self-efficacy) and conceptual (Crawl-Walk-Run-Fly and Bruner's spiral) frameworks are also discussed. At the pinnacle of this literature review ladder is the ultimate goal of computer science for all students in all grades as well as building capacity in the cohort teachers as advocates for computer science in K-12 schools and communities. Figure 2.1 provides a graphic representation of the literature review contents.

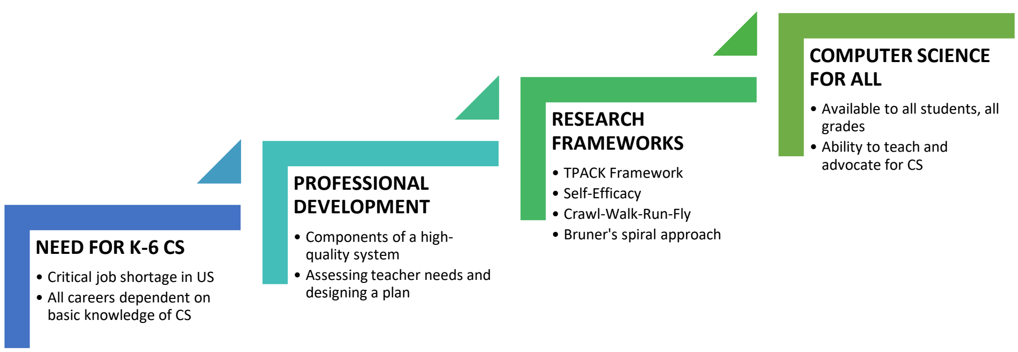


Figure 2.1. Graphical Representation of Literature Review

Before beginning the literature review, a more thorough explanation of computer science is necessary to lay a solid foundation for the argument that this subject and its related content is critical to the future success of our students and our country's workforce. There are many definitions of computer science, but this review was built upon Tucker et al.'s (2003) definition of computer science as presented in the *K-12 Computer Science Framework* (2016), developed in partnership with local and national education associations and national agencies advocating for the computer science education such as the Association for Computing Machinery, Code.org, the Computer Science Teachers Association and many other industry giants that support this critical initiative. Tucker et al.'s (2003) definition of computer science states:

> The study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society" (p. 6). Accordingly, computer science curricula includes all of the following: programming, hardware design, networks, graphics, databases and information retrieval, computer security, software design, programming languages, logic, programming paradigms, translation between levels of abstraction, artificial intelligence, the limits of computation (what computers can't do), applications in information technology and information systems, and social issues (Internet security, privacy, intellectual property, etc.) (p. 6).

Researchers Mason and Rich (2019) provided additional insight:

> Computer science is the broad discipline that encompasses computing, computational thinking, coding, and other branches dealing with computing connectivity and hardware. Computer science is the study of computers and computational systems (hardware software), their designs, capabilities and impact on society. It is far more than coding and has nothing to do with learning how to use a mouse, play video games or create a presentation. It is learning how to harness the power of a computer to solve a complex problem in a much shorter amount of time than humans can solve it (p. 792).

Need for K-12 CS

Computers play a vital part in our lives – from banking to education to healthcare and beyond. In the last 10+ years, STEM employment has grown by nearly 25% - over five times more than non-STEM employment over the same period, and throughout the 2020's, the U.S. Bureau of Labor Statistics projects that more than 1,000,000 STEM jobs will be added, a growth of almost 11% in the STEM field compared to less than 8% for all other industries combined (BLS, 2021). Still, the U.S. cannot fill all of the positions related to computer science now, which only adds to the concerns 10 years from now when additional STEM positions will be added. With this critical shortage, companies have turned to K-12 to try and grow a computer science workforce that will allow the United States to continue to be able to compete on a global stage. Even if students do not go into a computer science-specific field upon graduation, research has shown that almost every job available in this country will require some level of expertise in computer science or computational thinking (Brown & Brown, 2020; Salehi et al., 2020; Arfé et al., 2020).

Currently, the technology workforce in the United States is 90% white and 75% male (Scott et al., 2018). While there have been some gains in recent years, the disparity gaps continues to exist with Black and Hispanic workers as well as women in regards to STEM education (Pew Research Center, 2021). Black workers account for only 9% of the U.S.'s STEM workforce and Hispanics 8%, as Asian and White workers continue to dominate and while females consist of 47% of the workforce, they remain underrepresented in engineering (15%), computer (25%) and physical science (40%) occupations. For these trends to be reversed, a more rigorous approach to inclusivity in K-12 computer science education must be launched. Students of all colors, genders, ability statuses, and economic statuses must be afforded the opportunity to be exposed to these critical-need skills if they are to be successful citizens and workforce contributors. The entire school ecosystem – administrators, teachers, counselors, families and students – must be involved in this effort. Each has a specific role to play to ensure that computer science opportunities are available for all students and developing a sense belonging in STEM can have a positive impact on academic achievement and retention in STEM, particularly for women and students of color and may be positively influenced by relationships with mentors who look like them.

Exposure to ethnically diverse STEM professionals through multiple types of media (including social media) is a strong start to support and encourage underrepresented students to pursue STEM education and careers. Research has shown that women and men form gender roles in childhood, largely based on their exposure to media and popular culture (Olasson and Martiny, 2018) and that the types of media that students are being exposed to are rapidly changing, with the current generation of students spends a great deal of time interacting with media platforms and gaining exposure to social media influencers (Pew Research Center, 2019). Additionally, parents must be made aware of the critical-need for all students to pursue STEM-related courses and opportunities and the schools and school systems must find a way to bring this learning to every classroom if they are to prepare students to be successful post high-school.

Professional Development Plan

The proliferation of smart devices (tablets, phones, TVs, etc.) and the ease at which they are available have not only created a technological explosion, but has put innumerable amounts of information at the tip of their student's fingers that has changed the way learning has taken place. This has led to the need to provide teachers with additional resources and learning opportunities to better engage within this technological medium to support the needs of today's learners. In 2017, the Learning Policy Institute released their findings from an extensive review of professional development models that focused on positive relationships teacher professional development, teaching practices and student outcomes, and identified seven features of effective professional development, four of which will be identified and included in this paper. These include collaborating within the teaching community, providing coaching and expert support, offering feedback and reflection, and being built on a model of sustainability (Darling-Hammond, Hyler & Gardner, 2017).

Collaboration within the Teaching Community

Regarding collaboration, the authors asserted that by providing opportunities for teachers to share their ideas and collaborate in their learning, they could create communities that strengthened each other and that had the potential to positively change the culture and instruction for an entire grade level, department, school and/or district (Darling-Hammond, Hyler & Gardner, 2017). This collaboration could take on multiple forms: face-to-face, virtual, phone calls, online chats, blogs, social media pages, etc. and could also include opportunities to learn instructional tips and/or strategies for improvement and participate in lessons as a learner to better understand how the students will engage (Darling-Hammond, Hyler & Gardner, 2017). Collaboration could also be critical in helping overcome feelings of isolation that computer science teachers often feel in their own building as a result of being the only person on their campus who teaches their subject (Ahmad, 2017; Hamlen et al., 2018).

Coaching and Expert Support

Coaching and expert support in a professional development model involves the one-on-one sharing of expertise about content and strong instructional practices that can be directly focused on teachers' individual needs (Darling-Hammond, Hyler & Gardner, 2017). A system that has this component adds a layer of personalization for each teacher meets them where they are as an individual and works to provide the personal support that they need to be successful. In contrast to a traditional professional development model of a general workshop for multiple teachers, coaching occurs during the process of classroom instruction. Coaches are trained to observe teachers in their classrooms, provide feedback on the teaching and learning and engage in meaningful discussion with individual teachers about their lessons. (Quintero, 2019). This element is tied to building teachers' confidence in their own ability to both learn and teach the material.

Feedback and Reflection

Building in intentional feedback and reflection opportunities is also an element of a successful professional learning plan. In a cohort, this gives the learners the chance to not only think about, receive input on and make adjustments to their own learning (Darling-Hammond, Hyler & Gardner, 2017), but it gives them the chance to provide constructive comments to their cohort members in an effort to help the group as a whole be stronger in their practices. Personal and professional growth should be the focus and teachers should have an outlet through which to express concerns, share a-ha moments and be an active participant in their own learning and the learning of the cohort. With this project, each participant would create a blog and keep a record of their learning online. It was expected that every range of emotion from despair and frustration to victory and triumph would appear in their reflections as they move forward.

Model of Sustainability

Sustainable professional learning for teachers is rooted in research; it enables teachers to strengthen their personal professional networks, advocates for the formation of a learning community where teachers can safely share, cooperate, support and understand each other, while still allowing for systemic changes to occur (Villegas-Reimers, 2003). Recently, the movement towards self-directed teacher improvement has been encouraged, abandoning the traditional professional development models, which were typically focused towards teachers receiving knowledge (sit and get) rather than being active creators of knowledge (Makovec, 2018; Sumaryanta et al., 2019). Hargreaves and Fullan (2012) suggested that teachers improve at the workplace when they can exchange ideas, participate in study groups, research, etc. as well as share insight into problems that impact learning. But, they also warn that without a structure to the community, when teachers just discuss issues and share stories, that the learning piece may be ineffective (Hargreaves and Fullan, 2012). Finding a balance between the two would be critical to the success of this project.

Just as students should have personalized learning plans with inquiry-based opportunities embedded that are relevant to them, so should teachers. This should include adequate time to learn the content and the pedagogy, practice the learning, receive coaching, and reflect on their learning and students' outcomes as they continue in the cohort. Darling-Hammond et al. refers to this type of professional learning model as being of "sustained duration" (2017, p. 15), in that it lasts well beyond the initial "single, isolated encounter" (p. 15) and includes on-going trainings, a professional learning community and continued coaching/mentoring as part of the learning and support for the teacher learning a new subject.

Growing your own CS teachers

In recent years, computer science in K-6 classrooms has become more prevalent and school districts are steadily implementing new programs to support the successful development of computer science knowledge for all students. Current programs often include opportunities for students to code and create computer programs, develop problem-solving skills, study programming concepts, and develop computational thinking skills (Moreno-Leon et al, 2016). In order to prepare K-6 teachers who do not hold computer science credentials, school districts must find ways to provide training for these teachers that can not only teach them the concepts, but how to integrate those concepts into their subject matter to ensure sustainability (Howard, 2018). Researchers contend that the effective integration of computer science occurs as a distinct discipline and a school subject in the K-12 curriculum (Angeli et al. 2016; Fluck et al. 2016). Yet, despite the positive effects of integrating computer science concepts into daily instruction, teachers often struggle with unfamiliar pedagogical approaches and content (Howard, 2018).

Teachers' learning needs about technology integration have been historically documented as insufficiently supported through formal learning opportunities, and the formal and informal support they receive is imperative throughout the implementation process (Howard, 2018; Jones and Dexter, 2018). Although several of the challenges of teaching computer science in middle school and high school have been studied much more extensively, there is little known about the knowledge base or support strategies necessary to effectively support K-6 computer science teachers. This lack of clarity in regards to the start-up and on-going needs of K-6 computer science teachers and the current demand for training (Yadav et al., 2016) bring attention to the immediate need to understand the problems that K-6 teachers encounter when they attempt to integrated computer science into the classroom. Teacher resistance to such pedagogical changes, taken together with known barriers to computer science integration with current K-6 curriculum, suggest that the implementation of computer science programs may prove intimidating for teachers who have not undergone extensive training or who have a background in a related field.

Despite the potential resistance, preparing teachers to facilitate instruction is imperative and the only way to ensure that computer science is integrated in K – 6 classrooms. Specifically, what are the concepts that teachers new to coding most quickly pick up or struggle to understand? What computational thinking practices, tenets, and activities and are they most likely to engage in? Which of these principles and practices are most challenging? How long does it take for a teacher just starting to learn to feel comfortable teaching this new content? To effectively teach computer science in elementary schools, it is proposed that teachers need to have a strong foundation in computer science concepts, pedagogical knowledge, curriculum design including assessment and technology integration. A 2017 report by Yadev et al. asked readers to consider the following:

> Training teachers to teach computer science raises many questions for educators and educational researchers regarding the best practices for preparing teachers. How can we deliver content knowledge and promote effective instructional practices for in-service teachers? How do teachers acquire knowledge, including the knowledge of subject matter, pedagogy, and pedagogical content knowledge (PCK) in CS? What is the minimal set of computing skills that a teacher needs to be an effective computer science educator? (p. 3).

This research seeks to answer these questions through a grow-your-own model research study.

Research Frameworks

Research frameworks guide the design, execution and interpretation of research studies. They provide a structured approach to understanding a complex problem of practice, ensuring that the research is systematic and organized and establish a common language and form that invites collaboration, replication and further research. This research utilized both theoretical and conceptual frameworks.

Theoretical Frameworks

According to Creswell & Creswell (2018), theoretical frameworks consist of the theories or a set of concepts that underpin the research study and provide a theoretical perspective or lens through which the research problem is examined, often based on existing theories relevant to the research topic.

The TPACK Framework. Successfully integrating educational technology into subject-specific content is a three-pronged approach that requires a teacher to not only know their subject matter (knowledge) and how to teach it (pedagogy), but also which tool(s) (technology) can best be paired with it to enhance its effectiveness for students. This is where the TPACK Framework becomes a powerful instrument to guide teachers' use of technology in the classroom and how it works in tandem with the content and the way teachers present the content to students.

Starting with pedagogy, defined as the art and practice of teaching focusing on the accuracy of the teacher's knowledge and being able to use assess the effectiveness of instruction on student learning (Kohler & Mishra, 2009), the authors of the framework assert:

> [Pedagogy] focuses on teachers' deep knowledge about the processes and practices or methods of teaching and learning. It encompasses, among other things, overall educational purposes, values, and aims. This generic form of knowledge applies to understanding how students learn, general classroom management skills, lesson planning, and student assessment (p. 8).

When teaching computer science in elementary school, teachers need to be familiar with multiple pedagogical approaches and techniques that have been proven effective in disseminating this information to students such as inquiry-based learning, project-based learning and collaborative learning. Effective teachers with a strong pedagogical knowledge have the ability to design and implement engaging, interactive, and challenging learning experiences for their students that includes selecting appropriate instructional materials, designing age-appropriate learning activities and assignments and creating assessments that measure student learning (Kafai et al., 2017).

The technology being implemented must effectively communicate the content and support the pedagogy in order to enhance students' learning experiences in computer science and teachers need to be comfortable utilizing various technologies, such as programming languages, educational software, and digital tools, to support teaching and learning in computer science (Bell, et al., 2016). In the TPACK framework, technological knowledge (TK) also includes considering the possibilities needed for a specific subject area or grade level, learning to recognize when it will assist or impede the learning, and being continually open to learning and adapting new technologies and they become available (Kohler & Mishra, 2006).

Putting the three components together – content, pedagogy and technology – completes the TPACK Framework developed by Kohler and Mishra, which began in 2007 and has evolved over time because of its dynamics that allow for researchers and practioners to adapt its framework to multiple circumstances (Kurt, 2019). This adaptability can be seen in the overlapping intersections of the three areas of the framework. The technological content knowledge (TCK) demonstrates the relationship between technology and content and how they can both influence and oppose each other. This intersection focuses on understanding how computer science concepts can be learned through multiple technology offerings. The technological pedagogical knowledge (TPK) relates to the teachers' understanding of how a technology can affect both the teaching and learning experiences by the way it is utilized in the instruction. It also provides the opportunity for the teacher to demonstrate their knowledge by matching appropriate tools to pedagogies that are appropriate to the content being taught. The pedagogical content knowledge (PCK) focuses on the teachers' knowledge of instruction that includes curriculum development, instructional strategies, assessment, and student outcomes. Its primary objective is to improve instructional practices by bringing the content and pedagogy together with the most effective tool(s). Figure 2.2 shows Mishra & Koehler's TPACK Framework (2007) with the relationships between the constructs and Table 1provides the accompanying constructs and their definitions (Kurt, 2019).

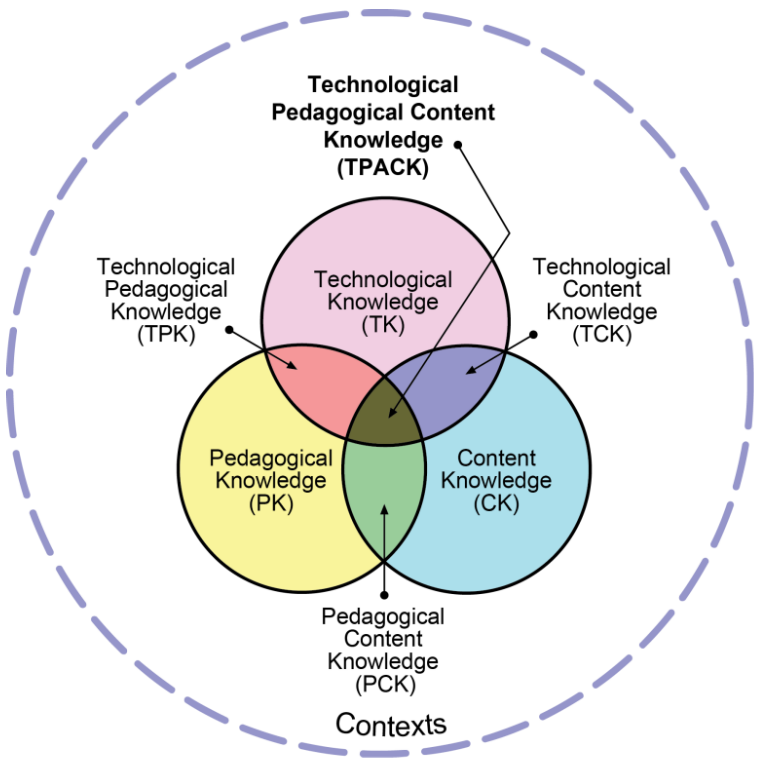


Figure 2.2. Mishra & Koehler's TPACK Framework (2007)

Note. Source: Reproduced by permission of the publisher, © 2012 by tpack.org.

Table 2.1. TPACK Framework Constructs with Definition (2019)

Note. Adapted from "TPACK: Technological Pedagogical Content Knowledge Framework" by S. Kurt, Source: **https://educationaltechnology.net/technological-pedagogical-content-knowledge-tpack-framework/**. ©2019 by Serhat Kurt.

TPACK Construct	Definition
Technological Knowledge (TK)	Knowledge of technology tools
Pedagogical Knowledge (PK)	Knowledge of teaching methods
Content Knowledge (CK)	Knowledge of subject matter
Technological Pedagogical Knowledge (TPK)	Knowledge of using technology to implement teaching methods
Technological Content Knowledge (TCK)	Knowledge of subject matter representation with technology
Pedagogical Content Knowledge (PCK)	Knowledge of teaching methods with respect to subject matter content
Technological Pedagogical Content Knowledge (TPACK)	Knowledge of implementing technology using multiple methods for different types of subject matter content

When it was originally developed, Koehler et al. (2007) asserted that development of TPACK contexts was focused on individual teachers as a "multigenerational process, involving the development of deeper understandings of the complex web of relationships between pedagogy, content and technology and the contexts in which they function" (p. 740). Meaning, that the framework measured the understanding, use and effectiveness of TPACK on a teacher-by-teacher basis. Recently however, research has shown that it can also be utilized effectively when teachers collaborate, or work together, thus theorizing that it can also be a form of knowledge shared by members of a teacher community (Carpenter et al., 2020). By developing a setting where teachers can collaborate around their own learning allows them the opportunity to share, learn from and with colleagues from across different levels of experience and expertise. It has the potential to bring together a wide range of knowledge and resources when utilized in this manner.

Bandura's Theory of Self-Efficacy. In order to deliver high-quality computer science education, teachers must be well prepared, have an established support system, and be professionally connected to one another. However, due to the aforementioned lack of common standards and clear certification pathways, school districts have continued to report struggles in hiring qualified computer science teachers (Google & Gallup, 2016; Yadav et al., 2016). The data was showing that the in-service teachers who are becoming computer science teachers were first having to be certified in another content area before being allowed to teach computer science (***Code.org,*** CSTA' & ECEP Alliance, 2020)! Further, these teachers who ultimately received approval to teach computer science reported facing a continuous lack of accessible and high-quality professional development opportunities, and resources to assist them in improving their computer science knowledge base and being able to better assess student understanding (Menekse, 2015; Yadav et al. 2016).

What had been routinely observed was that computer science teacher professional development that did exist was often short in duration (usually lasting about a week in the summer) and only sometimes had an ongoing engagement component during the school year (Darling-Hammond et al., 2017; Google & Gallup, 2015; Qian, Hambrusch, Yadav, & Gretter, 2018). These one-and-done professional development offerings sometimes lacked the necessary consistent follow-up opportunities needed for teachers to participate in order to feel that they could successfully learn new instructional techniques, and may left them them without the capacity to improve, often contributing to a sense of professional isolation and loneliness (Menekse, 2015; Franke et al., 2013; Yadav et al., 2016). This sense of isolation may have inhibited teachers' ability to take part in and share knowledge within a professional community in the same way as teachers in core subjects and could be compounded by a low sense of computer science teachers' self-efficacy—or the strength of teachers' beliefs in their abilities to teach CS. Many computer science teachers have reported low levels of self-efficacy (Yadav et al., 2016), which is related to teachers' institutional practices, occupational satisfaction, and job commitment.

Self-efficacy has been widely researched since the concept was pioneered by Albert Bandura as part of his Social Learning Theory in the late 1970's. Self-efficacy, according to Bandura, is a "personal judgement of how well one can execute courses of action required to deal with prospective situations" (1977), and that individuals develop self-efficacy from four main sources of influence – mastery experience (previous personal successes), vicarious experiences (seeing others successes), social persuasion (positive verbal feedback) and emotional state (well-being) (Bandura, 1977; Bandura & Cervone, 1986). In simpler terms, it was a person's belief in their own ability to be successful in a particular situation or task. Additionally, it reflected confidence in one's ability to control personal behavior and motivation. Bandura's position was that self-efficacy was the ability of a person to judge how they will react to a situation and/or the influence they have on the outcome of a situation (Bandura, 1977). Addressing self-efficacy issues and isolation concerns were critical to providing the high-quality computer science instruction.

Conceptual Frameworks

Ravitch and Riggan (2016) state that conceptual frameworks are a system of concepts, assumptions, expectations, beliefs, and theories that support and inform research. Further, that they help clarify the research question(s), and connects to existing literature.

Crawl-Walk-Run-Fly Model. The Crawl-Walk-Run-Fly (CWRF) model, a research-based approach commonly applied in core subject areas, provides a structured method of gradually introducing simple tasks before progressing to more complex ones. By focusing on mastering basic concepts initially and gradually scaling up to more advanced tasks, this strategy facilitates a step-by-step approach to learning. Figure 2.3 provides a graphical representation of this model.

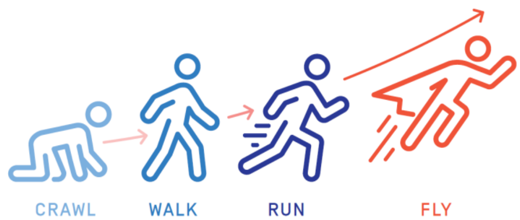


Figure 2.3. Crawl-Walk-Run-Fly (CWRF) model

Note. Source: LinkedIn, used by permission. ©2019.

The core principles of the CWRF model are rooted in the idea that learning is a gradual process, where individuals start with basic, foundational skills (crawl), then move to intermediate practice (walk), followed by advanced application (run), and finally achieve mastery and innovation (fly). This stepwise progression helps to prevent cognitive overload and supports sustained learning by building on previously acquired knowledge and skills (Dweck, 2006). According to Salas and Cannon-Bowers (2001), structured learning models like CWRF are effective in promoting both skill acquisition and retention, particularly in complex domains like computer science, where foundational knowledge is crucial.

The CWRF model aligns with several cognitive and educational theories. Vygotsky's (1978) Zone of Proximal Development (ZPD) highlights the importance of scaffolding in learning, which is closely aligned to the CWRF model. In each stage of the CWRF framework, learners receive the appropriate level of support to progress to the next stage, thereby operating within their ZPD. Research has also shown that the structured approach of the CWRF model can lead to increased improvements in learning outcomes. For example, a study by Wilson (2019) on professional development programs for educators found that participants who were trained using a CWRF approach demonstrated higher retention of knowledge and greater confidence in applying new skills compared to those who underwent traditional training methods. This is further supported by Kirschner, Sweller, and Clark (2006), who argue that structured instructional methods are more effective for beginning learners. In our research project, utilizing the CWRF model enabled teachers to spend ample time in each phase, thereby building the confidence and self-efficacy necessary to more effectively engage with their students in the subject matter.

There are some limitations to this model, however, that need to be considered when employing it in a training situation. One potential issue that needs to be taken in to account is the assumption of linear progression. Kirschner, Sweller, and Clark (2006) note that learning is often non-linear, with individuals needing to revisit earlier stages to reinforce understanding. To ensure the greatest success, it will also be imperative to take into account individual learning needs when employing this model. More on this limitation and a potential solution will be discussed in Chapter 6.

Bruner's Spiral Curriculum Approach. Jerome Bruner (1915 – 2016) was a renowned as an American psychologist who made important contributions to cognitive learning theory in educational psychology. Bruner advocated that learning was an ongoing process built on prior knowledge and experiences, and that children construct knowledge and make meaning through active experience with the world around them. He advocated that learning was an ongoing process built upon prior knowledge and experiences and one of his most widely known contributions was his spiral curriculum approach (1960). Bruner argued that learning was an iterative process, and that each time a topic was revisited, it not only reinforced what was already known, it deepened the knowledge, as new information was connected to previously known information about the topic. Figure 2.4 provides one graphical representation of Bruner's approach.

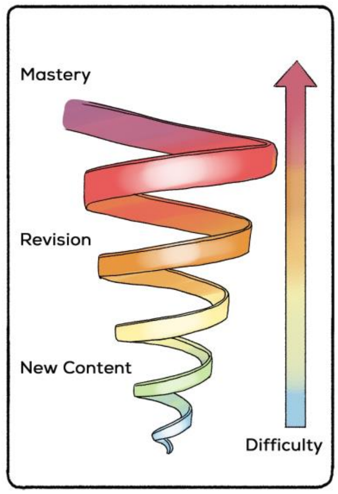


Figure 2.4. Bruner's Spiral Curriculum Approach (1960)

Note. Source: OER Commons, licensed under Creative Commons. ©2023 by Gagandeer Kaur.

Bruner's spiral approach is built on scaffolding, recognizing that learners may need varying levels of support at different stages. This support is flexible, adjusting to the learner's evolving needs and promoting independence over time as it enables them to take different paths and approaches to understand and apply knowledge. This adaptability supports personalized learning, catering to individual strengths, interests, and learning styles. This approach also promotes long-term retention and transfer of knowledge, as students continually apply and reapply concepts in various contexts (Bruner, 1977). Research has shown that this iterative process helps students develop a deeper and more flexible understanding of the subject matter, preparing them for higher-order thinking and problem-solving (Sawyer, 2006). Of this approach, Harden and Stamper (1999) wrote: "A spiral curriculum is not simply the repetition of a topic taught. It requires also the deepening of it, with each successive encounter building on the previous one." This framework was a late addition to this research project, but would end up as a powerful complement to the CWRF Model and will be discussed in greater detail in Chapter 6.

Summary

As there is currently no set certification pathway for elementary computer science, it is often left up to individual schools or districts to determine the content knowledge teachers must have to successfully teach this subject and oftentimes, it is in addition to other subject areas where they have received formal training in the instruction of it and have a strong knowledge base of the material being taught. There are many resources to guide and offer recommendations, but ultimately, each entity is able to set their own criteria for teacher knowledge in regards to content. While this non-specific approach has its advantages and disadvantages, there are basic computer science concepts that all who teach the subject should have a strong foundation of knowledge in such as algorithms, programming languages, data structures, computer systems and networks (Grover & Pea 2013; Guzdial, 2019). Referring back to the tenets of effective professional development discussed previously, opportunities must be also be provided for teachers to receive instruction in these areas, be able to demonstrate their own learning, observe model lessons being taught, teach lessons and be provided feedback, and be able to reflect on their work and adjust their practices as needed.

Chapter 3 - Methodology

Overview

Successful implementation of computer science programs at the elementary level that are effectively integrated into relevant curricula can lay a strong foundation to support students' growth and development in STEM-related fields. However, there is currently a lack of consensus on appropriate pedagogy not only for students, but also for teachers delivering the content (Franke et al, 2013; Liu et al., 2011). These known barriers, taken together with the historical resistance of teachers to curricular change (Fullan, 1993; Ford & Ford, 2010) suggest that the implementation of computer science programs may prove intimidating for teachers who have not undergone extensive training or who have a background in a related field. This project focused on adult learners who are all K-12 classroom teachers - their willingness to adapt to change by participating in a study of which they have little to no prior knowledge about, their ability to receive instruction in new subject matter and pedagogy and their capability to teach the content to students, and their capacity to lead change within their school communities.

Research Design

Qualitative research focuses on the experiences of an individual entity – e.g. a person, a community or an organization –and seeks to understand the nature of the research problem through their beliefs and practices (Strauss & Corbin, 1994). This approach has also been defined as an "iterative process in which improved understanding to the scientific community is achieved by making new significant distinctions resulting from getting closer to the phenomenon studied" (Aspers & Corte, 2019), which indicates that it can be an on-going opportunity to learn, analyze and share new information that will benefit others. While much of the research in K-12 computer science education has been focused secondary teachers and their experiences thus far, it is hoped that the lessons learned from the teachers in this project have the potential to help guide best practices for elementary computer science going forward.

The qualitative data that will be collected in this study included one-on-one interviews, pre-and-post survey results, anecdotal data from blog posts, and notes from my researcher journal. The importance of this textual (non-numerical) data is that it allows people to express their thoughts and beliefs in their own words and on their own terms and requires extensive interaction between the researcher and their subjects. An in-depth investigation in to this data will not only allow for the opportunity to understand the perspectives of the individuals living through the experiences (Creswell, 2013; Stake, 1995), but should also lead to the formulation of shared themes among the participants that could be used to inform future studies with this same group or others.

Therefore, in academic study, it is the responsibility of the researcher to engage in activities that generate and interpret their participant's data to answer specific questions. To do this, they must take a stance to decide what type of knowledge is possible and valid within their research. This project will blend constructivism and pragmatism through a series of document collections, interviews, surveys and research/participant reflections. Ontologically, it will follow the constructivist school of thought that there is no single reality or truth; that it is created by individuals or groups. It will also allow for the pragmatic ontology theory that reality is constantly negotiated. While slightly unconventional, as the study is searching for an answer not previously known, both of these ontologies can be applied. Epistemologically, it will lean towards pragmatism, where finding out what makes the learning and teaching successful is the means, and change is the underlying aim (Crotty, 1998). The objective is to look beyond the statistics (60% agree/strongly agree and 40% disagree/strongly disagree) of a Likert-scale survey of the participants and drill down to learn why and how, with the intent to try and replicate successful results to additional classrooms and participants and when able, identify a problem, its means and create a culture of change through research.

While some quantitative data will also be collected during this study, this data is limited in its ability to address "why" and "how" questions (Yin, 2014), so its primary function will be to validate the results collected from the qualitative data. Research from this project will be seeking to identify specific teaching resources and pedagogy that effectively contribute to the participants' willingness and confidence in themselves to embed CS-related activities into their daily instruction. It will also seek to gain an in-depth knowledge of the introduction of computer science in the district, investigating change to the school community as it becomes more a part of its culture.

Methodology

Research methodology is a logical, systematic plan that details how a researcher intends to carry out their research. It provides legitimacy to the findings of a study and is the foundational tool for the gathering and analyzing of data (Creswell, 2013). There are many different types of qualitative research methodologies that could be employed to guide this study, but the researcher chooses to use case study as the research methodology base.

Case Study Design

Yin (2002) recommends a case-study approach when investigating a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident. With the recent updates to the Technology Applications Texas Essential Knowledge and Skills (Tech Apps TEKS) that include coding and algorithmic-thinking instruction beginning in Kindergarten (Texas Education Agency, 2022), demand for increased elementary teaching and learning opportunities in computer science will likely have an impact on educators state-wide and many will be looking at solutions to fill those positions in their districts with personnel able to properly teach the material. And as was stated previously, while much of the research in K-12 computer science education has been focused at the secondary level, there currently exists a gap in the literature for elementary educators (Yadav et al, 2016; Kong, 2019), the research being conducted in this study can potentially help address that. Thus, a case study research design is appropriate for this study as teacher experiences at the elementary level have not been documented as thoroughly as those at the secondary level, and a case study approach could encapsulate these nuances.

Case studies are a preferred strategy when the researcher has little control over events as well (Yin, 2002). Such is the situation with this endeavor. The researcher does not have control over the school curriculum and subsequent pacing guides for the core subjects, nor does the researcher have control over what core content skills must be addressed and when. What the researcher does have some measure of influence towards is the way in which the content is presented to the students. With this study, we sought to determine what pieces and parts create a successful learning model for teachers to gain knowledge in a field that they are not familiar with as well as the knowledge of how to turn that learning into a teaching experience in their own classrooms. This is a very relevant endeavor in education today, with the continuing shortage of teachers in general and the lack of available computer science teachers currently in the field (NCES, 2022; Code.org, 2021).

Utilizing a case study approach will provided the flexibility to incorporate multiple types of data, while utilizing constructivism and pragmatism as research methodologies, whose theoretical perspective is rooted in design-based research. Rosenberg and Yates (2007) state that “[case study] has a practical versatility in its agnostic approach whereby it is not assigned to a fixed ontological, epistemological or methodological position.” In an educational setting, a community of teachers, researchers, coaches, administrators and other professionals would work together to approach a problem of practice. These methodologies often take an iterative approach to inquiry, which allows for the ontology and epistemology to be flexible and adjusted as needed through the design process. Engaging in multiple methodologies allows for constant renegotiation and signifies that this process is fluid and open to debate and that the decisions reached are not done so in isolation, but by a well-rounded committee that is actively engaged in and has a vested interest in the research.

For the context of this research project, the methodology used was referred to as a collective case study, in which a researcher will study a number of cases in order to provide insight into an issue or problem of practice (Stake, 1995; Stake, 2003) by collecting multiple forms of data from a specific group of people. The overall goal was to understand this one case and to maximize what can be learned to positively impact student and teacher development going forward. While this particular sample may have a relatively small sample size (between 10 and 20 maximum), for the purposes of investigating a representation of teachers, it will be sufficient. As Stake (1995) states; “Balance and variety are important; opportunity to learn is of primary importance” (pg. 18).

Study Context

This professional development study was derived from a larger computer science initiative that the participating school district had launched previously. Its primary purpose was to equip in-service teachers with the knowledge, skills, and pedagogy necessary to teach computer science at the elementary (K – 6) level. The teachers participating in this training initiative had agreed to meet regularly during the school year to share ideas and resources. They did not have a system in place for on-going professional learning, other than from each other, but when they did attend a professional development that focused on computer science, they were compensated for working off-contract.

To generate interest in this study, a recruitment flyer (Appendix A) was designed and deployed to all of the teachers currently participating in the larger district computer science initiative via email that included an abbreviated description of the overall project, the time frame it would be conducted within, and the requirements for completion. Two optional Q&A sessions (held via Zoom) were also offered and teachers who were interested in taking part in the research were then directed to an online form that would gather basic demographic information, and also give them the opportunity to respond to two open-ended questions regarding their interest in the study. Eighteen participants were identified and invited to attend this study. The next section will provided additional information about the participants.

Once the cohort was identified, it was announced that the teachers would participate in a year-long training program that includes both synchronous and asynchronous learning opportunities including, but not limited to, face-to-face and virtual sessions, classroom teaching experiences to practice teaching what they have learned, a summer computer science conference, and an on-going professional learning community, as well as keep a personal reflection of their own learning through online journaling. Data collected would include pre-and-post surveys, one-on-one interviews and notes from my researcher's journal.

Participants. Due to the high number of inquiries, the district agreed to allow up to 18 participants in the study. There were no specific requirements of subjects or content areas taught/teaching for participation; however, it was a personal goal of the researcher to create as inclusive of a group of participants as possible. It was hoped that building a group that was diverse not only in subjects or content taught, but also in color, gender, years of experience, schools currently teaching, and socio-economic backgrounds would help produce results that would resonate with multiple groups of staff and students within the school district. See Table 3.1 for participant demographics.

Table 3.1. CS Cohort Participant Data

GRADE(S)	SUBJECT(S)	YEARS OF EXPERIENCE
6th grade	Mathematics	1 – 5
1st grade	All subjects	6-10
8th grade	Science	6-10
4th grade	Mathematics, Science	6-10
5th grade	Mathematics	1-5
6th grade	Science, Social Studies	6-10
PreK – 6th grade	Art	11-15
PreK	All subjects	6-10
6th grade	Mathematics	6-10
4th grade	English/Language Arts	11-15
PreK – 6th grade	Instructional Coach	16-20
3rd grade	English/Language Arts	1-5
4th grade	English/Language Arts	1-5
2nd grade	All subjects	6-10
5th grade	Mathematics	6-10
6th grade	Science, Social Studies	1-5
6th grade	Science, Social Studies	1-5
PreK – 6th grade	Reading Specialist	16-20

This research project would employ voluntary-response sampling, a non-probability sampling method. For this sampling strategy, people would volunteer themselves to participate, which meant that the not everyone had a chance to be included. It was anticipated that multiple types and samples of data collected during this project would be utilized to further this growing wonder of teaching computer science in K-12, specifically in the K-6 realm where there are no certifications or exams required to do so. The project had been granted Institutional Review Board (IRB) approval and participants were provided with all relevant information before the project began and were asked to sign a statement of consent (Appendix B). This statement of consent also stated that participation may withdraw without penalty at any time. As a reminder, these teachers were already participants in a larger district computer science initiative for which they were receiving financial compensation, but they were not compensated for participating in this research study. Therefore, it was explicitly stated and understood that participation in the CS initiative was not dependent on participation in the research study and that participants were free to withdraw from the study at any time without losing their compensation for being a part of the initiative.

Structure and Funding of Materials. For non-trained teachers learning computer science skills, structure in professional development is especially crucial. As educators often begin with varying levels of familiarity with technology and coding concepts, having a structured approach can provide them with a clear and incremental learning path and learning goals, ensuring they build foundational knowledge before progressing to more advanced topics. By following a structured curriculum, non-trained teachers can gradually build confidence and competence in computer science, which is essential for integrating these skills into their teaching practices.

Moreover, structured professional development programs for non-trained teachers can offer the necessary support and resources to foster continuous learning. In their research, Fuller and Unwin (2004) highlighted the importance of robust learning environments in enhancing employee performance in the workplace, but their research also applies to teachers learning new skills in the classroom. For teachers, this means access to ongoing mentorship, collaborative learning opportunities, and practical applications of new skills through the integration of computer science skills and tools into the core curricular areas. The professional development also had to include checkpoints for feedback and reflection, allowing teachers to assess their progress and address any challenges they encounter. Building in this piece also allowed our teachers to look at their own learning more critically and set personal goals to continue to improve.

As was stated previously, this training would involve both synchronous and asynchronous learning opportunities that included self-exploration time, face-to-face and virtual training sessions, integration in to classroom teaching experiences, attendance at a summer computer science conference, and participation in an on-going professional learning community. The participants were also asked to keep a personal reflection of their own learning through online journaling. The tools and skills focused on in this professional development were selected using the newly revised 2023 Texas K-8 Technology Application TEKS vertical alignment (linked here) as a guide, but also included elements of choice for the teachers to self-select training on things that they were passionate about or wanted to learn about in greater detail. Additionally, elements from the district's current computer science curriculum for elementary were included. An overview of the professional development model used in this research is provided in Appendix C, but it is important to note that this should be considered as a suggestion or recommendation only, dependent on the availability of materials, funding, alignment requirements, etc. of the school or district that might attempt to replicate this study.

Data Sources and Collection

Data Sources. Data for this research was collected from the 18 teachers in the CS cohort. With the significant redesign of the Texas Education Agency's K-8 Technology Application TEKS that would be introduced in the 2024-2025 school year, there was a heightened sense of urgency to train teachers capable of instructing students in the area of computer science, which in turn had set up a fortuitous alignment with this case study, as the circumstances allowed for research in which some phenomenon was presently happening (Yin, 2014). In an earlier piece, Yin (2002) had also stated that case studies were a preferred strategy when the researcher has little control over events and when the focus is on contemporary phenomenon within some real-life context, which also helped make this research project so relevant and timely.

As case studies offer multiple approaches to instrumentation and data collection, there would be wide variety of available evidence to consider. However, to keep it manageable for one researcher three primary sources (pre-post surveys, semi-structured interviews, and notes from the researcher's journal) would be employed, as well as current research and policies regarding computer science. In addition to the full support of the school district's superintendent for this research project, approval from the Institutional Review Board (IRB) was also required for this research as it involved human beings as subjects and both have been been obtained. I also met with the teachers selected to participate and scheduled a meeting go to over the background information of the research, provided them with an opportunity to ask questions about the study and shared with them a link to the informed consent paperwork for them to sign indicating their willingness to participate. Presently, all 18 participants have agreed to the terms as outlined in the informed consent paperwork and have agreed to be a part of the research project as well as the professional development study.

Pre-Post Surveys. The surveys were administered at the beginning of the project and at the conclusion. The questions used in the survey were taken from the *Teacher Efficacy and Attitudes Towards STEM (T-STEM) Survey,* and were intended to "measure changes in teachers' confidence and self-efficacy in STEM subject content and teaching, use of technology in the classroom, 21st-century learning skills, leadership attitudes, and STEM career awareness" (Friday Institute, 2012). The initial survey results helped frame the first round of professional development for the program.

At the conclusion of the project, a follow-up survey was administered to assess the impact of the professional development program on the participants. This survey, like the initial one, utilized questions from the *Teacher Efficacy and Attitudes Towards STEM* (T-STEM) Survey to ensure consistency and comparability of results. The concluding survey aimed to measure any shifts in teachers' confidence and self-efficacy in STEM subject content and teaching, their use of technology in the classroom, 21st-century learning skills, leadership attitudes, and STEM career awareness. The data collected from this final survey provided valuable insights into the effectiveness of the program, highlighting areas of growth and identifying any remaining challenges. The results were instrumental in evaluating the overall success of the initiative and informing future iterations of the professional development efforts.

Semi-Structured Interviews. I anticipated conducting at least two thirty-minute to forty-five minute interviews with each participant during the course of this study to gain a comprehensive and thorough insight into their experiences. I chose to use a semi-structured protocol, which would allow my participants to expand on and elaborate on their responses during the interviews. As structured interviews do not provide this level of flexibility during responses, which can lead to additional probing questions and unanticipated discussion routes

(Creswell, 2013), I think that semi-structured interviews gave me the best chance to uncover and explore many facets, both intended and unintended.

The following questions were approved by IRB to be a part of the interview protocol.

- What new ideas have you gained thus far in your professional development training and how do you plan to implement these new ideas in your classroom?
- To what extent has the training impacted your belief in yourself to learn new content and new pedagogy?
- What additional resources or supports need to be put in place to ensure that computer science educational opportunities are accessible to all teachers and students?
- What concepts are you struggling with, either from the professional development or the implementation of the professional development?
- Is there anything else you would like to share?

The full protocol script is located in Appendix D.

Researcher's Journal. Maintaining a research journal was essential for me as it served as a comprehensive record of my research process, capturing the evolution of my thoughts, decisions, and reflections. This practice allowed me to document methodological choices, challenges encountered, and the rationale behind adjustments made throughout the study. According to Ortlipp (2008), keeping a reflective journal helps in making the research process transparent and aids in demonstrating the credibility and dependability of the study. By consistently recording my reflections, I was able to critically engage with my own biases and assumptions, thereby enhancing the rigor and depth of my analysis.

Further, my research journal acted as a valuable tool for personal and professional development. It provided a space for me to articulate my learning experiences, track my progress, and identify areas for further inquiry and skill development. Janesick (1999) emphasizes that journaling fosters a reflective practice that is crucial for continuous improvement and self-awareness in qualitative research. The journal also became a rich source of data in itself, offering insights into my perspective and the dynamic nature of my research journey. By systematically documenting my experiences, I was able to produce more nuanced and contextually grounded interpretations of my findings, ultimately contributing to the robustness of my research outcomes.

Data Collection. Stake (1995) defines analysis as “a matter of giving meaning to first impressions as well as to final compilations” which essentially means “taking our observations apart” (p. 71). He further contends that “Each researcher needs, through experience and reflection, to find the forms of analysis that work for him or her” (p. 77). As both theoretical and conceptual frameworks were utilized for this research, multiple types of coding were employed as well. A brief explanation of each follows.

A priori coding is the analysis approach that aligned most closely to this study for the theoretical frameworks (TPACK and Self-Efficacy). It is a qualitative data analysis technique where codes are determined before examining the data, based on pre-existing theoretical frameworks, literature reviews, or research questions. According to *The Coding Manual for Qualitative Researchers*, *a priori* codes are "derived from the research literature, theoretical constructs, or the specific objectives of your study” (p. 77). This approach allows researchers to systematically and consistently apply codes across data sets, facilitating comparison and pattern identification aligned with established theories or study aims (Saldaña, 2021).

Descriptive coding was employed for the data captured within the conceptual framework models, Crawl-Walk-Run-Fly and Bruner’s Spiral Approach. According to Saldana (2021), descriptive coding is appropriate for virtually every type of qualitative research, but especially appropriate for beginners (p. 88). He further asserts that descriptive coding is one approach to documenting from field notes the tangible products that participants create, handle, work with, and experience on a daily basis (p. 88).

With a research project that is collecting multiple sources of data, it is important to establish those protocols from the outset, along with the proper chain of evidence, in order to be able to use the synthesis of the data to answer the research questions. Establishing converging lines of evidence – data triangulation – is achieved when multiple independent sources all point to the conclusion. As this is the first research project of this magnitude that I have conducted, I anticipate following the structure set forth by Yin (2002), which asserts that case study researchers need to guarantee construct validity (through the triangulation of multiple sources of evidence, chains of evidence, and member checking), internal validity (through the use of established analytic techniques such as pattern matching), external validity (through analytic generalization), and reliability (through case study protocols and databases). This will help ensure that a novice researcher (like myself) maintains the integrity of the data collected and uses an established system to verify my findings.

Ethical Considerations

Ethical considerations are very critical to any research, and the responsibility of the researcher to safeguard the participants' rights and information is a key process that must be fully developed to maintain the integrity of the study. Prior to beginning, the participating teachers were provided with consent forms and information documents that outlines the specific data pieces that will be collected during this research. Before looking at the ethical considerations of the participants and their data, it is fundamental that the researcher(s) do all within their power to ensure the integrity of the entire research process. This means, among other things, that the research is trustworthy (valid and reliable), that the findings are based on evidence, that the arguments are justified and that the acceptable guidelines for conducting research within the chosen paradigm are followed (Ramrathan et al., 2016). It is a tight rope to walk – trying to be as transparent with research data as possible, while at the same time, ensuring the safety of your research participants. But for the research to be accepted by the educational community, these issues must be taken in to account and addressed to the greatest degree possible.

This research has another layer of ethics to manage as this study is being conducted within in the researcher's sphere of influence and they are deemed to be in the position of an 'insider' (Fleming and Zegwaard, 2018). While there are a number of benefits to being an insider, there are key challenges that cannot be ignored, including the potential for implicit coercion of participants, acknowledging the desire for positive outcomes, ensuring tacit patterns and regularities are not taken for granted and being sensitive to potential conflicts (Fleming and Zegwaard, 2018). To moderate this important piece of the research will require the services of an Institutional Review Board (IRB) to ensure participant safety and data safeguarding and the researcher is responsible for obtaining this permission. The researcher has received IRB training prior to beginning the research and has presented the required documentation to the Board for approval.

With human participants, it all begins with informed consent. Denzin and Lincoln (2011) state that the cornerstone of ethical research is 'informed consent' and that both elements – 'informed' and 'consent' require careful consideration. The 'informed' information was clearly stated from the outset and included basic details about the researcher, the intent of the research, what data would be collected, how it would be collected, used and reported, what would be required from the participants, and any potential risks. It would also include 'consent' information with how to "opt in" to the study, the right to withdraw at any time, assertions that all data will be kept confidential, their right to access their own data, and how a written process of how to register a complaint or concern. All information related to the participants in this study would be stored on a stand-alone USB drive that would be password-protected and unattached from the cloud. All physical items/artifacts will be kept under lock and key, accessible only by the researcher and both participant and school data will be de-identified through the district's student information system (SIS) and each participant will be assigned a multi-digit code identifiable only to the researcher.

Strategies for Ensuring Trustworthiness

In qualitative research, the criteria for quality work falls primarily under the heading of trustworthiness, meaning 'Can the findings be trusted?' (Lincoln & Guba, 1985 as cited by Korstjens & Moser, 2018) and the criterion referenced are often credibility, transferability, dependability and confirmability. When it comes to data validity and reliability, there are protocols that must be adhered to. "In our search for both accuracy and alternative explanations, we need discipline, we need protocols which do not depend on mere intuition and good intention to 'get it right.'" (Stake, 1995, p. 107). With a research project that is collecting multiple sources of data, it is important to establish those protocols from the outset, along with the proper chain of evidence, in order to be able to use the synthesis of the data to answer the research questions.

Establishing converging lines of evidence – data triangulation – is achieved when multiple independent sources all point to the conclusion. As this is the first research project of this magnitude that I have conducted, I anticipate following the structure set forth by Yin (2002), which declares that case study researchers need to guarantee construct validity (through the triangulation of multiple sources of evidence, chains of evidence, and member checking), internal validity (through the use of established analytic techniques such as pattern matching), external validity (through analytic generalization), and reliability (through case study protocols and databases). This will help ensure that a novice researcher (like myself) maintains the integrity of the data collected and uses an established system to verify my findings.

The critical piece of reflexivity – critical self-reflection about myself as a researcher – will be on-going throughout the research. Just as I will ask participants to document their journey, I will also document mine with a goal to address my own biases and preconceived notions in an effort to be as transparent as possible. As a first-time researcher, I anticipate that I will have to work diligently to monitor my own opinions of how I think things should be going in the research and what I believe is most important for teaching/learning. Making sure that I am accurately describing the setting and aspects of interviews, focus groups, observations, etc. will help to provide an accurate portrayal of the data collection and analysis.

Delimitations

Delimitations are defined as boundaries or limits that the researcher imposes on the study and are often imposed prior to the execution of the study and are meant to narrow the scope and make the study more feasible (Kraemer-Holland, 2022). For this study, I identified two delimitations. The first was the criteria for participant inclusion and the second was the notion that participants are being compensated for participation.

For inclusion into the study, the participants had to first agree to being a part of the CS teacher cohort, a district-led initiative. This included agreeing to take part in at least forty (40) hours of professional development off of their regular contract time, including two days in the summer for an out-of-town conference. For this, they would be a paid a sum of $2000.00. This payment would be made in three installments – May (for the work in the spring), August (for the work in the summer) and December (for the work in the fall). However, it was stressed repeatedly that participation in the study would not be required for participation in the cohort. But, participation in the cohort would be required for participation in the study.

Summary

The release of updated statewide computer science standards (K-8 Tech Apps TEKS) and increased adoption of computer science across elementary schools in Texas has created unique challenges for educators. Teachers with multiple-subject credentials are attempting to teach an entirely new concept with limited support resources and discovering many issues that they cannot move past. Furthermore, the majority of research on teachers' needs have been limited to secondary experiences, which has just exacerbated the already-tenuous situation at elementary and significant differences in teaching and learning from elementary to secondary perspectives have necessitated an urgent need for further research into elementary teachers' **e**xperiences. The research questions were designed to understand elementary teachers' experiences in learning and then teaching computer science in a large urban district. I anticipate gathering data through the phases and identified themes through coding and data analysis strategies shared in this chapter. The protocols employed will allow me to triangulate data to ensure validity and form a deeper understanding of elementary teachers' experiences with computer science education.

Chapter 4 - Teachers Cannot Teach What They Do Not Know: Using the TPACK Framework to Design, Implement, and Assess the Effectiveness of Computer Science Professional Development for Non-Computer Science Educators in K-6

Abstract

In a society dominated by technological advancements, the imperative for computer science education for all K-12 students has become undeniable, but identifying teachers capable of teaching the content has proven to be a formidable challenge. This article addresses the critical need for computer science education in elementary schools while tackling the challenge of preparing teachers with non-computer science backgrounds to effectively teach these concepts. Drawing upon the Technological Pedagogical Content Knowledge (TPACK) framework as a research-based model, which emphasizes the integration of technological, pedagogical, and content knowledge, the article advocates for a holistic approach that aligns teacher preparation with the demands of 21st-century education. This research contributes to the broader discourse on professional development in education, offering insights into effective strategies for overcoming barriers and fostering innovation in elementary computer science instruction for all.

Introduction – Statement of Problem

The integration of computer science education at the elementary level holds significant promise for equipping students with essential skills for the digital age. However, the potential success of this integration has been largely dependent on the teachers. It has come with a significant challenge to equip educators without a formal background in computer science with the necessary knowledge and skills to teach it. This requires a commitment to a comprehensive professional development plan that is grounded in research-based theories to identify needs, address knowledge gaps, provide time to reflect and practice, etc. and that is framed by continuous improvement practices to create an iterative process to evaluate the program to meet the evolving needs of educators and students.

The Technological Pedagogical Content Knowledge (TPACK) framework is a robust conceptual model for guiding professional development initiatives in technology. By understanding the dynamic interplay between technology, pedagogy, and content knowledge and how the relationship between the three can be used to enhance both teaching and learning experiences, the TPACK framework equips educators with a clearer picture of how technology intersects with pedagogical approaches and content knowledge, allowing them to discover meaningful ways to integrate technology tools and resources into their instructional strategies. This holistic understanding empowers educators to make informed decisions about selecting, implementing, and evaluating technology-enhanced teaching practices, thereby fostering more engaging and effective learning environments for students. This research project adopted the TPACK framework as its research-based model to use in the design, implementation, and assessment of a year-long professional development program tailored to teachers interested in teaching computer science, but who did not have the requisite background knowledge to do so.

Research Background

A review of the literature confirms that over the past decade, computer science efforts in K-12 have intensified and the need for in-service teachers to be properly trained in computer science content and pedagogy continues to be a challenge. A brief look at each perspective provides sufficient evidence to support the validity of this research project.

Increasing need for computer science in elementary education

Positive learning experiences in elementary grades are particularly important, as students who build personal confidence and competency in science, math, and technology classes are far more inclined to take additional coursework around these subjects (Krauss & Prottsman, 2017). Early exposure to computer science and its related skills has also been shown to foster essential cognitive skills and problem-solving abilities and contribute to the development of critical thinking and creativity skills in young learners (Resnick et al., 2017). Even before computers were in every home, Papert (1980) argued that computer science education in the early years encouraged a positive attitude towards technology, and empowered students to become active creators rather than just passive consumers of digital content, as they learned how and why computing technologies worked. Consequently, exposure to computer science and its related fields in K-12 could lay a solid groundwork for future academic and professional success, and allow students the opportunity to compete and thrive in a global society.

Multiple studies show that there continues to be a growing interest in and commitment to providing opportunities for all students to learn computer science in the United States (Code.org, CSTA & ECEP Alliance, 2020; Gallup & Google, 2020). According to a survey conducted by the Computer Science Teachers Association (CSTA) in 2019, 67% of K-12 school administrators expressed willingness to prioritize computer science education initiatives within their schools

(CSTA, 2019). As awareness has grown regarding the importance of computer science in fostering critical thinking, problem-solving, and computational literacy skills, educators are embracing opportunities to incorporate these principles across multiple content areas. Parents are also becoming more aware of the importance of computer science education, with nine out of ten parents supporting the inclusion of CS learning in schools (Google & Gallup, 2016). Even if students do not go into a computer science-specific field upon graduation, research shows that almost every job available in this country will require some level of expertise in computer science and its related fields (Brown & Brown, 2020; Salehi et al., 2020).

Challenges faced by non-computer science-trained teachers

Teachers cannot teach what they do not know, and a professional development plan tailored to their needs is vital if teachers are to obtain the necessary content knowledge and then integrate their newly acquired skills into teaching. Ni et al. (2021) underscored the critical role of ongoing training in enhancing educators' content knowledge, instructional practices, and confidence in delivering computer science education. The study emphasized that professional development initiatives built around the challenges faced by teachers without a computer science background are essential for promoting effective and sustainable integration into core content areas. Further, what had been routinely observed was that computer science teacher professional development was often too short in duration (usually lasting about a week in the summer), and only sometimes had an ongoing engagement component during the school year (Darling-Hammond et al., 2017; Qian et al.2018) and that one-and-done professional development offerings sometimes lacked the necessary consistent follow-up opportunities needed for teachers to participate in order to feel that they can successfully learn new instructional techniques (Menekse, 2015; Franke et al., 2013; Yadav et al., 2016).

Ensuring high-quality instruction in any educational domain necessitates a comprehensive approach extending beyond mere mastery of content knowledge. Educators must possess a deep understanding of concepts and effectively convey concepts to students. This foundational knowledge forms the basis for creating engaging and impactful learning experiences. Equally crucial is the need for a robust support system through professional learning communities (PLCs) for ongoing professional development. Lastly, establishing professional connections among the teachers facilitates collaboration and exchange of best practices. Altogether, the synergy of well-prepared content knowledge, a strong support system, and professional connections create a dynamic framework that empowers teachers to deliver high-quality instruction, ultimately enhancing the learning experiences of their students.

Utilizing the TPACK framework

In the current educational landscape, the purposeful integration of technology has become paramount for effective teaching and learning. This is where the Technological, Pedagogical, and Content Knowledge (TPACK) Framework can be employed to determine the depth and breadth of the teacher's personal accumulation of new knowledge, and it can also allow a deeper look into what the individual teacher does with the new content knowledge they have learned as they attempt to demonstrate that learning by teaching it to their students. Using this research-based model as the conceptual framework for this study provided a uniform basis to evaluate the design, implementation, and assessment of the professional development program.

The validity of the Technological Pedagogical Content Knowledge (TPACK) framework for professional development is evidenced by its ability to provide educators with a comprehensive understanding of how to integrate technology effectively into teaching practices (Koehler & Mishra, 2009). Proposed by Mishra and Koehler (2006) using Shulman's PCK

Theory (1986) as a starting point, the TPACK framework starts with three basic constructs of knowledge that expands to seven constructs as the three basic constructs interact with each other. The three basics are technological knowledge (TK), a knowledge of technology tools; pedagogical knowledge (PK), a knowledge of teaching methods; and content knowledge (CK), a knowledge of subject matter.

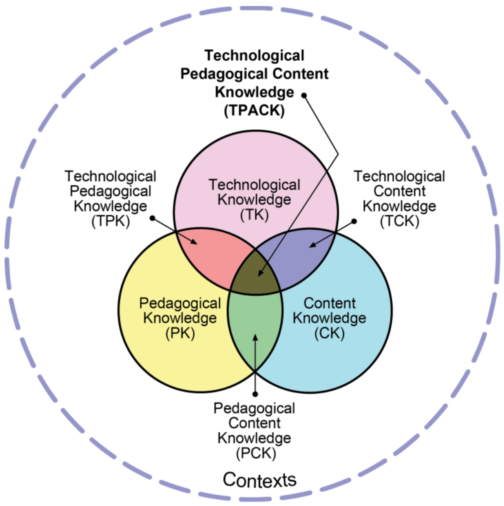


Figure 4.1. Mishra & Koehler's TPACK Framework (2007)

Note. Source: Reproduced by permission of the publisher. ©2012 by tpack.org.

As these three constructs interact with each other, four additional knowledge sources are formed: technological pedagogical knowledge (TPK) is the knowledge of using technology to implement teaching methods; technological content knowledge (TCK) is a knowledge of subject matter represented with technology; and pedagogical content knowledge (PCK) is a knowledge of teaching methods with respect to subject matter content. Finally, the "sweet spot", or place of intersection of all three sources, is Technological Pedagogical Content Knowledge (TPACK), which puts all three together at their maximum potential to create the highest impact on student learning. Research has underscored the necessity of ongoing support and training for teachers to successfully incorporate computer science into the elementary curriculum. It also emphasizes the importance of tailored professional development programs to enhance teachers' confidence and competence in delivering computer science education.

Research Design

Rationale for qualitative case study

Case study methodology offers a compelling approach to studying elementary computer science education due to its ability to provide rich, in-depth insights into complex educational phenomena within real-world contexts. According to Yin (2014), case studies allow researchers to explore the intricacies of computer science instruction in elementary settings, including curriculum design, teacher practices, and student learning experiences. Moreover, as highlighted by Stake (2006), case studies facilitate the exploration of multiple perspectives, enabling researchers to capture the diverse viewpoints of stakeholders involved, such as teachers with varying years of experience, teachers who teach different grades and subjects, etc. This holistic understanding is crucial for informing the development of evidence-based strategies to enhance computer science learning experiences at the elementary level (Weintrop et al., 2016) and enabled me to delve deeply into the complexities of this emerging field, yielding valuable insights that can inform both theory and practice.

Research Project Design

The purpose of this study was to determine if a year-long professional development program, built on the TPACK framework, could be an effective method for increasing teacher content knowledge, pedagogical skills, and technology acumen in the field of computer science. Research questions included:

1. How did utilizing the TPACK framework allow for a breakdown of the teacher's understanding of computer science into addressable components and identify existing gaps when designing the program? What impact does this information have on the program?

2. To what extent did aligning established professional development strategies with the elements of the TPACK in the implementation phase ensure that the teacher's gaps in knowledge were addressed in the plan?
3. How did the lens of the TPACK provide for a continuous improvement focus when assessing the overall effectiveness of the professional development project?

This study required a year-long commitment from the educators that was broken down into quarters and included a variety of training forms: face-to-face, online (asynchronous and synchronous), group and individual, and each teacher took part in an on-going PLC as well as maintained a blog of their experiences through the study. In the first quarter, the focus was on technological knowledge (TK), and the teachers were given a box of high/low tech devices to explore/play with on their own, complete tasks with each device, and blog about their experiences. The second quarter's learning objectives focused on content knowledge (CK) and included a variety of individual and group activities as well as bi-weekly PLC meetings. In the third quarter, teachers focused on pedagogical knowledge (PK), discovering strategies to integrate computer science into their curricular areas as well as differentiation strategies to meet the needs of multiple learning types. They also developed (or adapted) and taught a mini-lesson in their content area that included a set number of computer science skills. For the fourth quarter, they developed and taught a unit in their subject area.

Location & Demographics

This research project took place in a mid-sized, suburban Texas school district. The district has 21 elementary schools, 18 designated as Title I campuses, meaning that a minimum of 40% of the students must qualify for free or reduced lunch. There were approximately 11,000 students in K - 6, with a race/ethnicity breakdown as follows

Table 4.1. District Race/Ethnicity Statistics (October 2023)

Note. Source: Data obtained from Texas Education Agency. ©2023.

Race/Ethnicity	No. of students in K-6	Overall % of K-6 students
Latino/Hispanic	3,630	33%
White	3,300	30%
Black	2,240	22%
Asian	880	8%
Pacific Islander	330	3%
Two or more races	440	4%

Participants

The participants for this research project were recruited from the larger computer science initiative of the school district. For this project, a recruitment flyer was sent out to all K - 8 teachers in the district initiative in October 2022 and two Q&A meetings were held via Zoom in November and December 2022. Information was shared about the overall goals of the project, as well as timelines and participant expectations.

When recruitment was completed, participants included one male teacher and 17 female teachers. Their years of experience ranged from 0 years to 20+ years and their subject areas ranged from core curricular areas to art to instructional coaching. Additionally, at the first meeting, each was asked to share a one-word superpower or strength that they would bring to the group that was incorporated into a pseudonym for each one, as they preferred their comments to be anonymous. Table 4.2 lists the pseudonyms, subject(s) taught and years of teaching experience for the participants.

Table 4.2. CS Cohort Participant Data

Superpower Pseudonym	Subject(s) Teaching	Years of Experience
Newfangled Nancy	All (self-contained)	6-10
Consistent Cathy	Science	6-10
Present Patty	Math, Science	6-10
Creative Connie	Math	1-5
Empathetic Emily	Science, Social Studies	6-10
Crafty Campbell	Art	11-15
Curious Carla	All (self-contained)	1-5
Energetic Emma	Math	6-10
Engaged Elizabeth	Instructional Coach	16-20
Possibility Paula	English/Language Arts	16-20
Interested Irma	English/Language Arts	1-5
Passionate Pam	All (self-contained)	1-5
Positive Peg	Math	6-10
Excited Ella	Science, Social Studies	1-5
Driven Donna	Science, Social Studies	1-5
All-In Adam	Math	1-5
Learning Lisa	Reading	16-20

Data Collection & Analysis

Two primary types of data were collected during this project - surveys and interviews. The teachers were also encouraged to maintain a blog where they could continually reflect on their own learning and others could comment, offer suggestions/advice, etc.

The surveys were administered at the beginning of the project and at the conclusion. The questions used in the survey were taken from the *Teacher Efficacy and Attitudes Towards STEM (T-STEM) Survey,* and it was intended to “measure changes in teachers’ confidence and self-efficacy in STEM subject content and teaching, use of technology in the classroom, 21st-century learning skills, leadership attitudes, and STEM career awareness” (Friday Institute, 2012). The initial survey results helped frame the first round of professional development for the program as it provided data that aligned with the TPACK constructs of pedagogy, content, and technology knowledge in regards to computer science.

Two in-person interviews were conducted with each participant - one at the six-month mark and one at the end of the research. These questions were designed to gauge the effectiveness of the implementation and to see what suggestions the participants might have to improve the training.

After the interviews were conducted, the information was analyzed using an *a priori coding* process. This process involved beginning with a set of predefined codes, in this case, the constructs of the TPACK Framework, and then identifying patterns, themes, and relationships within the data. It was an iterative process and involved constant comparison and refinement as nuances embedded within the dataset were manifest. As these were discovered, they led to a narrative woven within the data.

The *a priori coding* process also provided me with a systematic way to organize and make sense of my findings. Using the TPACK Framework constructs helped ensure consistency throughout the analysis and provided a focused framework to help efficiently hone in on key themes and patterns within the data. It also provided a solid level of transparency in qualitative research.

Results

TPACK Alignment Data

Aligning the questions from the *T-STEM Survey* and TPACK Framework provided a research-based model to use to complete the initial assessment of each teacher's knowledge of technology, content and pedagogy based on their pre-survey results and also identified gaps that needed to be addressed in the initial design of the program. The training was conducted in three rounds, with each round having its own set of items to be completed and reflected on. I referred to the training as ***crawl-walk-run***. Using the concept of ***crawl-walk-run*** to teach new knowledge involved gradually introducing complex ideas in stages, starting with simpler concepts before progressing to more advanced ones (Mayer, 2019). Just as infants learn to crawl before they walk and eventually run, learners built a solid foundation of understanding before tackling more intricate topics. This incremental approach fostered comprehension, confidence, and mastery, ensuring that students are well-equipped to navigate and apply newfound knowledge effectively (Vygotsky, 1978).

The ***crawl stage*** (January - April) centered around the teachers learning to use the technology tools themselves. They were given a series of tasks to complete for each one, then they had to post evidence of their work in our shared drive, and finally, reflect on their own learning on their blogs throughout their assignments. The purpose of this stage was to let them play/explore at their own pace and build a solid foundation of personal understanding before attempting to teach it to someone else. The ***walk stage*** (May - August) was next and not only provided additional time for solo work, but added the component of designing a 30-minute to one-hour technology training session on any one of their new tools, providing all lesson plans and documentation of having taught a lesson - whether it be to a single teacher, a small group, at a faculty meeting, or to students. They were also to continually reflect on their own blogs and read to provide constructive feedback on fellow participants' blogs. Additionally, they were each able to choose a conference to attend - an in-person one within the state or a national one virtually. There were reporting requirements for the conferences as well. The ***run stage*** was the concluding quarter of the year and required the participants to significantly redesign an existing unit (minimum of 5 days in length) within their curriculum to include at least four computer science/computational thinking tools, including multiple strategies specifically designed to teach the tools and have the students produce a product reflective of their learning. As with the other stages, they also had to reflect regularly on their blogs and give feedback to others.

Technological Knowledge. Even though most educators rated themselves extremely high in using computers on the pre-survey, when discussing technological knowledge in their first interviews, multiple participants acknowledged a knowledge deficit in this area, particularly in terms of emerging technologies. Many cited lack of access to cutting-edge equipment as a major barrier to not only their learning, but to their students' learning as well. Adam felt that if computer science was "the way to move the district and its students forward" that they [the district] would need to invest in newer and more capable devices for students and staff. Ella also noted that, while the latest-and-greatest technology might not be available, that still "providing students with access of any kind to coding materials and resources at school helps close the gap with our low socio-economic students." Clearly this construct is one that the district must address as it continues to move forward.

At the end of the study, this area was still the weakest overall. Of the 18 respondents, 14 indicated some level of increased capacity regarding technological knowledge, saying that their expanded vocabularies of technical terms had helped them to "better articulate an issue" which had led to “better and quicker response times from IT’s end.” Other concerns also bubbled up, including digital citizenship, safety, and cybersecurity which had not been observed in the first round.

Content Knowledge. Preparing teachers for computing education continues to be a critical challenge as there is much confusion around state and national certification standards and what content teachers actually need to learn and what skills they need to develop to become successful CS teachers (Grover & Pea, 2013). In the first interview, participants highlighted the struggles of learning about computer science and computational thinking beyond just surface-level content knowledge. Several mentioned having never heard the term ‘computational thinking’ before starting the cohort. Cathy felt like she was “learning not only a whole new set of skills, but also a whole new vocabulary.” Nancy stated that as she learned more, her confidence grew and she felt empowered to use her new learning to “break down concepts for students to make the learning more manageable.” For these teachers, starting with the basics was critical.

In the second interview, the majority of the participants indicated that they had broadened their own personal knowledge of computer science and computational tools and how to use them, as well as many shared that they started to look forward to what others might be coming down the pike and how they could fit into the classroom.

Pedagogical Knowledge. When responding to questions about pedagogy in the survey and in their first interview, the participating teachers exhibited more confidence in their answers, citing specific examples of ways to integrate computer science concepts into the classroom including using task/problem cards, setting up stations for student exploration and creating student coding journals. Regarding student coding journals, Danika stated, “The journaling aspect is brilliant! Not only do they [the students] have a place to help keep them on-task and organized, but they have somewhere to reflect on their learning, including what worked and what didn't." Peg touted the instructional strategy of setting up stations in the classroom for students to explore and rationalized that letting students explore or play is a great way to see what interests them when learning about new concepts. Based on the first interviews with the participants, increased pedagogical knowledge enabled teachers to effectively integrate technology into their teaching, ensuring that digital tools enhance, rather than detract from, the learning process. This understanding of pedagogy within the TPACK framework emphasizes the importance of educators not only knowing their content but also possessing the skills to convey that knowledge effectively, fostering a positive and impactful learning environment.

In the second round of interviews, participants stressed using computer science and computational thinking tools more "purposefully" in the classroom and discovering the benefits of utilizing the same device of multiple grades and content areas. Still, Anna emphasized that "If I am not well-versed in the subject matter, I will not know where I can best put these new tools in the curriculum for maximum student impact" and Marge said that "for more mature teachers, it will be important to help them move past the 'this is always how we've done it mentality'" as they bring new ways of teaching into the classroom.

Pedagogical Content Knowledge (PCK). Pedagogical Content Knowledge (PCK) represents the unique intersection of pedagogy and subject matter expertise - what to teach and how to teach it - and it involves a strong understanding of how to make it accessible, meaningful, and engaging for students. Educators possessing strong PCK understand not only what to teach (content knowledge) but also how to teach it (pedagogical knowledge), integrating appropriate teaching strategies to convey complex concepts to students while utilizing technological tools to enhance instructional practices (Koh, Chai, & Tsai, 2014).

When first questioned, cohort participants stressed the importance of being able to integrate computer science and computational thinking into core curriculum classes. Grade-level core teacher Emma was emphatic in her interview stating “It is important that computer science (and computational thinking) not be seen as either ‘one more thing’ or as a ‘one-off’ to the rest of the grade-level curriculum…integration into all subjects will be critical to get the teacher buy-in we need.” Still others, who teach non-traditional classes, were appreciative of the fact that computer science does reach beyond the traditional core content areas and can be integrated into their classes as well.

In their second interview, the focus shifted slightly with multiple teachers expressing concerns about how digital citizenship would need to be incorporated into computer science skill sets, for both themselves and the students. Mimi summed up the thoughts of many in the cohort when she stated that “Digital citizenship will be a critical training piece for teachers and students who will both need to be trained in the ethical use of AI” in her interview and questioned her own ability not only to learn of best practices in digital citizenship but how to impart that learning to her students so that they understood the importance of ethical behavior online.

Technological Content Knowledge (TCK). This construct refers to teachers' ability to understand how technology and content connect. Specifically, in the context of elementary computer science education, it identifies educators' proficiency in both computer science concepts and the appropriate use of technology to teach these concepts effectively. Proficiency in TCK empowers teachers to select age-appropriate software, programming environments, and interactive tools that align with the curriculum and cater to the diverse learning styles of elementary students (Sentance & Csizmadia, 2017). It also enables them to design creative and interactive learning experiences.

In the first conversation, participants expressed some of the same concerns here that were expressed in the technological knowledge section, which included 1) a knowledge deficit in emerging technologies, 2) a lack of access for students and teachers, and 3) a shortage of up-to-date technological devices. However they also indicated positives regarding TCK, including several who referenced utilizing online simulations and game-based learning platforms to provide exposure to CS-related career fields and opportunities. Adam mentioned that his class was "going beyond traditional coding websites and showing them real-world online simulations in areas like Cybersecurity" to help them start thinking about possible pathways when matriculating to junior high. Lisa and several of her colleagues also stressed the importance of providing age-appropriate resources and the need to work with your IT department to ensure suitability for the younger ages.

When this was revisited in the second interview, many of the same issues arose again. While there was not much that the researcher could do about access and up-to-date devices, I did feel like I could affect the first concern - that of a knowledge deficit in emerging technologies - by providing the cohort with as many resources in the field of emerging computer science technology as possible. This might not have been the best practice, because Ally observed "More does not always equal better – I need to know what tool will best help facilitate the learning" was echoed by several in the group, leading me to rethink that particular strategy.

Technological Pedagogical Knowledge (TPK). Technological Pedagogical Knowledge (TPK) refers to teachers' understanding of how to effectively integrate technology into their teaching methods. Specifically, in the context of elementary computer science education, TPK involves educators' ability to combine their knowledge of pedagogy with technological tools to create engaging and effective learning experiences. It encompasses the skills to select appropriate digital resources, design interactive lessons, and foster collaborative learning environments, ensuring that technology enhances the teaching process and supports student learning (Archambault & Barnett, 2010). TPK plays a crucial role in shaping the learning experiences of young students, as effective integration of technology not only captures children's attention but also cultivates their interest and enthusiasm for computer science concepts. Teachers with strong TPK can employ various digital platforms, educational software, and online resources to present complex coding and computational thinking concepts in a comprehensible manner, making learning interactive and enjoyable (Voogt et al., 2013). Additionally, TPK enables educators to facilitate collaborative coding projects and interactive activities that encourage teamwork and problem-solving, essential skills in the digital age.

One recurring theme emerged from the first round of questioning – knowing when to introduce the technology into the classroom beyond exploration and play. Tiffany shared an experience of leaving a micro:bit on her desk and a student inquired about what it was and what it could do. She said she honestly did not know and the student asked to play with it; within ten minutes, he and his friends had watched a couple of videos, done some simple testing, and completed the “Hello World” tutorial! In her words, “The students are fearless and unashamed to ask for help when they cannot figure out what to do next.” From their answers, it would appear that the teachers would do well to follow the example of their students.

Technological Pedagogical Content Knowledge (TPACK). This is the “sweet spot” – where it all comes together and the true magic happens! The "sweet spot" in the TPACK framework represents the intersection of technological knowledge (TK), pedagogical knowledge (PK), and content knowledge (CK) and recognizes teachers' ability not only to comprehend the fundamental concepts of computer science (CK) but also to apply pedagogical strategies (PK) that cater to the diverse needs of young learners. Simultaneously, it involves leveraging appropriate technological tools and platforms (TK) to enhance the learning experiences and make the subject matter accessible and engaging (Niess, 2005). The importance of the "sweet spot" becomes evident when considering the dynamic and rapidly evolving nature of technology and available resources, and the multiple levels of cognitive abilities of elementary students. Teachers who are in this zone are adept at selecting age-appropriate technological tools that align with the curriculum, creating innovative lesson plans, and developing collaborative projects that not only deepen students' understanding of computer science but also foster critical thinking, problem-solving skills, and a positive attitude toward technology and learning.

Within the first round of interviews, participants indicated that getting to this level (and staying there) would be their ultimate goal as educators. Hannah remarked "It was awesome to see the same CS tools being used across grade and subject levels – the teachers could work and learn together! Great synergy!". In her first interview she focused on the need to be intentional in the integration of new materials. The theme of intentionality carried into the second round of interviews where Ellie talked about being intentional when using computer science terms and vocabulary and provided a short example:

> I teach ELA and my co-teacher teaches math…we committed to using the word 'debugging' in our instruction. For math, it was to figure out why they had the wrong answer and the students had to go back and look at each step of the problem – like a line of code – and find their error. For ELA, if their score on a comprehension was lower than normal, we would 'debug' and figure out where their comprehension (or fluency if that's what I was testing) had broken down.

Conclusion

In conclusion, this article advocates for the integration of the Technological Pedagogical Content Knowledge (TPACK) framework into elementary computer science education professional learning as a means to enhance teaching practices and promote deeper learning among students and teachers. Through qualitative analysis and teacher reflections, the study underscores the importance of aligning technological tools with pedagogical strategies and subject content and being able to make adjustments as dictated by the data. And now they can teach computer science, because they (better) know it.

Program Design

As a researcher, I felt like the overall design of the program was solid and not only took into account the knowledge that the teachers were coming into the program with, but also highlighted gaps in their learning, and provided a plan to address them. By incorporating hands-on experiences, collaborative learning opportunities, and real-world applications, I felt like the teachers developed a deeper understanding of computer science concepts while simultaneously honing their instructional strategies.

Further, the design of this program prioritized flexibility and customization to accommodate varying levels of prior knowledge and experience. Going forward, offering differentiated pathways that allow the educators to engage with content at their own pace and address specific areas of interest or need would better honor the educators' prior knowledge. Incorporating authentic, project-based learning experiences aligned with real-world contexts also helped enhance understanding and allowed for greater creativity and the development of personal problem-solving skills.

Lastly, integrating opportunities for interdisciplinary connections and cross-curricular collaboration provided an additional level of enrichment by encouraging educators to explore innovative ways of integrating computer science across subject areas. By emphasizing hands-on exploration and personalized learning pathways, I felt like the program design empowered teachers to become more confident in their own abilities and more proficient facilitators of computer science education in their classrooms. Empowered teachers gain confidence in their teaching strategies, allowing them to create more dynamic and engaging learning environments. With increased empowerment, teachers are better equipped to adapt their methods to meet diverse student needs, thereby enhancing their overall proficiency as facilitators. Moreover, a sense of empowerment encourages teachers to continuously pursue professional development, further honing their skills and effectiveness in the classroom.

Program Implementation

Incorporating critical elements such as professional learning communities (PLCs), collaborative learning opportunities, mentorships, and individual voice/choice laid a strong foundation for this program to be implemented with accuracy and consistency by the participants as implementation for teaching computer science to teachers requires careful planning and support mechanisms to ensure successful integration into educational settings. Beginning with comprehensive needs assessments and overall guidelines, this program tailored content and delivery methods to meet the diverse needs of educators. The professional development offered ample opportunities for hands-on exploration, peer collaboration, and ongoing support through coaching and mentoring, all of which were provided to strengthen teachers' confidence and competence in computer science learning and instruction. Moreover, embedding opportunities for reflection and feedback allowed me the chance to continuously refine practices and address any challenges encountered along the way.

By fostering a culture of experimentation, adaptation, and professional learning communities, this program helped develop the seeds of a potentially sustainable ecosystem where teachers could feel empowered to not only develop computer science skills of their own but also to incorporate computer science education effectively into their classrooms, ultimately equipping students with essential 21st-century skills.

Program Assessment

Even when every response from participants is glowing, there are always ways to improve the program to better meet the needs of the participants. Utilizing tenets of continuous improvement when assessing program effectiveness involves an iterative cycle of data collection, analysis, and action to adapt to the evolving needs of students and ensure that teaching methods remain effective and relevant. By consistently refining and enhancing instructional strategies, educators can foster a more engaging and supportive learning environment, ultimately leading to better student outcomes. The key is to start by regularly collecting feedback from stakeholders - the program participants and their students, when possible. Analyzing this data allows for informed decision-making and the implementation of targeted interventions to address identified challenges whether they are localized to one or two teachers or evidenced throughout the participants. Through ongoing reflection and adjustment, programs can adapt to evolving needs and circumstances, ultimately enhancing their effectiveness and ensuring sustained impact on teacher learning and student outcomes.

Limitations and Future Research

The pre-survey helped with the initial design of the program's contents and the data from the first round of interviews was used to make adjustments to the training as determined. Three participants did not complete their second interviews but did finish all of their assignments. I do not feel that this compromised the research and that what was reported was a fair and accurate representation of all of the participants.

A key goal of the research was to prepare additional district teachers to teach computer science at the elementary level so that when the program was expanded to include additional schools, there was a pool of candidates with sufficient pedagogical and technological know-how to be able to teach the curriculum. I feel like this was accomplished, particularly as this cohort of teachers requested to be able to continue to blog their learning and document their struggles and challenges, as well as their a-ha moments and successes.

The recruitment of teachers to this project was teacher-driven; the project appealed to teachers who were interested in teaching and learning computer science and they self-selected to participate. Upon reflection, I have identified two concerns with this approach that should be considered in the next round of research. First, the teachers in this cohort did not evenly represent the diverse population mix of the school district they serve. If the goal is to attract all students to computer science, then they must see themselves represented by the educators who teach it. Second, these teachers signed on because of their own self-interest, and perhaps went above and beyond expectations because they wanted to be considered for any expansion positions that might be available at a future date. A similar level of commitment might not have been present in teachers if they were assigned to be a part of a cohort. As such, should a project of this nature be undertaken in the future, adjustments will need to be made to the recruitment process to ensure that the representation of participants more accurately reflects the diversity of the school district and that support structures are put in place to compensate for the potential self-selection bias of the participating teachers.

Acknowledgements

I would like to express my sincere gratitude to Dr. Suzanne Porath (Kansas State University) for her valuable contributions and unwavering support throughout the research process. Dr. Porath's expertise, guidance, and assistance have been instrumental in shaping this study. Additionally, I extend my heartfelt appreciation to the participants who generously shared their time and insights, without whom this research would not have been possible, and for the assistance and encouragement of the cooperating school district. Lastly, I acknowledge the broader academic community for their ongoing efforts in advancing knowledge in this field.

References

Archambault, L. M., & Barnett, J. H. (2010). Revisiting technological pedagogical content knowledge: Exploring the TPACK framework. *Computers & Education, 55*(4), 1656–1662. https://doi.org/10.1016/j.compedu.2010.07.009

Brown, E., & Brown, R. (2020, March 4). The effect of Advanced Placement computer science course taking on college enrollment. West Coast Analytics. http://www.westcoastanalytics.com/uploads/6/9/6/7/69675515/longitudinal_study_-_combined_report_final_3_10_20__jgq_.pdf

Cheung, A. C. K., & Liu, A. M. M. (2018). Creating a collaborative learning community in a computer-supported collaborative learning environment: An examination of pedagogical design and interactional pattern. Interactive Learning Environments, 26(3), 370-383.

Code.org. (2021). *Texas Computer Science: State Advocacy Sheet*. Support K-12 Computer Science Education in Texas. Retrieved October 14, 2021, from https://code.org/advocacy/state-facts/TX.

Code.org, CSTA, & ECEP Alliance. (2020). 2020 State of computer science education: Illuminating disparities. Retrieved from October 14, 2022. https://advocacy.code.org/stateofcs

CSTA. (2013). *Bugs in the System: Computer Science Teacher Certification in the U.S.* CSTA - certification resources. Retrieved March 1, 2022, from https://codes-isss.org/csta_subdomain/ComputerScienceTeacherCertification/sub/CertificationResources

Darling-Hammond, L., Hyler, M. E., Gardner, M. (2017). Effective Teacher Professional Development. Palo Alto, CA: Learning Policy Institute. https://doi.org/10.54300/122.311.

Franke, B., Century, J., Lach, M., Wilson, C., Guzdial, M., Chapman, G., & Astrachan, O. (2013). Expanding access to K-12 computer science education: research on the landscape of computer science professional development. In *Proceedings of the 44th ACM technical symposium on computer science education* (pp. 541–542). Denver, CO: ACM.

Friday Institute for Educational Innovation (2012). Teacher Efficacy and Beliefs toward STEM Survey. Raleigh, NC.

Gallup Inc., (2019). *Current perspectives and continuing challenges in computer science education in U.S. K-12 schools*. Gallup.com. Retrieved October 1, 2022, from https://csedu.gallup.com/home.aspx

Google Inc. & Gallup Inc. (2015). *Searching for Computer Science Report - Google Search*. Current Perspectives and Continuing Challenges in Computer Science Education in U.S. K-12 Schools. Retrieved October 4, 2021, from https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf.

Google Inc. & Gallup Inc. (2017). *Encouraging Students Toward Computer Science Learning*. Results From the 2015-2016 Google-Gallup Study of Computer Science in U.S. K-12 Schools (Issue Brief No. 5). Retrieved October 4, 2021, from https://goo.gl/iM5g3A

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. Educational Researcher, 42(1), 38-43.

Koehler, M. J., & Mishra, P. (2009). What is technological pedagogical content knowledge? *Contemporary Issues in Technology and Teacher Education, 9*(1), 60-70.

Krauss, J., & Prottsman, A. (2017). Computational thinking and coding for every student. *Journal of Computer Science Education*, 25(3), 145-162.

Lee, K., Tsai, P.-S., Chai, C., & Koh, J. (2014). Students' Perceptions of Self-Directed Learning and Collaborative Learning with and without Technology. Journal of Computer Assisted Learning, 30, 425-437. https://doi.org/10.1111/jcal.12055

Mayer, R. E. (2019). How do students learn and what should teachers do? In R. E. Mayer (Ed.), The Cambridge Handbook of Multimedia Learning (2nd ed., pp. 3-32). Cambridge University Press.

Menekse, M. (2015). Computer science teacher professional development in the United States: A review of studies published between 2004 and 2014. *Computer Science Education*, 25(4), 325–350. https://doi.org/10.1080/08993408.2015.1111645

Mishra, P., & Koehler, M. J. (2006, June). *Technological Pedagogical Content Knowledge: A framework for teacher knowledge*. Teachers College Record. Retrieved February 6, 2023, from https://www.researchgate.net/publication/220041541.

Niess, M. L. (2005). Preparing teachers to teach science and mathematics with technology: Developing a technology pedagogical content knowledge. *Teaching and Teacher Education*, *21*(5), 509–523. https://doi.org/10.1016/j.tate.2005.03.006

Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas. Basic Books.

Qian, Y., Hambrusch, S., Yadav, A., Gretter, S. (2018). Who Needs What: Recommendations for Designing Effective Online Professional Development for Computer Science Teachers, Journal of Research on Technology in Education, DOI:10.1080/15391523.2018.1433565

Salehi, S., Wang, K. D., Toorawa, R., & Wieman, C. (2020, February). Can majoring in computer science improve general problem-solving skills? In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (pp. 156-161).

Sentance, S., & Csizmadia, A. (2017). Professional recognition matters: Certification for in-service computer science teachers.

Stake, R.E. (2006) Multiple case study analysis. The Guilford Press, New York.

Vygotsky, L. S. (1978). Mind in Society: The Development of Higher Psychological Processes. Harvard University Press.

Weintrop, D., Beheshti, E., Horn, M. *et al.* Defining Computational Thinking for Mathematics and Science Classrooms. *J Sci Educ Technol* 25, 127–147 (2016). https://doi.org/10.1007/s10956-015-9581-5

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, 28(4), 371–400. https://doi.org/10.1080/08993408.2018.1560550

Yin, R. (2014). Case Study Research: Design and Methods (5th ed.). Thousand Oaks, CA: Sage Publications, Inc.

Chapter 5 - The Little Learners That Could (And Did!): Getting On the Track to Computer Science Success in K-6 by Improving Teacher Self-Efficacy through Targeted Professional Development

ABSTRACT

In a society fueled by technological advancements, the imperative for computer science education for all K-12 students has become undeniable. Yet, identifying teachers with the confidence and competence to deliver this content has emerged as a formidable challenge. This article highlights the critical need for computer science education in elementary schools, while simultaneously delving into the pivotal role of teacher self-efficacy, which is a person's belief in their ability to accomplish a goal or task. It navigates the terrain of preparing educators, especially those without formal backgrounds in computer science, to effectively impart computer science concepts by examining the intersection of pedagogical and content knowledge training and the cultivation of self-efficacy beliefs as examined through the lens of Bandura's theory of self-efficacy.

This research offers insights into empowering teachers to navigate the digital landscape with confidence and competence, first as a learner, then as a teacher, thus situating them to be able to equip the next generation with essential skills for success in an increasingly tech-driven world.

Introduction - Statement of Problem

In the world of K-12 education, the integration of computer science into the curriculum has long stood as a formidable task, much like the journey depicted in Watty Piper's tale (1930), "The Little Engine That Could." In this beloved children's classic, a small blue engine bravely agrees to pull a long train over a mountain after larger engines refuse, motivated by her mantra, "I think I can, I think I can." Through perseverance and a positive attitude, the little engine succeeds, teaching the value of optimism and determination. Today's teachers have to be the small blue engines who believe that they can be the ones who will finally succeed in getting computer science integrated into elementary classrooms, even if they are not completely confident in their own abilities to teach the skills. One of the persistent challenges in this endeavor has been the shortage of adequately trained computer science teachers, which has led many schools to rely on educators with backgrounds in other disciplines to teach computer science. While these non-computer science trained teachers bring valuable skills and perspectives to the classroom, their ability to learn and then teach computer science has remained a topic of concern; and if they cannot learn it, they cannot teach it. That confidence in one's ability to achieve specific tasks or goals is often referred to as self-efficacy, and in this case, the task would be to learn the subject themselves, and then be able to turn around and successfully teach it to their students. This research project followed a group of elementary educators on a year-long journey of self-discovery, exploring the factors that propelled them forward and the barriers they encountered along the way in both learning and teaching elementary computer science.

As the little engine's journey serves as an allegory for perseverance and self-belief, so too does Albert Bandura's (1977) Theory of Self-Efficacy illuminate the concept of personal motivation in the realm of education. Drawing on this theory as a theoretical framework, this project explored the extent to which non-computer science-trained teachers perceived themselves capable of acquiring computer science knowledge and skills, as well as the factors that contributed to or inhibited their self-efficacy in this domain. I gathered a diverse sample of non-computer science-trained for a year-long professional development teachers to provide insights into their experiences, challenges, and strategies for learning and teaching computer science.

By examining their self-efficacy through the lens of Bandura's theory, this article will provide insights that can inform the development of tailored interventions and supports. Ultimately, much like the little engine's triumphant cry of "I thought I could, I thought I could," the goal is to empower non-computer science-trained teachers with the confidence and tools they need to navigate the challenging terrain of computer science education, ensuring that every child has the opportunity to embark on their own journey of discovery in the world of computer science.

Research Background

Elementary school is a time when children are naturally curious and open to learning new things, making it an ideal time to introduce computer science concepts, to prepare students for the demands of an increasingly technology-driven world (Engel, 2011). But teachers cannot teach what they do not know and equipping them with the needed content, pedagogical, and technological knowledge is the critical first step to this becoming a reality. Previous research has highlighted the importance of teacher self-efficacy in various educational contexts, demonstrating its significant impact on instructional practices, student engagement, and academic achievement (Bandura, 1997; Tschannen-Moran & Woolfolk, 2001). However, limited attention has been directed towards examining the self-efficacy of non-computer science-trained teachers specifically in the context of learning computer science. Existing studies have primarily focused on the self-efficacy of computer science teachers with formal training or have explored broader aspects of computer science education, neglecting the unique challenges and opportunities faced by non-computer science-trained educators transitioning into the field of computer science education (Margulieux et al., 2016; Tondeur et al, 2012). This indicates a pressing need for empirical research that investigates the factors influencing non-computer science-trained teachers' self-efficacy in learning computer science, as well as the implications of their self-efficacy beliefs for classroom practice and student learning outcomes.

The need for computer science education in elementary schools

Computer science education in elementary schools is increasingly recognized as crucial in preparing students for success in the digital age. Introducing those specialized concepts at an early age helps develop fundamental skills and knowledge that are essential in today's technology-driven society. As technology continues to play a significant role in various aspects of daily life, including communication, problem-solving, and innovation, equipping students with these skills becomes imperative (Education Corner, n.d.). Moreover, integrating computer science education into elementary schools has been shown to foster creativity, critical thinking, and collaboration among students. By engaging in activities such as coding, robotics, and algorithmic problem-solving, students learn to approach challenges systematically and develop resilience in the face of obstacles (School of Education Online). Research by Resnick et al. (2009) suggests that hands-on experiences with computer science concepts not only enhanced the students' problem-solving skills but also promoted their confidence and motivation to learn. These types of experiences lay the foundation for lifelong learning and empower students to become active creators and innovators in an increasingly digital world.

Finally, equitable access to computer science education in elementary schools is essential for addressing disparities in STEM (science, technology, engineering, and mathematics) participation and achievement, and it is also critical for innovation and economic growth (AAUW, 2015). It requires a concerted effort to challenge stereotypes, but in doing so, schools can help bridge the digital divide and promote diversity and inclusion in STEM . As emphasized by the National Science Foundation (2017), early exposure to CS education can inspire students from underrepresented groups to pursue further studies and careers in STEM, ultimately contributing to a more diverse and innovative workforce.

Challenges faced by non-computer science-trained teachers

Teachers require adequate content knowledge and ongoing professional development to effectively integrate new skills into their teaching practices. Niess et al. (2021) emphasized the importance of tailored professional development plans to address the specific needs of educators, particularly those without a background in computer science. Such initiatives play a crucial role in enhancing teachers' content knowledge, instructional practices, and confidence in delivering computer science education within core content areas. However, existing professional development programs often fall short in providing sustained support and follow-up opportunities for teachers. Darling-Hammond et al. (2017) and Qian et al. (2018) observed that these programs typically consist of short-duration workshops, often lacking ongoing engagement during the school year. Moreover, one-time professional development offerings frequently fail to provide consistent follow-up opportunities necessary for teachers to fully internalize new instructional techniques (Menekse, 2015; Franke et al., 2013; Yadav et al., 2018).

Effective instruction goes beyond content mastery and fosters critical thinking, problem-solving, and the ability to apply knowledge in real-world contexts. It involves creating an engaging learning environment that encourages students to explore, question, and connect concepts to their experiences, thereby promoting deeper understanding and lifelong learning skills (Bransford, Brown, & Cocking, 2000; Darling-Hammond et al., 2008). This foundational knowledge is essential for creating engaging and impactful learning experiences. Additionally, teachers benefit from robust support systems, such as professional learning communities (PLCs), which provide ongoing opportunities for collaboration, reflection, and growth. Establishing professional connections among teachers further facilitates the exchange of best practices and enhances the overall quality of instruction (DuFour, 2004). In summary, the synergy of well-prepared content knowledge, strong support systems, and professional connections empowers teachers to develop a strong sense of self-efficacy, which in turn helps them deliver high-quality instruction, ultimately enriching the learning experiences of their students.

Utilizing Bandura's Theory of Self-Efficacy

Bandura's theory of self-efficacy offers a valuable framework for evaluating teachers and their impact on student learning. Self-efficacy refers to individuals' beliefs in their capabilities to organize and execute actions necessary to attain desired goals (1986). In the context of education, teachers' self-efficacy beliefs significantly influence their instructional practices, student motivation, and overall classroom climate. By assessing teachers' self-efficacy beliefs, educational institutions can gain insights into their effectiveness as educators and identify areas for professional development.

Research has consistently shown that teachers with high levels of self-efficacy are more likely to adopt innovative instructional strategies, persist in the face of challenges, and create supportive learning environments. For example, a study by Tschannen-Moran and Woolfolk Hoy (2001) found that teachers' self-efficacy beliefs positively correlated with student achievement outcomes. Teachers succeed when their students succeed. By evaluating teachers' self-efficacy beliefs, valuable information can be obtained that contributes to effective teaching practices and provides targeted support to enhance teachers' confidence and effectiveness in the classroom. Additionally, Bandura's theory (1977) emphasizes the importance of vicarious experiences in shaping self-efficacy beliefs. Teachers serve as role models for their students, and their behavior and attitudes can influence students' beliefs about their own capabilities. By assessing teachers' modeling behaviors and their ability to foster a growth mindset in students, administrators can gauge their effectiveness in promoting positive self-efficacy beliefs among students. This approach not only enhances student motivation and academic performance but also contributes to the overall quality of education.

Bandura's theory of self-efficacy provides a valuable lens through which to evaluate teachers and their impact on student learning. By considering teachers' self-efficacy beliefs and their influence on instructional practices and student outcomes, schools can make informed decisions about teacher professional development and support. Integrating principles from Bandura's theory into teacher evaluations also fosters a more comprehensive understanding of teaching effectiveness and contributes to the continuous improvement of educational practices.

His theory highlights four major constructs - mastery experience, vicarious experience, social persuasion, and emotional state. The fishbone diagram, Figure 5.1, provides a visual explanation of Bandura's Theory of Self-Efficacy.

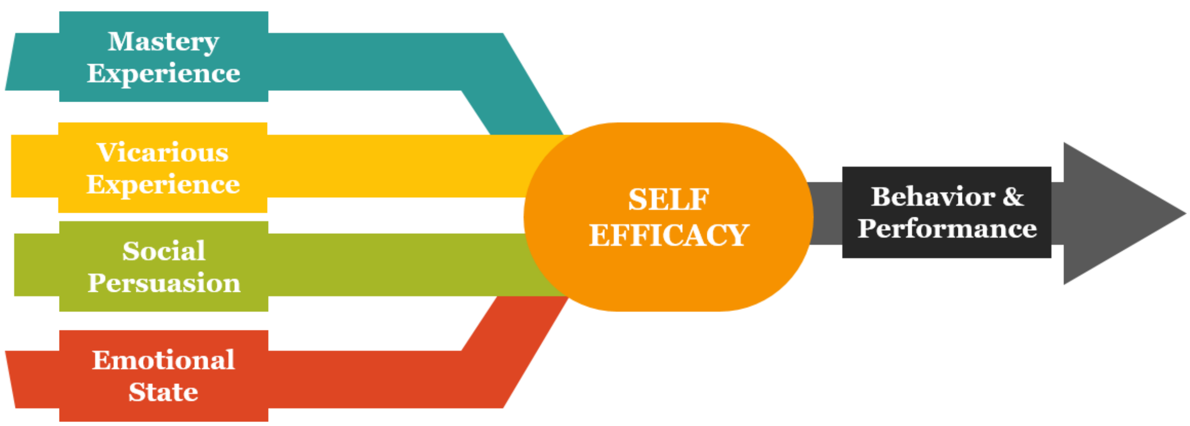


Figure 5.1. Visual Representation of Bandura's Theory of Self-Efficacy (1977)
Note. Source: Data from "Self Efficacy: Toward a Unifying Theory of Behavioral Change" (Bandura, 1977).

The first construct, ***mastery experiences***, emphasizes the importance of personal accomplishments and successes in building self-efficacy. When learners experience success in mastering tasks or skills, they develop confidence in their abilities to tackle similar challenges in the future. These successes serve as a powerful source of self-efficacy, shaping their beliefs about their own competence and fostering a growth mindset (Bandura, 1994).

The second construct, ***vicarious experiences***, highlights the role of observation and modeling in shaping self-efficacy beliefs. Learners gain confidence not only through their own experiences but also by observing others similar to themselves successfully perform tasks. In this instance, role models demonstrate skills and strategies, showing what can be achieved. Through vicarious experiences, learners can envision themselves mastering similar tasks, thereby strengthening their self-efficacy beliefs (Bandura, 1986).

The third construct is ***social persuasion***, and it focuses on receiving support from others. This support can include verbal and nonverbal encouragement, feedback, and motivation from others that can have a positive effect on an individuals' self-efficacy beliefs. Positive reinforcement and encouragement can bolster a learner’s confidence in their capabilities, even in the face of challenges or setbacks, which may result in them feeling more empowered to persist and overcome obstacles, thus strengthening their self-efficacy beliefs (Pajares, 2002).

The fourth construct, the ***emotional state***, focuses on the influence of an individuals' emotional well-being and physical state on their self-efficacy beliefs and performance. Anxiety, stress, fatigue, and other negative emotions can undermine self-efficacy, leading to reduced motivation and performance. Conversely, positive emotional states such as enthusiasm, calmness, and confidence can enhance self-efficacy and facilitate optimal performance. Addressing learners’ physiological and affective states can promote a positive mindset and cultivate resilient self-efficacy beliefs conducive to academic success (Bandura, 1997).

Research Design

Research design is the structured plan that guides the methods and procedures of a research study, ensuring that the research questions are effectively addressed. It is important because it provides a clear framework for collecting, analyzing, and interpreting data, which enhances the validity and reliability of the study. Research design also helps in identifying the appropriate tools and techniques for data collection, making the research process more efficient and systematic. Additionally, a robust research design anticipates potential challenges and includes strategies to mitigate them, thereby increasing the overall quality and feasibility of the study.

Why Qualitative Research

Qualitative research is a method used to understand things in their natural settings by focusing on people's experiences and perspectives. Instead of numbers, this approach collects data like words, images, or objects through interviews, focus groups, and observations. It's especially useful for exploring complex issues where understanding the context and details is important, helping researchers capture the richness and variety of human experiences that can't be easily measured with numbers.

One of the main advantages of qualitative research is its ability to provide a deep and detailed understanding of a situation from the viewpoint of those involved. Creswell and Poth (2018) note that it allows researchers to uncover patterns and themes in social phenomena that quantitative methods might miss. This is crucial for studying areas where different factors are closely linked and hard to separate, like in cultural studies, education, and healthcare. Additionally, qualitative research is key in developing new theories and ideas. By offering detailed descriptions and challenging existing beliefs, it helps build a thorough understanding of social processes. Denzin and Lincoln (2018) argue that this method is essential for creating a well-rounded understanding that often leads to further quantitative research and testing. Overall, qualitative research is vital for advancing knowledge by providing rich, context-based insights into complex issues.

Why employ a case study methodology

The utilization of case study methodology presents a compelling avenue for investigating elementary computer science education, offering profound insights into multifaceted educational phenomena within authentic settings. As articulated by Yin (2014), case studies afford researchers the opportunity to delve into the nuances of computer science instruction in elementary contexts, spanning curriculum design, teacher methodologies, and student learning encounters. Case studies are valuable because they allow researchers to gather detailed information from various sources, like interviews, documents, and observations. Yin also adds that, "The distinctive need for case studies arises out of the desire to understand complex social phenomena" (2018). In other words, case studies help researchers capture the full picture of a situation, including all the different factors that influence it.

How this research project was organized

This research was centered around 18 teachers who were interested in learning more about computer science - both for themselves and for their students. They agreed to participate in a year-long program of study that also included pre-and-post surveys, interviews, and personal and peer-to-peer reflections. The guiding questions being addressed in this research were:

1. How does the perceived self-efficacy of non-trained computer science teachers influence their ability to effectively learn new concepts and teaching methodologies?

2. To what extent does the self-efficacy of non-trained computer science teachers impact their confidence in teaching computer science topics, such as programming languages and algorithmic thinking, to students at various educational levels?
3. How do interventions aimed at enhancing the self-efficacy of non-trained computer science teachers, such as professional development workshops and mentoring programs, affect their motivation, competence, and performance in teaching computer science?

The training unfolded across three distinct phases, each tailored to specific learning objectives and reflective practices. Termed as "***crawl-walk-run***," this instructional approach aligns with developmental theories, progressively introducing complex concepts in stages akin to the developmental milestones of infancy (Mayer, 2019; Vygotsky, 1978). During the ***crawl stage*** (January - April), emphasis was placed on teachers familiarizing themselves with the tools through hands-on tasks. They engaged in exploration at their own pace, documenting their progress and reflections on personal blogs. This phase aimed to establish a robust foundational understanding before transitioning to instructional delivery. In the ***walk stage*** (May - August), teachers advanced to designing and delivering technology training sessions, integrating newly acquired skills into instructional contexts. This phase included opportunities for solo work, peer feedback exchange, and professional development attendance, fostering both individual growth and collaborative learning. The ***run stage*** marked the culmination of the training, challenging participants to redesign existing curriculum units to incorporate computer science tools comprehensively. Through iterative reflection and peer engagement, teachers refined their instructional strategies, ensuring alignment with learning objectives and promoting student-centered learning experiences.

Location. This study was conducted within a mid-sized, suburban Texas school district. As of November 2023, the district included 21 elementary schools, with 18 designated as Title I campuses, indicating that at least 40% of students qualify for free or reduced lunch. The district serves approximately 11,000 K-6 students, with the following racial/ethnic breakdown:

Table 5.1 District Race/Ethnicity Statistics (October 2023)

Note: Data obtained from Texas Education Agency, ©2023.

Race/Ethnicity	No. of students in K-6	Overall % of K-6 students
Latino/Hispanic	3,630	33%
White	3,300	30%
Black	2,240	22%
Asian	880	8%
Pacific Islander	330	3%
Two or more races	440	4%

Participants. Participants in this research were drawn from the broader computer science initiative of the school district, in which the author serves as a district STEM administrator. Recruitment occurred through the distribution of a flyer to all K-8 teachers involved in the district initiative in October 2022, followed by two Zoom Q&A meetings in November and December 2022. These sessions provided information on the project's objectives, timelines, and participant expectations. When recruitment was completed, there was one male teacher and 17 female teachers participating; their years of experience ranged from 0 years to 20+ years and their subjects taught ranged from core curricular areas to art to instructional coaching.

Additionally, at our first meeting, I asked each of them to share a one-word superpower or strength that they would bring to the group and I incorporated that into a pseudonym for each one, as they preferred their comments to be anonymous. Table 5.2 lists the participants by their superpower pseudonym (to preserve anonymity), what subject(s) each of the participants currently teach and the total number of years they have been teaching.

For complete transparency, it should be noted that the teachers selected to participate in this research project were recruited from the broader district initiative that was operating concurrently. Those participating in the broader computer science initiative were compensated for their work, but the teachers involved in this research project were not compensated for their time in this study.

GRADE(S)	SUBJECT(S)	YEARS OF EXPERIENCE
6th grade	Mathematics	1 – 5
1st grade	All subjects	6-10
8th grade	Science	6-10
4th grade	Mathematics, Science	6-10
5th grade	Mathematics	1-5
6th grade	Science, Social Studies	6-10
PreK – 6th grade	Art	11-15
PreK	All subjects	6-10
6th grade	Mathematics	6-10
4th grade	English/Language Arts	11-15
PreK – 6th grade	Instructional Coach	16-20
3rd grade	English/Language Arts	1-5
4th grade	English/Language Arts	1-5
2nd grade	All subjects	6-10
5th grade	Mathematics	6-10
6th grade	Science, Social Studies	1-5
6th grade	Science, Social Studies	1-5
PreK – 6th grade	Reading Specialist	16-20

Table 5.2. CS Cohort Participant Data

Data Collection & Analysis

Two primary types of data were collected during this project - surveys and interviews. The teachers were also instructed to set up and maintain an online blog where they could continually reflect on their own learning and others could comment, offer suggestions/advice, etc.

The surveys were administered at the beginning of the project and at the conclusion. The questions used in the survey were taken from the *Teacher Efficacy and Attitudes Towards STEM (T-STEM) Survey*, and it is intended to "measure changes in teachers' confidence and self-efficacy in STEM subject content and teaching, use of technology in the classroom, 21st century learning skills, leadership attitudes, and STEM career awareness" (Friday Institute, 2012). The initial survey results aided in the design of the first round of professional development for the program as it showed clear trends in teacher's perceptions of their own abilities and knowledge in regards to computer science.

Two in-person interviews were conducted with each participant - one at the six-month mark and one at the end of the research. These questions were designed to gauge the effectiveness of the training as well as assess teacher's comfort levels with the materials introduced.

After the interviews were conducted, the information was analyzed using an *a priori coding* process. This process involved beginning with a set of predefined codes, in our case, the constructs of Bandura's Theory of Self-Efficacy, and then identifying patterns, themes and relationships within the data. It is an iterative process and involved constant comparison and refinement as nuances embedded within the dataset were manifest. A story emerged from the data as these relationships were discovered.

The *a priori coding* process also provided me with a systematic way to organize and make sense of my findings. Having the predefined set of constructs (mastery experiences, vicarious experiences, social persuasion, emotional state) from Bandura's theory (1977) helped provide a focused framework to hone in on recurring themes and data points. It also ensured consistency throughout the analysis thus adding a solid layer of transparency to the qualitative research.

Results

Creating an alignment between the *T-STEM Survey* and the four constructs from Bandura's Theory of Self-Efficacy provided me with a research-based framework to begin the initial assessment of each teacher's perception of their own abilities. It also helped identify gaps in their knowledge that needed to be addressed in the initial design of the program.

Mastery Experiences (Past personal successes)

In the realm of teaching elementary computer science, the application of Bandura's theory of self-efficacy is particularly relevant for educators who are learning to teach this subject. Mastery experiences, reflecting past personal successes in mastering specific skills or tasks, play a pivotal role in shaping teachers' self-efficacy beliefs. As teachers engage in professional development activities, coding workshops, and collaborative curriculum planning, successful experiences in understanding and applying computer science concepts contribute significantly to their confidence in teaching the subject effectively (Bandura, 1997). Bandura's theory also posits that repeated mastery experiences that lead to an increased sense of self-efficacy, has a positive influence on teachers' willingness to persist in their own struggles and adaptations of their instructional strategies. It also builds the foundation of confidence in their own ability to convey complex concepts to young learners and it is essential for fostering a growth mindset in their approach to learning and teaching computer science, ultimately enhancing the overall quality of instruction (Tschannen-Moran & Hoy, 2001).

In the first interview, when questioned about past personal successes with computer science, the teachers reflected on previous positive experiences with leading student coding and robots clubs, as well as learning new skills in professional development training. Positive Paula raved about the fantastic after-school student club attendance that climbed each week as more and more students wanted to join and said that it "excited me and motivated me to continue my own learning!". Curious Carla remarked how inspiring it was that all students were afforded the opportunity to learn computer science when she stated "Seeing all students getting to work with the robots – not just the advanced academic students or those who finish their work first – is exciting and shows a strong commitment to equity and closing the gap for all of our learners, which makes me want to keep learning too."

When presented with the same question at the end of the professional development, the consensus was that the teachers really appreciated the time they had to 'play' with the devices and become more comfortable with them personally before having to tie them to their subject areas and introduce them to students. Generous Gina proudly stated: "Now that I have more confidence in myself, I am constantly looking for better resources and higher-quality materials that I can infuse into my lessons." That little engine was definitely on the right track!

Vicarious Experiences (Watching others succeed)

Vicarious experiences involve observing others succeed in a particular task or domain. In the context of teachers learning to instruct elementary computer science, vicarious experiences become vital as educators often look to colleagues, mentors, or expert instructors for models of effective teaching practices. When teachers see their peers successfully navigate coding challenges, implement engaging instructional strategies, or effectively manage technology-enhanced classrooms, it can significantly impact their own self-efficacy beliefs. The observation of others' achievements in teaching computer science serves as a powerful source of inspiration and confidence-building (Bandura, 1997) that can have substantial influence on a teacher's self-efficacy and instructional practices. Creating collaborative learning environments, where teachers can share successes and challenges, can provide rich opportunities for vicarious experiences, fostering a supportive professional learning community that benefits all involved (Tschannen-Moran & Hoy, 2001).

Several teachers in this cohort were at the same schools and more than one mentioned that being able to work with other cohort members, as well as interested teammates, to figure things out, gave the educators a built-in PLC and cheering section. In her first interview, Excited Ella remarked that she loved having another teacher on her campus involved in the cohort. She explained: “We are able to bounce ideas off of each other, show each other what we know, and celebrate together when we are able to figure something out!”. Present Patty commented that seeing students work together, take chances, and find success/failure gave hope to the “old folks”, as she called herself and her teammates! Those same feelings of working together and providing support also came through in their second interviews but focused on the success of the students. Motivated Molly noted how excited students were to share new knowledge with each other and with her and Empathetic Emily discussed moving towards a more facilitator-type role in the classroom as the gradual release of responsibility was employed in her classroom and she became more of a guide-on-the-side and less of a sage-on-the-stage type teacher. Her self-awareness of shifting to the facilitator role was echoed by several others in their interviews as well as Generous Gina remarked “I just needed to move out of their way and let them run with it!”. Emphasizing the guide-on-the-side approach proved to be a great way to foster active learning and critical thinking, and encourage students to engage with the material and develop their own problem-solving skills.

My researcher take-away from this construct was that I believe that being able to create that collaborative learning environment and safe space to fail, was an integral part in setting both student and teacher up for success.

Verbal Persuasion (Positive verbal feedback)

Verbal persuasion, the third element in Bandura's theory of self-efficacy, involves receiving positive feedback and encouragement from others. For teachers learning to instruct elementary computer science, positive verbal feedback becomes instrumental in shaping their confidence and belief in their teaching abilities. Supportive comments from mentors, colleagues, or professional development facilitators can serve as a potent source of motivation, fostering a sense of efficacy in navigating the challenges of teaching computer science effectively (Bandura, 1997). Positive reinforcement can come in various forms, such as acknowledgment of effective lesson planning or successful implementation of technology-enhanced activities or management of coding projects in the classroom. This type of encouragement not only boosts teachers' confidence but also reinforces their commitment to continuous improvement in teaching computer science (Tschannen-Moran & Hoy, 2001).

The power of verbal persuasion was affirmed by multiple participants in the first round of interviews, several of whom commented on the influence this had on their own self-efficacy. Engaged Essie noted how important it was that administrators could see the value of integrating CS/CT and encouraged attendance at relevant workshops, provided collaboration time, and let participants show off their learning at staff meetings and other key times. Empathetic Emily shared that she had to remind herself that she is a learner too and that it was “ok to struggle and even fail when I am trying to figure out these devices”. She also stated that “I feel like I am speaking to myself like I would to my students.” When trusted sources offer genuine encouragement, it can significantly enhance one's confidence and motivation to tackle challenges. Verbal persuasion holds the power to transform perspectives and inspire action by deftly appealing to emotions and logic. Used wisely, it can bridge divides, inspire movement and effect profound change.

Revisiting this question in the second interview highlighted a concept that emerged as a trend in the data - the positive verbal feedback for the teachers started coming from their students! Roles had reversed in these teachers' classrooms as the students took on the role of cheerleader to encourage their teachers to continue to learn and to teach them. Passionate Pam remarked how wonderful it was to have students ask about what she was learning and how she felt it was doing with her assignments. Consistent Cathy shared that she received an email from a campus administrator who had done a walkthrough in her classroom on a day she was introducing a computer science-based concept. The administrator remarked that it "warmed his heart to see the students asking leading questions or offering targeted suggestions" to keep the lesson moving. Parents in one class also chimed in, with Energetic Emma stating in her interview that "parents have even called/emailed to let me know the positive changes they are seeing in their children towards school, with them even going so far as to share with them how they 'helped' the teacher in class that day!" Clearly, word was getting out that something special was happening in the classrooms of the cohort teachers.

Emotional State (Personal well-being & awareness)

The last element of Bandura's theory of self-efficacy involves the emotional state, or personal well-being and awareness of the participants. The complex nature of teaching computer science, especially in the rapidly evolving landscape of educational technology, can be challenging, so being able to navigate the uncertainties associated with learning and implementing new pedagogical approaches and technologies effectively requires emotional resilience and grit (Bandura, 1997). Cultivating emotional intelligence in teachers enhances their ability to adapt to changing circumstances, promote a positive classroom environment, and foster a growth mindset among students (Brackett & Katulak, 2006) and allows them to have positive emotional experiences, such as enthusiasm, joy, and a sense of accomplishment, that contribute to increased self-efficacy beliefs and motivation to persevere through challenges.

In their first interviews, participants were quick to highlight the need for a growth mindset, greater empathy, and professional responsibility. Learning Lisa acknowledged that struggles are real for all learners and that letting her students see her struggle to learn helped them see her in a different light – not as the know-it-all at the front of the class, but as a learner who struggled like they did at times. She and others expressed that their own levels of empathy were increased, which led to decreased stress levels in the classroom for all and promoted a sense of helpfulness and optimism. Interested Irma summed up their feelings with this powerful observation: "I felt myself in my student's shoes…it reminded me how hard learning something new can be, but it also gave me confidence once I learned how to do it." Acknowledging the struggle and then pushing through it perfectly typified the little blue engine's mantra of "I think I can, I think can" (1930) as the teachers continued to move forward. How grateful I am as a researcher that these teachers agreed to participate, of course, but also were so thoughtful in their feedback and reflections.

Looking back over the whole program in the second interview, two themes emerged. The first prevailing message from the participants was that establishing a positive classroom environment that promoted a growth mindset and allowed for mistakes, by both students and teachers, was the critical first step that could not be ignored. Twelve of them, independent of each other, acknowledged that their classrooms seemed to run much more smoothly and that stress levels had decreased for both students and teachers. A number of them also indicated that their colleagues who were not part of the project had noticed some of these positive changes and wanted to learn more.

The second theme centered on their responsibility to meet the students where they were. Newfangled Nancy spoke of the cohort's professional collective determination to lead the way for their students, even if it meant they would struggle. She remarked "I recognize that this is just the tip of the iceberg when it comes to technology – so many new things are being developed every day like AI and Chat GPT. I will be doing my students a disservice if I can't share at least a passing reference to get them to hopefully want to learn more." And, like the little blue engine that could, Excited Ella positively proclaimed "I CAN do this! I actually have been doing it, I just did not realize it until I learned more about it…and I am good at it!" Just as the determined little engine faced daunting challenges on its ascent up the mountain, non-trained computer science classroom teachers have been confronted with their own uphill climb of incorporating computer science education into classrooms worldwide. As these educators reach the top of their mountains, their pride is evident as is the excitement of what integrating computer science into their curriculum can look like.

Conclusion

As “I think I can” (Piper, 1930) became “I thought I could”, the project concluded and the teachers reached the peaks of their own professional development mountains,, a host of insights into teacher self-efficacy, targeted professional development and computer science integration at the elementary level were illuminated.

Self-Efficacy and Teacher Learning

As a researcher, I found the program's overall design to be more than acceptable for addressing both the existing knowledge base of participating teachers and identifying areas for improvement and new learning, as it aligned with the four constructs of Bandura’s theory of self-efficacy. It offered a combination of hands-on experiences, collaborative learning, and real-world applications, with the goal of fostering a deeper understanding of computer science concepts and refining instructional approaches. Flexibility and customization (emotional state) were key features, catering to diverse levels of expertise of those involved. By allowing educators to progress at their own pace and focus on specific interests (mastery experiences), the program promoted self-engagement and personal skill development. Having them continually reflect on their own practice as well as the experiences of others (vicarious experiences), also helped build a sense of community and support.

Additionally, the inclusion of authentic, project-based learning experiences encouraged creativity and problem-solving. Interdisciplinary connections were also emphasized (social persuasion), enabling the teachers to integrate computer science into their various subjects. Through this approach, teachers gained confidence and proficiency in facilitating computer science education, ultimately benefiting their students' learning experiences (as well as their own!).

Self-Efficacy in Teaching Computer Science

As evidenced by their unfiltered responses in interviews and blog reflections, these teachers had to accept that there were things that they did not know that their students already had knowledge of, such as the one teacher handing her micro:bit over to her students. But, they also had to believe that they were capable of learning it as well. So, increased self-efficacy in teaching computer science served as a cornerstone for educators, fostering a robust sense of confidence and competence in navigating new and unfamiliar learning.

From the data, as teachers developed a deeper understanding of the concepts through personal exploration, their efficacy in delivering instruction was amplified. Moreover, as they witnessed the tangible impact of their teaching, their belief in their own ability to facilitate meaningful computer science education was enhanced. This led to a positive feedback loop of continued teachers' personal learning of computer science concepts as well as refined teaching practices which bolstered confidence and set the expectation for students to do likewise.

Importance of targeted professional development systems

This professional development program placed a strong emphasis on nurturing teacher self-efficacy in computer science education. Through personalized support, ongoing feedback, and opportunities for reflection, teachers were encouraged to build confidence in their own ability to effectively learn and then teach computer science concepts. By providing a supportive environment where they could take ownership of their learning and development, the program aimed to enhance teacher self-efficacy, ultimately leading to more empowered and effective instructors in the classroom. As teachers gained confidence in their own skills and abilities, they were better equipped to inspire and motivate each other as well as their students

, fostering a culture of curiosity and achievement in computer science education. One teacher even noted that participating in the cohort was the “coolest thing I’ve ever done as a teacher!”, which led me to believe that this training and subsequent learning and teaching opportunities truly did have a positive impact on the participants! High teacher engagement is crucial when learning is meaningful, as it directly impacts student motivation and achievement. Research indicates that teachers who find their work meaningful and are deeply engaged create a positive learning environment that fosters student engagement and success. According to the EdCan Network, teacher engagement is driven by factors such as autonomy, mastery, and purpose, which align with what motivates most people. When teachers feel they have control over their work, can perform it well, and see it as contributing to a larger purpose, they are more engaged and effective in the classroom (Bearisto, 2012).

By integrating key components like professional learning communities (PLCs), collaborative learning opportunities, mentorships, and individual voice/choice, this program established a solid foundation for participant fidelity. Teaching computer science to educators necessitates meticulous planning and supportive structures for successful integration into educational settings. Through thorough needs assessments and comprehensive training guidelines, the program tailored its content and delivery methods to suit the diverse needs of educators. Having the opportunities to reflect and provide feedback also allowed for an on-going assessment of the program’s effectiveness and gave space for the resolution of any challenges or barriers that developed.

Fostering a culture of experimentation, adaptation, and professional collaboration, the program laid the groundwork for a potentially sustainable ecosystem. Here, teachers were empowered to develop their own computer science skills and effectively integrate computer science education into their classrooms, thus equipping themselves and their students with indispensable 21st-century competencies.

Limitations and Future Research

One limitation of my research was that less than 100% of participants completed the program. The non-finishers did complete and turn in all of their assigned work, but did not participate in the second face-to-face interview. Although I did not thoroughly explore the reasons for non-completion, I do not believe that the missing interviews affected the overall validity of the findings. Despite these limitations, the insights I gathered from those who did complete the program still offered valuable information for refining and enhancing the training. The primary aim of the research was to equip district teachers from non-computer science trained backgrounds to teach elementary-level computer science, and success was evidenced by the enthusiasm of the over 80% who did complete the program to continue documenting their learning experiences, indicating increased self-efficacy in their learning and teaching abilities.

While recruitment from a group that received financial compensation might have seemed like trying to stack the deck to ensure success, it resulted in a cohort that didn't fully mirror the diversity of the district's student population. Making a more targeted approach to recruit teachers who look more like the students should be considered when deploying this plan. This approach also raised concerns about participants' motivations and potential bias towards future opportunities Future projects should address these issues by refining recruitment strategies to ensure a more representative participant pool and by implementing support mechanisms to mitigate self-selection bias among teachers, and seek to develop a broader sense of self-efficacy within the teaching community.

Acknowledgments

First, I would like to express my sincere gratitude to Dr. Suzanne Porath (Kansas State University) for her valuable contributions and unwavering support throughout the research process. Dr. Porath's expertise, guidance, and assistance have been instrumental in shaping this study and the ensuing scholarly articles.

Additionally, I extend my heartfelt appreciation to the participants who generously shared their time and insights, without whom this research would not have been possible, and for the assistance and encouragement of the cooperating school district.

Lastly, I acknowledge the broader academic community for their ongoing efforts in advancing knowledge in this field.

References

American Association of University Women. (2015). *Solving the equation: The variables for women's success in engineering and computing*. https://www.aauw.org/research/solving-the-equation/

Bandura, A. (1997). Self-efficacy: The exercise of control. New York: Freeman.

Bandura, A. (1986). Social foundations of thought and action: A social cognitive theory. Prentice-Hall, Inc.

Bandura, A. (1994). Self-efficacy. In V. S. Ramachaudran (Ed.), Encyclopedia of human behavior (Vol. 4, pp. 71-81). Academic Press.

Beairsto, B. (2012, March 3). Teacher engagement is the key to student engagement. EdCan Network. Retrieved from https://www.edcan.ca/articles/teacher-engagement-is-the-key-to-student-engagement/

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (2000). *How people learn: Brain, mind, experience, and school*. National Academy Press.

Brackett, M. A., & Katulak, N. A. (2007). Emotional Intelligence in the Classroom: Skill-Based Training for Teachers and Students. In J. Ciarrochi & J. D. Mayer (Eds.), Applying emotional intelligence: A practitioner's guide (pp. 1–27). Psychology Press.

Darling-Hammond, L., Austin, K., Cheung, M., & Martin, D. (2008). *Thinking about thinking: Metacognition*. Stanford University School of Education.

Darling-Hammond, L., Hyler, M. E., Gardner, M. (2017). Effective Teacher Professional Development. Palo Alto, CA: Learning Policy Institute. https://doi.org/10.54300/122.311.

Engel, S. (2011). *Children's need to know: Curiosity in schools*. Harvard Educational Review. https://elimindset.com/wp-content/uploads/2022/08/engel2011.pdf

Franke, B., Century, J., Lach, M., Wilson, C., Guzdial, M., Chapman, G., & Astrachan, O. (2013). Expanding access to K-12 computer science education: research on the landscape of computer science professional development. In Proceedings of the 44th ACM technical symposium on computer science education (pp. 541–542). Denver, CO: ACM.

Friday Institute for Educational Innovation (2012). Teacher Efficacy and Beliefs toward STEM Survey. Raleigh, NC.

Margulieux, L. E., Catrambone, R., & Guzdial, M. (2016). Computer Science Education: Where Are the Software Engineers of Tomorrow? ACM Transactions on Computing Education, 16(1), 1-37.

Mayer, R. E. (2019). How do students learn and what should teachers do? In R. E. Mayer (Ed.), The Cambridge Handbook of Multimedia Learning (2nd ed., pp. 3-32). Cambridge University Press.

Menekse, M. (2015). Computer science teacher professional development in the United States: A review of studies published between 2004 and 2014. Computer Science Education, 25(4), 325–350. https://doi.org/10.1080/08993408.2015.1111645

National Science Foundation. (2017). Broadening participation in computing. Retrieved from https://www.nsf.gov/cise/bpc/.

Niess, M. L. (2005). Preparing teachers to teach science and mathematics with technology: Developing a technology pedagogical content knowledge. Teaching and Teacher Education, 21(5), 509–523. https://doi.org/10.1016/j.tate.2005.03.006 Tschannen-Moran, M., & Woolfolk Hoy, A. (2001). Teacher efficacy: Capturing an elusive construct. Teaching and Teacher Education, 17(7), 783-805.

Pajares, F. (2002). Overview of social cognitive theory and of self-efficacy. Retrieved from http://www.uky.edu/~eushe2/Pajares/eff.html

Qian, Y., Hambrusch, S., Yadav, A., Gretter, S. (2018). Who Needs What: Recommendations for Designing Effective Online Professional Development for Computer Science Teachers, Journal of Research on Technology in Education, DOI:10.1080/15391523.2018.1433565

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., & Kafai, Y. (2009). Scratch: programming for all. Communications of the ACM, 52(11), 60-67.

Tondeur, J., Van Braak, J., Sang, G., Voogt, J., Fisser, P., & Ottenbreit-Leftwich, A. (2012). Barriers to Teaching Computer Science: A Study of High School Teachers' Perspectives. Educational Technology & Society, 15(2), 69-78.

Vygotsky, L. S. (1978). Mind in Society: The Development of Higher Psychological Processes. Harvard University Press.

Weintrop, D., Beheshti, E., Horn, M. et al. Defining Computational Thinking for Mathematics and Science Classrooms. J Sci Educ Technol 25, 127–147 (2016). https://doi.org/10.1007/s10956-015-9581-5

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. Computer Science Education, 28(4), 371–400. https://doi.org/10.1080/08993408.2018.1560550

Yin, R. (2014). Case Study Research: Design and Methods (5th ed.). Thousand Oaks, CA: Sage Publications

Chapter 6 - One Size Does Not Fit All: Fusing Linear & Non-Linear Learning Approaches for a K-6 Computer Science Professional Development Initiative

Abstract

This article proposes a professional development training plan designed to empower non-computer science (CS) teachers to effectively integrate CS concepts into their curriculum in K-6. Drawing upon the strengths of both the Crawl-Walk-Run-Fly (CWRF) Model of professional development and Jerome Bruner's Spiral Curriculum Approach, the training plan offers a structured and scaffolded framework to guide teachers through the process of learning CS. The crawl-walk-run-fly model provides a progressive pathway, starting from foundational basics and advancing to more complex topics, while Bruner's spiral approach emphasizes the iterative revisiting of key concepts to gain a deeper understanding of them. By integrating these pedagogical frameworks, an effective professional development plan can provide differentiated instruction, support diverse learning needs, and promote deep understanding and long-term retention of CS knowledge.

Through the collaborative professional development initiative, non-CS teachers can gain the necessary knowledge, skills, and instructional strategies to cultivate computational thinking and problem-solving skills in themselves, and gain the abilities and confidence to teach them to their students, preparing them for success in an increasingly digital and interconnected world.

Introduction to the Problem

As computer science (CS) continues to gain prominence in educational settings, there is a growing need for non-CS teachers to integrate computational thinking and coding skills into their curriculum. However, many educators face challenges in effectively teaching CS due to limited background knowledge and training in the subject. To address this gap, this article describes the development of a professional development training plan tailored specifically for non-CS teachers, drawing upon the strengths of the crawl-walk-run-fly model and Bruner's spiral curriculum approach (1966). By combining these two pedagogical frameworks, educators can provide comprehensive and scaffolded support to facilitate the learning and teaching of CS concepts in a manner that aligns with learners' developmental stages and promotes deep understanding.

The crawl-walk-run-fly model offers a structured progression that guides teachers and students through the process of learning CS concepts, starting from foundational basics and gradually advancing to more complex topics. This model acknowledges the diverse needs and starting points of learners, allowing educators to differentiate instruction and provide tailored support to meet individual learning goals. Integrating Bruner's spiral curriculum approach (1966) further enhances this framework by emphasizing the iterative revisiting of key concepts in a progressively more sophisticated manner to develop a deeper understanding. This iterative approach promotes deeper comprehension and long-term retention of CS knowledge by encouraging learners to build upon prior learning experiences and make connections between different concepts and applications.

By developing a professional development training plan that leverages the strengths of both the crawl-walk-run-fly model and Bruner's spiral approach, non-CS teachers can gain the necessary knowledge, skills, and instructional strategies to effectively teach CS concepts in their classrooms. This integrated approach fostered a supportive learning environment that promoted active engagement, risk taking, and a lifelong passion for learning new things. It also encouraged participants to view challenges as opportunities for learning and continuous improvement.

This article will demonstrate how fusing the two models to prepare non-CS trained educators yielded a positive return on a school district's investment in a year-long professional development plan. This systematic and empowering approach provides both content and pedagogical knowledge that will better prepare educators for the increasingly important role of teaching computer science in elementary schools.

Research Background

A comprehensive examination of scholarly research reveals a noticeable surge in computer science initiatives within K-12 education over recent years. Despite this momentum, the persistent challenge of adequately equipping in-service teachers with the requisite computer science knowledge and pedagogical skills remains evident. Each perspective examined in brief in this section sheds light on the pressing need addressed by this research endeavor. It includes a focus on the increasing need for computer science education in elementary, the lack of properly trained in-service teachers to teach the subject matter, and the importance of employing the crawl-walk-run-fly model as a research-based framework and blending it with Bruner's spiral approach to create a robust training plan grounded in research that promotes a growth mindset and encourages continual reflection and improvement.

The increasing need for computer science education in elementary schools

It takes little effort to recognize the pivotal role of computer science education in elementary schools and that it has surged in recent years. Teaching digital concepts to students at an early age is increasingly recognized as vital for their future success and has been shown to help children develop essential skills for navigating the modern world and prepares them for technology-driven careers. Engaging with STEM (Science, Technology, Engineering, Math) subjects from a young age has also shown to help develop critical thinking, curiosity, and problem-solving skills, which are crucial in today's digital age (School of Education Online). According to Resnick et al. (2009), hands-on experiences with computer science concepts enhance problem-solving skills while bolstering students' confidence and motivation to learn, laying the groundwork for lifelong learning and empowering them to become active creators and innovators, not just consumers of technology. This perspective aligns with research by Grover and Pea (2013), which underscores the importance of early exposure to computer science in fostering computational thinking skills essential for navigating a technology-driven world.

Ensuring equitable access to computer science education in elementary schools is crucial for addressing disparities in STEM participation and achievement. By affording all students, regardless of background or socioeconomic status, opportunities to engage in CS learning experiences, schools can help mitigate the digital divide and promote diversity and inclusion in STEM fields (Misha, 2023). As highlighted by the National Science Foundation (2017), early exposure to CS education inspires students from underrepresented groups to pursue further studies and careers in STEM, contributing to a more diverse and innovative workforce. This perspective echoes findings by Margolis and Estrella (2010), emphasizing the transformative potential of inclusive CS education initiatives in fostering equitable access and participation in STEM.

Finally, research suggests that early exposure to computer science education can have long-term effects on students' academic and professional trajectories. By instilling computational thinking skills and digital literacy from a young age, students are better equipped to navigate an increasingly technology-dependent society and pursue advanced studies and careers in STEM fields. Studies by Guzdial and Ericson (2015) highlight the positive impact of sustained engagement with computer science education throughout the K-12 years, correlating with increased interest and persistence in STEM disciplines at the collegiate level. Therefore, integrating computer science into elementary education not only fosters immediate benefits in terms of skill development and creativity but also lays the groundwork for future academic and career success in a rapidly evolving digital landscape.

Challenges of equipping non-CS trained teachers to effectively teach computer science

Teachers' effectiveness in integrating new skills into their teaching practices hinges on both adequate content knowledge and ongoing professional development to learn how to best impart that content knowledge to their students (Niess et al., 2021). Thus, tailored professional development plans are crucial, particularly for educators without a background in computer science, as they focus on enhancing content knowledge, instructional practices, and teacher self-confidence in delivering computer science education within core content areas (Niess et al., 2021). However, research has shown that existing professional development programs often lack sustained support and follow-up opportunities. Short-duration workshops, most often during summer months, dominate these programs, with limited ongoing engagement opportunities during the school year (Darling-Hammond et al., 2017; Qian et al., 2018). Additionally, one-time offerings frequently overlook consistent follow-up opportunities, hindering teachers' ability to fully internalize the new material that they have learned, which in turn may prevent them from successfully imparting that new knowledge to their students (Menekse, 2015; Franke et al., 2013; Yadav et al., 2018). Effective CS instruction must go beyond content knowledge and include teachers' firm understanding of concepts as well as their skillfulness and confidence in conveying them to students. This foundational knowledge is pivotal for building and delivering engaging and impactful learning experiences.

Teachers benefit from robust support systems, such as professional learning communities (PLCs), which foster ongoing collaboration, reflection, and growth (Niess et al., 2021). Establishing professional connections among teachers further facilitates the exchange of best practices, enriching the overall quality of instruction. It also provides teachers with a learning partner that they can reach out to when they are struggling. In essence, the combination of well-prepared content knowledge, robust support systems, and professional connections empowers teachers to cultivate a strong sense of self-efficacy, a belief in themselves that they are capable of learning and teaching new content. This, in turn, enables them to deliver high-quality instruction, ultimately enhancing the learning experiences of their students.

Research also underscores the importance of ongoing mentorship and coaching as integral components of effective professional development for teachers. Mentoring programs provide personalized support and guidance, allowing educators to receive targeted feedback and tailored assistance in implementing new instructional strategies (Kraft et al., 2020). By pairing novice teachers with experienced mentors, perhaps by local industry or higher-ed partnerships, these programs create opportunities for collaborative learning and skill development. Additionally, coaching interventions offer real-time support and reflective dialogue, empowering teachers to refine their practice and address specific instructional challenges (Kraft et al., 2020). The inclusion of mentoring and coaching within professional development initiatives not only enhances teachers' capacity to integrate new skills but also fosters a culture of continuous improvement within educational communities.

Devoting time and effort to designing a professional development plan that encompasses each of these components from the start is a critical first step. Doing so will help realize the end goal of developing teachers whose backgrounds are not in CS to be effectively skilled in CS content, as well as instructional strategies, that can deliver meaningful and impactful learning experiences for their students.

Importance of a research-based model in professional development design

A research-based model is imperative when designing professional development initiatives because it ensures that training plans are grounded in evidence and aligned with best practices in education. Research-based models provide a systematic approach to professional development, guiding the selection of activities and resources that have been shown to be effective in improving teaching practice and student learning outcomes (Garet et al., 2001). Additionally, research-based models promote accountability and continuous improvement by encouraging educators to regularly evaluate the impact of professional development initiatives and make data-informed adjustments as needed (Wei et al., 2009). Ultimately, a research-based approach to professional development fosters a culture of evidence-based decision-making and ensures that resources are invested in initiatives that have the greatest potential for success.

The Crawl-Walk-Run-Fly (CWRF) approach is one such research-based model that has been used in core subject areas to introduce simple tasks then advance to more challenging ones, and the foundation on which the professional development plan to deploy CS at the elementary level for this research project was based. This strategy emphasizes mastering the fundamentals first, then slowly taking small steps to scale up to success. Employing it in our research project allowed the teachers to have extended amounts of time in each area to gain the confidence and self-efficacy needed to successfully engage with the students. Figure 6.1 illustrates the four phases of the CWRF; a brief explanation of each phase follows.

Figure 6.1. Visual Representation of Crawl-Walk-Run-Fly (CWRF) Model

Note. Source: LinkedIn.com, used by permission. ©2019.

In the ***crawl*** phase, learners are introduced to fundamental concepts and skills through structured instruction and guided practice, laying the groundwork for deeper exploration by the individuals (Smith & Johnson, 2018). This phase serves as a crucial building block, fostering a solid understanding of fundamental principles and building a firm foundation for subsequent growth (Herold, 2017). If this step is skipped over or enough time is not devoted to it, the learner is starting with a handicap right out of the gate. In this introductory phase, participants are introduced to the foundational concepts and skills needed to teach elementary computer science. Just as infants learn to crawl before they walk, the crawl stage focuses on exploration, experimentation and acquisition of basic competencies (Smith & Johnson, 2018). Additionally, this stage provides an opportunity for the learners to play and develop a sense of curiosity in the subject matter (Jones, 2017).

Transitioning to the ***walk*** phase, the learners start to test out their newly-acquired knowledge and skills. They do this by integrating them into their teaching practices, through hands-on activities, project/problem-based lessons and through interdisciplinary connections (Smith & Johnson, 2018). This phase encourages experimentation, collaboration, and on-going reflection, as well as gathering feedback and making some refinements in their instruction. As they experience small successes, the learners tend to demonstrate increased autonomy and competence during this phase, as they continue to refine their skills and further develop their critical-thinking abilities (Stevens & Anderson, 2018).

As they quicken their stride and advance to the ***run*** stage, the learners reach that level of proficiency where they demonstrate that they can confidently lead computer science instruction, design innovative computer science curriculum and serve as advocates for computer science in their schools and communities (Smith & Johnson, 2018). This phase marks a significant advancement, showing how far they have come through the crawl and walk phases, to being able to successfully apply complex concepts and techniques related to computer science (Brown & Johnson, 2017). The learners also feel equipped to tackle challenging tasks and projects with confidence, and want to engage in learning and developing additional computer science content (Garcia & Anderson, 2020). Overall, the run phrase represents a job well-done, as the non-trained CS teacher has now successfully gained the requisite knowledge and skills to make significant contributions to CS education at the elementary level.

The addition of the ***fly*** phase represents the pinnacle of mastery and innovation within the educational context being studied. In this phase, the learners not only demonstrate high levels of proficiency but also excel in pushing boundaries, driving change, and contributing to advancements in their field (Garcia & Smith, 2019). This phase is defined by a strong sense of continuous improvement, where learners engage in self-directed learning, creative problem-solving, and personal leadership development. By integrating the fly phase, the crawl-walk-run-fly model provides a comprehensive framework for lifelong learning and academic growth, empowering students to thrive in a rapidly evolving educational landscape.

Bruner’s Spiral Curriculum Approach to complement the CWRF Model

At its core, Jerome Bruner’s spiral curriculum approach, grounded in his theory of cognitive development, emphasizes the importance of scaffolding learning experiences to match the learner's current level of understanding while progressively introducing more complex concepts (Bruner, 1960). The spiral metaphor reflects the idea that learners continuously spiral back to previously encountered concepts but at increasingly sophisticated levels, thereby expanding and refining their understanding with each iteration (Bruner, 1966). This approach aligns with the notion that learning is not always a linear progression but rather a dynamic and ongoing process of constructing meaning through interaction with one's environment as well as prior learning experiences and knowledge. Figure 6.2 illustrates a view of Bruner’s spiral model.

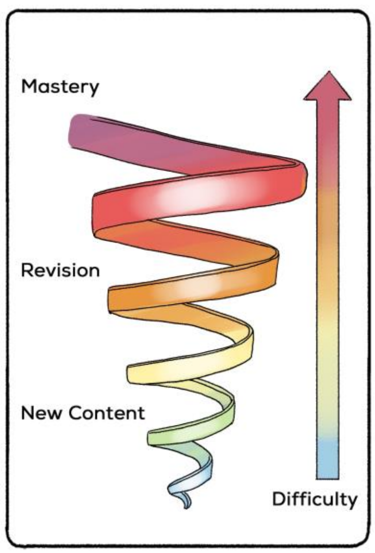


Figure 6.2. Bruner’s Spiral Curriculum Approach (1960)

Note. Source: Creative Commons, used by permission. ©2023 by Gagandeep Kaur.

In computer science education, Bruner's spiral curriculum approach can be effectively applied to facilitate the learning of introductory concepts and skills while laying a foundation for the development of higher-order thinking abilities essential for computational thinking and problem-solving. His model emphasizes revisiting key concepts in a progressively more complex manner, and by revisiting and expanding upon these foundational concepts iteratively, learners not only solidify their understanding but also develop the ability to apply them in increasingly sophisticated contexts, mirroring real-world software development scenarios (Feldman et al., 2017). Further, in one study, the spiral approach was shown to encourage active engagement and exploration and enabled the learners to discover connections between various subjects and understand how knowledge can be integrated and applied across multiple disciplines (Haden & Wilensky, 2014)..

As the teachers engaged in learning CS concepts, they were able to spiral back to foundational ideas while progressively exploring more advanced topics (Bruner, 1966). This approach highlighted that CS education is not merely about imparting a new set of skills but also about fostering deep conceptual understanding and problem-solving abilities that evolve over time. The professional development program for teachers clearly defined a culture of continuous learning and innovation as well as personal reflection, ultimately empowering educators to inspire and prepare the next generation of digital citizens.

Research Design

Research Project Design

The purpose of this research project was two-fold: first, to design, implement and assess a professional development program for in-service elementary teachers to learn computer science skills and pedagogy and second, to chronicle the experiences of a district computer science cohort of teachers through a year-long professional development journey. These teachers volunteered to take part in the project and were not compensated for their time, except in the form of my undying gratitude.

The following sections provide justification for the design of the project, a review of the research questions with a short explanation of each one and their importance to the project, detailed information about the research participants and their backgrounds, and the data collection and analysis protocol employed.

Utilizing a qualitative case study approach

Because this project dealt with human subjects and their feelings/emotions centered around a certain phenomenon, a qualitative case study approach was selected.   According to Merriam (1998), qualitative case study approaches are highly adaptable, and allow researchers to modify their strategies as new data emerges. This flexibility turned out to be crucial when dealing with dynamic environments such as classrooms of elementary students or unexpected developments within the case. I found it to also allow for deeper engagement with participants, fostering a stronger sense of trust and openness, which I felt led to more authentic and insightful data.   Employing this methodology enabled me to thoroughly investigate this evolving field, yielding valuable insights for both theory and practice without recycling my previous work.

Review of the Research Questions

Guiding questions are fundamental in a research project as they help to establish the study's focus, ensuring alignment with its objectives and scope. In this project, they provided a structured framework for data collection and analysis, assuring a systematic exploration of the research topic. They also helped delineate the critical aspects of the phenomenon that was being investigated and kept the investigation focused on specific targets. The guiding questions driving this research project were:

1. What was the researcher's experience designing and implementing a professional development program that is grounded in the Crawl-Walk-Run-Fly model layered with Bruner’s Spiral Curriculum Approach model?
2. What was the overall learning experience of the participating teachers in a professional development program that is grounded in the Crawl-Walk-Run-Fly model layered with Bruner’s Spiral Curriculum Approach model?

Regarding the first question, this actually became a two-part question that altered my entire project. First, it prompted me as the researcher to begin the project by reflecting on what I considered to be key elements of an effective professional development program and how those could be incorporated into the training I was preparing. Three components emerged that became the basis of my training plan. Second, it led me to define criteria to use in selecting established research-based models on which to build said plan. Both of these questions and their subsequent answers will be discussed in the Results section.

The second question gave me the opportunity to do a deep dive into the teachers’ initial survey responses and blogs and note how their individual experiences, beliefs, and backgrounds influenced their approach to this professional development. I also thought knowing that their self-reflections would not be read or discussed outside of our group gave them courage to be brutally honest about their learning and about the training pieces. By examining their personal journeys individually and collectively, I was able to gain valuable insight into how they preferred to learn, and what activities and/or strategies yielded the best outcomes, and was able to incorporate many of their recommendations into the training as the year progressed. In the Results section, I will identify the trends that emerged from the data and share what that looked like in the training and how it will be used going forward.

Research Participants - Background Information

Location. This study was conducted within a mid-sized, suburban Texas school district. As of April 2024, the district had 21 elementary schools, with 18 designated as Title I campuses, indicating that at least 40% of students qualified for free or reduced lunch (Texas Education Agency, 2023). The district served approximately 11,000 K-6 students, with the following racial/ethnic breakdown shown in Table 6.1.

Table 6.1 District Race/Ethnicity Statistics (October 2023)

Note: Data obtained from Texas Education Agency, ©2023.

Race/Ethnicity	No. of students in K-6	Overall % of K-6 students
Latino/Hispanic	3,630	33%
White	3,300	30%
Black	2,240	22%
Asian	880	8%
Pacific Islander	330	3%
Two or more races	440	4%

Participants. The participants in this study were selected from a larger computer science initiative within the school district, overseen by the author in their role as a district STEM administrator. Recruitment commenced with the distribution of a flyer to all K-8 teachers engaged in the district initiative in October 2022, followed by two Zoom Q&A sessions held in November and December 2022. These sessions conveyed details about the project's objectives, timelines, and expectations for participants.

When recruitment was completed, there was one male teacher and 17 female teachers participating; their years of experience ranged from 0 years to 20+ years and their subjects taught ranged from core curricular areas to art to instructional coaching. Additionally, at our first meeting, I asked each of them to share a one word superpower or strength that they would bring to the group and I incorporated that into a pseudonym for each one, as they preferred their comments to be anonymous. Table 6.2 shows a summary of their pseudonyms, what subject(s) each of the participants currently teach, and the total number of years they have been teaching.

GRADE(S)	SUBJECT(S)	YEARS OF EXPERIENCE
6th grade	Mathematics	1 – 5
1st grade	All subjects	6-10
8th grade	Science	6-10
4th grade	Mathematics, Science	6-10
5th grade	Mathematics	1-5
6th grade	Science, Social Studies	6-10
PreK – 6th grade	Art	11-15
PreK	All subjects	6-10
6th grade	Mathematics	6-10
4th grade	English/Language Arts	11-15
PreK – 6th grade	Instructional Coach	16-20
3rd grade	English/Language Arts	1-5
4th grade	English/Language Arts	1-5
2nd grade	All subjects	6-10
5th grade	Mathematics	6-10
6th grade	Science, Social Studies	1-5
6th grade	Science, Social Studies	1-5
PreK – 6th grade	Reading Specialist	16-20

Table 6.2. CS Cohort Participant Data.

Data Collection & Analysis

During this project, data collection encompassed two primary methodologies: surveys and interviews. Additionally, teachers were tasked with establishing and maintaining an online blog to facilitate continuous reflection on their learning journey, while also inviting input from others in the cohort through comments and suggestions. These provided me with a great insight on preferred learning styles, comfort levels in sharing out to the group raw emotions as they struggled to learn. The blogs were not made public beyond the participants of this research project and myself.

Surveys were administered at the project's outset and its conclusion. Drawing questions from the *Teacher Efficacy and Attitudes Towards STEM (T-STEM) Survey*, the aim was to assess changes in teachers' confidence and self-efficacy across various domains including STEM subject matter, technology integration, 21st-century learning skills, leadership attitudes, and STEM career awareness (Friday Institute, 2012). Initial survey results guided the design of the initial professional development phase, revealing significant trends in teachers' perceptions of their own proficiency in computer science.

Each participant also underwent two in-person interviews: one at the six-month mark and another at the research's culmination. These one-on-one interviews sought to evaluate the training's effectiveness and gauge teachers' comfort levels with the introduced materials. For each question in this research project, I used *descriptive coding* to analyze the qualitative data and followed the same set of steps each time. I read and reread the transcripts as well as my research journal to understand the content deeply (Saldaña, 2016), and noticed patterns and themes emerging naturally, without forcing them into predetermined categories (Charmaz, 2014). I refined the codebook through repeated coding and analysis to accurately capture the subtleties in the data. After coding, I grouped related codes into broader categories or themes to interpret the findings (Saldaña, 2016). By examining the relationships between codes, I uncovered connections that shed light on overarching concepts. This systematic approach helped me describe the research topic in detail, providing valuable insights.

Results

In this section, the findings of the study are presented, offering insights garnered from the analysis of collected data. Through a systematic examination of the data, key themes, patterns, and trends emerged, providing a comprehensive understanding of the research questions. The results are organized to address each specific research objective, offering a detailed account of the outcomes observed. These results not only contribute to the existing body of knowledge in the field but also offer practical implications for stakeholders and practitioners alike. I will address each question individually.

RQ1: What was the researcher's experience in designing and implementing a professional development system using the Crawl-Walk-Run-Fly model layered with Bruner's Spiral Curriculum Approach model?

It is important to note that the professional development program for non-computer science teachers learning computer science began by following a crawl-walk-run-fly approach, mirroring the gradual progression of skills and knowledge acquisition. But I soon discovered that it might not be enough on its own to get my teachers where they needed to be, so I returned to the drawing board to see what other options might be available to help the teachers have a firmer grasp of the subject knowledge.

In my research journal, I noted several times after reading the teachers' reflections that they seemed to have drifted off the linear crawl-walk-run-fly model due to factors such as individual learning styles, prior knowledge, and environmental influences. For instance, some of the teachers pointed out that they had skipped certain stages because they already possessed the relevant skills or knowledge with All-In Adam telling me "I already knew that, so I skipped over it and went on to something more challenging." I also noted several instances where they revisited earlier stages to reinforce understanding or address gaps in learning. Crafty Campbell remarked about how grateful she was for the time between assignments, since she had to “flip back a couple of times” to remember how to complete some of the work. On top of these instances, preferred instructional methods, resources, and support systems also influenced the pace and direction of learning trajectories (Anderson & Krathwohl, 2001) and caused minor deviations.

So while the crawl-walk-run-fly model offered a valuable framework for scaffolding ***learning experiences*** for the teachers, I do not think it was able to take into account and accommodate for the oftentimes non-linear nature of individual ***learning processes***. To provide a model that would account for both experiences and processes, I decided to introduce another approach - Bruner’s Spiral Curriculum (1960). His curriculum approach was built on three key concepts - **cyclical** (students should return to the same topic several times), **increase in depth** (each time it is revisited it should be learned at a deeper and more complex level) and ***access* prior knowledge** (should be utilized each time so students build on their foundation rather than starting anew) (Drew, 2023). It turned out to be the missing piece of the PD puzzle that I was looking for!

In a perfect CWRF-model world, participants would begin with the "crawl" phase, where they familiarize themselves with fundamental computer science concepts and terminology. Through introductory workshops and guided activities, they would lay the groundwork for more advanced learning. As they gained confidence and proficiency, they transitioned to the "walk" phase, where they delved deeper into specific topics and explored practical applications in their teaching contexts. Hands-on exercises and collaborative projects allowed them to refine their skills and develop a deeper understanding of computational thinking principles and greater confidence in teaching these principles to their students. Subsequently, participants entered the "run" phase, where they demonstrated autonomy and initiative in applying their knowledge to design and implement computer science-infused lessons. Finally, in the "fly" phase, participants showcased creativity and innovation by integrating advanced computational concepts into cross-curricular projects and real-world scenarios, effectively elevating their teaching practice to new heights. This crawl-walk-run-fly approach should have enabled participants to progress systematically from foundational understanding to mastery, empowering them to confidently incorporate computer science education into their classrooms. While all of the boxes were checked, as a researcher, I still felt that something was missing in the overall design of the professional development and that more might be needed!

As indicated earlier, I began to see references in the teacher's reflections and interviews to needing to go back and revisit topics, spend longer time on concepts than I had allowed for, skipping over concepts they felt they already understood, etc. and I felt like I needed to try and address these issues before moving forward. Incorporating a new strategy into an existing professional development (PD) plan involves careful planning, collaboration, and alignment with the overarching goals and objectives of the program. I reviewed the overall goals of the professional development and looked at best practices available that could be integrated without having to begin again, as well as revisited the pre-survey questions and informally polled the group for their suggestions.

All of this led to additional research about other possible options and after evaluating several prospects, I settled on a plan to layer a modified version of Bruner's Spiral Curriculum Approach on the current Crawl-Walk-Run-Fly model. My primary goal in integrating the new strategy into existing PD sessions and workshops was for it to be a seamless implementation and provide additional opportunities for hands-on practice and reflection. By systematically incorporating new strategies into an already existing PD plan, I was also able to model a continuous improvement mindset towards professional and personal growth, ultimately enhancing teaching effectiveness and student learning outcomes.

Figure 6.3 is my visual interpretation of the fusion of the two frameworks showing the linear (learning experiences) and nonlinear (learning processes) and how they work together. The learning experiences are linear on the left side and step up from crawl to fly as indicated by the straight line. The learning processes are non-linear on the right side and the double-ended arrows represents moving back and forth between individual stages of the model and the dashed line represents the spiral approach of moving between the stages as needed.

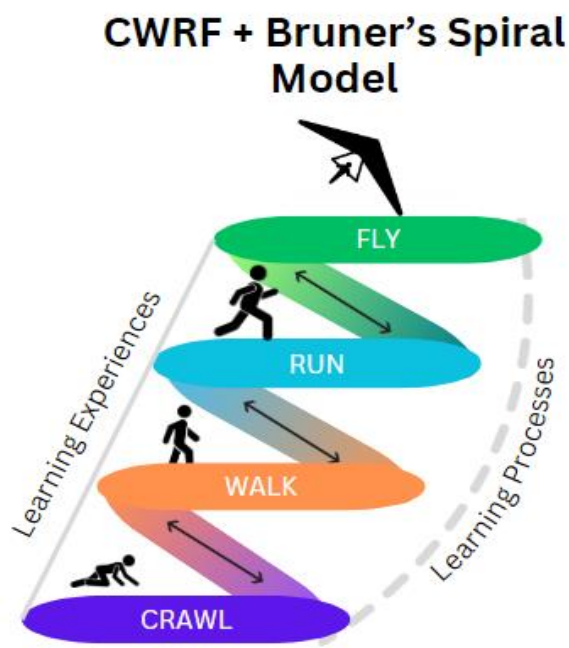


Figure 6.3. CWRF + Bruner's Spiral

Note. Source: Kiera Elledge. ©2024.

Overall, I felt that the professional development program aimed at equipping non-computer science teachers with foundational knowledge and skills in computer science yielded promising results. Through a series of workshops, hands-on activities, and collaborative discussions, participants demonstrated significant growth in their understanding of key computer science concepts and pedagogical strategies. Pre-and post-assessment data revealed a substantial increase in participants' confidence levels regarding their ability to not only learn the computer science skills, but also ways to integrate computer science concepts into their teaching practices. Moreover, qualitative feedback highlighted the effectiveness of the program in demystifying complex computer science topics and fostering a supportive learning environment. Participants reported feeling more empowered and enthusiastic about incorporating computational thinking principles across various subject areas as well. These findings underscored the potential of targeted professional development initiatives to bridge the gap between computer science education and other disciplines, with a goal of giving teachers as many tools as possible to reach their students.

It is my opinion that the other adjustments I made during the professional development were minor and not sufficient enough to warrant discussion of. Additional time to complete the assignments, extra options for an individual or two to get them more comfortable with a particular device or topic and increased one-on-one coaching were the primary types of adjustments I had to make, which were accomplished by adding in Bruner's spiral approach.

RQ2: What was the overall learning experience of the participating teachers in a professional development program that is grounded in the Crawl-Walk-Run-Fly model layered with Bruner's Spiral Curriculum Approach model?

How the participants felt the professional development went is a little harder for me to answer as the researcher. 15 of the 18 teachers completed the whole program and provided detailed reflections, took both the pre and post surveys, and participated in both interviews. The three that did not complete everything completed the majority of the pieces - 1 did not take the post survey and 2 did not participate in their second interviews. No reasons were given for these anomalies.

The culminating project was to redesign a current curriculum unit (at least 5 - 10 days in length) and integrate a minimum of 4 computer science skills/tools. Then they had to teach the unit, gather any evidence they wanted to share and create a post-unit reflection video that they shared on their blogs. There was no rubric or assigned criteria - rather, it was what the teachers deemed to be the most important take-aways from completing the work.

The reflections were thoughtful as well as thought-provoking, and led to some excellent exchanges online and during their second interviews. These end-of-project remarks and suggestions for change/improvement showed great insight and maturity. They were brutally honest about their struggles in the beginning and very appreciative of the opportunities in the latter half of the program to go back and revisit topics from the beginning. Many of them expressed gratitude for being able to participate in the professional development, stating how much they had enjoyed working with me and with the other teachers. They also praised the set up of the entire project and asked if there was any way they could continue it moving into 2024!

One teacher in particular was effusive in her praise of the professional development. Excited Ella noted: “I have learned so much about myself as a teacher and as a learner this year! Working with you [Kiera] and the other teachers has given me a fresh perspective of how to better engage my students and meet them where they are. We are so excited to implement some of these computer science tools in our classroom and our after-school STEM clubs as soon as possible.”

In regards to the pre/post surveys, one question in particular stood out. It read: "Please rate yourself on the following statement - I know the steps to teach computer science effectively." They were given a Likert scale of 1 to 5, with 1 being strongly disagree and 5 being strongly agree. Of the 17 that completed the pre/post surveys, 100% of them scored themselves higher on this question in the post survey than the pre survey! Granted, one moved from strongly disagree to disagree and others from disagree to somewhat agree, but every one of them moved in a positive direction on that question. I will take that as a win!

Conclusion

Overall Program Assessment

In conclusion, I felt that fusing the crawl-walk-run-fly model with Bruner's spiral approach offered a well-rounded framework for professional development that provided clear goals to mastery (linear processes) and offered choice to the learners in how they learned it (non-linear processes).

The crawl-walk-run-fly model provided a solid start to the professional development plan. It offered a clear, staged pathway for skill acquisition and mastery. This model ensured that the learning opportunities were both structured and scalable. .Bruner's philosophy of revisiting topics at increasing levels of complexity ensured that learners deepened their understanding over time, which enhanced their ability to apply their newfound knowledge in various contexts (Bruner, 1960). This cyclical nature of learning not only reinforced the foundational skills of computer science, but also encouraged continuous growth and adaptability, which are crucial in a rapidly evolving professional landscape in which technology is so prevalent.

By synthesizing Bruner's spiral curriculum with the incremental progression of the crawl-walk-run-fly model, it is possible to create robust professional development programs that are both comprehensive and flexible. I felt that this dual approach not only addressed the immediate learning needs of the participating teachers, but also equipped them with the skills and mindset necessary for long-term success. Ultimately, it is hoped that this integrative strategy will foster a culture of continuous learning and improvement, essential for maintaining competitive advantage in today's dynamic educational environment.

Limitations and Future Research

Despite the promising integration of the crawl-run-walk-fly model and Bruner's spiral approach in professional development, I found several limitations. One significant limitation was the variability in individual learning paces, which did not always align perfectly with the structured phases of the crawl-walk-run-fly model. This misalignment sometimes resulted in teachers feeling either rushed or held back, which could have potentially affected their engagement with the material and overall learning of computer science. Additionally, while the spiral approach was beneficial in reinforcing concepts, it occasionally led to redundancy, where some learners felt they were revisiting material multiple times without enough new challenges. In an attempt to mitigate some of these limitations, it was important to assure that all participants had access to the same instructional materials and resource opportunities. As those may vary widely within a school or district, ensuring equitable access will be critical to the success of any professional development initiative.

In considering future research, I believe it will be important to explore ways to potentially customize these models even further to better accommodate diverse learning styles and paces. Utilizing adaptive learning technologies that tailor the progression through the crawl-walk-run-fly stages based on real-time performance data could enhance the personalized learning experience. Additionally, further studies should examine the long-term impacts of integrating these models on teacher professional growth and career advancement, providing empirical evidence to support their efficacy. Lastly, researching the scalability of these models in other educational settings would offer valuable insights, ensuring that professional development programs could be effectively implemented across multiple grade levels and subject areas.

Acknowledgments

First, I would like to express my sincere gratitude to Dr. Suzanne Porath (Kansas State University) for her valuable contributions and unwavering support throughout the research process. Dr. Porath's expertise, guidance, and assistance have been instrumental in shaping this study and the ensuing scholarly articles.

Additionally, I extend my heartfelt appreciation to the participants who generously shared their time and insights, without whom this research would not have been possible, and for the assistance and encouragement of the cooperating school district.

Lastly, I acknowledge the broader academic community for their ongoing efforts in advancing knowledge in this field.

References

Anderson, L. W., & Krathwohl, D. R. (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. Longman.

Brown, A., & Johnson, R. (2017). Advancing proficiency: The run phase in the crawl-walk-run model. Educational Development Journal, 22(3), 145-158.

Charmaz, K. (2014). Constructing grounded theory (2nd ed.). Sage.

Darling-Hammond, L., Hyler, M. E., & Gardner, M. (2017). Effective teacher professional development. Learning Policy Institute.

Drew, C. (October 15, 2023). *Bruner's Spiral Curriculum – The 3 Key Principles*. Helpful Professor. https://helpfulprofessor.com/spiral-curriculum/

Education Corner. (n.d.). The importance of digital literacy in K-12. Education Corner. https://www.educationcorner.com/importance-of-digital-literacy.html

Franke, M. L., Kazemi, E., & Battey, D. (2013). Mathematics teaching and classroom practice. In J. Hattie & E. Anderman (Eds.), International guide to student achievement (pp. 223-225). Routledge.

Garcia, M., & Anderson, L. (2020). Mastering complexity: The role of the run phase in professional development. Journal of Professional Growth, 35(2), 78-92.

Garet, M. S., Porter, A. C., Desimone, L., Birman, B. F., & Yoon, K. S. (2001). What makes professional development effective? Results from a national sample of teachers. American Educational Research Journal, 38(4), 915-945.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. Educational Researcher, 42(1), 38–43. https://doi.org/10.3102/0013189X12463051

Guzdial, M., & Ericson, B. (2015). A Tale of Two AP CS Principles Pilots: Factors Affecting the College Readiness of Georgia Students. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (pp. 640-645). ACM. https://doi.org/10.1145/2676723.2677254

Guskey, T. R. (2009). Closing the knowledge gap on effective professional development. Educational Horizons, 87(4), 224-233.

Johnson, R., & Smith, K. (2019). Building a path to proficiency: The crawl-walk-run approach in professional development. Journal of Continuing Education in Nursing, 50(3), 132-136.

Jones, S. (2017). Beyond the run: The fly phase in education. Educational Leadership, 65(4), 30-35.

Jones, S., & Smith, K. (2019). Leading the way: Leadership development in the run phase of professional growth. Leadership Quarterly, 40(4), 321-335

Kaur, G. (2023, June 16). *Bruner's Spiral Curriculum | OER Commons*. OER Commons. https://oercommons.org/courseware/lesson/105463/student/?section=1

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. Educational Psychologist, 41(2), 75–86.

Kraft, M. A., Blazar, D., & Hogan, D. (2020). The effect of teacher coaching on instruction and achievement: A meta-analysis of the causal evidence. Review of Educational Research, 90(3), 1-36. https://doi.org/10.3102/0034654320915597

Margolis, J., & Estrella, R. (2010). Stuck in the Shallow End: Education, Race, and Computing. MIT Press.

Menekse, M. (2015). Supporting teacher change in integrating computational thinking in elementary science classrooms: A model of action research and professional development. Journal of Science Education and Technology, 24(5), 624-640. https://doi.org/10.1007/s10956-015-9560-1

Misha, A. (2023, December 15). *Can affordable computing education bridge the global digital divide?* EdSurge. https://www.edsurge.com/news/2023-12-15-can-affordable-computing-education-bridge-the-global-digital-divide

National Science Foundation. (2017). Broadening Participation in Computing (BPC) (NSF 17-567). Retrieved from https://www.nsf.gov/pubs/2017/nsf17567/nsf17567.htm

Niess, M., Graziano, K. J., & Gao, X. (2021). Professional development in K-12 computer science education. Springer.

Qian, Y., Youngs, P., & Frank, K. A. (2018). Learning to teach mathematics and to teach within the curriculum: A comparative study of novice teachers' experiences in China and the United States. Journal of Mathematics Teacher Education, 21(4), 301-328. https://doi.org/10.1007/s10857-017-9389-6

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... Kafai, Y. (2009). Scratch: Programming for All. Communications of the ACM, 52(11), 60–67. https://doi.org/10.1145/1592761.1592779

Saldaña, J. (2016). The coding manual for qualitative researchers (3rd ed.). Sage.

Smith, J., & Johnson, R. (2018). The crawl-walk-run-fly model: A comprehensive framework for educational development. Journal of Educational Research, 42(3), 211-225.

Stevens, M., & Anderson, L. (2018). Applying the crawl-walk-run model to employee training and development. Human Resource Development Review, 17(2), 223-240.

Texas Education Agency. https://tea.texas.gov/finance-and-grants/grants/essa-program/swtrainingqa-t1a-campus-elig-allocation.pdf

Wei, R. C., Darling-Hammond, L., Andree, A., Richardson, N., & Orphanos, S. (2009). Professional learning in the learning profession: A status report on teacher development in the United States and abroad. National Staff Development Council.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127-147.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2018). Computational thinking in elementary and secondary teacher education. ACM Transactions on Computing Education (TOCE), 18(1), 1-16. https://doi.org/10.1145/3168744

Chapter 7 - Discussion and Conclusion

In the ever-evolving landscape of public education, the need to continuously adapt and innovate has never been more pronounced if the US wants to produce a workforce capable of filling the hundreds of thousands of open computer science-related jobs in this country. With this goal in mind, it is critical that schools embrace change as a means of not only surviving but thriving in an increasingly complex world. As society's needs continue to shift and technological advancements continue to accelerate, we must position our schools to be adaptable to the changing conditions in the world in order to stay relevant.

Having a clear vision and strong leadership at the top of the district is a critical first step in being able to adapt and change to meet today's workforce demands. When implementing new policies or restructuring curriculum or current processes, it is also vital to involve the stakeholders who have a vested interest in the school's success, including parents, students, local government, and community organizations. In addition, developing a plan that allows for on-going teacher professional development and phased implementation helps to manage any resistance in the classroom and ensures that teachers and students can gradually accept the changes. Lastly, continuous monitoring of the system as it is being overhauled helps identify potential challenges and allows for on-going adjustments. Realizing that all of the pieces of the system are interconnected and that change can have a ripple effect across the entire system can help guide schools to adjust in ways that benefit students, teachers, and the community as a whole.

Review

Problem of Practice

In today's digital age, providing elementary students with foundational computer science skills is critical. However, a significant challenge exists as many elementary school teachers lack formal training in computer science and its related fields, which hinders their ability to effectively teach these essential skills. This gap in teacher preparation affects students' early engagement with computer science, limiting their development of critical thinking, problem-solving, and computational skills that are vital for future learning and career opportunities (Barr & Stephenson, 2011). Addressing this issue requires identifying and empowering the hidden potential of current elementary school teachers to become proficient in both learning and teaching computer science

To bridge this gap, it is essential to provide professional development and training programs that equip current elementary teachers with the knowledge and resources needed to teach computer science effectively. Many of these teachers, despite lacking formal training in the subject, possess a passion for learning and a dedication to their students' success. They understand the importance of modeling a growth mindset for their students by being life-long learners themselves, willing to learn what they do not know.

This research study followed a group of eighteen teachers on a year-long journey of professional learning. It assessed gaps in their current knowledge, identified needed professional development, and provided resources and tools to support the teachers in their efforts to learn computer science skills. Their experiences – both positive and not-so-positive – are reviewed in this section. Implications for schools, districts and beyond are also presented, as well as recommendations for future research.

Methodology

Qualitative research, often exploratory in nature, aims to ferret out feelings, reasons and meaning behind human behavior. It has the power to produce detailed, narrative data through interviews and observations, but it also has the potential to produce change as it is being implemented (Hatch, 2023). With that, electing to conduct a qualitative research project on this computer science professional development cohort allowed me the opportunity to gain a deeper understanding of technology development and use. Unlike quantitative methods, which focus on numerical data and statistical analysis, qualitative research explores the experiences, motivations, and perceptions of individuals involved in or affected by these computer science innovations. It rich insights into user behaviors, developer practices, and the socio-cultural impacts of technology, leading to more user-centered designs, inclusivity, and ethically aware technological advancements. By capturing the nuanced realities of the digital world, qualitative research can significantly enhance the field's ability to address complex, real-world challenges and improve the overall efficacy and acceptance of technological solutions (Hammersley, 2013).

Review of Research Design and Questions

This qualitative research project employed a case study design to explore the implementation and impact of a targeted professional development model focused on computer science curriculum for K-6 teachers. The goal was to understand the experiences of the teachers throughout the professional development, identify challenges and successes, and provide insights for future curriculum development and teacher training programs. The case study method was selected for its ability to provide an in-depth understanding of the complex phenomenon within a real-life context (Yin, 2018).

The research questions are crucial in a dissertation as they guide the entire study, defining the focus and direction of the research. They also help to establish clear objectives and ensure that the research is purpose-driven and relevant. Two over-arching questions framed this project:

1) What are the effective components of a professional learning model that includes the necessary academic and social supports for teachers to successfully learn and learn to teach computer science?

2) In what ways, if any, are the participants of this cohort able to take their new knowledge and implement it in their classroom or school as a way to increase awareness of the need for all students to have access to computer science education?

Each question (or a slight variation of it) was investigated through three different lens – the Technological Pedagogical and Content Knowledge (TPACK) Framework, Albert Bandura's Theory of Self-Efficacy and a blended model of Crawl-Walk-Run-Fly (CWRF) and Bruner's Spiral Curriculum Approach. Those results were detailed in chapters four through six and will be briefly discussed again in this section.

Discussion of Major Findings

This section examines the key results from the study on professional development for elementary computer science (CS) teachers, interpreting their significance in light of the research questions and existing literature. The findings reveal crucial insights into the effectiveness of current professional development programs, highlighting both the strengths and areas needing improvement. Specifically, the study uncovers the importance of sustained, hands-on training and the role of collaborative learning communities in enhancing teachers' confidence and competence in delivering CS education. By discussing the implications of these results, we gain a deeper understanding of how to optimize professional development for elementary CS teachers, addressing the challenges they face and improving student outcomes. Additionally, this section will explore the limitations of the study and propose directions for future research to further advance the field.

Research Question 1

This question dealt with identifying the components of a professional learning module that included both academic and social supports for teachers to be able to learn and to teach computer science. Administering a pre-survey at the beginning of the research provided great insight into the teacher's current state regarding computer science – both learning and teaching – and helped guide the first round of professional development.

TPACK. The Technological, Pedagogical and Content Framework (TPACK) measures a teacher's ability to use technology in a way that enhances learning and teaching of specific content. This framework helps in identifying how well educators can incorporate digital tools into their teaching to improve educational outcomes and evaluates the intersection of three primary forms of knowledge: technology (how to use technology tools and resources), pedagogy (teaching methods and practices) and content (subject matter).

In this study, from start to finish, the teachers ranked the lowest in technological knowledge. In interviews and on their blogs, many stated that they felt most inadequate in knowing how to not only use some of the tools and resources, but also how to teach them to students. Many of the tools provided were ones that they were not familiar with, even though they are well-known and widely accepted classroom teaching tools. Two tools in particular, the micro:bit and Makey Makey, were completely unfamiliar to over 80% of the cohort, and this was a major source of frustration for the teachers. Still, they found value in the productive struggle and concluded that having to strengthen their own growth mindset and stretch to learn something new helped them have greater empathy for their students when they struggle. This aligns with research from Tschannen-Moran & Hoy (2001) which stated that for teachers, the productive struggle is essential for fostering a growth mindset in their approach to learning and teaching computer science, ultimately enhancing the overall quality of instruction.

Content knowledge of their subjects taught was the highest rated area of the study, with teacher’s expressing strong confidence in knowing the material they were teaching. The prevailing comments in this section indicated that the teachers felt very comfortable with the curriculum components, but also did confess that their level of confidence in the content of computer science was not as strong. Even some of the basic CS terminology, such as computational thinking and algorithms, were considered foreign terms to over 50% of the teachers. For them, laying a strong foundation for all things computer science would be critical to the success of the study.

Self-Efficacy. Bandura's theory of self-efficacy emphasizes the belief in one's capabilities to achieve a goal or an outcome (1977). In the context of teachers learning computer science (CS), this theory suggests that teachers who believe in their ability to learn and teach CS effectively are more likely to engage in professional development and persist in overcoming challenges. Self-efficacy influences their motivation, effort, and resilience, which are crucial for mastering new CS concepts and technologies. Teachers with high self-efficacy in CS are more confident in integrating CS into their curriculum, thereby enhancing student learning. Therefore, fostering self-efficacy in CS education can lead to more effective teaching practices and improved student outcomes in computer science.

The teachers in this study were very open in their reflections of their own learning and these reflections led to not only some powerful discussions online and in person, but also to many opportunities for the teachers to provide support and encouragement to each other. With a blend of self-directed and collaborative experiences, they had the flexibility to experience multiple types of learning opportunities and being able to pick/choose what worked best for them (emotional state), which promoted the development of personal skills (mastery experiences). The on-going reflections and evidence of learning on their blogs provided the opportunity for their peers to provide support and encouragement (vicarious experiences) and build a sense of community. As they worked together on projects, the chance to socialize and interact among disciplines (social persuasion) also strengthened the teachers and led to an increase in self-efficacy.

Crawl-Walk-Run-Fly (CWRF)/Bruner's Model. Blending a linear (CWRF) and non-linear (Bruner) approach to learning computer science provided yet another way to customize the professional development for each participant. Through the interviews, online reflections and my researcher journal notes, I was able to discern that not all of the teachers progressed at the same pace and that they did not all have as firm of a foundation in the basics of computer science as was necessary for them to be successful going forward as they needed to be. I addressed this situation by making an adjustment to the CWRF Model to bring in Bruner's Spiral Approach, which gave the teachers the opportunity to move among the checkpoints as needed. This change affirmed a statement about qualitative research by Hatch (2023) who said that this type of research also has the potential to bring about change. It did and that change added a whole new dimension to this study. I felt that putting the linear and non-linear processes together provided the greatest opportunities for the teachers to be successful. It did add additional work for me as the researcher, but in the end, I felt that the return on that investment was worth it.

For the participants, they were given the end goal and provided with checkpoints that increased in difficulty as they progressed (crawl to walk to run to fly). They were also given the freedom to stay at a checkpoint as long as they needed to and/or revisit earlier checkpoints to solidify a concept. Combining these models supported differentiated instruction, catering to diverse learning styles and paces. The CWRF model's clear stages (crawl, walk, run, fly) aligned well with the spiral approach's iterative nature, and allowed educators to customize learning experiences based on individual needs. This dual approach also encouraged continuous engagement and motivation, as learners experienced both the satisfaction of mastering immediate tasks and the intellectual stimulation of revisiting and expanding on previous knowledge (Dweck, 2006). Comments from the participants affirmed that the combination of linear and non-linear added more flexibility for them and gave them confidence to move forward if they were ready and stay behind a little longer if they were wanted.

An additional aspect that was not anticipated, but pleasantly accepted, was the overall strength of the professional learning community (PLC) that emerged as the professional development requirements ramped up throughout the year. In the beginning, I would hold optional online PLC meetings. My intent was to have them elaborate on any comments that they had given through feedback and also provide a recommendation or two of other sources that might could help a colleague who was struggling. What I witnessed was a tremendous bond of friendship being formed by several in the group, who seemed to take the most time providing thoughtful and constructive feedback through the blogs, then wanted to elaborate and expound on it in the online forum. They took great care to always lead with positive statements and downplay the challenges. I appreciated the professionalism that they brought to the online and eventual in-person discussions and felt that the strength of the PLC added another dimension of confidence to those who fully participated.

Research Question 2

This question addressed the issue of the teachers being able to take their newfound knowledge and implement it in their classroom or school to increase awareness of computer science education for all students.

TPACK. None of the teachers participating in this study had heard of the TPACK Framework prior to now. As a result, a quick overview of the framework and its constructs, as well as why it was being used in this study was provided in one of the early face-to-face trainings. Some very solid discussions ensued and the teachers were asked to reflect on this overview in their blogs and do an internal assessment to see where they thought their strengths and weaknesses were and why. Their reflections showed a great deal of honesty and very closely mirrored their answers from the pre-survey. At the end of the study, they were encouraged to revisit their original post and do another internal assessment to see if anything had changed, and reflect on their answers as part of their final blog post. One particular quote really resonated with me and let me know that these teachers "got it" when it came to computer science the importance of making sure all students have access. Newfangled Nancy declared: "As technology continues to evolve in education, someone is going to have to stand up and be willing to try new things…that is going to be me."

Self-Efficacy. Bandura (1977) stated that self-efficacy is the belief in one's capabilities to organize and execute the courses of action required to manage situations. This statement succinctly encapsulates his theory that self-efficacy influences how people think, behave and feel, and that high self-efficacy can lead to greater motivation, perseverance and success in achieving goals. Referring to Newfangled Nancy's statement above in the TPACK section, these teachers were on fire and ready to share their new learning with anyone who would give them an audience. Two of the participants at the same school wrote a grant to start a robotics club in the fall and another wrote a grant to purchase more robots so she could expand her club in the upcoming school year. Four of the teachers requested to attend a three-day K-5 elementary CS summit (the last week of school!) that as part of their stipend, required them to provide training to an additional 30 teachers in the school year. And yet another applied for a computer science position in another school district because the district's plan to expand in the 2024-2025 school year had been delayed! These and other actions provide solid evidence that strengthening their self-efficacy is leading to some major changes for these teachers.

CWRF/Bruner's Spiral Model. The CWRF model, when blended with Bruner's framework, maintains learner engagement by offering clear, achievable milestones. Each stage (crawl, walk, run, fly) aligns with Bruner's progression from concrete to abstract learning, providing continuous challenges and successes. This fosters a sense of accomplishment and motivates learners to advance through the stages, ultimately leading to a deeper and more sustained interest in the subject matter.

Of the three areas discussed in this research, this area was the most personal to me as a researcher. Once a new need was identified, and deemed critical to the success of the project, it allowed for the change to take place and the project to be able to continue. It also offered the teachers a chance to more completely work on what they wanted, how they wanted. Allowing for differentiated learning opportunities was a catalyst to affect positive change and give the teachers the impetus to complete this professional development journey!

Current Impact of Computer Science in Elementary

The impact of these findings for education today are significant for elementary computer science. In today's digital age, equipping elementary students with fundamental computer science (CS) skills has become absolutely essential. This knowledge forms the foundation for future learning and active (not just passive!) participation in a technology-driven world. However, a significant challenge lies in the fact that many elementary school teachers lack formal training in computer science, which hinders their ability to effectively teach these crucial skills. Empowering non-trained CS teachers with the necessary knowledge and resources to teach elementary CS can bridge this gap, ensuring that young students receive a comprehensive education that includes critical thinking, problem-solving, and digital literacy from an early age. But the teachers cannot teach what they do no. And many of the teachers are ready to learn – they want to know how to best prepare and support their students, and are willing to put the work in to learn what they need to know.

In preparing non-trained teachers to confidently teach computer science in K-6, we are building a more adaptable, future-ready education system. As we unmask these "CS heroes" in K-6 classrooms can inspire the next generation of innovators and problem-solvers, demonstrating that anyone can become proficient in technology and contribute to the digital world.

Implications for Future Research

The findings from this study underscore the need for future research in elementary computer science (CS) education to focus on the creation and evaluation of targeted professional development programs for teachers. These programs should aim to enhance both the self-efficacy and pedagogical skills necessary for effective CS instruction. It is essential to continue to explore which elements of professional development—such as hands-on workshops, ongoing mentorship, and collaborative learning communities—most significantly impact teacher confidence and competence in CS. Longitudinal studies could provide valuable insights into how sustained professional development influences instructional practices and student learning outcomes over time. By investigating these aspects, researchers can develop evidence-based strategies that support teachers in delivering high-quality CS education in elementary schools (Yadav, Hong, & Stephenson, 2016).

In addition to professional development, future research should examine the integration of CS concepts across various subjects in the elementary curriculum. Interdisciplinary approaches could reveal effective strategies for embedding CS education within traditional subjects like math, science, and literacy, thereby enriching the overall learning experience. Further, studies could investigate the long-term impact of early CS education on students' academic trajectories and career choices, providing insights into how early exposure can shape future proficiency and enthusiasm for the field (Grover & Pea, 2013). These research directions will contribute to a comprehensive understanding of how to optimize CS education in elementary schools, ensuring that students are well-prepared for a technology-driven future.

Conclusion

For almost 25 years, I have called one school district in the southern part of the United States and its surrounding community home. During that time, I have served in various capacities within the school system and the community. In the school system, I have served as a classroom teacher, teacher and student mentor, school (Title I) librarian and district administrator. I have also been very active in the community, participating in various civic, school, and religious organizations whose primary focus is to serve the students and families in my school district, often working alongside other district personnel, parents, and community members. I consider myself to be fairly well-known and respected for my strong work ethic, and for my commitment to creating a culture of inclusivity by seeking to engage with multiple community groups when gathering input on district and/or city initiatives. Both school and community leaders point to multiple honors and awards I have earned as evidence of my ethics and commitment to cultivating school-community connections.

I acknowledge that the primary participants in the study are district and community leaders, students, their guardians, and practicing teachers. And although I share leadership experiences with district and community leaders and teaching experiences with the practicing teachers, I do not share the experiences of a challenging upbringing as some of the students. Employment experiences have provided an insider's perspective into the inner workings of the school system, from the expectations of classroom teachers to a seat at the table where policy is decided upon. But viewed from the perspective of the student and their parent/guardian, these experiences place me in the role of an outsider.

I understand the critical need to address the outsider perspective issue from the outset in order to have the support of the home as this project unfolds. As a white, native English speaker in a majority-minority school district, I operate from the belief that the school earning the trust and respect of parents/guardians is vital to a student's overall success, particularly as it pertains to this research project. Research affirms that when working together, parents/guardians and schools can create the best possible environment to foster physical, emotional and intellectual well-being for students and that his happens most effectively when there is open, honest and reciprocal cultural exchanges between school and home (Wardlow, 2013; Eberly et al., 2007). This provides the opportunity for school personnel to gain insight into individual families and their cultures in an effort to see beyond their own biases, experiences and upbringing, and work with an open mind to respect practices that may conflict with their own core values.

For this project, the researcher I have committed to including an on-going reflection process that ensures both my voice and the primary participants' voices are proportionately represented in the project.

References

Angeli, C., Voogt, J., Fluck A., Webb, M., Cox, M., Malyn-Smith J., Zagami, J., (2016). A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. Journal of Educational Technology & Society, 19(3), 47–57. http://www.jstor.org/stable/jeductechsoci.19.3.47

Aspers, P., & Corte, U. (2019, February 27). What is qualitative in qualitative research - Qualitative Sociology. SpringerLink. Retrieved September 21, 2022, from https://link.springer.com/article/10.1007/s11133-019-9413-7

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. Psychological Review, 84(2), 191–215. Retrieved February 2, 2022, from https://psycnet.apa.org/doi/10.1037/0033-295X.84.2.191

Bandura, A., & Cervone, D. (1986). Differential engagement of self-reactive influences in cognitive motivation. Organizational Behavior and Human Decision Processes, 38, 92-113. Retrieved February 2, 2002, from https://doi.org/10.1016/0749-5978(86)90028-2

Bowen, G. (2009). Document Analysis as a Qualitative Research Method. Qualitative Research Journal. 9. 27-40. Retrieved November 4, 2022, from 10.3316/QRJ0902027.

Bruner, J. S. (1960). The Process of Education. Harvard University Press.

Bruner, J. S. (1977). The Culture of Education. Harvard University Press.

Carpenter, J., Rosenberg, J., Trust, T., & Dousay, T. A. (2020, June 17). What should teacher educators know about technology? perspectives and self-assessments. Teaching and Teacher Education. https://www.sciencedirect.com/science/article/pii/S0742051X19314908?via%3Dihub

Chai, C. S., Koh, J. H. L., & Tsai, C.-C. (2012, November 30). A review of technological pedagogical content knowledge. Educational Technology & Society. Retrieved March 1, 2023, from https://eric.ed.gov/?id=EJ1016563

Code.org. (2021). Texas Computer Science : State Advocacy Sheet. Support K-12 Computer Science Education in Texas. Retrieved October 14, 2021, from https://code.org/advocacy/state-facts/TX.

Code.org, CSTA, & ECEP Alliance. (2020). 2020 State of computer science education: Illuminating disparities. Retrieved October 14, 2022 from https://advocacy.code.org/stateofcs.

Creswell, J.W. (2009). Research Design: Qualitative, Quantitative, and Mixed-Methods Approaches (3rd ed). Thousand Oaks, CA: SAGE. Retrieved November 2, 2023 from https://www.ucg.ac.me/skladiste/blog_609332/objava_105202/fajlovi/Creswell.pdf.

Creswell, J.W. & Poth, S. (2013). Qualitative Inquiry & Research Design: Choosing among Five Approaches (3rd ed.). Thousand Oaks, CA: SAGE.

Crotty, M. (1998). Foundations of Social Research: Meaning and perspective in the research process (1st ed.). Routledge. https://doi.org/10.4324/9781003115700

CSTA. (2013). Bugs in the System: Computer Science Teacher Certification in the U.S. CSTA - certification resources. Retrieved March 1, 2022, from https://codes-isss.org/csta_subdomain/ComputerScienceTeacherCertification/sub/CertificationResources

Darling-Hammond, L., Hyler, M. E., & Gardner, M. (2017). Effective Teacher Professional Development. Learning Policy Institute.

Dennon, A. (2021, November 10). Biden's 2022 education budget prioritizes stem, students of Color. BestColleges.com. Retrieved November 24, 2021, from https://www.bestcolleges.com/blog/biden-2022-education-budget-stem-students-of-color/.

Fleming, J., & Zegwaard, K. E. (2018). Methodologies, methods and ethical considerations for conducting research in work-integrated learningERIC. ERIC, ed.gov. Retrieved May 7, 2022, from https://files.eric.ed.gov/fulltext/EJ1196755.pdf

Fluck, A., Webb, M., Cox, M., Charoula, A., Malyn-Smith, J., Voogt, J., & Zagami, J., (2016). Arguing for Computer Science in the School Curriculum. Journal of Educational Technology & Society, 19(3), 38–46. http://www.jstor.org/stable/jeductechsoci.19.3.38

Ford, J.D., & Ford, L.W. (2010). Stop blaming resistance to change and start using it. Organizational Dynamics, 39(1), 24-36.

Franke, B., Century, J., Lach, M., Wilson, C., Guzdial, M., Chapman, G., & Astrachan, O. (2013). Expanding access to K-12 computer science education: research on the landscape of computer science professional development. In Proceedings of the 44th ACM technical symposium on computer science education (pp. 541–542). Denver, CO: ACM.

Gal-Ezer, J., & Stephenson, C. (2010). Computer science teacher preparation is critical. ACM Inroads, 1(1), 61–66. doi:10.1145/1721933.1721953.

Fullan, M. G. (1992, November 30). Why teachers must become change agents. Educational Leadership. Retrieved October 18, 2022, from https://eric.ed.gov/?id=EJ459419

Gaetz, L., & Phadke, S. (2018). The Writer's World: Paragraphs and essays, with Enhanced Reading Strategies. Pearson.

Gallup Inc., (2020). Current perspectives and continuing challenges in computer science education in U.S. K-12 schools. Gallup.com. Retrieved October 1, 2022, from https://csedu.gallup.com/home.aspx

Google Inc. & Gallup Inc. (2015). Searching for Computer Science Report - Google Search. Current Perspectives and Continuing Challenges in Computer Science Education in U.S. K-12 Schools. Retrieved October 4, 2021, from https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf.

Google Inc. & Gallup Inc. (2017). Encouraging Students Toward Computer Science Learning. Results From the 2015-2016 Google-Gallup Study of Computer Science in U.S. K-12 Schools (Issue Brief No. 5). Retrieved October 4, 2021, from https://goo.gl/iM5g3A

Grant, C. and Osanloo, A. (2014). Understanding, selecting, and integrating a theoretical framework in dissertation research: creating the blueprint for your house. Administrative Issues Journal: Vol. 4: Iss. 2, Article 4. Retrieved January 27, 2022 from https://dc.swosu.edu/aij/vol4/iss2/4

Hagge, J. (2017, July). Scratching Beyond the Surface of Literacy: Programming for Early Adolescent Gifted Students. Gifted Child Today. Retrieved November 26, 2021, from https://www.proquest.com/docview/1912846664/fulltextPDF/A8703F27F07448C4PQ/1?accountid=11789.

Hatch, J.A. (2023). *Doing qualitative research in educational settings. Second edition*. SUNY Press.

Holmes, A. (2013). Direct Observation. In: Volkmar, F.R. (eds) Encyclopedia of Autism Spectrum Disorders. Springer, New York, NY. DOI: 10.1007/978-1-4419-1698-3_1758

Howard, N. R. (2018). Edtech leaders' beliefs: How are K-5 teachers supported with the integration of computer science in K-5 classrooms? Technology, Knowledge and Learning. Retrieved October 8, 2021, from https://link.springer.com/article/10.1007/s10758-018-9371-2.

Hubwieser, P., Magenheim, J., M€uhling, A., & Ruf, A. (2013). Towards a conceptualization of pedagogical content knowledge for computer science. In Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research (pp. 1–8). New York, NY: ACM. doi:10.1145/2493394.2493395

Janesick, V. J. (1999). A journal about journal writing as a qualitative research technique: History, issues, and reflections. Qualitative Inquiry, 5(4), 505-524. http://dx.doi.org/10.1177/107780049900500404

Kamii, C., Manning, M. M., & Manning, G. L. (1994). Early literacy: A constructivist foundation for whole language. NEA Professional Library, National Education Association.

Kim, C., Kim, D., Yuan, J., Hill, R. B., Doshi, P., & Thai, C. N. (2015). Robotics to promote elementary education preservice teachers' STEM engagement, learning, and teaching. Computers & Education, 91, 14–31. https://doi.org/10.1016/j.compedu.2015.08.005.

Koehler, M. J., & Mishra, P. (2009). What is technological pedagogical content knowledge? Contemporary Issues in Technology and Teacher Education, 9(1), 60-70.

Koehler, M., Mishra, P., & Yahya, K. (2005, December 27). Tracing the development of teacher knowledge in a design seminar: Integrating content, pedagogy and technology. Computers & Education. https://www.sciencedirect.com/science/article/pii/S0360131505001752?via%3Dihub

Kong, S. T. (2019). A Case Study on Elementary Teachers' Experiences Teaching Computer Science (Thesis, Concordia University, St. Paul). Retrieved September 28, 2022, from https://digitalcommons.csp.edu/cup_commons_grad_edd/316

Korstjens, I. & Moser, A. (2018). Series: Practical guidance to qualitative research. Part 4: Trustworthiness and publishing, European Journal of General Practice, 24:1, 120-124, DOI: 10.1080/13814788.2017.1375092

Kraemer-Holland, A. (2022, October). Chapter 3: Putting it all together. EDCI 991 Class Discussion. Manhattan; Kansas.

Kurt, S. (2019, September 16). TPACK: Technological Pedagogical Content Knowledge Framework. Educational Technology. https://educationaltechnology.net/technological-pedagogical-content-knowledge-tpack-framework/

Lichtman, M. (2013). Qualitative research in education: a user's guide (3rd ed.). Thousand Oaks, CA: SAGE Publications.

Liu, J., Hasson, E. P., Barnett, Z. D., & Zhang, P. (2011). A survey on computer science K-12 outreach: teacher training programs. In Proceedings of the 2011 frontiers in education conference (pp. T4F-1- 6). Aizu: ICIET.

Manches, A., and Plowman, L. (2017). Computing education in children's early years : A call for debate. British Journal of Educational Technology, 48(1), 191-201. https://doi.org/10.1111/bjet.12355

Maor, D. (2013) Does the use of the TPACK model enhance digital pedagogies: We don't understand the present so how can we imagine the future? In: 30th ASCILITE Conference: Electric Dreams, 1 - 4 December 2013, Macquarie University, Sydney

McGee, S., McGee-Tekula, R., Duck, J., Greenberg, R., Dettori, L., Reed, D., Wilkerson, B., Yanek, D., Rasmussen, A., & Chapman, G. (2017). Does A taste of computing increase computer science enrollment? IEEE Xplore. Retrieved January 27, 2022, from https://ieeexplore.ieee.org/document/7914599/

Menekse, M. (2015). Computer science teacher professional development in the United States: A review of studies published between 2004 and 2014. Computer Science Education, 25(4), 325–350. https://doi.org/10.1080/08993408.2015.1111645

Mishra, P., & Koehler, M. J. (2006, June). Technological Pedagogical Content Knowledge: A framework for teacher knowledge. Teachers College Record. Retrieved February 6, 2023, from https://www.researchgate.net/publication/220041541.

Motamedi, J., Leong, M., & Yoon, S. Y. (n.d.). Strategies for designing, implementing, and evaluating grow-your-own teacher program for educators. Retrieved February 2, 2023, from https://ies.ed.gov/ncee/edlabs/regions/northwest/pdf/strategies-for-educators.pdf

Office of the President of the United States. (2015). Progress report on coordinating federal science ... - archives. Obama White House Archives. Retrieved November 23, 2021, from https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/stem_ed_budget_supplement_fy16-march-2015.pdf.

Olofson, M., Meredith, J., Swallow, C., Naumann, M. (2016, January 15). TPACKing: A constructivist framing of TPACK to analyze teachers' construction of knowledge. Computers & Education. Retrieved March 2, 2023, from https://www.sciencedirect.com/science/article/pii/S0360131515301007

Ortlipp, M. (2008). Keeping and Using Reflective Journals in the Qualitative Research Process. The Qualitative Report, 13(4), 695-705. https://doi.org/10.46743/2160-3715/2008.1579

Pajares, F. (1996). Self-Efficacy Beliefs in Academic Settings. Review of Educational

Research, 66(4), 543–578. https://doi.org/10.3102/00346543066004543

Piaget, J. (1952). The Origins of Intelligence in Children. New York, NY: W.W. Norton

& Co. Retrieved March 12, 2022, from https://doi.org/10.1037/11494-000

Piaget, J. (1953). Genetic Psychology and Epistemology. Diogenes, 1(1), 49–63. https://doi.org/10.1177/039219215300100105PriceWaterhouseCooper. (2018, September 18). Are we preparing our kids for the jobs of tomorrow? Are We Preparing Our Kids for the Jobs of Tomorrow? | ASTRA. Retrieved October 14, 2021, from https://usinnovation.org/reports/technology-us-schools-are-we-preparing-our-kids-jobs-tomorrow.

Qian, Y., Hambrusch, S., Yadav, A., Gretter, S. (2018). Who Needs What: Recommendations for Designing Effective Online Professional Development for Computer Science Teachers, Journal of Research on Technology in Education, DOI:10.1080/15391523.2018.1433565

Ramrathan, L. & Mzimela, J., 2016, ‘Teaching reading in a multi-grade class: Teachers’ adaptive skills and teacher agency in teaching across grade R and grade 1‘, South African Journal of Childhood Education 6(2), a448. http:// dx.doi.org/10.4102/sajce. v6i2.44

Ravitch, S. & Riggans, M. (2017). Reason and Rigor: How Conceptual Frameworks Guide Research (2nd ed). Thousand Oaks, CA: SAGE Publications.

Rosenberg, J.P. and Yates, P.M. (2007), Schematic representation of case study research designs. Journal of Advanced Nursing, 60: 447-452. DOI: 10.1111/j.1365-2648.2007.04385.x

Saldaña Johnny. (2021). The coding manual for qualitative researchers + qualitative data analysis: A methods sourcebook (4th ed.). Thousand Oaks, CA: SAGE Publications.

Schanzer, E., Fisler, K., & Krishnamurti, S. (2018). Assessing Bootstrap:Algebra students on scaffolded and unscaffolded word problems. In Proceedings of the 49th ACM SIGCSE Technical Symposium on Computer Science Education (pp. 8–13). https://doi.org/10.1145/3159450.3159498

Sentance, S., & Csizmadia, A. (2017). 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). New York, NY: ACM.

Stake, Robert. (2011, January). Stake2003 Chapter 5: Case Studies. Scribd. Retrieved October 5, 2022, from https://www.scribd.com/document/487066305/Stake2003-CaseStudies

Stake, Robert E. (1995). The art of case study research. Thousand Oaks, CA: SAGE Publications

Stanton, J., Goldsmith, L., Adrion, W. R., Dunton, S., Hendrickson, K. A.,

Peterfreund, A., Zinth, J. D. (2017). State of the states landscape report: State level policies supporting equitable K–12 computer science education. Retrieved from Education Development Center website: https://www.edc.org/state-stateslandscape-report-state-level-policies-supporting-equitable-k-12-computerscience

Strauss, A., & Corbin, J. (1994). Grounded theory methodology: An overview. In N. Denzin & Y. Lincoln (Eds.), Handbook of qualitative research (pp. 273-285). Thousand Oaks, CA: SAGE Publications

Texas Education Agency. (2022). Title 19 Education Part 2 Texas Education Agency Chapter 126, Texas Essential Knowledge and Skills for Technology Applications, Subchapter A, Elementary. Texas Administrative Code. Retrieved November 1, 2022, from https://texreg.sos.state.tx.us/public/readtac$ext.ViewTAC?tac_view=5&ti=19&pt=2&ch=126&sch=A&rl=Y

Trump, D. J. (2017, September). Expanding Access to High Quality STEM and Computer Science Education Provides More Pathways to Good Jobs. Trump White House Archives. Retrieved November 24, 2021, from https://trumpwhitehouse.archives.gov/briefings-statements/expanding-access-high-quality-stem-computer-science-education-provides-pathways-good-jobs/.

U.S. Bureau of Labor Statistics. (2022, September 8). Employment in STEM occupations. U.S. Bureau of Labor Statistics. Retrieved March 12, 2022, from https://www.bls.gov/emp/tables/stem-employment.htm

U.S. Department of Education. (2022). NCES 2022 School Pulse Panel - Staffing. 2022 School Pulse Panel. Retrieved November 21, 2022, from https://ies.ed.gov/schoolsurvey/spp/

Webb, M. E., Fluck, A., Cox, M., Angeli-Valanides, C., Malyn-Smith, J., Voogt, J., et al. (2015). Thematic working group 9: curriculum - advancing understanding of the roles of computer science/informatics in the curriculum. In K.-W. Lai (Ed.), EDUsummIT 2015 summary report: Technology advance quality learning for all (pp. 6069). Thailand: Bangkok.

Yadav, A. (2017). Computer science teacher professional development: towards a research agenda on teacher thinking and learning. In Proceedings for the 12th workshop in primary and secondary computing education (pp. 1–2). Nijmegen: WiPSCE. DOI: https://doi.org/10.1145/3137065. 3137066 .

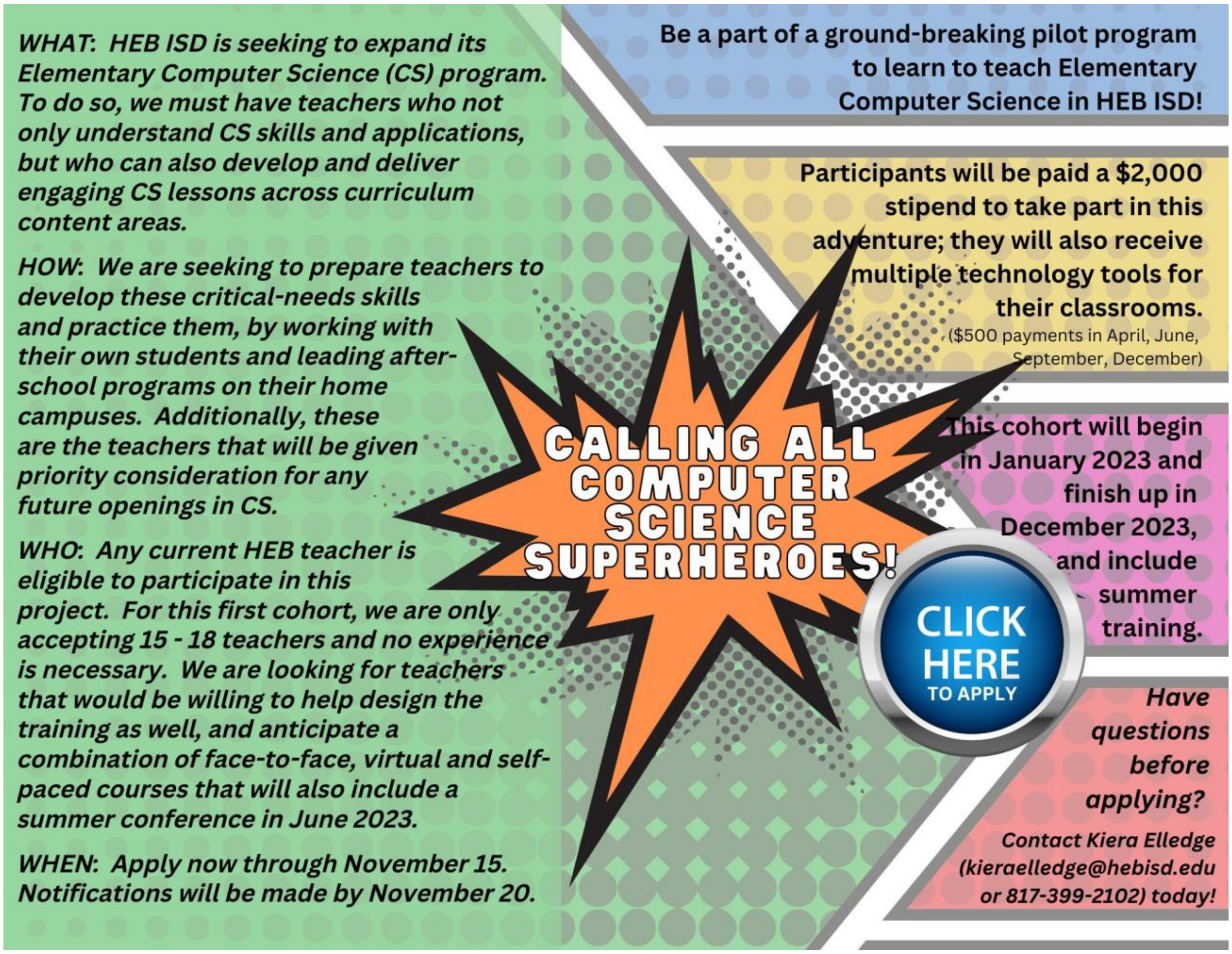
Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. Computer Science Education, 28(4), 371–400. https://doi.org/10.1080/08993408.2018.1560550

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. Tech Trends (60) (p. 565-568). Retrieved September 23, 2022, from https://link.springer.com/content/pdf/10.1007/s11528-016-0087-7.pdf

Yin, R.K. (2002) Applications of Case Study Research. Thousand Oaks, CA: SAGE Publications

Yin, Robert K. (2014). Case study research: Design and methods. Thousand Oaks, CA: SAGE Publications

Appendix A - Computer Science Recruitment Flyer



Appendix B - Participant Informed Consent (3 pages)

K ELLEDGE - DOCTORAL THESIS PROJECT
RESEARCH PARTICIPATION EXPLANATION AND CONSENT FORM

Your Name (please print): ____

Project Title: *Hiding in Plain Sight: Unmasking Computer Science Superheroes in K-6 Core Classrooms*

Principal Investigators
(Including contact info): Kiera Elledge - kselledge@ksu.edu, 817-980-2729
Dr. Suzanne Porath - sporath@ksu.edu, 920-858-3010

Introduction: Please carefully read the following information about the *Hiding in Plain Sight: Unmasking Computer Science (CS) Superheroes in K-6 Core Classrooms*, which is a research study of the effects of Computer Science professional development on non-CS trained teachers. This consent form includes explanations of:

- What is the purpose of the research?
- What are the research procedures?
- Will the research be recorded?
- What are the risks and benefits of participation?
- What are the costs of participation?
- Will you be compensated for the study?
- What about confidentiality?
- Will this research be shared with other researchers?
- What are your rights?
- Who do you call with questions or problems?

It is important for you to understand all the information in this form because it will tell you what the study is about and how it will be done. If you do not understand something in this form, please talk with Mrs. Elledge or Dr. Porath to answer your questions. Do not sign this consent form unless all your questions have been answered and you feel comfortable with the information you have read. You will be given a copy of the form to keep.

What is the purpose of the research study?
The purpose of this research is to gain an in-depth understanding of the experiences of a cohort of teachers who are learning through professional development and teaching computer science in elementary classrooms and how this experience impacts their ability to lead change within their school communities.

What is involved in the study (research procedures)?
The activities you are participating in for professional development will generate some artifacts that you will be asked to provide as part of the research. For example, you will also be asked to fill-out a survey at the beginning of and at the end of the program. Additionally, you will be expected to keep an online journal of your experiences and provide artifacts of your work upon request. Also you will be asked to schedule, at your own convenience, two 30-minute semi-structured interviews that will be conducted during the training at established intervals.

The questions on the survey will include demographic questions, attitudinal questions and questions about what you learned during the activities. The semi-structured interviews will include questions focused on assessing on-going learning, technology comfort as well as knowledge and pedagogy.

Will the research be recorded?
Yes. The survey is conducted through Qualtrics, which is a secure location. The interviews will be conducted via Zoom, which records both audio and video. But only the audio file will be downloaded, transcribed and stored on a password protected device and within a password protected document. The video, only recorded as a convenience, will not be downloaded and deleted immediately.
Artifacts and reflections will be conducted in a password-protected platform; if downloaded, will be stored on a password protected device and password protected document.
All data files may be stored for a maximum period of three years and then permanently deleted; only the researcher will have access to the information.

What are the Risks and Benefits of Participation?
There is minimal risk associated with this research study. To the best of my ability your answers in this study will remain confidential. A potential risk of the study is that it may be possible to identify you through the survey data, although the survey will not have your name on it and only the principal investigator will have the identifying data pieces.

There are no anticipated potential benefits for you in this research. However, it is hopeful that a potential benefit of the research is that it will help identify new teachers who will have the knowledge and pedagogy to teach computer science at the elementary level when the district expands the current program. The research will also hopefully lead to better ways to teach students computer science in core content classrooms.

What are the costs to the participants for taking part in the research?
There is no cost to you for taking part in this study.

Will you be compensated for participating in the research?
No, you will not be compensated for participating in this research study.

What about confidentiality?
You have a full right to privacy.

The results of this study may be published in a doctoral thesis, scientific book/or journal or presented to other people. If this is done, all names and identifying information will be anonymized. If you consent we may include quotes but pseudonyms will be used. Data from this research project will be kept in the locked office of Mrs. Elledge. Data kept on computers will be password-protected.

Will this research be shared with other researchers?
Participant information collected as part of the research, even if identifiers are removed, will not be used or distributed for future research studies.

What are your rights as a participant?
Taking part in this study is voluntary. You do not have to take part in this study and it is okay to refuse to sign this form. If you agree to take part and then change your mind, you can withdraw for any reason. Simply notify Mrs. Elledge by phone or in person. Deciding not to be in the study, or leaving the study early, will not result in any penalty from the school or other organizations nor will it impact your participation in the professional development cohort. If you decide to withdraw, all of your information will be destroyed.

Who do you contact if you have questions or concerns about this research?
If you have concerns or questions about the study or have a research-related injury, contact:

Kiera Elledge - kselledge@ksu.edu or 817-980-2729
Dr. Suzanne Porath - sporath@ksu.edu or 920-858-3010

If you have questions about your rights as a participant or feel you have been placed at risk, you may contact:

Dr. Lisa Rubin, Kansas State University IRB Chair
comply@k-state.edu

Confirmation of Consent by Research Participant:

Note to participant: When you sign this form, you are giving your permission to be in the study. By signing this form, you have not given up any of your legal rights or released anyone from liability for negligence, and you are at least 18 years of age.

The participant acknowledges that: Mrs. Elledge has explained to me the purpose of the research project, the study procedures, and the possible risks that may happen. I have read this consent form. I have been given a chance to ask questions about the research study and the procedures involved. I believe that I have enough information to make my decision. I have also been told my other options. I agree to the following as a participant in this research cohort:

- answer survey questions pertaining to this study
- answer semi-structured interview questions pertaining to this study
- keep an online journal of my experiences
- share reflections of teaching and/or artifacts pertaining to this study

______________________________ ________________________
Signature of Participant Date and Time

____
Printed Name of Participant

______________________________ ________________________
Signature of Researcher Printed Name of Researcher

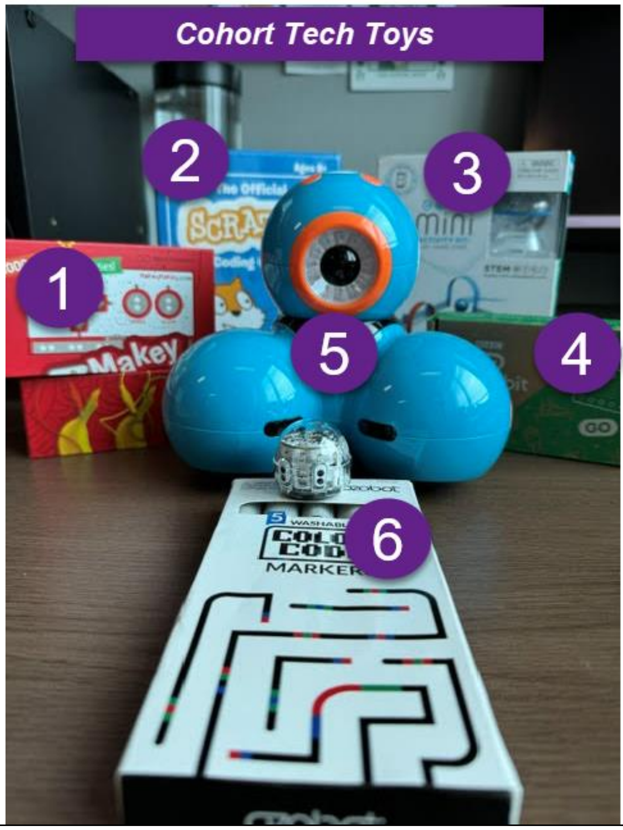
2

Appendix C - District Professional Development Plan

Researcher's Note: This training plan should be considered as a suggestion or recommendation only - dependent on the availability of materials, funding, alignment requirements, local and state computer science guidelines, etc. of the school or district that might attempt to replicate this study.

Title	***Objective(s)***	***Assignment(s)***	***Due Date***
In-Person Welcome Meeting (~2 hours)	• Overview of cohort expectations • Solicitation for research participation • Review of calendar dates for assignments	✓	January 16
Self-Paced Getting Started (~1 hour)	• Conduct group and individual needs assessments • Gather input on training preferences • Set up online journals	✓ Complete baseline surveys ✓ Complete training survey preferences ✓ Set up free account at blogger.com (see January 16 Action Items doc for specifics)	January 31
Video Instructions (~20 minutes)	Preview of upcoming training and assignments		
Self-Paced Personal Play Time (~3 hours)	• Spend some time (last entry of Appendix C) exploring the tools provided for this cohort • Reflect on your blog what you learned about each tool (Can do as individual posts or one long post, broken down in to readable chunks) • Reply to at least two of your colleagues' posts	✓ Reflection entries for each tool must include at least one photo or video and a two-paragraph (minimum) reflection of your learning ✓ Reply entries must be at least one paragraph in length	February 17
Zoom PLC (~1 hour)	Online PLC (optional)		
Video Instructions (~20 minutes)	Preview of upcoming training and assignments		
Self-Paced Intro to Online Resources (~8 hours)	• Explore three online resources that provide free computer science training to teachers and students • Work through modules on each platform	✓ Code.org – complete Teaching Computer Science Fundamentals Course ✓ Infosys/Pathfinders Institute – complete Intro to Computational Thinking, Building 21st-Century Skills through STEM **OR**	April 14

	• Document on your blog as indicated in the handout	Developing Inclusive CT Pathways, Micro:Bit Lessons for CS ✓ Scratch – complete at least three tutorials, select three coding cards and complete the tasks indicated	
Zoom PLC (~1 hour)	Online PLC (optional)		
In-Person Meeting (~6 hours)	• Scratch Week 2023 • Summer Conferences • CS in Public Schools Presentation • Tech-Tool One Pager Presentations	✓ Scratch – review resource guide, planning kit, virtual conference attendance ✓ Summer – WeTeach-CS (in-person) OR CSTA (virtual); look at classes and make a tentative group schedule so not everyone attends the same sessions ✓ CS in Public Schools – presentation by UT-Austin (John B. Owen and Amy Carrell) ✓ Presentations – which one(s) are you using in your class now; reply on blogs	May 6
Summer Conferences (~12 hours)	WeTeach_CS Summit (in-person) June 16-17; Houston, TX CSTA 2023 (virtual) July 11-13; online		
Face-to-Face Meeting (~2 hours)	• Report on conference sessions…everyone to present at least one new thing learned • Integrate 1 – 2 CS resources into a curriculum lesson • Create a video reflection of your personal Plan/Do/Study/Act	✓ Revise and teach one curricular mini-unit (2 – 3 days) to include at least 1 or 2 CS resources (online or physical) ✓ Post a video reflection on your blog ✓ Comment on at least two other blogs	September 29
Face-to-Face Meeting (~2 hours)	• Final group meeting, final project instructions and assignment details	✓ Revise and teach one curricular unit (7 – 10 days) to include at least 5 CS resources (online or physical) ✓ Upload lesson file and all ancillary materials to our Google drive ✓ Post a reflection on your blog	December 22

		✓ Comment on at least two other blogs	
Personal Playtime – Technology tools purchased for each member of the cohort 1 – Makey Makey Invention Kit 2 – Scratch Coding Challenge Cards 3 – Sphero Mini 4 – Micro:bit 2.0 5 – Dash Robot 6 – Ozobot w/marker set			

Appendix D – Interview Protocol Script (4 pages)

I. OPENING SCRIPT

Welcome ________________ and thank you for your participation today. My name is Kiera Elledge and I am a student at Kansas State University conducting a research project for my doctoral dissertation. This interview will take approximately 45 minutes and will focus on your experiences as a teacher participating in the school district's elementary computer science cohort. I will be inquiring about your personal journey to education, what you have learned about computer science in elementary thus far, and what you plan to do with your new learning this coming school year. And, all of the identifying information you provide will be kept confidential as has been explained previously. The aim of the study is to provide teachers from a non-computer science background with the training and resources necessary to integrate it into their core subject classrooms.

As a reminder, this information has also been covered in the participation consent form, which you were given a copy of and signed in the affirmative prior to this interview. It provided a brief introduction of the research study and answered the questions:

- Why is this study being done?
- How many people will take part in this study?
- What are the risks and benefits of participation in this study?
- What are my rights as a participant in this study?
- What about confidentiality of information in this study?
- Whom do I contact if I have questions or problems?

I would like your permission to video record this interview, so I may accurately document the information you convey. If at any time during the interview you wish to discontinue the use of the recorder or the interview itself, please feel free to let me know. If at any point you want me to turn off the recording or keep something you said off the record, just let me know. Do I have your permission to record this interview? ___Yes ___No

If yes: thank you! If no: Thank you for letting me know. I will only take notes of our conversation.

If any other questions arise during the interview, you can feel free to ask them at any time. I am more than happy to answer your questions. Please remember that your participation in this interview is completely voluntary. If at any time you need to stop or take a break, let me know. You may also withdraw your participation at any time without consequence. Before we begin the interview, do you have any questions? [Discuss questions]

We will now begin the interview.

II. INTERVIEW ROUND

You have been selected to be a part of this research project because you applied and were selected to be a part of the district's elementary computer science cohort of teachers. Let's begin with some questions about your educational background and life experiences, these will help me to get to know you better. Afterwards, we'll transition to questions about your teaching experiences in general, then move on to more specific questions in regards to elementary computer science.

GET-TO-KNOW-YOU – ROUND 1

You are currently a ______ teacher.
- How long have you been in the classroom?
    - What subjects have you taught during your career?
    - Did you have a favorite and why?

What brought you to teaching?
- Is there any other career choice you would have liked to have made?
    - What might that have looked like for you?

TRANSITIONAL – ROUND 2

Tell me about your current school and your classes. Include things like the physical location (urban, rural, etc.) number of students in the school, the race and ethnicity demographics (it doesn't have to be exact), the socioeconomics, and any other factors you'd like to share.
- Have these statistics changed since you started teaching these?
    - What examples could you share with me of how they have changed?

Q1: What new ideas have you gained thus far in your professional development training and how do you plan to implement these new ideas in your classroom?	**Q2: To what extent has this training impacted your belief in yourself to learn new content and pedagogy?**
•	• Could you provide me with one or two examples?
Q3: What additional resources or supports need to be put in place to ensure that computer science educational opportunities are accessible to all teachers and students?	**Q4: What concepts are you struggling with, either from the professional development or the implementation of the professional development?**
• What suggestions would you have to get more elementary schools interested in bringing computer science to their students?	• Are there any specific concepts you would like additional training on or like to see us add to our cohort training plan?

III. CLOSING SCRIPT

This concludes our interview. Before we finish, are there any final thoughts or questions about participating in this cohort that you would like to share or any additional questions about the research study that I can clarify for you?

Yes: Please share

No: Move on

Then again, I thank you for your time and willingness to be a part of this research project. I truly appreciate your candor and forthrightness in answering the interview questions. Have a nice day.