UNIVERSITY OF CALIFORNIA

Los Angeles

Computer Science for Middle School (CS4MS):
How Middle School Administrators and Teachers Implement Computer Science Curricula

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Education

by

Verjinia Mayer

2019

ABSTRACT OF THE DISSERTATION

Computer Science for Middle School (CS4MS): How Middle School Administrators and Teachers Implement Computer Science Curricula

by

Verjinia Mayer
Doctor of Education
University of California, Los Angeles, 2019
Professor Linda P. Rose, Co-Chair
Professor Linda J. Sax, Co-Chair

Computer science education in K-12 schools is a popular topic of study, especially in quantitative research. Many findings concur that computer science education should be introduced at lower grade levels to provide students with early exposure to computing skills or computational thinking more broadly. While there is substantial research about computer science education implementation at the elementary and high school levels. However, there is a paucity of qualitative research about computer science curriculum implementation at the middle school level.

The current case study concentrated on the computer science implementation stories of six teachers and six administrators at six middle schools in a large urban public school district with a large number of socioeconomically disadvantaged and traditionally underrepresented minority students. Through the theoretical framework of diffusion of innovation in organizations, the study investigated the essential elements of a computer science curriculum implementation, as well as the essential elements of sustaining a computer science education implementation. The study also focused on the challenges of computer science education implementation and how the challenges, if any, varied by student population. Lastly, the study sought data on solutions to implementation challenges and how solutions, if any, varied by student population.

Study findings show that when middle school teachers do not receive computer science curriculum training, their implementation of the curriculum does not reach the higher stages of diffusion, such as redefining, clarifying, and routinizing. Results also indicate that the middle school computer science teacher’s dedication to and enthusiasm for teaching the subject is essential to sustaining the implementation. Furthermore, when the administrator and teacher in charge of computer science education are both committed to the implementation, then they are more likely to exhibit a problem-solving mindset.

Challenges to implementing a middle school computer science curriculum include securing funding for training to teach the curriculum, in addition to finding enthusiastic teachers to teach the topic. A further challenge was the lack of teacher support from administrators who were not completely sold on any particular computer science curriculum or plan of implementation. Challenges pertaining to students include teachers not having any assistance with special education students who are not necessarily ready for a computer science classroom environment. Teachers also expressed challenges in terms of student behavior and students’ reluctance to work in teams, especially when there is not enough equipment for every student.

Solutions to the funding challenge that worked for participants of the study included continuous fundraising and grantwriting. Administrators who were more dedicated to implementing computer science education at their middle schools had all read research about the importance of early exposure to computer science opportunities as well as project-based learning curricula. Solutions related to special education students and others who were not ready for the advanced curriculum included having multiple levels of the work available for students to feel challenged at their level.

Teacher enthusiasm and ability to teach CS can be found among existing middle school teachers. Administrators who are informed about middle school CS education are in a better position to make teacher-centered and student-centered decisions about CS implementation. When administrators' visions of CS implementation align with the middle school teacher's vision of CS implementation, the social network that is generated by the administrator and CS teacher working collaboratively makes a CS program more sustainable. The alignment of middle school CS education research with existing research about diffusion of innovations makes these insights more accessible. Consequently, the implementation of middle school CS education can be better understood by using the five stages of diffusion of innovations as a theoretical framework.

The dissertation of Verjinia Mayer is approved.

Miryung Kim

William A. Sandoval

Linda P. Rose, Committee Co-Chair

Linda J. Sax, Committee Co-Chair

University of California, Los Angeles

2019

DEDICATION

I dedicate this dissertation to the four people who encouraged and inspired me to take on and finish this odyssey: to my mother, Mary, whose many sacrifices for my education are too many to list and too priceless to forget; to the memory of my late stepfather, Anghel, whose belief that I could pursue this degree gave me the confidence to do so; to my husband, Eric, whose love, patience, silly puns, and many printer purchases kept me going; and to our son, Joshua, who let me take him along for the journey from the beginning.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

Tables

Figures

LIST OF ABBREVIATIONS

Advanced Placement Computer Science A (AP CS A)

Advanced Placement Computer Science Principles (AP CSP)

Association for Computing Machinery (ACM)

Computer Science (CS)

Computer Science Education (CSE)

Computer Science STEM Network (CS2N)

Computer Science Teachers Association (CSTA)

Computing, Science, Technology, Engineering, and Mathematics (C-STEM)

United States Department of Education (DoED or ED)

Every Student Succeeds Act (ESSA)

Exploring Computer Science (ECS)

Information and Communications Technology (ICT)

Information Technology (IT)

Project GUTS (Growing Up Thinking Scientifically)

Science, Technology, Engineering, and Mathematics (STEM)

Science, Technology, Engineering, Arts, and Mathematics (STEAM)

Underrepresented Minorities (URMs)

VEX Educational Robotics (VEX EDR)

ACKNOWLEDGEMENTS

I was fortunate to learn from so many gifted professors in the Educational Leadership Program. My inspiring and tremendously accomplished co-chairs, Dr. Linda Sax and Dr. Linda Rose, guided me in developing a project that I could be passionate about and that enriched and transformed me as an educator. I was also privileged to have Dr. Miryung Kim and Dr. William Sandoval on my committee. Thank you to my entire committee for your time and support. Dr. Diane Durkin, Dr. Lillian Utsumi, and our dissertation writing group provided valuable insights and made me a better writer. Dr. Cindy Kratzer was an unforgettable role model of an organized, disciplined, and innovative educational leader. I am indebted to Dr. Stuart Biegel for his generosity and his captivating lessons on education law. I am grateful to Dr. Eugene Tucker and Dr. Richard Wagoner for broadening my learning with organizational theories and insights that helped me see the big picture in education. I appreciate Dr. Robert Cooper and Dr. Tyrone Howard for the social justice and educational equity narratives they provided that gave all this work an anchor. Thank you to Shan Boggs who kindly helped me navigate the numerous technical aspects of the process, such as forms, signatures, and deadlines. Shan, you are an ELP treasure.

To the amazing Cohort 21, thank you for all the starfish reminders and the many birthday and holiday celebrations, and for making this journey fun and unforgettable. Thank you to my friends who supported me along the way. Lisa, thank you for your calming stories and sage advice. Thanks to Taline for being there during monumental and challenging times. Thanks to Doug Dall for being my mentor and for entrusting me with opportunities to be an educational leader. Thanks to my brilliant and beautiful soul sister Mary for all the life lessons and guidance. And thanks to my kind and gifted brother Ara for the music, the road trips, the movies, and the

tech memories that united us. I am so lucky to be connected to you by love and laughter, no matter how many miles separate us. I wouldn’t be where I am without all of you.

This project would not have been possible without the generous participation of the district, principals, and computer science teachers who were a part of this study. Thank you for your time and for all that you shared about your experiences with computer science education. I learned so much from all of you. I am also grateful to all the computer science professional development providers I was fortunate to meet and learn from as a result of this project: Dr. Steve Cooper (Adventures in Alice), Dr. Emmanuel Schanzer (Bootstrap), Steven Zanotti (Code.org), and so many others at the CUE, CSTA, and CSPD Week conferences. I could not have become the CS educator I am today without your expertise.

I would also like to thank my current principal, Dr. Tom Crowther, for being so supportive and nurturing. Thank you for your guidance and for giving me a playground where I could grow and help to excite students to learn. Thanks to my colleagues at Toll for making me a part of the family. Finally, I am fortunate to have the most brilliant students who make teaching a pleasure. We get to learn from playing, tinkering, and designing. It is a privilege to watch you create like young inventors. I learn to be a better teacher from each of you every day.

VITA

1996	B.S. Mathematics and Philosophy University of California Los Angeles, California
1998-2001	Mathematics Teacher Roosevelt Middle School Glendale, California
2000	Single Subject Teaching Credential Mathematics with CLAD Emphasis National University Sherman Oaks, California
2002-2003	Master of Education National University Sherman Oaks, California
2001-2010	Mathematics Teacher Clark Magnet High School La Crescenta, California
2004-2009	Beginning Teacher Support and Assessment (BTSA) Support Provider Glendale Unified School District Glendale, California
2010-2016	Teacher Specialist in Educational Technology Educational Technology and Information Services Glendale Unified School District Glendale, California
2015-present	Google for Education Certified Trainer
2016-present	VEX IQ Certified Robotics Instructor
2016-present	Robotics, 3D Design and Printing, and Applied Technology Teacher Toll Middle School Glendale, California
2018	Designated Subjects Career Technical Education Teaching Credential Information and Communication Technologies Commission on Teacher Credentialing Sacramento, California

CHAPTER 1: INTRODUCTION

“Perhaps in the past it was possible to undereducate a significant portion of the population without causing serious harm to the nation. No longer. Education, today more than at any time in the past, is the key to successful participation in society” (Ravitch, 2001).

Statement of the Problem

After years of eluding female and underrepresented minority students in grades K-12, computer science education (CSE) is at a critical juncture in education history wherein educators understand the need to prepare all children to learn advanced computing in high school and are willing to implement CSE in their schools. CSE curriculum reform has reached a tipping point, with many schools using free online resources or paying for computer science curricula while struggling to find teachers willing to teach computer science in grades K-12. Despite the lack of research on effective CSE implementation in the traditional K-12 classroom, early adopters (i.e., school administrators and teachers) of relatively new CS curricula are yielding to the pressures of producing enough students with computer science knowledge to help meet the needs of the future job market. To compound their early efforts, school administrators and teachers have the added pressure of achieving ethnic and gender equity in CSE, because recent data reveal that the current technology workforce is predominantly White and male-dominated. Due to a lack of early opportunities to learn foundational computer science concepts, traditionally underrepresented student populations have been less prepared to learn advanced computer science in high school and, consequently, less likely to pursue computer science majors and careers. Reinforcing the importance of early CSE opportunities and the need for gender and ethnic diversity in CSE, the United States Department of Education (DOE) endorsed computer science in the 2015 Every Student Succeeds Act (ESSA) as part of a “well-rounded education”. However, despite all the recent hype about the importance of teaching computer science to all K-

12 students, the feasibility and cost of long-term CSE implementation have yet to be well documented.

Research supports the need to start computer science interventions as early as possible, and specifically in middle school, before students' perceptions of gender and career roles are solidified (Barker, Snow, Garvin-Doxas, & Weston, 2006; Whitecraft & Williams, 2010). Traditionally, the majority of public school students in America have not been exposed to any computer science curriculum prior to high school (Google & Gallup, 2015a, 2015b; Kaczmarczyk & Dopplick, 2014). As a result, the vast majority of K-12 students, particularly the underrepresented populations, lack the essential computer science exposure and socialization that would help them thrive in the field—namely, available credit-earning courses, a computer science social network, parental support, and teacher support (Margolis, Estrella, Goode, Holme, & Nao, 2008). Absent an easy-to-access computer science pathway, most underrepresented female and minority students choose career pathways outside the technical fields (i.e., science, technology, engineering, and mathematics) (Denner, 2011; Lightbody, Siann, Tait, & Walsh, 1997; Trigg & Perlman, 1976). Margolis et al. (2008) found that the numerous barriers to computer science access (including lack of trained computer science teachers and little prior student exposure to rigorous computer science curriculum) result in more homogeneous and male-oriented high school computer science courses (Lee, 2015; Rodger et al., 2014; Ryoo et al., 2012). "Chilly climate/classroom," "brain drain," and "leaky pipeline" more accurately describe female and minority students' K-12 experiences with computer science (Eccles, 1989; Frenkel, 1990; Goode, 2007; Hilton & Lee, 1988). Yet, in reality computer science is intimately linked to science, technology, engineering, and mathematics (STEM).

Computer science is defined as “the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society” (CSTA, 2009). The high school Advanced Placement Computer Science A (AP CS A) course, historically not offered in many high schools, is an example of a rigorous computer science course that exposes students to complex problem solving and computational thinking through computer programming (Cuny, 2012). Descriptions of computer science may seem dry and boring to many students, but various K-12 computer science programs have seen success in motivating students to pursue higher levels of computer science. The most notable of these improvements in student access to and engagement in CS education is the new AP Computer Science Principles (CSP) course introduced in high schools in 2016. A variety of AP CSP curricula have been designed to make the study of computing more appealing to traditionally underrepresented students. AP CSP courses’ purpose is to raise high school students’ interest in computing through the use of topics relevant to high school students’ lives (College Board, 2017).

Nonetheless, despite their high interest and academic achievement in mathematics and science throughout their elementary and middle school education, many girls and underrepresented students do not pursue more rigorous STEM courses like AP Computer Science in high school or major in computer science in college (CSTA, 2006, p. 72; Glenn, 2000; Goode, 2007; Sax et al., 2017; Stout, Dasgupta, Hunsinger, & McManus, 2011). Consequently, the rate of openings in STEM careers worldwide is growing at a higher rate than female and minority STEM college graduates (Lemons & Parzinger, 2007; Ma, 2011; Stout et al., 2011; Hill, Corbett, & St. Rose, 2010; Papastergiou, 2008; Rosenbloom, Ash, Dupont, & Coder, 2008; Whitecraft & Williams, 2010). The billions of dollars spent on filling elementary and middle schools with technology and Internet access will not have as great an impact on our future economy if more and diverse students do not immerse themselves in rigorous computer courses in high school (Cuny, 2012; Margolis & Fisher, 2002; Stephens, Jelenac, & Noack, 2010). While access to AP Computer Science courses in high school is important, students also need access and exposure to formal (in-school) and informal (after-school) computer science curriculum throughout their K-12 education, not just at the tail end of their secondary schooling (Denner, 2011; Denner, Martinez, Thiry, & Adams, 2014; Goode, 2007; Google, 2014; Google & Gallup, 2015a, 2015b; Ryoo, Margolis, Lee, Sandoval, & Goode, 2012; Wilson, Sudol, Stephenson, & Stehlik, 2010).

The College Board provides some of the most detailed statistics about the status of computer science education in the United States. College Board AP CS A and CSP exam results from 2018 indicate an upward trend for Black/African American, Hispanic/Latino, and female students (College Board, 2019). While these data show a rise in some minority AP CS A and AP CSP test takers and passers, the overall ratios of Black to White and Hispanic to White students taking the exams remain low—1 to 12 and 1 to 4 for AP CS A and 1 to 6 and 1 to 2 for AP CSP, respectively. Similarly, male test takers outnumber female test takers on the AP CS A exam by a ratio of 3 to 1 and on the AP CSP exam by a ratio of 2 to 1. While these data are encouraging, the lack of parity between minority and female test-takers and White and male test-takers indicates a need for CS education interventions to begin earlier than high school. As computer science education policy reformers continue to gain momentum, teacher and administrator preparation for implementing the new curriculum at the middle school level will become more and more important.

Research has established that science, technology, engineering, and mathematics—collectively known as "STEM"—have traditionally been male-dominated subjects (Dryburgh, 2000; Anderson, Lankshear, Timms, & Courtney, 2008; Cheryan, Plaut, Handron, & Hudson, 2013). Girls and minorities are underrepresented in certain STEM subjects, STEM college majors, and STEM careers despite their interest and academic achievement in these subjects in the earlier years of schooling (Cheryan et al., 2013; Tan, Barton, Kang, & O'Neill, 2013). For example, girls outnumber boys in biology and environmental science, however boys heavily outnumber girls in computer science and physical science (Anderson et al., 2008; College Board, 2014c; Diekman, Brown, Johnston, & Clark, 2010; Rosenbloom et al., 2008). The historical gender and racial imbalances in computer science education have allowed negative stereotypes to solidify wherein females and minorities are perceived as less naturally able to study computer science. Studies confirm the access barriers girls and minorities face in studying computer science posed by stereotype threat (Cheryan et al., 2013; Hill et al., 2010; Rosenbloom et al., 2008; Tan et al., 2013). Stereotype threat is a fear that one who belongs to a group will confirm the truth of negative stereotypes about his or her group if he or she performs poorly on an important task. The said group is stereotypically considered weaker in such tasks (Aronson, Fried, & Good, 2002; Sanders, 2005).

In addition to a lack of access to computer science curriculum, another common thread that runs through most K-12 computer science research is the lack of role models and mentors available to female and minority students (Aronson et al., 2002; Reinen & Plomp, 1993, 1997; Frenkel, 1990; Simard & Gammal, 2012; Hill et al., 2010). Programs that bring successful women computer scientists to K-12 classrooms as mentors have been effective in recruiting more girls to the field (Finson, 2009; Minogue, 2010). Girls report that field trips to see computer scientists at work have had a positive impact on how they perceive women in computer science (Bamberger, 2014). However, Bamberger (2014) also discovered that when the knowledge of the women mentors is far too advanced for the female mentees to identify with, the mentees make a vital distinction—group-as-target and self-as-target stereotype threats—"we (women) can do it, however *I* cannot" (Shapiro, Williams, & Hambarchyan, 2013). Research has shown that the extent of a female mentor's computer skills makes a difference in the impact she can make on middle school female students (Reinen & Plomp, 1993, 1997). Betz and Sekaquaptewa (2012) argue that mentors need to be less stereotypical of a computer science student, and female mentees need to be able to identify with their female mentors and their computing skills (as cited in Cheryan, Drury, & Vichayapai, 2012, p. 76). Absent a robust pipeline of available CS mentors, the pressure lies on CS teachers to establishing mentoring relationships with their students and encourage greater student interest in the field.

Urgency for Proficiency in Computer Science in the United States

National labor statistics paint a bleak picture of the long-term effects of not recruiting more females and underrepresented minorities into computer science courses. According to the May 2014 United States Bureau of Labor Statistics' report on Women in the Labor Force, of the roughly 67 million women who are employed in the civilian labor force, a mere 1.5%—approximately 1 million women—are working in computer and mathematical occupations. These women represent only one-third of the entire computer and mathematical occupations work force. The Labor Bureau also reports that currently the top three highest paying computer science occupations are software developer, computer network architect, and information security analyst. Figures 1a, 1b, and 1c below show that by the year 2024, the Labor Bureau predicts that these three top-paying computer science careers will grow at higher rates than the national average for all other occupations, with information software application developer jobs predicted to grow by approximately 19%.

Yet, today, women and underrepresented minorities represent disproportionately low percentages of those employed in these three technical and high skill-based professions (see Table 1). Major tech corporations, who have released their diversity statistics annually since 2013, confirm what can happen if access to computer science does not become a priority in public education soon (see Figure 2).

Table 1

Examples of CS-Related Professions Where Women and Minorities Are Underrepresented (U.S. Department of Labor, 2015a, 2015b, 2015c)

Profession	**Women**	**Black or African American**	**Hispanic or Latino**
Software Developers	17.9%	5.0%	5.4%
Computer Network Architects	12.1%	8.9%	6.5%
Information Security Analysts	19.7%	3.0%	5.2%

In the interest of "serving humanity" (Diekman et al., 2010) and as a matter of national economic security, achieving greater social equity in computer science will help fill the hundreds of thousands of computer science jobs that will dominate the work force of the near future. The Labor Bureau states, "analysts will be needed to create innovative solutions to prevent hackers from stealing critical information or causing problems for computer networks" (U.S. Department of Labor, 2015c, www.bls.gov). If the top-paying computer science occupations are predicted to grow from 12% to 19% by the year 2024, an additional 300,000 or so high-paying and high-skilled technology jobs will be available in less than ten years (Goode, 2007; Lee, 2015; Millar & Jagger, 2001; Rodger et al., 2014). However, the rate of girls and underrepresented minorities pursuing these careers is not growing at a comparable pace, which may result in the United States outsourcing computer science jobs (CSTA, 2006).

The nation is looking to K-12 public education to ameliorate the paltry diversity statistics in computing professions (Byars-Winston, 2014). District and site administrators, counselors, and classroom teachers can be enlisted to help girls and underrepresented minorities understand that in less than ten years, confidential activities of American citizens' lives will be outsourced to qualified computer scientists in other countries (Byars-Winston, 2014; Charles & Bradley, 2006; Zarrett & Malanchuk, 2005; Dryburgh, 2000; Hess & Miura, 1985). Female and minority students have expressed greater interest in studying computer science when computing is taught as a means to achieving greater social justice and benefiting society (Denner et al., 2014; Melguizo & Wolniak, 2012; Ryoo et al., 2012).

Barriers and Traditional Solutions

Several reasons explaining the low numbers of girls and minority students in advanced computer science courses have been offered in the past. Some researchers have posited that girls would rather study subjects other than computer science, because girls see other subjects as more direct pathways to making a difference in the world (Diekman et al., 2010; Stout et al., 2011). In addition, girls have often complained of boredom and irrelevance in their computer science coursework—not seeing the real-life applications of programming and other aspects of computer science (AAUW, 2000; Anderson et al., 2008; Carbonaro, Szafron, Cutumisu, & Schaeffer, 2010; Diekman et al., 2010; Margolis et al., 2008). Trivialized curricula in K-12 technology courses, such as cutting and pasting or desktop publishing, lead students to being underprepared to pursue rigorous computer science coursework in college. Research shows that improving computer science curriculum and cross-curricular integration of computer science in core subjects (i.e., science, English, social sciences, etc.) can help girls and minorities see the real-life relevance of computer science (Carruthers, Milford, Pelton, & Stege, 2010; Rodger et al., 2014).

Researchers who have conducted studies in racial and gender diversity in careers report that despite evidence that women and minorities truly prefer other careers over computer science, it is possible that this is a result of reverse causation—that feeling unwelcome in the profession is what causes women's and minorities' lower interest in computer science (Byars-Winston, 2014; Rosenbloom et al., 2007). Ryoo et al. (2012) and Byars-Winston (2014) showed that counter-narratives provided by inquiry-based, culturally relevant computer science programs helped increase women's and underrepresented minorities' interest in computer science. Notwithstanding, a challenge to providing more inquiry-based, culturally relevant computer science education programs is a lack of experienced computer science teachers (Carruthers et al., 2010). For example, the CS 10K Project, funded by the National Science Foundation (NSF), focused on creating a sequence of computing classes to prepare students for AP Computer Science at 10,000 high schools by the year 2015. CS 10K, now known as the CS for All Teachers Community, however, also planned to find 10,000 well-prepared educators who would teach these pathway courses—a herculean feat considering that most K-12 computer science teachers are self-taught techies and have never had formal computer science training (Margolis et al., 2008).

Due to the impending shortage of computer scientists, computer science education with an emphasis on increasing the number of traditionally underrepresented minorities and females is a burgeoning topic in educational research. The National Science Foundation (NSF) has funded dozens of programs that focus on increasing the number of underrepresented students taking computer science courses (NSF, 2006). One such program is the Computer Science Computing and Mentoring Partnership (CS-CAMP) led by mathematician Richard Tapia at Rice University in Texas. Another example with a long waitlist is the Technology Education And Literacy in Schools (TEALS) program backed by Microsoft, which sends high-tech professionals to partner high schools to teach students advanced computer science courses that many schools do not offer (www.tealsk12.org). Notwithstanding the expansion of CS outreach programs, much of what is written on the topic of CSE consists of quantitative studies evaluating a specific CS curriculum and how that curriculum raised students' interests in CS. Curriculum implementers' stories about challenges and successes of offering CSE are missing, leaving a gap in the research that can be filled by more studies that share administrators' and CS teachers' narratives about what it means to teach CS.

Learning from Administrators and Teachers Implementing CSE

In general, middle schools are not preparing students to take AP Computer Science in high school. The lack of middle school computer science courses under educates students in advanced computing, and by the time most students reach high school they are not interested in learning a difficult computing language for which they see little relevance to real life. Despite years of research on the topic of non-traditional (i.e., after-school or out-of-school) CS interventions, female and ethnic underrepresentation in advanced high school computer science courses remains an unsolved issue.

There is a gap in the literature about how CS programs are implemented in traditional K-12 public classrooms, specifically at the middle school level. The phrase foundational computer science curriculum is used to refer to middle school coursework, including introductory computer programming, that prepares students for learning computer programming in Java if they choose to take AP Computer Science Principles or AP Computer Science A in high school (see Appendix J for sample middle school curricula). Research has shown that learning Java without any prior exposure to computer programming results in greater student attrition along the computer science pipeline. To study how CS programs are implemented, this study will focus on the following three research questions.

Research Questions

This study will attempt to fill the gap in middle school CSE implementation research by investigating the following research questions:

1. According to middle school administrators and computer science teachers in a district with a large number of socioeconomically disadvantaged (SED) and traditionally underrepresented minority students (URM), what are the essential elements (i.e., teacher, technology, funding, etc.) necessary to offer computer science courses to its middle school students?
    a. What are the essential elements necessary to sustain a computer science education program?
2. According to middle school administrators and computer science teachers in a district with a large number of SED and traditionally URM students, what are the challenges they faced while implementing a computer science curriculum in their schools?
    a. How did the challenges vary, if at all, by student population (i.e., URM, SED, special education students, female students, male students, etc.)?
    b. According to middle school administrators and computer science teachers, how has the school and/or district addressed the challenges?
3. What supports and resources do middle school administrators and computer science teachers say they need to help them integrate computer science into the middle school curriculum?

a. How do the supports and/or resources vary, if at all, by student population (i.e., URM, SED, special education students, female students, male students, etc.)?

Through a qualitative multi-site comparative case study, I investigated how administrators and teachers at six middle schools in California provided access to computer science education to their middle school students. Through this study, I interviewed six administrators and six teachers about CSE implementation at their sites. Districts preparing to offer CSE in grades K-8 can learn from early implementers about how to offer a foundational computing curriculum that allows students to construct their own identities, without imposing on them how to fit into gender and/or ethnic roles (Cassell & Jenkins, 2000). In the process of focusing on students' reactions to foundational computer science curricula, teachers' and administrators' voices have not been heard in the research. Little has been published about the experiences and perspectives of administrators and teachers who are leading computer science interventions in the United States. The few studies that have included teachers' perspectives about teaching CS—many of which are either literature reviews or quantitative surveys—have come from outside the United States (Black et al., 2013; Hubweiser, Armoni, Giannakos, & Mittermeir, 2014). Therefore, this study focused on the stories shared by 12 interviewees at six middle schools that have implemented a computer science curriculum within the past three years. Data was collected through CS curriculum analysis and telephone interviews with each school principal (or administrator with the greatest knowledge about the site's CSE implementation) and the computer science instructor at each site.

Methodology

Sites and Access

In the Fall and Winter of the 2015-16 school year, I spoke to CS curriculum writers, CS professional development providers, and administrators planning to offer CS at the middle school level throughout Southern and Northern California. Many of the administrators I spoke to were planning to begin implementing middle school computer science courses for the first time in the 2016-17 school year. One large urban district in California was further along in the middle school CS implementation process than any other district I contacted. At least 50% of the students at each school were underrepresented minority students, and at least 60% of the students at each site were socio-economically disadvantaged (see Table 3).

Data Collection and Analysis

Interviews. I conducted in-depth semi-structured interviews with administrators and computer science teachers at each of the six middle schools in Sunrise Unified School District (SUSD). Interviews elicited the types of challenges and successes that administrators and teachers faced in implementing a computer science curriculum at the middle school level.

Document Analysis. I analyzed the computer science curricula implemented in the district in order to help with understanding the interview data. I also looked at other materials and online resources that the administrators and computer science teachers said they used to facilitate the learning of CS concepts.

Public Engagement and Dissemination

The findings of my multi-site case study were shared with stakeholders at SUSD and at each of the participating middle schools. The findings were also shared with stakeholders in my own district, in order to guide my district in moving forward with computer science courses at the middle school level. In the future, I plan to share key findings of my study nationally, if possible, through publications in educational technology journals, as well as through presentations at educational technology conferences.

CHAPTER 2: REVIEW OF THE LITERATURE

Introduction and Roadmap

We have learned that there is a constant need to build and nurture advocacy networks within the school, district, and state level. And, this is often best accomplished with local people who know the district and state culture, have "skin in the game," and are truly committed to equity within a particular district (Margolis, Goode, & Chapman, 2015).

As indicated by the data in chapter one, the United States currently cannot keep up with the demand for workers with computer science knowledge. Most US public schools are not teaching computer science—an essential 21st century skill students will need when they enter the work force (Aesaert, Vanderlinde, Tondeur, & van Braak, 2013; Rodger et al., 2014). Some have even been calling computer programming "the new literacy" (Prensky, 2008). As teachers in the US grapple with the challenges of teaching the traditional language literacy through the Common Core State Standards (CCSS), teachers in other countries (i.e., United Kingdom (2012), New Zealand (2012), Russia (1970s), France (1960s), Sweden (1960s)) have been implementing computer science curricula in their classrooms for decades (Hubweiser et al., 2014; Sentance & Csizmadia, 2015). The US Department of Education made computer science education a priority when President Barack Obama signed the Every Student Succeeds Act (ESSA) into law on December 10, 2015. ESSA is the reauthorization of the 1965 Elementary and Secondary Education Act (ESEA), and it specifies that federal money is available to districts that teach computer science as part of the "well-rounded subjects" (see Appendix I). However, the road to proper implementation of computer science education in K-12 schools is long and paved with challenges. While the ESSA's mention of computer science implies progress in mainstreaming computer science education as a STEM subject, it may take years from its 2016-17 implementation year to see successful CS implementations. In the meantime, the experiences of administrators and teachers who have already implemented CSE in their schools will be valuable to districts and schools looking to implement CSE in the near future.

My conceptual framework was based on Everett Rogers' (2003) theory of diffusion of innovation and how Rogers' theory could be implemented to achieve greater computer science access for traditionally underrepresented groups. Diffusion of computer science curriculum into mainstream K-12 public education with gender and ethnic equity in mind involves a heavy reliance on social systems and collaboration—two basic tenets of Rogers' (2003) diffusion of innovations theory. Understanding how educators leverage their social networks and leadership roles to adopt and implement CS curricula will help accelerate the diffusion of CSE in the entire country. The innovation is the computer science curriculum, and the diffusion is the spread of CS implementation ideas as they apply to middle school CS courses through social communication channels (Rogers, 2003).

Before discussing the CSE literature, I define the key terminology used in CSE research with an explanation of why Rogers' (2003) theory of diffusion of innovations applies to middle school CS implementation. Next, I provide background data illustrating the need to expand CSE access to promote diversity in CS in general and increase diversity in middle school CS courses in particular. Then, I summarize several calls to action that lay out evidence of the lack of middle school access to CSE. Finally, I provide examples of successful CS reform efforts, despite persistent barriers to CS implementation. I conclude my literature review with a summary that emphasizes the need for more studies documenting the CSE implementation experiences of middle school administrators and teachers in the United States.

Terminology

Throughout this literature review I use the terminology and abbreviations routinely seen in research about computer science education (CSE). The terms STEM (Science, Technology, Engineering, and Mathematics) and STEAM (Science, Technology, Engineering, Art, and Mathematics) are often used in research involving computer science education, especially about CSE as it relates to robotics and engineering courses. Information technology (IT) and computer science (CS) are sometimes used interchangeably to refer to a general preparation for a variety of careers in technology. CSE literature from countries other than the United States usually refers to CS and CSE as ICT or Information and Communication Technology (Falkner, Vivian, & Falkner, 2014; Kori, Pedaste, Leijen, & Tõnisson, 2014). The 2007 College Board AP Computer Science Teacher's Guides, defines computer science as:

> The systematic study of computing systems and computation… [and] theories for understanding computing systems and methods; design methodology, algorithms, and tools; methods for the testing of concepts; methods of analysis and verification; and knowledge representation and implementation (NCO-HPCC, 1996, p. 116).

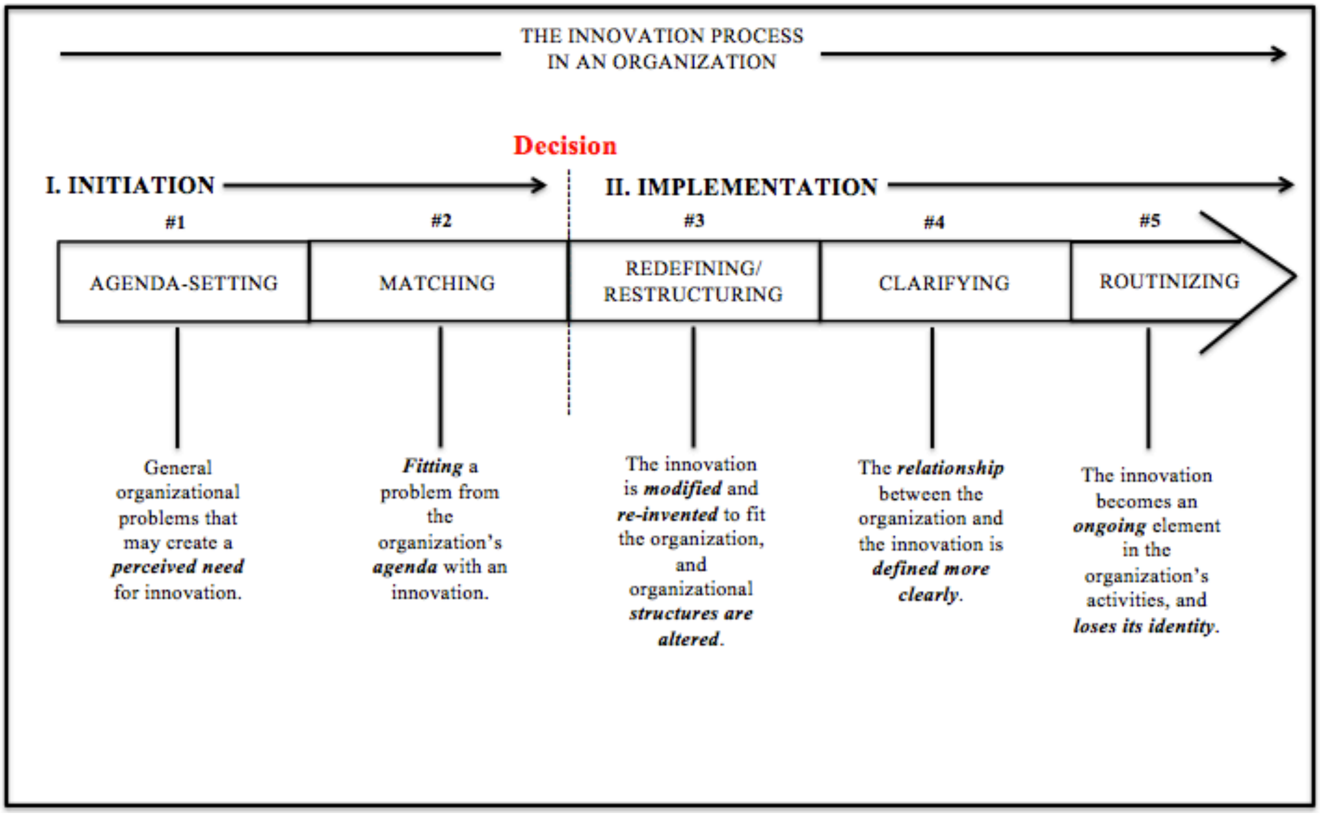
When I refer to the technical College Board definition of CS in this literature review, I also mean that CS is the gateway to advanced careers such as software developer, computer network architect, and information security analyst (U.S. Department of Labor, 2015a, 2015b, 2015c). As mentioned earlier, these careers are multiplying at a much faster rate than the rate of students, particularly women and underrepresented minorities, entering these professions.

Two other terms often used synonymously in CS literature include "coding" and "computer programming" (Duncan, Bell, & Tanimoto, 2014). While they have varying meanings, coding refers to writing basic computer instructions and programming refers to the more comprehensive task of designing a computer program and debugging the program.

Theoretical Framework

An analysis of curricular implementation practices would not be complete without a grounding theory. Its connection to a theoretical framework makes the implementation analysis more replicable and the implementation itself less fleeting (Strang & Meyer, 1993). Israel, Pearson, Tapia, Wherfel, and Reese (2015) suggest two possible frameworks to study CSE implementations: implementation science theory (Fixsen, Naoom, Blase, Friedman, & Wallace, 2005; Harn, Parisi, & Stoolmiller, 2013; Odom, 2009) and diffusion of innovations theory (Norris, Sullivan, Poirot, & Soloway, 2003; Rogers, 2003). Sometimes used to imply diffusion, implementation science is defined as the “scientific study of methods to promote the systematic uptake of research findings and other evidence-based practices into routine practice, and, hence, to improve the quality and effectiveness of health services” (Eccles & Mittman, 2006). On a less technical note, “diffusion connotes the socially mediated spread of some practice within a population” (Strang & Meyer, 1993, p. 487). In addition, implementation science is a relatively new framework for analyzing the adoption of practices, therefore it is a theory with limited applications, thus far. On the other hand, diffusion of innovations theory has been used for almost a century to analyze the spread of such innovations as kindergarten and driver training in public schools (Rogers, 2003). Furthermore, implementation science relies heavily on Rogers’ (2003) theory of diffusion of innovations, yet implementation science has a more complex and nebulous topography (Fixsen et al., 2005) than diffusion’s five basic stages of innovation (see Figure 3). Due to the newness of computer science education implementation in middle schools at this time, it is more appropriate to use a time-tested framework like diffusion of innovations theory that has been shown through a variety of applications to be a generalizable theoretical framework. Once middle school computer science courses become more mainstream, then improvement efforts on established CSE implementation practices may benefit from the application of an implementation science framework. I also chose to analyze CSE implementation through a diffusion lens rather than an implementation science lens, because of diffusion theory's foundation in social networks. "[S]chools are fundamentally social organizations", especially when it comes to implementing computer technologies (Frank, Zhao, & Borman, 2004). Surprisingly, many CSE studies describe a variety of social aspects of CSE interventions without attributing any social theories to the implementation process. Without a unifying social theory, these various CSE studies remain isolated instances of CSE intervention.

Figure 1. Five Stages in the Innovation Process in Organizations (Rogers, 2003)



In addition to the importance of social channels, Strang and Meyer (1993) further emphasize that widespread adoption of an innovation depends on how "compelling [the innovation is] to relevant audiences" (p. 494). CSE suffers from not yet being a compelling enough curriculum reform to more district- and site-level administrators and teachers. In other words, the slow diffusion of CSE can partly be explained by the fact that for a long time the most vocal CSE reform advocates have been the "highly specialized and marginal" (p. 495) populations—namely, CS teachers, CS professors and researchers, and CS professional organizations.

Finally, identifying and overcoming the challenges of CSE implementation also requires a look at K-12 institutional barriers. Institutional challenges to CSE implementation include maintaining student engagement, providing CSE opportunities for all, funding for curricula, for resources, and for teacher preparation that promote rigor, recruitment, and diversity, and changing traditional images of CS among administrators, parents, and society at large (Black et al., 2013). Educators facing this complex mesh of challenges would benefit from a unifying theory that helps them make sense of the CSE implementation trajectory.

To date, few studies apply the diffusion of innovations theory to K-12 technology integration, and none have applied diffusion of innovations to K-12 computer science education reform. In one of its earliest applications to education research, Carlson (1965) used a diffusion of innovation model to explain how several popular superintendents' "interpersonal networks" helped spread the adoption of a new math program in Pittsburgh (as cited in Rogers, 2003, p. 61). Since Carlson (1965), hundreds of education studies have shown how diffusion in schools follows Rogers' S-shaped adopter distribution curve. The left tail of the S begins at the early stages of adoption. Over time and through the social process of diffusion, the S-curve "takes off" to reach a peak number of adopters (p. 272). Rogers (2003) also developed a bell curve upon which to map different levels of adopters. He wrote, "*Adopter categories*, the classifications of members of a social system on the basis of innovativeness, include: (1) innovators [venturesome], (2) early adopters [respect], (3) early majority [deliberate], (4) late majority [skeptical], and (5) laggards [traditional]" (pp. 20, 283).

Figure 1 above illustrates how diffusion of an innovation can be broken down into two phases: initiation (pre-decision) and implementation (post-decision) (Rogers, 2003). During the pre-decision initiation phase, which is comprised of two stages—agenda-setting and matching—the organization identifies a problem in need of innovation and the problem is matched to the innovation. Once the problem and innovation have been aligned to each other, the decision to implement is made and the next three stages—redefining/restructuring, clarifying, and routinizing—make up the implementation phase of the innovation process. During the three implementation stages, the organization first adapts the innovation to its needs and adapts itself appropriately to implement the innovation. Then, the organization can more clearly define its relationship to the innovation. In the final stage of the implementation, the innovation has become so organically woven into the organization's fibers that the innovation "loses its identity" (Rogers, 2003, p. 421).

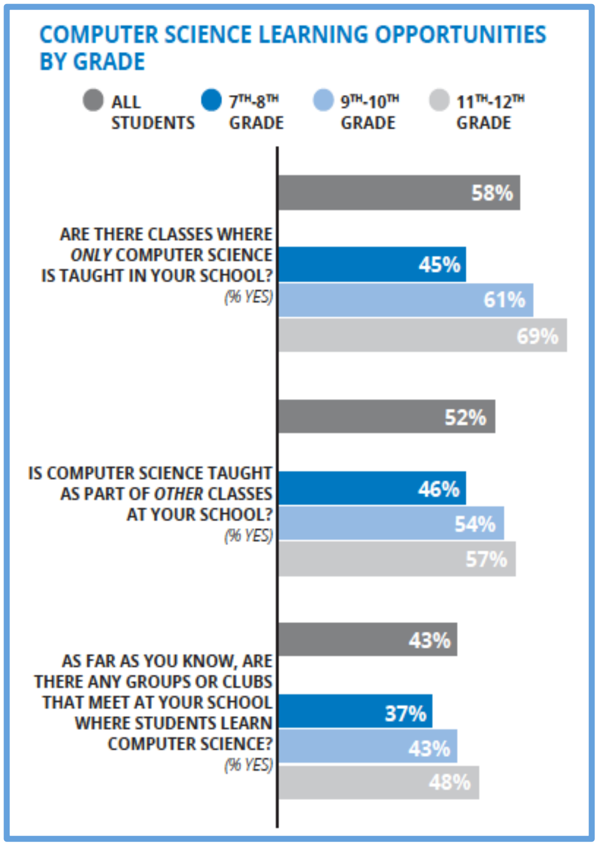
A detailed and large-scale example of a CSE study that follows the diffusion of innovations model is the six-year Georgia Computes! (GaComputes) research project (Guzdial, Ericson, McKlin, & Engelman, 2014). At the agenda-setting phase, the state of Georgia realized a need for innovation in CS teacher professional development when the AP CS A curriculum changed from the C++ programming language to Java. Two major goals of the Georgia Computes! initiative were to increase the number of high schools offering AP CS and to increase the number of female and underrepresented minority students taking AP CS. In the matching phase, the researchers participated in the development of new CS course standards for Java and in finding universities that would offer the certification necessary to increase the number of CS teachers who could teach the AP CS A course in Java. The matching phase proved to be complicated, because of the unpredictable, inconsistent, and shifting nature of state policy. The redefining/restructuring phase involved the design of CS teacher professional development based on data gathered from teacher observations and interviews. During the clarifying phase in Year Four, the Georgia Computes! initiative hired a program evaluation group to help determine how the CS professional development impacted teacher practices and student learning in AP CS courses. Finally, at the routinizing phase, GaComputes helped develop CS summer camps for students in grades 4-12, developed strategic partnerships to help deliver robotics kits to schools, and had developed an extensive library of CS professional development resources for teachers. Even though the GaComputes researchers did not deliberately map their six-year study to Rogers' five stages of innovation, doing so makes the Guzdial et al. (2014) study easier to compare to other states' large-scale CS intervention initiatives.

Rogers' five-stage innovation implementation model aptly applies to K-12 CSE research that has found how linking CSE to mainstream skills, such as writing and storytelling, can improve both computing skills and English literacy skills in middle school (Burke, 2012; Rodger et al., 2014). Even as they write about the importance of advisory adoption committees in schools, many of these innovative CSE researchers and staunch K-12 CSE reform advocates fail to apply a social model, such as Rogers' diffusion of innovations theory, to help future researchers evaluate which of the five stages of diffusion pose the greatest challenges for administrators and teachers involved in K-12 CSE implementation (Anthony & Patravanich, 2014). For these reasons, K-12 CSE reform research would benefit from the innovation initiation and implementation stories told by middle school administrators and teachers who have already implemented CS curricula in their schools.

Why Access to Middle School CS is a Social Justice Issue

Whether or not they study the most sought-after fields in college or not is largely determined by students' foundational preparation in grades K-12, which is driven by the choices of administrators and teachers (Strang & Meyer, 1993). Public K-12 districts in the United States offer college preparatory courses in mathematics, science, English language arts, and social science. Most of these topics are introduced in elementary school and get more and more advanced by the time students enter high school. However, when it comes to learning computing languages (those which control what a computer does), a skill that many researchers today call essential (Bernier & Margolis, 2014; CSTA, 2010), students' first encounter with programming a computer is with AP Computer Science A in high school—an elective course that sometimes does not count towards the University of California's A-G graduation requirements (CSTA, 2012a). Not receiving college credit for taking a difficult computer science course is a disincentive for high school students to pursue CS studies. Furthermore, many students who take the AP Computer Science A course without prior CS preparation struggle to learn Java—a difficult first computing language for absolute beginners. Many students do not pass the AP CS A exam for which the course is supposed to prepare them. With such a

Figure 2. Google & Gallup (2015a)



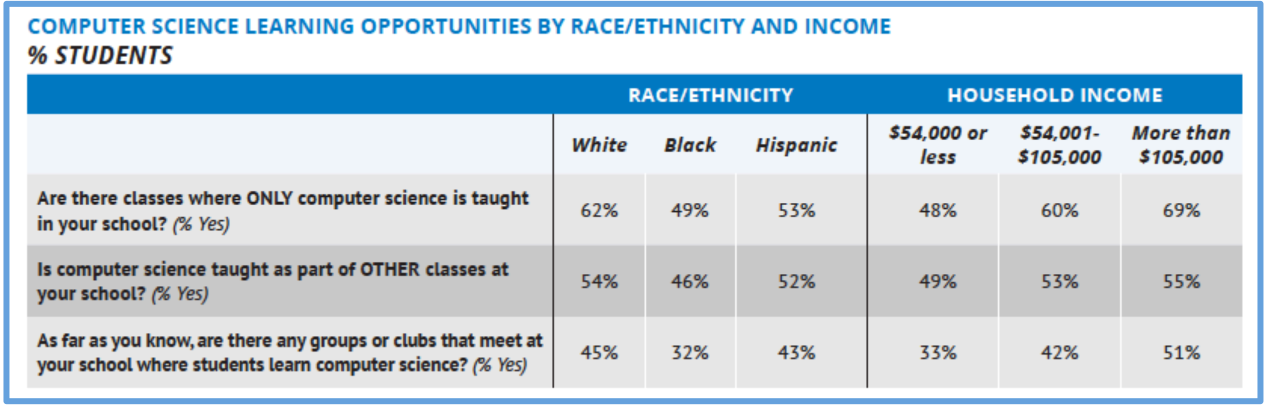
disconnect between the importance of CSE and how it is first introduced to most students, students' adverse reactions that the traditional high school CS curriculum is boring and irrelevant are not surprising (Carbonaro et al., 2010). This downward trend in high school computer science has been stemmed by the 2016 introduction of the AP CSP course which has higher AP exam passage rates than the traditional AP CS A exam (College Board, 2018).

Based on National Center for Education Statistics (NCES) graduation data, the percentage of overall high school graduates who earned computer science credits dropped from 25% to 19% from 1990 to 2009 (Nord et al., 2011). In 2009, 14% of high school computer science credit earners were girls—10% lower than male CS credit earners. Meanwhile, the rate of high school students taking advanced mathematics courses continues to rise. In fact, advanced mathematics courses demonstrate the gender parity that is lacking in computer science. Seventy-eight percent of high school graduates who earned Algebra II credit in 2009 were girls. Algebra II is a prerequisite to taking a high school computer science course (Cooper & Weaver, 2003), yet by the time they can choose to take an elective computer science course, girls and minority students have opted to take other classes instead.

While clear and consistent high school mathematics course pathways exist at most public high schools, computer science course offerings are rare (Carbonaro et al., 2010; Google & Gallup, 2015a). At the middle school level, computer science courses nationwide are even less common (Figure 2). The College Board, which administers and reports data on Advanced Placement (AP) examinations, indicated in their 2018 summary report that nationwide only 3,751 secondary schools out of over 20,000 (or less than 20%) of America's high schools reported offering an AP Computer Science course, compared to over 6,000 schools that offered AP Biology, AP Chemistry, and the more traditional AP English and mathematics courses.

Granted that the number of schools offering AP CS has increased by 53% from 2010, when only 2,457 were offering the course, is cause for some optimism (College Board, 2010, 2014a). However, Margolis et al. (2008) found that the majority of high schools in the Los Angeles Unified School District that serve socio-economically disadvantaged youths cannot afford to offer AP CS, and that during economically challenging times administrators cut computer science offerings first. More recent data in Figure 3 below show that minority and socioeconomically disadvantaged students are less likely than White students to have access to computer science courses (Google & Gallup, 2015a). In order to bring more underrepresented students into the CS pipeline, starting in the 2016-17 school year school were able to offer AP CSP classes in high school, assuming administrators could find teachers qualified and willing to teach this new introductory CS course. Meanwhile, researchers have shown that valuable recruitment efforts to bring more underrepresented groups to CS are needed in middle school, when many students who have not yet decided their career trajectory need to be shown the pathway to taking CS in high school (Barker et al., 2006; Whitecraft & Williams, 2010).

Figure 3. Google & Gallup (2015a)



COMPUTER SCIENCE LEARNING OPPORTUNITIES BY RACE/ETHNICITY AND INCOME

% STUDENTS

	RACE/ETHNICITY			HOUSEHOLD INCOME		
	White	*Black*	*Hispanic*	*$54,000 or less*	*$54,001-$105,000*	*More than $105,000*
Are there classes where ONLY computer science is taught in your school? *(% Yes)*	62%	49%	53%	48%	60%	69%
Is computer science taught as part of OTHER classes at your school? *(% Yes)*	54%	46%	52%	49%	53%	55%
As far as you know, are there any groups or clubs that meet at your school where students learn computer science? *(% Yes)*	45%	32%	43%	33%	42%	51%

Data support the need to recruit students to CS earlier than high school. Over the ten-year period from 2008 to 2018, the increase in AP CS A test takers increased 319%. Although College Board (2018) data show a recent upsurge in computer science course offerings in high school, female and minority students continue to be underrepresented in these courses (College Board, 2018). On a positive note, female AP CSP test takers rose by 183% from 2016 to 2018, African American AP CSP test takers increased by 256%, and Hispanic AP CSP test takers increased by 229% over the same period (College Board, 2018). While these data indicate that the increase in CSE opportunities can have a positive effect on female and traditionally underrepresented minority students' participation in CS, achievement results remain troubling. Unfortunately, female, African American, and Hispanic students' confidence levels have not kept up with the increased national average, and these underrepresented students' passage rates (i.e., receiving a score of 3 or higher on an AP exam) in 2018 were far below the national average passing rate on the AP CS A and AP CSP exams (College Board, 2018). Of all the 2018 AP CS A exam takers, only 2% were African American students who scored a 3 or higher, or passed the exam. Only 8% of all 2018 AP CS A examinees were Hispanic students who passed. On the 2018 AP CSP exam, only 4% of test takers who passed were African American and 15% of those who passed were Hispanic. On the other hand, even though females represent 36% of AP CS A and 44% of AP CSP test takers, 65% passed the AP CS A exam and 68.5% passed the AP CSP exam. These data indicate that CS interventions are more effective with female students than they have been with traditionally underrepresented minority students.

Across the 50 states, a variety of organizations have provided and continue to provide pockets of computer science exposure to workshop and camp participants. However, most public school districts in the United States have failed to make computer science—an essential 21st century skill—a part of the standard K-12 curriculum. Districts' delay in providing access to computer science education for all students in grades K through 12 is a matter of social inequity and economic injustice (Parker & Guzdial, 2015). Researchers have found that a lack of access to computer science education is directly correlated with the socioeconomic makeup of a district's student population (Google & Gallup, 2015a). Wealthy parents have the resources to provide their children with computer science opportunities in elementary, middle, and high school. Meanwhile, minority Latina/o and African-American students, as well as females, miss out on CS courses and experiences that equip students from more affluent neighborhoods with foundational computer programming skills. Researchers have pondered the semantics of calling the imbalance in CS access either a matter of inequality or a matter of injustice. The following explanation can help CS intervention providers determine whether their CS program provides equal access and/or equal results:

> Equality is privilege agnostic and implies giving every student equal opportunities no matter where they start. Justice is privilege sensitive and involves giving some students more opportunities that others based on how disadvantaged the student might be. Issues of equality and justice make us ask: what goals do we design our interventions for? Do we want to mitigate privilege to the point that we reach 'equality of input school resources or equality of results of schooling'? Which approach benefits the student more? (Parker & Guzdial, 2015, p. 4).

While the semantic debate may have no immediate resolution, it is evident that CS intervention providers need to be mindful of what students have access to CS and how those students perform in their CS classes. School districts that offer advanced computer science courses strictly in their more affluent schools help sustain unwanted socioeconomic inequalities by limiting CS exposure for girls and minorities in lower SES schools. Surprisingly, teachers' and administrators' voices, particularly those at schools with significant numbers of socioeconomically disadvantaged students of color, are largely missing from the computer science education literature. As a result, it is difficult for districts eager to implement computer science education to replicate the CSE reform efforts that have been successful in the trailblazing districts already teaching CS (i.e., New York, San Francisco, Iowa).

Despite volumes of research attempting to explain why the leaky STEM pipeline starts with 78% girls in Algebra II and ends with only 14% girls in computer science in high school (Barker & Aspray, 2006; Margolis et al., 2008; Moses, Howe, & Niesz, 1999), recent university statistics show a declining trend in female advanced computing enrollees. The 2013 Taulbee Survey, administered by the Computing Research Association (CRA) since 1974 to collect data on "enrollment, production, and employment of Ph.D.s in" computer science (CS), computer engineering (CE), and information (I) (Zweben & Bizot, 2014), reported an all-time high in computing degrees as well as a rise in male computing graduates, who earned 82% of all doctoral computing degrees compared to 18% that were earned by women—a decline from 19.2% in 2012 (Zweben & Bizot, 2014). Since it began disaggregating bachelor's, master's, and doctoral degrees by gender in 1994 through the 2007-2008 school year, the Taulbee Survey has shown a 6% drop in female computing bachelor's degree earners and a 4% and 8% rise in female master's and doctoral degree earners, respectively (Zweben, 2014).

In 2006, scientific philosopher Thomas Kuhn articulated the paradigm shift happening in science thanks to computing science (Carbonaro et al., 2010). Kuhn emphasized the fact that science does not happen in "isolation" and that in order for science research to be "relevant to society" it must incorporate the use of computing science (Carbonaro et al., 2010; Cohoon & Aspray, 2006). As a result of the ubiquity of computing in solving society's problems, female and minority underrepresentation in the physical sciences can no longer be overlooked.

Benefits of working in a computer science career include versatility of job options (Eubanks, 2011). Individuals with computer science knowledge will be in demand, because machines need programmers to function. Design creativity is also in high demand, as CS can be applied to solve complex problems in all scientific areas (U.S. Department of Labor, 2015a, 2015b, 2015c). Carbonaro et al. (2010) argue that scientific innovation depends on improving CSE and increasing the number of students interested in pursuing CS majors and careers. Considering the intimate link between other STEM subjects and computer science, some K-12 schools have realized that to make their computer science offerings more appealing to underrepresented groups, they need to publicize that CS learning can be applied to solving social issues—a reason often cited for why many girls gravitate towards the life sciences (Diekman et al., 2010; Emmott & Rison, 2006; Goode, 2010).

Increasing female and minority participation in computer science to promote innovation and diversity in perspectives would be a benefit to society (Bartol & Aspray, 2006; Cheryan et al., 2013). Of personal economic significance to females and underrepresented minority students should be the increased earning potential offered by a computer science career (Carruthers et al., 2010; Cheryan et al., 2013; Melguizo & Wolniak, 2012; Simard & Gammal, 2012). Through her action research study, Eubanks (2011) witnessed the social injustice in under-educating females and minorities in computer science. After conducting four years of participatory action research in a YWCA, Eubanks found that women need a more powerful voice in high-tech computing, and that women without access to advanced computing knowledge tend to stay in low-tech jobs (Eubanks, 2011).

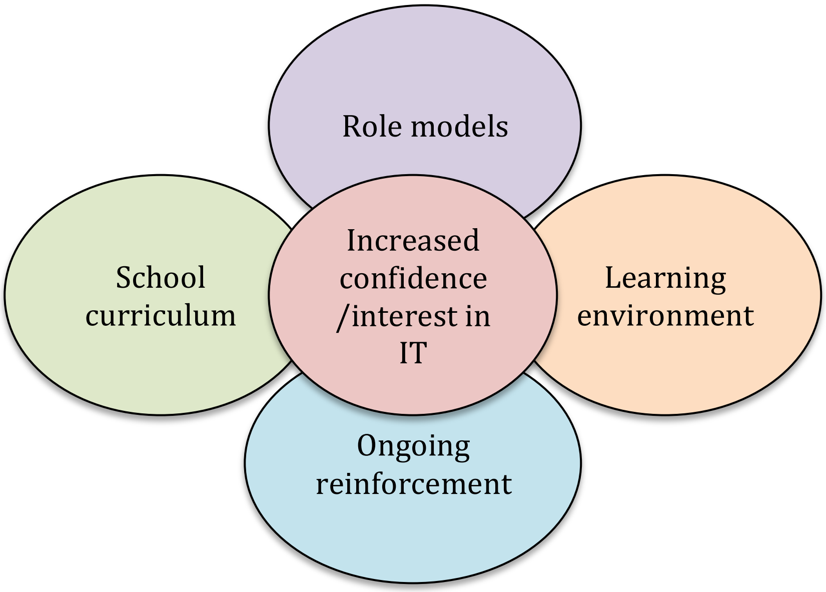
Computer Science Education Calls to Action

Early warnings about the shrinking computer science pipeline were published over 29 years ago. In a November 1990 communication of the Association for Computing Machinery (ACM), for example, computer scientist Karen Frenkel referred to an “academic pipeline” that is not producing enough post-secondary computer science graduates. While recognizing the need for more research to determine if the shrinking pipeline uniquely applies to computer science, Frenkel (1990) argued that the shortage of female computer science advanced degree holders results in a shortage of qualified computer science teachers and, consequently, a lack of computer science courses and curriculum for middle and high school students. Frenkel (1990) went on to say that the "chaos" surrounding computer science in middle schools and high schools is further exacerbated by the cumulative effects of tracking, as well as a lack of "role models, parental engagement, science-related opportunities, … [and access to] computers" (p. 39), of female and other under-represented student populations.

Close to three decades later, the bleak outlook for secondary computer science education exposed by Frenkel (1990) as negatively affecting underrepresented populations in computer science remains a persistent issue (Google & Gallup, 2015a; Margolis & Fisher, 2002; Margolis et al., 2008). However, as mentioned earlier in this chapter, the most vocal advocacy for CSE reform has thus far been limited to CS researchers and CS professional organizations. In 2006, the Computer Science Teachers Association (CSTA) published a report indicating that school administrators and education policy leaders did not understand the value of a computer science education or the urgency of improving high school computer science curriculum. Further research resulted in another CSTA (2010) call to action. Fueled by Margolis et al.'s (2008) research, the 2010 CSTA call to action claimed that a "lack of access to K-12 computer science education, or 'privileged knowledge,' is what education researchers have described as a significant social justice issue for the 21st century" (CSTA, 2010, p. 9).

The CSTA's latest research and call to action (CSTA, 2012b), commissioned by a group aptly named ACCESS (Alliance for California Computing Education for Students and Schools),

Figure 4. Student Barriers to CS Access (Fisher et al., 2015)



reports some progress in increasing high school CS course offerings. However, the latest CSTA (2012b) report reaffirms that a uniform and consistent middle school CS curriculum is still missing from the nation's schools. The report blames difficulties in CS teacher credentialing, not awarding graduation credits to CS courses, and a failure to allocate proper funding to CS for students being continually denied access to a computer science education (CSTA, 2012b).

Evidence of Successful CSE Reform Despite Persistent Barriers

In addition to the barriers cited by the CSTA in its calls to action, underrepresented students face other challenges in CSE. Figure 4 shows how Fisher, Lang, Craig, and Forgasz (2015) summarize often-cited CS barriers faced by girls and minorities. Current research about CSE reform can be summarized with the following four themes, which I will explain in detail below: 1) access to CSE is a matter of social justice and needs to address gender and ethnic equity; 2) rigorous CS curricula need to involve experiential learning; 3) CS teachers need regular professional development (PD); and 4) grassroots CS teacher networks and ongoing coaching will help CSE be a sustainable and permanent part of the overall middle school curriculum. Despite the many cited barriers, however, a multitude of interventions have shown positive results in recruiting more underrepresented students to CS and improving students' espoused image of CS.

Promoting an Image of a Diverse and Supportive CS Landscape

Much of the recent CS research about broadening access to CS opportunities for all students has focused on after-school interventions and the use of mentors to change adolescents' negative perceptions about CS. Black et al. (2013) conducted a questionnaire evaluation of their secondary CS initiative *cs4fn* in the United Kingdom (UK)—a CS magazine. They also gathered qualitative data from 115 teachers about how to motivate students to take CS. Teachers in the Black et al. (2013) study confirm the importance of after-school computing clubs to broaden CS access for elementary students or provide enrichment for older students, in addition to further diffusing the CS curriculum. Other researchers echo Black et al.'s (2013) views about the challenges of expanding CS access by emphasizing the importance of convincing local and federal education agencies of the need for CSE expansion, as well as the need to promote a CS image of "programming and rigorous analytical thinking, with a consistent set of core principles" (learning to produce software) rather than simple technology literacy (learning to use software) (Brown et al., 2013, p. 274). According to Brown, Sentance, Crick, and Humphreys (2014), computing education in the UK has benefitted from three changes to their 1988 national curriculum, which was updated in the years 2012-2014: 1) CS is now compulsory for students ages 5-16, CS is now a part of the English Baccalaureate, and the new term "Computing" has replaced the old term "ICT". Unfortunately, US K-12 CS curriculum reform policy lags behind, and until CS is made a part of the traditional K-12 course landscape, akin to English language arts and mathematics, few other reform efforts to increase underrepresented groups in CS will have lasting effects (Goode, 2010). Black et al. (2013) also document the need for improving

CSE’s image with administrators, parents, and society at large—groups of people who in most cases need convincing about the importance of CS in the middle grades. Research-based methods of summer workshop curriculum development rely on findings that girls participate in CS when they enjoy the computing environment and are encouraged by their teachers, parents, and peers to pursue IT careers (Zarrett, Malanchuk, Davis-Kean, & Eccles, 2006).

According to recent studies on recruitment efforts to increase the number of girls taking computer science courses in high school, teacher trainings on how to recruit girls in computer science have been very successful (Cohoon, Cohoon, & Soffa, 2011). Furthermore, since advanced mathematics courses serve as a reliable indicator of success in computer science, one can easily identify students with potential to succeed in a computer science course using math grades and math test scores. However, students cannot be expected to know what courses are best for them without proper guidance.

> One of our more disturbing findings was that some of the African American and Latino/a students we interviewed actively chose to stay in regular-tracked classes because they thought that the high grades they were earning would give them an advantage in terms of college competitiveness. What was not conveyed to many of these students was the weight that college admissions officials place on honors and AP classes. And in the absence of mentors or a network of peers to tell them otherwise, these students’ misguided notions persisted (Margolis et al., 2008, p. 92).

In addition to receiving academic guidance from their teachers and counselors, students also benefit from mentoring relationships. Research supports the need for a supportive CS classroom environment, and much of the research confirms that the CS teacher or mentor plays a key role in providing that support. Middle school girls need guidance to navigate the intimidating waters in their computer science classes (Simard & Gammal, 2012). Moreover, middle school girls’ stories about their interests in science need to be heard and recognized by those around them in order for the middle schoolers to feel confident about pursuing their science interests (Tan et al., 2013). Tan et al. (2013) also argued for the need to reconcile girls' "narrated identities—who I think I am and want to be" with their "embodied identities—what I do" (p. 1170). Support from club teachers, who Tan et al. (2013) call "authority figures," provides "positive reinforcement" for middle school girls who are at a critical and vulnerable age for deciding their future career pathways (p. 1172).

Mentorship delivered by older female students to middle school girls has shown improvements in mentees' attitudes towards science and mathematics (Reid & Roberts, 2006). Reid and Roberts (2006) studied a mentorship program in Detroit, Michigan, called Gaining Options: Girls Investigate Real Life (GO-GIRL) in which 33 female university students mentored 74 predominantly African-American female seventh graders over a period of ten Saturdays. Reid and Roberts collected data through pre- and post-surveys and journal prompts. The Wayne State University student mentors of the GO-GIRL mentorship succeeded in introducing the middle school girls to new STEM careers, increasing their confidence levels in mathematics, bonding with their female mentors, and learning about what advanced mathematics courses to take in high school.

A supportive CS environment also involves opportunities for students to learn problem-solving strategies that will help them persist through difficult CS challenges. Werner et al., (2010) found that the social interaction between mentoring pairs during problem-solving on computers influenced middle school girls problem-solving skills—a trait that is necessary in order to persist in computer science courses. They concluded that more research is need in the types of peer mentoring activities that will lead to more middle school girls pursuing computer science courses in high school. Other computer science studies that focus on mentoring of ethnic minority students support Werner et al.'s (2010) findings (Clark & Sheridan, 2010). Furthermore, studies have shown that adolescent girls lack confidence in their computer science skills as compared to boys (Halpern et al., 2007; Mosatche et al., 2013), even among those aspiring to major in computer science (Lehman, Sax, & Zimmerman, 2017). Werner et al. (2010) conducted a two-year study analyzing middle school computer programming dyads for 160 middle school students. Their findings indicate that a friendship between partners results in greater confidence in computing and greater programming knowledge.

The value of having a mentor guide a student through CS has been confirmed at the college level as well. In a survey study of 214 undergraduate STEM majors (22.9% of whom were computer science majors) who reported on the effects of informal mentorship they received from a peer, Holland, Major, & Orvis (2012) found that participants receiving peer mentoring were more satisfied with their major, had greater affective commitment to their major, were more involved with their major, and were more willing to mentor others. Holland et al. (2012) failed to categorize their findings by STEM major, so it is not possible to pinpoint the strength of the regression for computer science majors specifically. However, their general recommendations to promote retention in STEM are that university staff provide academic socialization opportunities, such as group study sessions and collaborative assignments, specifically to underrepresented groups.

Availability of CS Curricula and Resources—Rigor, Recruitment, & Diversity

While early CS reform efforts are taking place in parts of the country, many public school districts continue to allocate significant human and economic resources to implementing the Common Core State Standards (CCSS) in English, mathematics, and science. Therefore, any new computer science curriculum that reinforces topics in the CCSS should be welcomed by districts as a way to increase students' academic achievement (Modekurty, Fong, & Cheng, 2014). According to the CSTA, an introductory computer science education course must include the following topics: Collaboration (CL), Computational Thinking (CT), Computing Practice and Programming (CPP), Computers and Communication Devices (CD), Community, Global, and Ethical Impacts (CI), and Comprehensive Curriculum (CC). These topics provide educators with opportunities to teach the CCSS through CSE, which inevitably requires a constructivist and experiential learning model (i.e., learning through doing). In 2006, Jeannette Wing popularized the term *computational thinking* when she proclaimed that people in general, not just computers, need to learn how to solve complex problems using processes that are usually attributed to computer science. Manches and Plowman (2015) (citing Furber, 2012, p. 29) define computational thinking as "'the process of recognising [sic] aspects of computation in the world that surrounds us, and applying tools and techniques from computer science to understand and reason about both natural and artificial systems and processes'" (p. 6).

Computer science courses naturally apply mathematics principals that many students struggle to grasp. Computer science courses have been known to engage students when the material was fun. Many recent curricula that teach coding to students involve hands-on applications and project-based learning. Often students learn introductory computer science principles by programming robots or designing video games. Organizations nationwide are mobilizing to recruit more girls and minorities into computer science through engaging activities. For example, the Hour of Code and Black Girls Code are just two examples of hundreds of initiatives geared to increase female and minority involvement in computer science. Girls who were recruited into a computer science course based on their above-average performance in advanced mathematics expressed positive attitudes towards the computer science course (Margolis et al., 2008).

An example of a CS curriculum that is welcoming to underrepresented student groups, that also integrates hands-on computing activities and real-life applications with common core mathematical concepts, is C-STEM (Modekurty et al., 2014). Survey results after a one-week summer computer science camp using the C-STEM curriculum indicated that over 50% of the middle school female participants were interested in pursuing computer science further (Modekurty et al., 2014). Fisher et al.'s (2015) findings from a longitudinal study of close to 200 middle school girls participating in the Digital Divas computer science program further confirm that a female-friendly CS curriculum can increase adolescent females' interest and participation in CS. Female-friendly CS interventions are needed during middle school to counteract the negative effects of what Andersen (2000) calls gendered curricula, tracking, and other gender-related messages perpetuated by schools (as cited in Sax & Harper, 2005).

Confirming the importance of teacher engagement and buy-in for the successful delivery of secondary CS curriculum, Black et al.'s (2013) qualitative results also reveal several curricular qualities of successful foundational CS courses in secondary schools. In terms of student engagement, Black et al.'s (2013) data identify the need for teachers to offer recreation (fun), relevance (real-life applications), regard (being a part of well-respected group), and reward (prizes and future work potential). Black et al. (2013) emphasize the value of giving students opportunities to create tangible products through creative and interactive computing—also known as *constructionism* or *experiential learning* (Papert, 1993; Sentance & Csizmadia, 2015). Sentance and Csizmadia (2015) highlight how problem-based conversations and "active participation" engage students in the experiential learning process in a computer science classroom (p. 2). According to Van Gorp and Grissom (2001), active learning experiences such as "code walkthroughs, writing algorithms in groups, insert[ing] comments in pairs into existing code, develop[ing] code from algorithm in pairs, [and finding] the bugs in code" make difficult computer programming challenges more accessible to all learners (as cited in Sentance & Csizmadia, 2015, p. 2). These active learning experiences are defined as follows:

> Experiential learning that stems from constructivism describes the design of activities which engage learners in a very direct way. Working with tangible real world objects is a central tenet of Papert's constructionism (Papert, 1991) (which builds on constructivism). Thus, constructivist principles support the strategies of using more kinaesthetic [sic] and active approaches to teaching in the computer science classroom (Sentance and Csizmadia, 2015, p. 2).

However, CS curricula are not always delivered in the manner in which they are intended. A look at UK CSE efforts illustrates discrepancies in how the ICT curriculum is traditionally taught in UK schools. What has surprised researchers is that policymakers have proposed CS curriculum with an emphasis on computational thinking, yet teachers deliver CS curriculum with a lack of computational thinking (Black et al., 2013). To remedy the differences in delivering CS curricula, Goode, Margolis, and Chapman (2014) developed a CS curriculum called Exploring Computer Science (ECS) funded by the National Science Foundation (NSF). ECS is a foundational CS course offered in over 40 LAUSD high schools. ECS has seen female participation rise to over 45%, which is over 20% higher than female participation in the traditional and rare AP CS course. ECS data also indicate that Latinos/as make up over 70% of the program's enrollment. Margolis, Goode, and Binning (2015) assert that higher enrollment of traditionally underrepresented populations in ECS is a result of ECS schools having to replicate their school demographics in their ECS courses. Goode et al. (2014) also credit the ECS course's scaffolded curriculum and built-in problem-solving opportunities with raising student confidence and helping traditionally underrepresented female and minority students with achieving a deeper understanding of CS principles—factors that will help students be more successful in AP CS as well as in college CS courses. Margolis et al.'s (2015) and Goode et al.'s (2014) findings are consistent with Eccles et al.'s (1993) Expectancy Value Model findings about the importance of girls and minorities finding success when studying a subject before they commit to studying that subject in greater depth. Numerous recent studies have focused on the value of offering girls and minorities the opportunity to take a foundational CS course that will ensure their preparation to succeed in more advanced high school CS courses. The premise behind this scaling up or foundational hypothesis is that girls and minorities are more likely to stay in CS if they experience prior CS successes to build upon (Ware & Stein, 2013).

One way to invite students to be successful in their CS experiences is to introduce CS topics via fun challenges that are relevant to adolescent students' lives—namely, gaming and storytelling. Clark and Sheridan's (2010) gaming design intervention to recruit underrepresented students in computer science proved effective for African-American middle school boys from over 30 schools in Washington, D.C. The Game Design through Mentoring and Collaboration (GDMC) after-school program was a voluntary 10-week Saturday program that offered students instruction in the latest game design software, including "Maya (3-D modeling and animation), GameMaker (2D game design logic), Alice (introductory computer programming), Civilization (3D game logic), …, flash animation, and MissionMaker (3D game design logic)" (p. 8). The Carnegie Mellon University team that created the Alice software named it after Alice in Wonderland, in order to convey their goal of using a storytelling environment to attract beginners to computer science (www.alice.org). Culturally sensitive CS curricula, such as the Alice program and its accompanying resources, can help break down structural barriers faced by female and underrepresented minority students when considering taking CS courses. Culturally sensitive interventions are necessary when interventions which work for White students do not necessarily work for Black and other minority students (Olszewski-Kubilius, 2003).

Interventions like Clark and Sheridan's (2010) GDMC program illustrate the importance of providing non-traditional and social justice-minded CS interventions that help recruit more minority students to CS, because, as (Olszewski-Kubilius, 2003) points out, traditional color- and gender-blind interventions that focus purely on increasing access to programs for "gifted children" only help to widen the achievement gap for minorities and girls (p. 304). As part of non-traditional, culturally sensitive CS interventions, social justice-minded educators may also have to equip their female and underrepresented minority students with coping strategies to help retain these much-needed students in the White male-dominated CS workforce.

Funding for CS Teacher Preparation—Professional Development Matters

One of the persistent challenges to training a sufficient number of CS teachers to teach rigorous computing is developing a feasible professional development program (Brown et al., 2014). Notwithstanding positive findings with their CS intervention study of middle school girls in Australia, Fisher et al. (2015) fail to address the need for teacher preparation to teach CS and the cost of a sustainable CS professional development model. The well-documented CS teacher shortage is further exacerbated by high CS teacher attrition, which further affects districts economically (Bernier & Margolis, 2014). The need for adequate school funding to sustain STEM professional development in general, and CS teacher preparation in particular, has been reinforced by other research (Bernier & Margolis, 2014; Campbell, Jolly, & Perlman, 2004; Menekse, 2015).

Research supports the idea that CS professional development can change teachers' classroom practices in a way that brings more girls to CS courses (Goode et al., 2014). Goode et al.'s (2014) ECS program requires teachers interested in teaching an introductory computer science course to participate in a summer professional development course. The ECS professional development model brings "gender, preparatory privilege, and other equity considerations" to the forefront of participating CS teachers' minds. The ECS PD model directly addresses the need to make "chilly" and traditionally White male-centered CS classrooms more inviting to female and minority students. As Goode et al. (2014) caution, new computer science curricula will not suffice to ameliorate the shortage of computer science students. Goode et al. (2014) emphasize that teacher preparation to teach computer science is tantamount to providing CSE access (Hubweiser et al., 2014).

Over the past couple of decades, various universities nationwide have been offering summer CS teacher trainings on increasing girls in CS classes. Evaluations done on these CS professional development opportunities found that teachers had an opportunity to learn recruitment methods, creative CS lessons, and share "challenges and successes" with other CS teachers. Research on summer CS teacher workshops has also found that time for training teachers is a major challenge, as well as the need for more training opportunities. For this reason, training high school mentors who have had advanced CS experiences to recruit more middle school students in CS courses would alleviate the stress felt by teachers to find time to be trained (Israel et al., 2015). Older CS mentors would also instill leadership qualities in the mentors—a quality found by many CS researchers to be a benefit to underrepresented groups pursuing careers in IT.

Grassroots CS Teacher Networks and Ongoing Coaching

Introducing “authentic [computing] experiences” in a variety of subject areas becomes an even greater challenge when teachers have no prior computer science experience (Israel et al., 2015). Israel et al. (2015) found that elementary school teachers inexperienced in CS saw the lack of classroom expertise as a barrier to implementing CS curriculum. This stage of the Israel et al. (2015) study could be dubbed the redefining/restructuring stage. It is the time of the implementation when the teacher may realize that she does not necessarily need to know the entire CS curriculum in order to begin implementing it. Subsequently, the novice CS teachers were able to persist through their first year of teaching CS when they accessed CS information through communication channels such as Twitter chats, online resources, and on- and off-site technology specialists—the clarifying phase. Sentance and Csizmadia (2015) confirm that the grassroots CS community Computing at School, or CAS group, in the UK is the conduit that helps teachers leading the UK CS movement diffuse CS teaching strategies more effectively.

Even UK schools have faced CSE challenges, such as offering affordable and continuous CS teacher professional development, despite the 2008 establishment of the CAS group—a UK CS network similar to the CSTA in the US (Brown et al., 2013). Nonetheless, Brown et al. (2014) offer the CAS group’s establishment of a “national training network, whereby ‘lead’ schools with expertise in delivering computer science can assist those nearby schools without it” (p. 20) as a model for US efforts, such as CS10K, trying to increase the number of trained CS teachers in America. To further illustrate the benefits of a CS teachers’ social network, Sentence and Csizmadia (2015) provide a glimpse at how 300 UK in-service teachers use their pedagogical expertise to deliver CSE, including “contextualized learning, computational thinking skills development, code manipulation, working collaboratively and learning away from the computer" (p. 1), as well as how these experienced CS teachers use their membership in the CAS group to diffuse their knowledge to other teachers. To further expand the CS teacher network, Falkner, Vivian, & Falkner (2014) suggest that students would benefit from more sharing of best practices between K-12 CS teachers and higher education CS experts.

Black et al. (2013) confirm Ni's (2009) earlier results indicating that teacher excitement about teaching CS is key to successful curriculum implementation. Ni's (2009) results, based on the survey answers of approximately 30 teachers, are consistent with Black et al.'s (2013) findings that the key to CS curriculum innovation lies in a bottom-up (or grassroots) approach, rather than top-down mandates. Alarmingly, teachers in Ni's (2009) study expressed concerns about generating excitement among their CS colleagues, especially when a comprehensive CS curriculum (or resource package including robots) was not available at their school site due to inadequate funding.

Summary

As illustrated in this chapter, computer science education reform efforts are at a critical juncture with more national and local support for increasing CS access than ever before. Volumes of research show the benefits of teaching advanced computing concepts to students in grades K-12, especially in the middle grades as a preparation for the AP CS A and AP CSP courses and exams. The data show the urgency of teaching CS to traditionally underrepresented minority and female students earlier than high school.

CHAPTER 3: METHODOLOGY

Introduction

As illustrated in chapters one and two, research on the topic of computer science education warns that, for a variety of reasons, female and underrepresented minority students drop out of the CS pipeline during the transition from middle school to high school (Eccles et al., 1993; Margolis & Fisher, 2002; Wang & Degol, 2013). This study focused on how middle schools with a large percentage of underrepresented minority students deliver computer science education to help prepare their students for advanced high school computing courses. The following research questions guided the study:

1. According to middle school administrators and computer science teachers in a district with a large number of socioeconomically disadvantaged (SED) and traditionally underrepresented minority students (URM), what are the essential elements (i.e., teacher, technology, funding, etc.) necessary to offer computer science courses to its middle school students?
    a. What are the essential elements necessary to sustain a computer science education program?
2. According to middle school administrators and computer science teachers in a district with a large number of SED and traditionally URM students, what are the challenges they faced while implementing a computer science curriculum in their schools?
    a. How did the challenges vary, if at all, by student population (i.e., URM, SED, special education students, female students, male students, etc.)?
    b. According to middle school administrators and computer science teachers, how has the school and/or district addressed the challenges?

3. What supports and resources do middle school administrators and computer science teachers say they need to help them integrate computer science into the middle school curriculum?
    a. How do the supports and/or resources vary, if at all, by student population (i.e., URM, SED, special education students, female students, male students, etc.)?

Research Design

I used a qualitative cross-case study design that is rooted in a social justice and diffusion of innovation framework to answer my research questions. I chose a multiple-case study method, because I wanted the findings to inform future computer science education policymakers' and decision-makers' choices in supporting CS education (Patton, 2015). Furthermore, comparative case studies with rich descriptions of the phenomenon taking place in middle schools teaching computer science can "afford the reader the vicarious experience of having been there" (Merriam, 2009; Mertens, 2015). Stake (2006) refers to the phenomenon being studied in a case study the *quintain*, and the multiple sites in this case study were chosen for what they could contribute to understanding the quintain (p. 7). These thick descriptions of the six cases were essential to prospective computer science-implementing middle school administrators and teachers who may have little to no exposure to computer science education. Merriam (2009, citing Donmoyer 2000) points out that among the advantages of a descriptive case study are access to otherwise inaccessible situations, the ability to see novelty in a somewhat familiar situation, and a "decreased defensiveness" in trying to accept an innovation that might otherwise generate hostility (pp. 258-259). Learning new ideas from descriptive case studies is less threatening than learning from first-hand experiences.

Quantitative methods for a small sample size (N=6 schools and Interviewees = 12) would not have yielded rich data about administrators' experiences in implementing and teachers' experiences in teaching computer science (Maxwell, 2013). However, qualitative methods allowed me to descriptively unveil the perspectives of middle school administrators and CS teachers in providing computer science education (Marshall & Rossman, 2011).

A qualitative case study with detailed data from 12 interviews also allowed for more precise triangulation or "crystallizing" of emerging themes in the data (Maxwell, 2013; Merriam, 2009). The transformative purpose of the current study (i.e., an intent to influence educational policy about computer science curriculum access, as well as inform future computer science curriculum implementations) also lent itself to a qualitative design (Mertens, 2015). Consequently, a qualitative comparative case study with *thick descriptions* of computer science education provided the amount of detail necessary for other administrators and teachers to make more intelligent decisions about how to implement CSE in their own districts.

Methods

Research Sites

My search for research sites began with an online search for public school districts, which resulted in a list of all the middle schools in these districts. I emailed a brief survey (Appendix A) to the middle school administrators from these districts asking whether computer science courses that prepare students to take AP CS A in high school are taught at their middle school. The survey also asked if teachers at schools teaching computer science would be willing to participate in my case study. From those that said they were teaching CS and were willing to participate in this study, I chose the six schools with the highest population of students underrepresented in CS, as well as the schools that had the highest number of socioeconomically disadvantaged students.

The district I worked with was the Sunrise Unified School District (SUSD), a large urban district in California. The participants were administrators and CS teachers at six middle schools in SUSD that had implemented CS for at least one year.

Site, Access, and Participants

To conduct the case study, I first obtained permission to conduct my research in SUSD from SUSD's research office. Once I secured permission from SUSD, then I obtained permission from UCLA's Internal Review Board (IRB) to conduct my study at SUSD. Finally, I sent emails to all the middle school principals in the district requesting their participation in the study. Of the twelve middle school administrators who responded to my email with interest in participating in my study, I chose six middle schools for their number of socioeconomically disadvantaged students and their underrepresented minority student populations. SUSD is a pseudonym, and pseudonyms were used for all sites and participants in order to protect their identities.

I interviewed administrators and CS teachers at six middle schools in SUSD—Arbor MS (largest by population), Bloom MS (smallest by population), Crest MS (largest population of socioeconomically disadvantaged and Hispanic students), Delta MS, Eagle MS (largest population of African-American students), and Falcon MS. One administrator and one CS teacher at the participating middle schools were asked for their permission to be interviewed for the study. The participants at each site had to have been involved with implementing CSE at their school for at least one year. Studying an administrator and a CS teacher at each site provided important insight into the social networks used to diffuse CSE at the school.

The six administrators I interviewed had between 17 and 31 years of experience in education and between one and 17 years of experience at their respective site. The six CS teachers I interviewed had between three and 43 years of experience in education and between three and 16 years of experience at their respective site. Four of the administrators were males and two were females, and two of the CS teachers were males and four were females.

Data Collection Methods and Analysis

To answer my research questions through a diffusion of innovation framework, data collection methods for this comparative case study included: six semi-structured administrator interviews, six semi-structured CS teacher interviews, and document analysis. In addition to helping me triangulate all my data, my multiple modes of qualitative data collection helped me understand how six middle schools harnessed the power of social networks to spread the implementation of CSE, especially for student populations that are underrepresented in CS.

Interviews

I conducted 12 semi-structured, one-on-one telephone interviews that lasted from 45 to 70 minutes each. Face-to-face interviews were not possible in this study because of the participants' busy schedules. However, I applied what Rubin and Rubin (2012) call *responsive interviewing* by allowing interviewees' answers guide some of my follow-up questions to help me better answer my research questions (p. xv). I asked interviewees to describe in detail their computer science curriculum implementation process, as well as the benefits, challenges, and resources that are necessary for CS curriculum implementation. My interview questions focused on how administrators and teachers use social communication channels to diffuse CSE as an innovation in their schools.

Interviews were audio-recorded using a digital recording device, and interview recordings were transcribed using an online transcription service called Rev. Once interviews were transcribed, I used HyperResearch to code the transcripts for emergent themes. In order to get the most out of my interviews, I listened to my interview recordings and wrote field notes and memos to guide my data analysis and follow-up data collection. I then used Microsoft Excel to color code and organize all the themes with field notes and quotations from the interview transcripts. Microsoft Excel made it possible for me to filter data in multiple ways, including by theme and by school, as well as by administrators and by CS teachers. After printing theme reports of my Excel data, I used Stake's (2006) and Rubin and Rubin's (2012) data analysis theories to guide me with more in-depth analysis and cross-case analysis:

Document Analysis

I analyzed the computer science curriculum used at each middle school. By asking the middle school computer science teachers to guide me through each unit in their CS curriculum, I gained greater insight into the more and less challenging topics to teach. Furthermore, I concentrated on what communication and collaboration tools are embedded in the curricula to promote greater diffusion of CS at the school. I also learned to what extent, if at all, the CSTA standards helped and/or hindered the teaching of CS topics. Finally, I ascertained what additional resources teachers use or would like to have access to in order to teach rigorous middle school CS standards to prepare students for more advanced CS courses in high school.

Table 2

Research Question and Data Collection Crosswalk

Research Question	Data Collection Method (when)	Data Type
1) According to middle school administrators and computer science teachers in a district with a large number of socioeconomically disadvantaged (SED) and traditionally underrepresented minority students (URM), what are the essential elements (i.e., teacher, technology, funding, etc.) necessary to offer computer science courses to its middle school students?	• Interviews and document analysis (during)	• Qualitative
1a) What are the essential elements necessary to sustain a computer science education program?	• Interviews	• Qualitative
2) According to middle school administrators and computer science teachers in a district with a large number of SED and traditionally URM students, what are the challenges they faced while implementing a computer science curriculum in their schools?	• Interviews and document analysis (during)	• Qualitative
2a) How did the challenges vary, if at all, by student population (i.e., URM, SED, special education students, female students, male students, etc.)?	• Interviews and document analysis (during)	• Qualitative
2b) According to middle school administrators and computer science teachers, how has the school and/or district addressed the challenges?	• Interviews and document analysis (during)	• Qualitative
3) What supports and resources do middle school administrators and computer science teachers say they need to help them integrate computer science into the middle school curriculum?	• Interviews and document analysis (during)	• Qualitative
3a) How do the supports and/or resources vary, if at all, by student population (i.e., URM, SED, special education students, female students, male students, etc.)?	• Interviews and document analysis (during)	• Qualitative

Validity and Reliability

The validity of the study's findings was established by triangulating the qualitative data obtained from all three of my data sources—namely, administrator interviews, CS teacher interviews, and CS curriculum document analysis (Stake, 2006). Threats to construct validity were minimized by extrapolating common themes from multiple sources of data (Yin, 2014). I attempted to increase the internal validity of my case study by triangulating all three sources of data. I worked to minimize threats to the internal validity of my case study by avoiding inferences not directly supported by the data (Yin, 2014). I used rigorous methods of qualitative data analysis that answered my research questions (Stake, 2006; Rubin & Rubin, 2012). By conducting follow-up interviews, I was able to check for the existence of errors and/or bias in my interpretation of my qualitative data.

To maximize the reliability of my research, I maintained constant transparency in my research by maintaining a research journal. I kept thorough notes and other documentation throughout data collection and analysis (i.e., an auditable trail or "case study database"), so that my work could easily be replicated (Yin, 2014, p. 49).

Ethical Considerations

I followed ethical protocol throughout the data collection and data analysis phases of my study. I sought UCLA Internal Review Board (IRB) approval prior to conducting my case study. I avoided bias by using member-checking to stay true to the evidence I received from interviews and document analysis.

In order to further mitigate ethical concerns, I made sure that participants understood that their participation in this study was optional. In order to sustain a trusting relationship between me and all the study participants, I maintained transparency and regular communication with study participants. As such, I obtained permission from interviewees to audio-record interviews and made sure that interviewees understand that they could stop the recording if they felt the need to do so. All participants' and sites' anonymity were maintained on all data records and reports.

CHAPTER 4: CASE STUDY FINDINGS

This multi-site case study investigated the implementation of computer science education at six middle schools in a large urban district with a 50% or higher minority and/or socioeconomically disadvantaged student population. During the data collection and analysis phases of this study, I sought answers to the following research questions:

1. According to middle school administrators and computer science teachers in a district with a large number of socioeconomically disadvantaged (SED) and traditionally underrepresented minority students (URM), what are the essential elements (i.e., teacher, technology, funding, etc.) necessary to offer computer science courses to its middle school students?
    a. What are the essential elements necessary to sustain a computer science education program?
2. According to middle school administrators and computer science teachers in a district with a large number of SED and traditionally URM students, what are the challenges they faced while implementing a computer science curriculum in their schools?
    a. How did the challenges vary, if at all, by student population (i.e., URM, SED, special education students, female students, male students, etc.)?
    b. According to middle school administrators and computer science teachers, how has the school and/or district addressed the challenges?
3. What supports and resources do middle school administrators and computer science teachers say they need to help them integrate computer science into the middle school curriculum?
    a. How do the supports and/or resources vary, if at all, by student population (i.e., URM, SED, special education students, female students, male students, etc.)?

The SUSD Case Study

Introduction

Sunrise Unified School District (SUSD) is home to approximately 475,000 non-charter school students, 84% of whom qualify for the free or reduced lunch program, 8% of whom are African-American, and 77% of whom are Hispanic. Due to its size, SUSD is divided into semi-independent local districts based on geographic location. The six middle schools that participated in this multi-site case study were from half of the local districts. I conducted telephone interviews with participants from the six sites over a period of three months. Once data collection was completed, I analyzed the data for patterns and emergent themes that would shed light on the phenomenon of middle school computer science implementation. Common themes emerged between sites when I aligned the sites by type and similarity of implemented curriculum. Sites One, Two, and Three (Arbor, Bloom, and Crest middle schools, respectively) had implemented similar curricula to each other—namely, block-based coding lessons without physical computing. By contrast, Sites Four, Five, and Six (Delta, Eagle, and Falcon middle schools, respectively) implemented curricula with the same programming language—namely, RobotC—and physical computing of VEX robots. Once I split the six sites into two categories—non-physical computing and physical computing—then I organized the three sites within each of my two groups by the CS teacher's and administrator's level of CS expertise. As a result, a hierarchical grouping emerged with least to greatest level of CS expertise. For the purposes of this study, implementing a coding class with no physical computing opportunities, such as programming a robot, is the lowest level of computer science education implementation [least trouble-shooting and formal language learning]. Conversely, a site that implemented a robotics course where students built robots and used a computer programming language to program their robots represents the most rigorous computer science education implementation [competition against other middle schools].

The Schools

The six middle schools from SUSD chosen for this case study had all offered some type of computer programming instruction for at least one school year. While I contacted close to all 80 SUSD middle schools to request their participation in this study, approximately ten of the district's middle schools agreed to participate. Of these ten schools, the six sites shown in Table 3 below were chosen because they served at least 60% socioeconomically disadvantaged or at least 50% underrepresented minority students (African American or Hispanic students).

Table 3

Case Study Middle School Sites with Demographics

Site No.	**School**	**Total (Non-Charter) Middle School Student Population**	**Socio-Economically Disadvantaged**	**African-American Student Population**	**Hispanic Student Population**
S1	Arbor MS	1,291	70%	57%	30%
S2	Bloom MS	254	60%	24%	33%
S3	Crest MS	890	92%	3%	91%
S4	Delta MS	676	81%	4%	81%
S5	Eagle MS	680	70%	60%	28%
S6	Falcon MS	735	88%	3%	78%

While the six sites represent less than 12.5% of the district's middle schools, they reflect a cross-section of the types of computer programming implementations happening or being planned in other large urban districts with large populations of minority and socioeconomically disadvantaged students (Century, Lack, King, Rand, Heppner, Franke, & Westrick, 2013; Guzdial, Ericson, McKlin, & Engelman, 2014; Lang, Galanos, Goode, Seehorn, Trees, Phillips,

& Stephenson, 2013; Margolis, Ryoo, & Goode, 2017). As the most knowledgeable decision-makers in the computer science implementation, a computer programming instructor and at least one administrator from each site were interviewed. These interviewees shared information about their early adoption of available computer science curricula. To keep their identities anonymous, all the schools', teachers', and administrators' names have been changed to names with the letters A to F.

Overview

Each of the six sites in this case study offered one of several computer science curricula available to middle school CS teachers. Table 4 lists each site with the name of the CS curriculum, as well as the computer programming environment, implemented at the site. I have organized the curricula, and consequently each site, in order of least difficult to most difficult to learn based on grade level recommendations made by the curriculum writers. Some Google CS First (i.e., Storytelling and Music & Sound) and Code.org (CS Fundamentals) curricula are easily accessible to students as young as nine with no prior coding experience because they offer user-friendly blocks and visually clear feedback for debugging. In order to complete a challenge—usually moving a character known as a sprite from a starting point A to an ending point B—Google CS First and Code.org students will need to drag-and-drop simple commands (i.e., left, right, up, down), and many students will arrive at either identical or very similar solutions for getting their sprite from point A to point B.

Table 4

CS Curriculum Offered at Each Site

Site #	Site Name	Computer Science Curriculum	Computer Programming Environment
S1	Arbor MS	Google CS First	Scratch (block-based drag-and-drop)
S2	Bloom MS	Code.org/ECS & Course 2	Code Studio (block-based drag-and-drop)
S3	Crest MS	Code.org/*CS in Science*/Project GUTS	Starlogo Nova (agent-based simulation)
S4	Delta MS	No Curriculum with VEX IQ Kits	RobotC Graphical (block-based drag-and-drop)
S5	Eagle MS	Project Lead the Way (PLTW) with VEX EDR Kits	RobotC (text-based natural language)
S6	Falcon MS	Project Lead the Way (PLTW) with VEX EDR Kits	RobotC Graphical (block-based drag-and-drop)

By contrast, Starlogo Nova and RobotC are more complex programming environments to dive into without prior programming experience. RobotC is much closer to technical programming languages like C++ and Python and contains over 50 commands that students can use to control a robot. Debugging a RobotC program is more time-consuming, because it requires more steps to run a program and sometimes many iterations to get a robot to the desired location to complete a challenge. Of all the coding environments in this case study, only courses using RobotC (and LEGO, used at Site #4) can prepare middle school students for robotics and programming competitions. At the time of this study, the other coding environments did not provide middle school students with opportunities to participate in international programming competitions, which expose middle school students to physical programming challenges requiring a greater level of mastery and rigor (not to mention soft skills such as leadership and teamwork) than challenges completed individually on a tablet or computer screen.

Physical programming of a robot is considered a more complex implementation of computer programming principles than is programming digital images (or sprites) to move on a computer screen. Problem-solving or troubleshooting physical computing tasks often take more time and skill (Brinkmeier & Kalbreyer, 2016). For example, the results running the same commands multiple times on a screen are always the same. However, identical programs run on a physical robot on a classroom floor or on a robotics competition table can have significantly different outcomes each time the program is run. Robots' sometimes unpredictable behaviors can result from motor or wheel performance (e.g., "mechanical faults") (Brinkmeier & Kalbreyer, 2016), physical characteristics of the course that the robot must traverse such as dirt or other type of friction, or simply a need to reset the robot brain or recompile the program to reset all the robot components. Opportunities to cover these essential robotics programming topics do not arise in an entry-level Google CS First or Code.org course. Furthermore, CS First and Code Studio students need much less teacher guidance to begin a challenge than do Starlogo Nova and RobotC students.

The Computer Science Curricula

Arbor Middle School: Google CS First with Scratch

Google CS (or Computer Science) First is a theme-based curriculum specifically designed by Google CS curriculum designers to attract middle school students to computer science. Some of its themes include music, art, storytelling, sports, and fashion design—all topics that are known to be appealing to middle school students (ISTE Seal of Alignment, 2018). Due to the ready-to-use nature of the CS First website, Google does not offer face-to-face CS First training for teachers on a regular basis. Teachers with CS First experience sometimes share their experiences with the curriculum and provide basic training at educational technology conferences or Google Summits. Google CS First introduces students to coding using the Scratch block-based coding environment from MIT. Scratch is a coding platform that allows students to easily design computer games and create interactive stories. Google’s CS First thematic courses offer the contexts for student projects created in Scratch.

The Google CS First curriculum is available free online without any software installation. CS First allows a teacher to setup a class with a unique course code that the teacher can share with students. Students can use Chromebooks, tablets, or desktops/laptops to sign up with the course code online and begin watching video tutorials about the theme their teacher has chosen. The unique course codes generated by Google’s CS First website provide teachers and students with a secure course management environment where student progress is saved and teachers can track student progress. Logging into the CS First website requires that each student have a Google account—preferably a district-provided Google account. Once students enter their Google username and password, CS First provides them with an alphanumeric username and password to use to login to Scratch 2.0. The video tutorials explain what students will need to program using the Scratch drag-and-drop coding environment available for free through a Massachusetts Institute of Technology (MIT) website (scratch.mit.edu).

Bloom Middle School: Code.org CS Fundamentals (formerly Course 2)

The second type of computer science education curriculum referred to in this study is also block-based and is available for free on the Code.org website without any need for installing software. The environment where students solve coding challenges is called Code Studio. At the time of this study, the Code.org curriculum used by one of the teachers interviewed was called Course 2. In 2017 Code.org restructured their online course catalog into three main offerings—CS Fundamentals Courses A-F (for grades K-5), CS Discoveries (for grades 6-8), and CS Principles (for grades 9-12).

Code.org offers 1-day, face-to-face, free training courses for teachers on a regular basis throughout the year. Teachers can find free trainings on the Code.org website. Teachers can also create a Code.org login and start coding in the CS Fundamentals courses without any prior coding experience.

Crest Middle School: *CS in Science* with Code.org and StarLogo Nova

CS in Science is a science curriculum that incorporates block-based coding and is available on the Code.org website through a collaboration with Project GUTS (or Growing Up Thinking Scientifically). StarLogo Nova, or SLNova, is the *CS in Science* coding environment developed by MIT's Scheller Teacher Education Program. Students use the website www.slnova.org to login to SLNova and to use coding blocks to model scientific phenomena, such as the spread of disease and chemical reactions, for example. Students code the characteristics of agents, such as bacteria, and agent behavior in a specific environment. SLNova then generates a three-dimensional model of the student's simulation to give the impression of a real-life scientific experiment. In addition to creating their simulation projects, students can also analyze data generated by their simulations.

SLNova is available for free online without any need for installing software. Teachers and students can share their projects publicly or add collaborators to work on projects as a team. SLNova modeling is similar to creating a simple video game from scratch, so it is not as easy to learn independently as Google CS First's more intuitive Scratch coding environment or Code.org's scaffolded Code Studio coding environment. Project GUTS offers teacher professional development to learn to use their curriculum. An archived online training is available for anyone who creates a Project GUTS account on their website.

Delta, Eagle, and Falcon Middle Schools: VEX IQ and VEX EDR with RobotC

RobotC is a C-based programming language designed to program various types of robots. VEX IQ and VEX Educational Robotics (EDR) are two robotics curricula that both use RobotC programming. VEX IQ's plastic robot parts snap together like LEGO robot parts, while VEX EDR's metallic beams and screws require screwdrivers, wrenches, and hex keys for robot assembly. A VEX IQ robot is controlled by a robot brain, and a VEX EDR robot is controlled by its Cortex microcontroller. In order to program the VEX IQ brain and EDR microcontroller, the free RobotC software must be installed on a Windows 7 or higher computer. Once a year the RobotC developers usually provide a free software update online, which means that all RobotC student computers need to be updated. RobotC programming software is available in two interfaces—graphical and natural language. RobotC Graphical offers drag-and-drop block-based programming very similar in appearance to the Scratch and Code Studio programming environments. On the other hand, the text-based version of RobotC requires actual typing of C programming language commands, and syntax mistakes (i.e., missing commas and semicolons) and spelling errors can result in a program not working due to errors printed on the screen that students have to decipher. Students need to understand error messages in RobotC in order to debug their code. While the order of the commands in a program matter in most programming languages, they matter more in RobotC than they do in Scratch and Code Studio. RobotC's greater reliance on code line order is its more rigid, technical, and less forgiving nature.

Carnegie Mellon University (CMU) has created a robotics learning management system for teachers to use with their classes. The system used to be called CS2N or Computer Science Student Network and is now referred to as the CS-STEM Network. For students to add themselves to a teacher's CS2N course, students need to create and activate a CS2N account with an email address. This can be problematic for students who are too young to have a personal email address and who are not provided with an email account through their school district. Without the ability to login to an email account, the student cannot activate their CS2N account. Recently, the CS2N team at CMU has given teachers the ability to upload their own student lists without the need for student email accounts.

CS Teachers and Site Administrators

Table 5 describes the five administrator, one magnet coordinator, and six computer science teachers who were interviewed at the six sites.

Table 5

Administrators and Teachers Interviewed

No.	**School**	**Title (Subject) (Pseudonym)**	**Gender**	**Years in Education (Years in SUSD)**	**Years at Site**	**Academic Field**	**CS Training**
1-1	Arbor MS	Principal (Aaron)	M	31 (25)	5	Math	None
1-2	Arbor MS	Teacher (CS/Science) (Alan)	M	33 (28)	9	Psychology/Computers	Self
2-1	Bloom MS	Principal (Brenda)	F	28 (28)	1	History	None
2-2	Bloom MS	Teacher (CS/Math) (Becky)	F	3	3	Math	Face-to-Face
3-1	Crest MS	Principal (Chuck)	M	27 (14)	8	Science	None
3-2	Crest MS	Teacher (CS/Science) (Carol)	F	6 (4)	4	Psychobiology	Face-to-Face
4-1	Delta MS	Magnet Coordinator (Dan)	M	27 (25)	1	English/Film/Tech	None
4-2	Delta MS	Teacher (CS/Science) (Doris)	F	43 (16)	16	Chemistry & Physics	Self
5-1	Eagle MS	Principal (Evelyn)	F	17 (17)	17	Biology; Molecular Genetics	None
5-2	Eagle MS	Teacher (CS/English) (Emily)	F	7 (4)	4	English	Face-to-Face
6-1	Falcon MS	Principal (Fred)	M	25 (25)	6	Economics; Film Production	None
6-2	Falcon MS	Teacher (CS/Science) (Felix)	M	18 (9)	4	Geology	Face-to-Face

Site 1: Arbor Middle School, SUSD

Site Description and Background

Despite having the largest student population of the six middle schools, Arbor Middle School has declining student enrollment due to migration of students to charter and private schools. Arbor is situated in a trendy neighborhood with a lot of modern townhouses and condominiums. Arbor Middle School is home to a diverse and high-performing student body. Its students have traditionally scored above the district average on state English Language Arts (ELA) and mathematics standardized exams. In addition to being the largest middle school in this study, Arbor MS also had the second highest number of African American students—seven times the district average. Its diversity was one of the characteristics Alan liked most about teaching there. Its principal, Aaron, had been at Arbor five years at the time of this study. In terms of technology access, Arbor MS was a pilot iPad one-to-one school, which meant that every student had access to an iPad throughout the day.

After some informal discussion about incorporating coding into the school curriculum, Alan decided, “I might as well jump into that, because I wanted to see if there was an opportunity for me to actually add a full [class], like another elective.” Due to parent perceptions that charter and magnet schools offer a better education and small class sizes, students are leaving Arbor MS for other nearby charters and magnets. To minimize this migration, Aaron is actively pursuing another magnet that incorporates technology, either with medicine or with robotics, depending on what parents decide to approve and support.

Implementation Elements and Timeline

Aaron acknowledged that he wanted more computing opportunities for Arbor students. A couple of years earlier, an Arbor parent who worked as a computer programmer at a local university had success introducing Arbor students to the Google CS First curriculum in an after-school club setting. Understanding that parents wanted more coding opportunities for their students, for the 2015-16 school year Aaron hired a for-profit organization to lead an after-school program that introduced students to a variety of STEM activities. Aaron first offered the after-school program to eighth graders and later to sixth and seventh graders. According to Aaron, the after-school program could not take as many students as were interested, so the program had to turn some students away.

The same parent who oversaw the 2014-15 after-school coding club recommended the CS First program to Alan in 2016 and suggested that Alan try the simplest activities with a group of his own students. At the same time, Alan said that he wanted Arbor MS to be more competitive with other schools, so he wanted to bring on coding for the benefit of the school. Aaron agreed to let Alan teach a ten-week Google CS First course in the last quarter of the 2016-17 school year.

Alan saw this as an opportunity to secure a new course for his own teaching future, so he said that he took on the challenge without any hesitation. “I know some JavaScript and coding for web pages.” Alan had the freedom to put together whatever CS curriculum he chose. The Google CS First curriculum was decided upon after it had been offered successfully as an after-school club. The advice of the PTA parent who worked as a UCLA programmer was the main reason that Alan decided to go with the Google CS First curriculum. Easing into CS First, Alan used Storytelling, then Music & Sound, and Art. Looking to the future, Alan’s plan was for the students to learn to animate and to learn to make their own games.

About his first attempt at using Google CS First, Alan said, “I was tentative and afraid to try something too advanced.” Even though Alan self-selected himself to teach the ten-week tech wheel, he expressed that he felt like “such a beginner right now. It was just kind of overwhelming at the beginning. I almost feel like I'm a student teacher right now.” While Alan recalled having some computer programming in his background, he admitted that it is “nothing like what I'm having to teach my students now”. Alan hoped that as a novice CS teacher he had had someone, like a CS guru, assisting him in delivering the CS First curriculum. “There’s no one right now. I’m just basically on my own and just trying to survive.”

Implementation Challenges and Solutions

The early stages of the CS implementation at Arbor MS were full of challenges with few solutions at the time I interviewed Aaron and Alan. For example, Aaron attributed the difficulty in bringing more CS opportunities to Arbor students to funding shortages. “I think, number one, that the district really needs to listen to schools, because in a large district different communities have different needs and wants. I think a lot of the parents in our local district want to be right at the forefront of whatever is new and exciting and innovative for their kids. And, they want technology, they want things that are new for their kids, and that’s where the challenge of money comes in, because it is very expensive to buy these things.” I asked Aaron what support staff he had to help bring in more CS opportunities to Arbor students. He mentioned having a magnet coordinator, a college and career coach, and numerous classroom teachers who focus on writing grants.

Alan had the impression that other schools were more competitive in teaching technology classes, and specifically computer science classes, than Arbor MS had been during his nine years at the site. Alan expressed a desire to help the school and the students become more competitive by giving them some computer science background. “All the schools are now competitive, so I thought that it would help the school become more competitive and help me personally if I began teaching this coding class." With over 20 years in the district, Alan also felt the pressure of making himself more academically diverse and competitive to his site and district by being the only one to take on computer science instruction.

Of the seven CS First curricula available to him at the time, Alan chose Storytelling, Music & Sound, and Art. "I'm just going from the easiest ones," said Alan, who followed the advice of the parent who was guiding him. The parent strongly urged Alan to start with the easier curriculum. Aaron mentioned that two female teachers were eager to bring robotics to students as an after-school club, which is how the for-profit company was hired by Aaron.

Prior to the start of the spring quarter CS First class, Alan said he did not have a chance to try out the curriculum himself. Throughout the quarter, he tried to keep up with the students, but he eventually gave that up because he discovered that the students were much quicker at getting ahead than he was. Instead, Alan planned to spend his summer learning at least five additional Google CS First modules to get ahead for the next CS course he would teach. Part of what helped Alan keep a positive outlook on his CS course was the students' creativity. Alan said, "I find that they're a whole lot more creative than I am. Their minds are so open." Despite voluntarily jumping into teaching Google CS First, Alan declared that not having a background in computer science was a problem for him. "I've done some coding, but nothing like what I'm doing right now."

Alan suggested that part of the benefit of diving into a new curriculum such as CS First is that it makes him somewhat of an expert in an area where there is a need for teachers. Even when he retires, he hopes that he will be able to consult on a CS curriculum like CS First. One of the major selling points of CS First for Alan was the lack of paperwork to correct like there would be in a traditional subject course. Alan turned to the Google CS First forum to gather advice from other teachers implementing the same CS curriculum. He was surprised and encouraged that he had not faced the same obstacles that teachers in the forum said they faced.

Implementation Supports and Resources

Alan was not discouraged by the limited support he received during his first CS First course, but he said that he continued to be optimistic. Of the course gurus and volunteers that were offered on the CS First website that Alan didn't receive, he said, "I found that I actually have not needed them and when the kids get stuck, if I can help them I will. But I find that the other kids are the best thing, the other kids help each other. It's so amazing! I'm so big on like those shout outs, students like to give each other shout outs." Alan was referring to the posters Google provides to CS First instructors that Google calls CS First Community Boards. Google advice to CS First teachers is to encourage students to "post words of thanks or encouragement, called a 'shout-out', to a [fellow student] who helped them." Some students progressed at a faster pace than Alan, and they are teaching him the curriculum. Among the curriculum resources available on the CS First website are posters where students can post shout outs to each other, and Alan found that this type of positive encouragement gave his middle school class a positive boost. Alan capitalized on students helping each other to make sure the class survived despite a lack of other teacher collaborators.

Alan said he saw the crossover between the storytelling module and English and math. Alan witnessed the open-mindedness of his students, which to him was "simply amazing". What really struck him was his students' level of creativity--a level that he himself could not possibly match. He credits the structure of the CS First challenges created by Google for giving the students the opportunity to really express themselves creatively. In addition to enjoying the CS First curriculum, Alan enjoyed the opportunity to interact with his students in ways that hadn't been possible in more traditional math and science courses. In order to accommodate diverse learners, Alan gives students who are struggling extra time to complete challenges as well as time at home to work on their CS First modules.

When asked to compare what he observed in the after school program versus what he observed in Alan's coding class, Aaron recalled that in the after-school program students had to work collaboratively to first solve a problem with a model and then to control that model using their computer. Aaron picked up on the physical computing aspect of what the after school students were doing. He noticed students having to communicate to design a viable model. "So I saw more students engaging with somebody else."

Aaron was motivated by the loss of enrollment to focus on marketing his school to parents. In addition to having some supportive PTA parents like the computer programmer who first taught CS at Arbor MS and the life coach who helped Aaron see how to better market his school, Aaron also mentioned a science partnership with a nearby university. Aaron attributes the success of the university science partnership "to the hard work, naturally, of our teachers here, but also with the help of the university."

Summary

Over a period of two years, Arbor MS had multiple teachers eager to bring an entry-level CS course to its middle school students. A CS club was first introduced at Arbor by a parent, and that club became an official course one year later. A combination of a lack of CS teacher training, a lack of a formal CS implementation plan, and an absence of CS partners and collaborators have slowed the potential CS implementation that was temporarily powered by an expensive commercial organization. In its third year of parent pressure to create more CS experiences and to help sustain its student enrollment, Arbor’s main plan was to continue offering the Google CS First curriculum as a semester-long course taught by Alan.

Site 2: Bloom Middle School, SUSD

Site Description and Background

Being a girls-only middle school has made Bloom MS more attractive to CS professional development providers like Exploring Computer Science (ECS), an organization actively engaging female CS students. Consequently, the Bloom CS teacher (Becky) agreed to participate in ECS training, even though ECS training prepares teachers to teach high school CS.

Bloom Middle School is a relatively new school site and was designed to encourage girls to be entrepreneurial. Being a single-gender middle school using the latest educational research, Bloom MS has a socioeconomically and ethnically diverse female demographic and offers innovative course curricula. Girls are bussed in from their home areas to get to the school that shares a campus with a high school.

Implementation Elements and Timeline

Becky came to teaching sixth grade computer science with a mathematics degree and extensive mathematics instructional training through Math for America. Math for America paid for her master’s degree in mathematics education. “The principal, Brenda, wanted students to have computer science instruction all three years that they attended Bloom Middle Schools. As a credentialed mathematics teacher, it was placed into my lap and it was like a ten-week rotational wheel. So I got to see groups of students ten weeks at a time.” In addition to CS expertise, Brenda looked for a teacher who would be good with students. Becky received training to teach ECS (Exploring Computer Science), which is a high school CS curriculum. As a result of her ECS training, Becky knew the topics that her CS students would need in high school CS, but she was never formally trained in middle school CS topics or on how to customize the ECS lessons for middle school students.

While ECS guided everything she did in the middle school course, Becky said that she needed more material designed for middle school students. She stayed faithful to topics such as equity, inquiry and principles, but she also indirectly discovered the more age-appropriate CS course she eventually used with her students, namely Code.org's Computer Science Fundamentals (formerly known as Course 2). Code.org offers a range of CS curriculum choices for grades K-12. Becky was familiar with Code.org, because it was a resource that was used in the ECS training she received.

According to Becky, students loved doing binary activities, like translating computer code into plain English, and designing magazine covers. Binary challenges and magazine cover designs are examples of the many "unplugged" activities Becky took advantage of, especially while she was waiting for the tech hardware to be delivered to her classroom where students could actually do computer programming. The ten-week course she taught, according to Becky, was a "mish mosh" of topics and she was told that whatever she found she should use. So she had "full reign" of what to teach students during the quarter. Becky also had the unique opportunity of teaching her sixth grade female students mathematics as well as computer science, which gave her more insight into her female students' computational thinking abilities and problem-solving skills. Becky shared that formal assessments were not a focus. The pair programming driver and navigator structure meant that students were actively involved at every level.

To illustrate her lack of preparation to implement all the materials available in her CS classroom, Becky said that when donors would visit the school to see what was being done with their donations, she would spend a weekend learning how the LittleBits electrical circuit kits worked. Her feeling about having curriculum resources but not having training was “I wish I had been forced to do this earlier.”

Brenda said that her past experience as a counseling coordinator overseeing 34 schools gave her a chance to connect with school counselors. She also stated that in her past administrative role, counselors saw her as a technology leader, because she had brought innovative technology to her school. At some point Brenda thought, “There are four or five excellent all girls’ private schools in the Sunshine city area, but the amount that it costs to go to those is out of the reach of most people.” It was her research about all-girls schools that led her to discover the National Coalition of Girls’ Schools, which led to her connecting with NCWIT. Brenda brought NCWIT’s Counselors for Computing to the counselors of the 34 SUSD schools she oversaw. Having been a high school assistant principal, Brenda was intrigued enough by NCWIT’s work in recruiting women to computer science to read the book *Stuck in the Shallow End* (Margolis et al., 2008). Commenting about the content of the book, Brenda said, “I was totally unaware, until I read the book. I was like, ‘Oh, okay. That's really interesting.’ And again, I’m not … I’m a history teacher. I don’t know a lot about all this, but I learn as much as I can.”

Becky hesitated when thinking about the impact of computer science education on the demographically diverse groups in her class. She said that special education students were quick to catch on and were proud of themselves. They had a more positive experience than they are accustomed to have. Becky observed that girls with individualized education programs (IEPs), who had more difficulty in math class, were quicker with understanding code. Becky further witnessed how their coding experiences gave special education students a greater sense of belonging in the classroom.

Implementation Challenges and Solutions

Trying to implement computer science without computers was a challenge, so Becky felt fortunate that she had been trained in unplugged activities. However, the momentum that Becky was building with students' CS knowledge through unplugged activities could not be transferred to computer programming on devices because the devices were not ready until the end of her second quarter of teaching the class. Even though her students eventually gained access to some Dell laptops, ninth grade science classes had the priority in using these laptops. As a result, Becky's CS class benefited from mastering the method of pair programming, which allowed for two students to alternate in controlling one computer. Pair programming resulted in greater student collaboration, but Becky said, "I think they wished they'd had laptops. The unplugged activities were intended to prepare them for laptop activities." Brenda made Becky's and the other computer teacher's classes a priority during the first quarter when hardware was limited to laptop carts shared by all teachers. "What we said as a school was [Becky's] class and the other computer teacher's class had the priority on using those carts, so everybody else had to kind of work around them."

The combination of not having access to a middle school CS curriculum and lacking any prior CS background made the implementation all the more challenging. Becky had to learn as she went. Furthermore, she did not have a professional CS network to turn to since the other ECS training participants were all high school CS teachers. Meanwhile, she and the only colleague at Bloom with whom she could have collaborated would each have had to teach more than four preps in order to have a chance to plan CS courses together. Ultimately, fewer preps was the option chosen over having two CS teachers learning from each other and reflecting on each other's practice of implementing new CS lessons. Becky called this challenge too much freedom due to a lack of a comprehensive curriculum. “As a math teacher, I would have enjoyed it [having freedom to plan math courses]. But because I was less familiar with computer science, and what was available, that was definitely the biggest challenge—just not knowing what [my students] were going to do. And then not really having a clear idea of what they were supposed to do in seventh or eighth grade, either. Just knowing that we wanted them in computer science in seventh and eighth grade. I wanted to vertical plan, but there was no one to vertical plan with.”

A lesson-planning challenge Becky faced resulted from the longer than average 90-minute class blocks. Becky made sure that if she planned two different coding lessons that both lessons did not require a lot of sitting. She tried to avoid having her students sit in front of a screen for all 90 minutes of a class.

During the first iteration of the elective wheel, Becky planned out “meticulously every minute of every day”. Becky recalled, “I would do every question myself the day before, just to make sure to answer the questions they had and then towards this last cycle of ten weeks, it’s been kind of the exact opposite. It’s actually more beneficial when they struggle with it and then succeed, as opposed to me just saying, ‘Oh, yeah, did you think about this?’ I have learned that me taking my hands off of the situation gives them a lot more room to grow and learn, but at the same time if I could do it over again, I would do the meticulousness over the summer before the year started.”

It wasn’t until the fourth cycle of the ten-week wheel that Becky finally mastered several key components of her CS implementation. She decided that programming partners would switch at the completion of each Code.org Course 2 level. The benefits of this constant partner switching were that neither the driver (student coding) nor the navigator (student debugging) ever became the “dominant” partner. Becky engineered a way for every CS student in her class to become more actively involved in the problem-solving process.

Implementation Supports and Resources

Becky credited student collaboration as a great support and resource for her classroom. According to Becky, switching seats and partners gave students a fresh start each day, so that focus was on new beginnings rather than past failures. According to Brenda, there was a lot of student collaboration happening in Becky’s classroom, which Becky called productively noisy. She said that as a result students’ problem solving skills became more efficient and concise, debugging lessons forced students to look at what was wrong and how to look at what was wrong. Slowing down the execution of code helped students see the value of looking at their solutions slowly. Becky observed that this new skill carried over into students’ mathematics problem-solving, and she heard her students saying “I need to look at this problem one step at a time” to locate the mistake.

As stated by Becky, 70% of the girls were excited and wanted more, and one student was able to complete a challenge in six steps even though the challenge called for 15 steps. “I was blown away!” exclaimed Becky. One benefit that Becky observed was how sixth grade students slowed down their debugging process. Through the Code.org activities, students were forced to look at their buggy code step by step in order to locate the bugs that needed fixing. This ability to slow down when looking for mistakes and hone in on a specific mistake was something Becky saw as a great benefit “not only in computer science, but in math.”

Seeing students problem-solving outside the mathematics classroom and seeing them be more successful in one or the other motivated Becky as a CS teacher. By her fourth iteration of the wheel, Becky figured out a way to help her CS students have mostly positive experiences with coding. The fact that she learned to allow her programming pairs to switch often meant that no student would be branded as being bad at coding. Brenda had multiple opportunities to visit Becky's CS classroom of 25 girls. Becky's CS course having been a ten-week wheel meant that Brenda could visit the same lesson multiple times over the school year and see the evolution of Becky as a CS teacher and the evolution of the lesson that was being delivered. One such lesson that Brenda made of point of seeing during each quarter was the "relay race". She saw students engaged in the work they were doing. She recollected the excitement in the room during the relay races. "Every time I've gone in that room, they've been engaged. Whether it's paper work or computer work, they're really engaged. They do a lot of group work, so there's a lot of questioning each other."

Brenda facilitated Bloom's students going to a movie screening of the film *Hidden Figures*, brought role models in engineering to the school, and she has partnered with state legislators to expose students to software that is still in development. Brenda is a firm believer in partnerships, and she actively pursues and maintains her partnerships. Whether it is an opportunity from Sprint or Girls Build LA or a local university, she said she has decades of experience in leveraging partnerships that bring about more computing exposure to students, especially students from disadvantaged backgrounds. However, despite all the support she had, Brenda still had to do a lot of her own groundwork. Becky's perspective on partnerships involved taking students on field trips to YouTube and JPL. Becky said it was difficult for her to partner with any of the high school CS teachers at the same site due to the already immense workload.

Summary

Over a period of a year and a half, Bloom MS had one teacher trained to teach CS to its middle school students. Becky was trained in a high school CS class, but she had to teach a middle school version of her training during four ten-week quarter courses. A combination of a lack of middle-school specific CS teacher training, a lack of a formal middle school CS curriculum and a delay in obtaining computers, as well as an absence of CS partners and collaborators caused teacher burnout in Becky. In its second year of trying to implement CS, Bloom's principal opted to hire a CS professional trainer to teach the sixth grade CS quarter courses.

Site 3: Crest Middle School, SUSD

Site Description and Background

Crest Middle School is situated in a bustling business region. It differs from the other five middle schools in this study by the fact that it rests in a very commercial neighborhood. In fact, one wouldn't know that Crest was a middle school just by looking at its outer façade, where it appears to be more of an apartment building in a bustling mixed-use neighborhood than a traditional public school site.

Crest MS opened its doors to students in 2009 as a co-educational middle school with a STEM focus that offered single-gender core courses like science and mathematics. With the goal of maximizing student learning, at Crest Middle School students are separated by gender for core classes. Chuck, the principal at Crest, has also secured educational trainings to help his teachers understand how boys and girls differ in how they learn in grades six through eight. As a result, Crest teachers have benefited from training in teaching methods spanning from project-based learning to collaborative classroom strategies.

Implementation Elements and Timeline

Carol, a science teacher at Crest, was recruited to attend an exclusive Code.org training in New Orleans where she learned to use a coding curriculum called Project GUTS (Growing Up Thinking Scientifically) that also teaches middle school science topics, such as the spread of epidemics and other so-called complex-adaptive systems (CAS). Carol did not need much convincing to use the curriculum. She was excited to be recruited as a GUTS teacher and as a potential GUTS trainer to other teachers. What she didn't anticipate is that SUSD would eventually leave its partnership with Code.org, and that left her place in the Code.org pool of trainers in uncertainty. She said that she did not feel that she should continue pursuing a professional development relationship with Code.org once its partnership with her district ended.

With tech support, students are paired up with teachers. Principal Chuck believes that "you just have to build that culture at the school where ... the kids come and set it up. [Teachers and parents] see the kids, and it is part of the culture. You have to de-mystify technology. These kids have more technology skills than adults, we all know that." Chuck has built a school where students have access to cutting edge hardware, and he trusts the students to take the lead with the equipment. He mentioned that during parent nights, the school's Mouse Squad students are in charge of the technology setup. Chuck proudly shared that his core teachers all hold single subject credentials, which he said is not always the case with middle school teachers. He was proud of the fact that he was a science teacher, and that he is a staunch advocate for STEM education.

An example of such a program that Carol gave was the *CS in Science* course she was trained to teach. She said, "I really like how computer programming can really teach the idea of logic and problem solving. I think a lot of our kids need that. And even though I may not know as much as some other people, I've tried to at least give the students a taste for it, if I can." Carol said that she did not receive any other CS training after her participation in the New Orleans Code.org training. However, Chuck said that Carol "is a national Code.org trainer." Carol said that she had looked forward to becoming more involved with Code.org when she was approached by SUSD's Instructional Department to be a Code.org trainer for other teachers in the district. "I almost had an opportunity to do, but then I realized, well, I would love to but all I know is the *CS in Science*, if I'm allowed to teach that [to other teachers]." Carol followed up with the district, but her district contact told her, "We're still trying to figure out if we can do that."

At the time, Carol did not know what the CS Discoveries curriculum was about. She learned throughout the 2016-17 school year that CS Discoveries was a CS curriculum that was general enough to be incorporated into any class. "CS Discoveries is something that should be more of an elective class", while "*CS in Science* is something a science teacher can integrate." She went on to say that CS Discoveries might be too much to expect a core teacher to incorporate into their busy core curriculum, but *CS in Science* made sense to use in a core science class because it covered the science standards. "I feel like [CS Discoveries] would be a hard sell to [other science teachers].

Some of Carol's science students were interested in *CS in Science* and others were not. However, Carol observed students actually creating scientific models through the block-based coding in StarLogo Nova, so she gave her students science credit for completing those coding challenges. Carol observed that her two worst behaved students in science class were totally engaged by the *CS in Science* coding lessons. Chuck believes that a middle school's STEM focus needs to start with sixth graders and with teachers with college degrees in the topics they teach.

He also believes in empowering the underrepresented children of immigrants and that STEM courses are more empowering than perhaps history and English classes. Carol also wished that students were exposed to coding platforms earlier than sixth grade. She saw that some of her students were frustrated when they could not solve a coding challenge, and they did not possess the necessary skills to bounce right back and persevere to a solution.

During her implementation of the *CS in Science* curriculum, Carol had access to MacBook Pro laptops for her seventh grade science classes and her STEM lab course. There was one laptop for every two students in her science classes and one laptop for every student in her STEM lab class. Unlike her single gender science courses, Carol's STEM lab class, where she implemented the *CS in Science* curriculum, was not a core class and was mixed gender. Carol observed that during the *CS in Science* instruction some of her misbehaving male students were better behaved when it was time to do coding. "They were argumentative and defiant students, but when we did the coding, they were really into it." Meanwhile, Carol observed that some of the other boys in her class struggled with the logical order of placing the blocks in their StarLogo Nova programs. "A lot of the boys were getting frustrated. Sometimes they were really having difficulty with getting how the blocks come together, I guess the logic part of it. I felt like the girls had an easier time with it. They were kind of neutral about it. There were some girls who were kind of into it and some girls who were not into it." To provide her students with continued encouragement, Carol said she did not play the role of strict enforcer when students were working on scientific models in StarLogo Nova. She gave them credit when she observed the general behavior of creating an agent-based model.

Implementation Challenges and Solutions

Without an official partnership between SUSD and Code.org to train teachers in *CS in Science*, Carol didn't see how sites would find funding to train their science teachers to use *CS in Science*. However, Chuck would rather have less teachers and larger class sizes and allocate the money to professional development instead.

Time to prepare CS lessons is something that Carol finds to be a deal breaker for teachers who already have too much to do and insufficient time to cover all the standards for their subject area. “The thing that really helped me integrate *CS in Science* in my classes is the fact that it was a curriculum that was easily accessible. I got trained for it and they basically gave us the resources, and I could literally go into my classroom the next day and with a little bit of work that night, you know just modify a few things. I think if I had to learn everything, I probably would not be where I am right now.”

In Carol’s opinion, for her to consider any CS curriculum in the future it is important to see the curriculum in action. “I think if maybe there was a video, where I saw it being implemented in the classroom, and then, I think I would do it. I think I would take the time to like mull over the curriculum. But if I were just to see the book that they gave me [in the CS for Science training], maybe not.” What Carol found made the *CS in Science* curriculum accessible was its prepackaged nature. “The fact that they literally did that prepackaging for me ‘you need to teach this part first, this part first, this part first, then this part’, they literally broke it down for you and gave you the lesson plan and resources, that definitely eased any anxiety I have about computer science.” Chuck believes that passionate teachers will take ownership of whatever they want. He finds that his job is to expose the teachers to as much as possible, and not to impose a specific program on a teacher. This is EC's organic way of empowering teachers and students to grow technologically.

In addition to adjusting to a new school in her second year of teaching, Carol was asked by her new principal at Crest to attend a Code.org training for science teachers. Chuck's suggested that Carol incorporate the *CS in Science* material in a STEM lab class, which is a more free-form science class than a traditional seventh grade science class. In reality, Carol's experience was that the Code.org training aligned more with her integrated NGSS seventh grade science class. Carol took what was provided by the *CS in Science* curriculum and modified it to suit her courses.

Chuck was seeking a curriculum that teaches the mechanics of robotics. "I haven't done a formal robotics program. I'd like to, but the ones that I see are kind of little gimmicks and packages, like build this packet, but it's not really...it kind of does all the creativity for you. It doesn't allow kids to really kind of, I don't know. Maybe I don't know a lot about it, but it sounds like a recipe. 'Just do the recipe and you do a program,' versus really learning the mechanics of robotics."

Implementation Supports and Resources

Carol said that she collaborated with two other SUSD teachers at the New Orleans Code.org training. Then she went on to say, "We were not able to collaborate again after that." After her summer 2015 *CS in Science* training, Carol set goals for herself as to which of the five Code.org modules she would incorporate into her science classes. She planned to do more than she eventually did. "I wanted to do ecosystems and chemical reactions, but for my first year I did the chemical reactions only. My goal for this year was to do the spread of diseases, chemical reactions, and ecosystems, but I only got through the spread of diseases." She had to pilot other curriculum from the district, which resulted in her not having time to implement the ecosystems and chemical reactions modules.

Carol expressed some ambivalence when she described how she tried to incorporate pair programming as a learning strategy. “I tried to as best I can. I introduced the idea of driver and navigator. Whether the kids truly did it was a different story. I'm a little bit torn about [pair programming]. I think that some kids were like, ‘you drive, I’ll navigate…you drive and I’ll kind of watch along.” She recalled not being stringent with the roles and asking students to reverse rolls. “I guess my reasoning was I wanted them to do whatever was comfortable for them.” However, in terms of verifying which student was doing the work and the problem solving, “I feel that [pair programming] would definitely be a problem, because I would be wondering if [the navigator] was really practicing their problem solving skills.”

Notwithstanding the ambiguity Carol mentioned in the partnerships that have approached her to pursue more CS opportunities for her students, Chuck mentioned having partnerships with several local universities and that the strength of those partnerships was important. “I think the number one is you’ve got to get Silicon Valley people to look like our kids, women and people of color. Silicon Valley does not look like our city and it’s mostly white and Asian guys. It’s hard for these people to come speak to our kids and have any grab. You can invite them, but they tend to go to schools that look like them. They tend to go to the schools with white and Asian kids versus brown and black kids. Everybody wants to volunteer in [the more affluent part of town], but they don't want to volunteer in our area. We are just on the wrong side of the city.”

Carol was satisfied with the *CS in Science* teaching resources and did not feel intimidated by the computer science concepts embedded in the science. She did not seek out other resources to supplement what *CS in Science* offered.

Summary

Crest MS had an enthusiastic science teacher trained to offer a cutting edge CS curriculum. After a promising first year of CS implementation, the district changed course in its CS focus, and that derailed the *CS in Science* implementation that was underway in Carol's classroom the Project GUTS curriculum. A combination of a lack of shared decision-making and communication between site administration, district administration, and the CS teacher, as well as a shortage of CS partners and collaborators has disrupted the *CS in Science* implementation that was catalyzed by a reputable Code.org training.

Site 4: Delta Middle School, SUSD

Site Description and Background

Delta Middle School is situated in a bedroom community with relatively older and affordable housing. Delta is several miles away from some well-known aerospace companies that have partnered with other middle schools to provide field trip opportunities for students.

Delta was the only middle school whose principal I was not able to interview for this study. Instead, Delta's magnet school coordinator, Dan, participated in the study as the administrator with the most knowledge of the CS implementation at Delta. Furthermore, Dan was just wrapping up his first year at the site, so he had the least amount of history with the site of the administrators I interviewed for this case study. In contrast, Doris, Delta's CS teacher, had the most teaching experience of all six teachers interviewed for this case study. What distinguished her further from other teachers in this study was that she was also a National Board Certified Teacher (NBCT) in science. She earned special distinction for her expertise in science curriculum design.

Implementation Elements and Timeline

In his role as magnet coordinator, Dan needed to “reboot” the magnet programs at Delta. It fell on him to support Doris’s robotics’ implementation. According to Doris, she ended up teaching VEX because no one on her middle school campus had the expertise to teach a VEX robotics course. “It fell on me, … because I have a lot of experience teaching science and mathematics.” Doris already had at least two years of success implementing LEGO Mindstorms robotics with Delta MS students. With the LEGO robotics training she received, she was comfortable and successful in designing robotics activities centered around the LEGO Mindstorms robotics kits. What changed in the 2016-17 school year was that her principal introduced two new robotics kits in Doris’s classroom from the VEX IQ platform. However, Doris did not get any VEX IQ training. Her many years of prior teaching experience, as well as her success with LEGO robotics, coupled with her enthusiasm to expose students to robotics and computer science made her the prime VEX IQ teaching candidate at Delta. She also had the support of the magnet coordinator, who had no training in implementing VEX IQ robotics. As a result, Dan reached out to a graduating high school student with high school VEX experience to spend one day a week working with students on implementing VEX IQ in Doris’s classroom.

According to Doris, “Two years ago (2014-15) it was an after school club. Last year (2015-16) it was an elective without the VEX though, just the Mindstorms. But still, the group of kids that I had last year were more motivated and they wanted the challenges.” Dan mentioned that identifying the right students for Delta’s robotics program is essential. He also made a point about distinguishing between students who are good with building robots and students who are more talented in programming robots. His ability to distinguish between these two robotics student groups is a result of his own experiences.

Students were not as independent in the 2016-17 class, according to Doris. “The students weren’t very motivated to take it as a challenge. They didn’t say, ‘Let’s try this. Let’s find out how this works.’ They all like sitting and waiting for me to give them instruction and literally do some stuff for them. It wasn’t like that with the year before. The students were so motivated with the Mindstorms. That’s what motivated me. Okay we can do this with the students being so motivated and finding out how things work.”

Dan identified the way a middle school student’s mind works and how that compares to what a robotics curriculum expects from students. He was mindful of the need for middle schoolers to have instant gratification and that robotics challenges can take more persistence than most middle schoolers are ready to offer. “We didn’t do much because the students were ... they weren’t ready. They weren’t mature enough to handle these things. They broke the chargers. Something happened to the battery and we replaced it and the whole year it was a challenge. It was a serious challenge for us so we couldn’t participate in any of the competitions but then we have to start somewhere. We took it that way.” Dan was also aware that the type of students who were scheduled to Doris’s sixth period robotics class were not necessarily the right fit for the course. He admitted that “scheduling the right kids into the class is essential. I don’t think that happened this year.”

Implementation Challenges and Solutions

The first challenge with Delta’s implementation of VEX IQ robotics and RobotC programming was the lack of teacher training. VEX IQ is significantly different from the LEGO Mindstorms robotics curriculum that it is not possible to learn on one’s own without professional guidance. The second challenge was how the teacher would incorporate the VEX IQ curriculum into her established LEGO curriculum while addressing the limited number of VEX IQ kits. She felt pressure to include the two VEX kits, but she didn’t know how to make them fit into a class that had already been doing LEGO robotics. She had two VEX kits for 34 students in a classroom that already had 15 LEGO kits for students to use in teams of two and three. She went from coaching multiple middle school teams to participate in LEGO robotics competitions to not having any students be able to compete. Among other things, she had to first figure out which of her students would get to use the VEX kits. While having the high school student with VEX experience in the classroom once a week may have seemed like some support for Doris, she did not feel it helped at all because the student herself was trying to figure out what the VEX IQ kits were all about.

With only two VEX IQ kits in her classroom, Doris could not see herself as implementing VEX and RobotC with her entire class. To her, VEX is still this extra thing that she can't see how to fit into her LEGO regime. Meanwhile, Doris also had to move the location of her class because of the VEX kits.

As far as access to a curriculum, Doris said, “We had a link. We didn't have software. It was a link. We went to a website and got the information.” When she recalled her prior experience with the LEGO robotics kits, Doris said, “The Mindstorms is okay. Even with the Mindstorms I didn’t have a curriculum. I created the lessons for them. Programming the robot to go straight and then make a left turn and then hit the wall and then back up. Things like that.”

However, venturing into a new curriculum without training was a “huge, huge challenge” for Doris. While she had developed a viable LEGO course during the 2014-15 and 2015-16 school years, in 2016-17 when she had two VEX IQ kits introduced into her classroom she said, “I couldn’t focus much on the VEX IQ.” She realized that LEGO and VEX could not easily be taught in the same class. “Separating the Mindstorm from the VEX because it’s so hard. You know you want to focus on the VEX, especially I don't have experience and I want to focus to see what's going on and then ignoring 32 students around you in the same room. It's like impossible." Dan mentioned that providing summer training for teachers to teach robotics is also a challenge, especially if it is not planned and scheduled far in advance.

Dan's honesty about having more questions and ideas than there are answers for led him to admit that he personally doesn't see how all schools that want to offer robotics can quickly do so. He indicated that schools faced an impossible feat without the immediate intervention by the school district to offer "ongoing and consistent" professional development to support the neophyte robotics teachers.

Implementation Supports and Resources

The teacher at Delta did not express any benefits of the school having tried to implement VEX IQ robotics. She bemoaned that students used to compete with LEGO robots, but there was no way to compete in VEX IQ competitions without a more comprehensive implementation—more VEX kits and teacher training.

Two students had to share each of the LEGO kits and six students shared the two VEX IQ kits. "Having the two classes in one was a challenge," said Doris with respect to using two robotics platforms in one class. Students participating in robotics competition also appears to be on Dan's radar for the future of CS education at Delta MS. Funding for competitions is fourth on his wish list after a makerspace, professional development, and field trips.

Summary

Over a period of three years, Delta MS had one veteran science teacher eager to bring more CS challenges receive LEGO robotics training to support an after-school robotics club. Then her LEGO program fell apart when her principal asked her to incorporate two grant-funded

VEX IQ robotics kits. A combination of a lack of VEX IQ robotics training, a lack of a formal plan on how to implement two different robotics platforms, as well as an absence of robotics partners and collaborators has negatively impacted Doris's CS implementation. Going into their fourth year of CS implementation, Delta's magnet coordinator and CS teacher did not have formal plan on how to improve their CS implementation.

Site 5: Eagle Middle School, SUSD

Site Description and Background

Eagle MS serves approximately 700 students in grades six through eight, many of whom are socioeconomically disadvantaged (66%) and reside outside a neighborhood surrounded by million-dollar homes. The neighborhood surrounding Eagle MS is predominantly white, but white students represent only 8% of the school's student population. The school's 60% African-American students and 28% Hispanic/Latino students come from outside the neighborhood mostly by school busses.

Student enrollment at Eagle Middle School (EMS) has decreased by about 13% over the past six years. As a result, critics refer to EMS as under-enrolled and cite the relatively static mathematics proficiency rate of about 10% for eighth graders as a reason parents would not want their children to attend this STEM magnet. However, science achievement results for eighth graders over the last two years show an improvement of about 5% (2015-16 School Accountability Report Card). Reform was not an option for Eagle MS. It was a necessity in a neighborhood of affluent families who would rather send their students to private schools than send them to Eagle MS across the street. Essential partnerships such as the one Eagle forged with nearby Mount Sunshine University (MSU) sent a message to parents that Eagle was serious about reform and that parents should enroll their children in their local school and maintain the pipeline between local feeder schools.

Implementation Elements and Timeline

When Eagle's principal, Evelyn, realized that parents were opting to send their children to local middle schools that offered robotics and computer science opportunities, she decided to visit those schools. Evelyn discovered that these nearby schools were offering Project Lead The Way (PLTW) STEM courses. It was the need to stem the loss of students to these schools and receiving iPads for every student that helped Evelyn and her team to choose to offer PLTW courses. With the urgency and the hardware in hand, Evelyn and her team applied for a multi-million dollar STEM grant.

In 2013, Eagle MS was one of a network of magnet schools to receive a three-year $2.6 million dollar federal grant to help with its magnet implementation of several PLTW curricula, including an automation and robotics course that incorporates computer programming. PLTW is an organization that offers STEM-focused teacher training and certification. According to Evelyn and Emily, one of Eagle's PLTW teachers, PLTW is one of the more expensive options available for VEX robotics training. When teachers at Eagle were asked if they would be interested in being trained in and teaching any PLTW courses, Emily volunteered. She recalled, "I said 'yes' just because I was familiar with the program. My own children participate in Project Lead The Way classes in their school. That's why I knew about it, and I felt like it was a great program that I wanted to be a part of."

In summer 2014, Emily attended a two-week "intense training" to get PLTW certifications in Computer Design and Modeling and Automation and Robotics—two of PLTW's foundational middle school STEM courses. Emily remembered, "I had no idea what I was getting myself into. I guess I could have been tipped off by the 20 hours of homework I had to do just to be able to take the class. I had all these assignments that I had to complete before I was even able to really register for the class." Emily admitted, "I had never done any type of robotics or computer programming or anything before [the PLTW training] at all. I didn't know about gear mechanisms, I didn't know about gear ratios, I didn't know any of that stuff. I was at a huge disadvantage in the class."

Despite being overwhelmed by the two-week PLTW trainings she received, Emily felt optimistic and excited about going back to Eagle and teaching her students everything that she had learned. "I just thought it would be a breeze. I would just go in and I would teach it and I would do it just like we did in training and everything would be fantastic. I did it. I passed the classes, I got 100%, I can teach them now." At the start of the 2014-15 school year, Emily said she was disappointed that she would not be teaching the intended PLTW course first semester. At the same time, she was told that she needed to figure out a way to weave the PLTW material into the actual course she was scheduled to teach. When she was able to teach the intended PLTW Automation and Robotics course in the second semester, Emily continued to identify as an "absolute beginner" at robotics.

Evelyn and Emily both confirmed that the PLTW curriculum was not lengthy enough to fill an entire semester. As a result, Evelyn said that a teacher has to be dynamic enough to want to stretch some lessons and be able to "expand and improvise or bring in your own lessons. It came down to those persons who had the desire, not so much the knowledge base." Evelyn learned that selecting potential PLTW teachers strictly by their credentials would be a mistake. Her advice was, "Don't go by what the credential is." It was evident to Evelyn from the time Emily agreed to participate in PLTW training that Emily had the requisite level of commitment.

According to Evelyn, teacher credentialing becomes an issue when a school is not a magnet school. “So they would like the teachers to have the tech education credential, which obviously my teachers do not, but because we’re a magnet school we’re able to have teachers teaching out of their program. I know of some schools that are not magnet but offering Project Lead The Way, having a tough time with making sure their teachers have the appropriate credentials.”

Evelyn noted that the PLTW training was content specific, and that middle school PLTW teachers need additional training to customize the robotics curriculum to middle school students’ developmental needs. Evelyn could have let her teacher figure out how to bring PLTW classes to the middle school classroom level, but instead she invested in Kagan and Buck Institutes trainings. So Emily had access to Kagan’s collaborative learning structures and the Buck Institute’s Project Based Learning strategies in order to implement the PLTW with greater fidelity and age-appropriate collaborative learning strategies.

In the computer lab, only approximately half of the Windows desktops had the RobotC programming software installed on them. As a result, Emily’s students had to share a computer to program in RobotC, which meant that students could not always program when they wanted to. She added, “Our school is a one-to-one iPad school, but we don’t use any of the apps for RobotC, but [students] use iPads to look at the assignment sheet during building.”

Implementation Challenges and Solutions

Eagle’s process of providing students with access to computer science opportunities came with several challenges. During her training, Emily felt pressured to succeed in her PLTW classes from multiple sources. First, each course was costing the school $2,000 of grant funds, and each of her two classes had to be passed by at least a 75% success rate. To Emily it felt more like she had to get a 100%, “otherwise you don’t get certified to teach the class.”

Emily had no budget constraints with the approximately three million dollar grant, and she requested whatever she could and it was all purchased for her. She observed high schools that compete in and win robotics competitions with only a quarter of the extra materials that her students got with the grant. Budget constraints couldn't fund a full-time PLTW elective teacher, so early in the grant teachers had to continue to teach their core subjects.

Emily credited her skillfulness in teaching low income student populations with behavior issues. Her ability to keep discipline in her classes is what she named as the key secret ingredient in persevering through the many obstacles of teaching robotics. Emily specifically advised future teachers taking on a new robotics curriculum to persevere through that very rough first year.

Due to funding constraints embedded in the magnet grant, Evelyn could not use the magnet money to fund new electives teachers. Another obstacle to offering the intended computing curriculum at the middle school level were non-existent course codes for the PLTW class that Emily was teaching. For example, an advanced robotics class could not be offered on its own, because there was no district course code for advanced robotics in the district course catalog. Instead, Emily had to simultaneously teach the advanced students and the regular robotics students in one class, which made it difficult for her to deliver the intended curriculum.

Her average robotics class size was 40 students in grades six, seven, and eight mixed together in one class. When it came to special education students mixed into the robotics elective, Emily saw issues with the lack of support she received as far as accommodations for these students. According to Emily, special education students were used to small special day classes (SDC) of up to nine students similar to themselves. Emily said, "They come to my class where there's 45 of them and they're sixth through eighth grade and it is a very different class." Some of her students were non-verbal, autistic children. "They really struggle with the class, [because] they don't sit and read from a book or sit and fill out a worksheet. They've got to get up, they've got to move around, they have to manage their time." This group of special needs students made Emily's class a whole other situation, which she dealt with mostly on her own. "Elective classes are not the priority for the special education department, so I don't get assistance. No one pushes in to help them other than my two autistic students." Emily also believed that the site administration needed to be extremely patient and supportive of their new robotics teachers in understanding that the novice robotics teacher is learning as she's teaching it. "It's unfortunate that you've got to figure it out while you're teaching students. If your administrator is understanding about that and not expecting some world-class robotics program in the first year, then I feel like for me it went a long way to letting me figure it out and get comfortable with it so then the kids could be comfortable with it."

Emily's biggest challenge was not having any choice in who gets to take robotics, as well as not being able to separate the younger 6th grade students from the older 7th and 8th grade students. She felt that 6th graders are very different from 8th graders, and that mixing these two groups who are at completely different places cognitively can create a very counterproductive atmosphere in a robotics class.

According to Emily, there were so many challenges that she had to juggle in terms of classroom management, that it was impossible for her to develop as a seasoned CS instructor. After teaching PLTW for three years, Emily wanted to make CS a greater focus than just 25-30% of the class. Emily also taught 6th grade English and history, in addition to her four robotics courses. She missed teaching high school English, and she admitted that she would be happier if she didn't have to teach any middle school English content. She had a lot to prepare for her robotics students, so to her having to teach a core class was exhausting.

Emily was initially under the impression that it would be “super easy” to instantly implement what she learned from PLTW during her first robotics class. She was shocked when her first attempt at teaching the robotics class was much more challenging than she had initially anticipated. Emily was confident enough to decide not to require homework for her elective classes to prevent student apathy and resentment for the class.

Implementation Supports and Resources

Five teachers attended the PLTW trainings, however only one other persevered to year four of the grant like Emily did. Emily regularly texted or emailed with at least two other robotics teachers from two of the other grant schools. She recalled, “We leaned on each other a lot in the beginning, especially to decide how to store the materials and student projects.” Evelyn sought district technology support by applying to be part of a technology integration cohort. “We were a little confused as to why we were denied and then we were told that our needs were a little more advanced than what the cohort was going to focus on.” Evelyn continued, “But I still need support.”

Not all of Emily’s students flourished in robotics, because some students were scheduled into Emily’s classes by school counselors even though these students might prefer to take a dance or a drama class instead. She also pointed out that a large class with 45 students did not accomplish nearly as much in one semester as her smaller 18-student class from an earlier semester. Her lack of control over who got into the robotics class was a big challenge for Emily.

Yearly field trips include Northrop Grumman, Boeing, and LAX. Evelyn shared that female engineers are involved with career day and come into Eagle MS to train teachers on how to prepare students for the future engineering workforce. Skills such as collaboration. One obstacle to collaborating with other middle school principals is that CTE information is “very much a high school thing”, according to Evelyn. So it isn't uncommon for middle school principals to be entirely unaware of CTE. The shortage of partnerships put the brunt of the work on Evelyn’s teachers. Evelyn made sure that engineers come to Eagle MS to talk to students about what it takes to succeed in engineering, because “we really wanted to make sure that we weren’t teaching our kids something that would be obsolete by the time they got to the workforce.”

One of the disadvantages for Emily was that her PLTW trainings were scheduled back to back and they happened towards the end of her 2014 summer break. In addition, she indicated that she did not have access to a robot kit to use during training. Instead, she planned but wasn’t able to practice what she learned in training during planned trips to her school before the summer ended. Her school had already purchased all the VEX kits needed to teach the course. Furthermore, she discovered at the start of the 2014-15 school year that she wouldn’t be teaching the class during the fall semester. At some point Emily was informed that the she would be the only robotics teacher at Eagle MS, so she inherited 20 VEX EDR robotics kits and ordered 20 more and took it upon herself to organize thousands of robot parts herself. Emily admits that the “teacher part of me took over and I got a little crazy.”

According to Evelyn, one of the benefits of the MSAP funding was being able to train the tech-minded new staff in Kagan’s collaborative learning and Buck Institute project-based learning (PBL) strategies. Evelyn felt fortunate that those teacher trainings occurred early in the grant implementation timeline and before the technical, content-driven PLTW trainings. By the time PLTW trainings happened, staff had experience with specific learning strategies that they could fall back on that would make implementing PLTW curriculum more successful. Evelyn was unequivocal about the fact that none of these expensive pedagogical trainings would have been possible without the MSAP STEM magnet grant.

After ten weeks of studying the robotics parts, the students are then qualified to receive the robot parts they need to build their robot. So during the first ten weeks of the semester, Emily had access to curriculum that allowed her to teach students about all the parts on their inventory sheet.

Meanwhile, Evelyn underscored the importance of encouraging her PLTW teachers to pursue training and collaboration. “We're still continuing to have [our science teacher] go to [PLTW] training even though he’s not currently teaching it so that next semester when we offer it he knows if there’s been any changes to the curriculum.” Through a partnership between SUSD and Trash for Teaching, Emily received a more engineering-minded classroom that Evelyn called a ‘flex engineering lab’, that is similar to a makerspace. Consequently, Emily’s remodeled robotics classroom was designed with students in mind. She had access to a computer lab and an adjacent space that housed six large rolling tables.

Summary

Over a period of three years, Eagle MS had one enthusiastic English teacher eager to bring more CS challenges receive VEX EDR robotics training to teach Project Lead The Way automation and robotics courses. Emily and Evelyn have witnessed the positive effects on their students of having a chance to learn to build and program robots in middle school. Despite her unfailing perseverance to succeed as a robotics teacher and the autonomy given to her by her principal, Emily continued to struggle with having to prepare for and teach middle school English and having students in her robotics classes who did not seem to want to be there.

Site 6: Falcon Middle School, SUSD

Site Description and Background

Falcon Middle School (BMS) is surrounded by charter schools and elite private schools, several bars, and a variety of fast food establishments. Just a few blocks away from the school is a freeway separated by cement walls riddled with graffiti. Almost 90% of the students attending Falcon qualify for the free and reduced lunch program.

Felix, the computer science teacher at Falcon, teaches several one-year Project Lead The Way (PLTW) courses, including robotics and automation. Felix has a background in geology, has coached basketball, and worked for almost a decade with what he called “troubled youth”. With 18 years in education, he was told by some who observed his coaching style that he would be a good teacher. He used to teach physical science at Falcon before he became PLTW certified. Now he only teaches PLTW courses. Felix recalled, “When they moved me to teach [robotics], I thought ‘I can’t believe they’re paying me to play with robots all day.’ That’s what I would say. The principals, they love coming into my room because we’re always doing stuff. People say, ‘Man, it looks like you’re having fun all the time.’” For the past three years, Falcon’s robotics teams—coached by Felix—have been prominently highlighted on the school’s website and in district publications. Articles written by the principal, Fred, feature star female robot programmers and use words such as “excellence”, “prestigious”, “commitment”, “dedication”, “hard work”, and “champions”. During the same period of time, student enrollment at Falcon has increased by 19%.

According to Fred, Falcon’s STEM theme meant that all core subjects would include STEM concepts or that the school would develop electives that would be dedicated to teaching the STEM theme. “Once the MSAP team established what was the most important, namely

English, math, and electives, we realized where we had to dedicate the most instructional time to get the biggest bang for our bucks and stay within the 16-hour constraint that was part of our magnet theme.” The result was a STEM-focused humanities pathway made up of the English and history cores using informational text from science and history. “We also incorporated the engineering design process—normally taught in an elective class—into our science classes.”

“Pushing the bell curve forward” was Fred’s goal, and this is what he said state policymakers could learn for future school design. Ultimately, by widening the net to capture many more STEM students in the STEM electives, Fred wants to make sure that the school’s STEM focus is not reserved for the “best and brightest”. By increasing the opportunities for exposure, Fred wants the STEM experience to be available and accessible to a diverse group of students. “With the proliferation of computer science lessons in various STEM electives, such as film, dance, and graphics, the chance that students at Falcon will get computer science exposure during their middle school years is so much higher than at the average middle school.”

Implementation Elements and Timeline

In 2013, Falcon MS was among a network of magnet schools to receive a three-year $2.6 million dollar federal grant to help with its new full magnet implementation of several PLTW curricula, including an automation and robotics course that teaches computer programming. When word got out that the school needed a robotics teacher and coach, Felix went straight to the principal and declared that he *had* to be that teacher.

Recognizing that teacher recruitment to a STEM focused magnet middle school would be difficult, Fred and his magnet “came up with a great plan to ask candidates what their hobbies were.” Fred remembers how soft-spoken Felix was during his interview. “It wasn’t until he said he liked building houses on the side that we realized we had someone special,” recalls Fred.

During the 2014-15 school year (Year 1) and the summer of 2015, Felix was required to continue Project Lead the Way (PLTW) training with his MSAP cohort. Felix mentioned that follow-up trainings during Year 1 were not as productive for him, “because everyone is so busy. During the summertime, when they actually force us together, we actually do stuff.” During the school year, Felix had questions that he could not answer for himself or through other means. In terms of getting his own questions answered, Felix said that he did not benefit from many of the follow-up trainings. He found that the other teachers in his cohort who struggled with robotics turned to him for help during the follow-ups. “They always ask me about robotics stuff, that’s what I mostly teach them.”

Having finished his second full year of teaching robotics, Felix admitted that he was happy to have only regular and advanced robotics, as well as a design and modeling class. Felix confirmed his principal’s belief that teachers who teach electives with a passion for the subject should be rewarded with more sections of what they love to teach.

Implementation Challenges and Solutions

Felix had to choose what parts to implement and when to implement each part of his robotics training. He said he didn’t get enough of the coding training with his PLTW cohort, so he had to learn much of the coding on his own. Felix did not have any computer programming experience before his PLTW training, and his chance to learn more programming in his PLTW training was diverted by fellow trainees who asked a lot of robot-building questions of the instructors and Felix. Eventually, a veteran STEM trainer at one of the mandatory district follow-up trainings helped Felix learn to be a better RobotC programmer.

At the beginning of year three, he had set a personal challenge for himself as a computer science teacher and for his robotics teams to become “top ten teams in programming [in VEX robotics competitions], because some of the advanced students are really gifted programmers." He was disappointed that Falcon robotics' students did not come in top ten in programming during their Year 3 competition season, but he was optimistic about reaching this goal in Year 4. He said he wished he had learned how to access the Computer Science STEM Network or CS2N portal, where students could have done more programming challenges. However, learning to use CS2N was not part of Felix's PLTW training.

Meanwhile, Fred had a plan for how to scale up the motivation that Felix has seen in his classroom. "Our experiment [in Year 3] was to take Felix's class and instead of having 30 students build five competition robots, that we have 150 students build 30 competition machines and we compete right here on our campus before we go to competitions [against other schools], not because we think it's more fun but we think it's more rigorous." While writing the three-year MSAP grant, Fred had the opportunity to do extensive research on PBL. He also attended the Buck Institute's PBL training. PBL was something that he felt was right in his gut, but the training confirmed "that there was something wrong with the [traditional] middle school curriculum design." So Fred transformed Falcon's structure to fully embody the STEM magnet theme. As a result, Falcon began to offer students two electives as a way to help middle schoolers discover their interests or specialize in what they love.

Fred said that Falcon also wanted to obtain VEX IQ kits with mostly LEGO-like plastic parts because they are so popular and even more accessible than the more rugged, metallic VEX EDR kits. "I never put in computer labs to put in computer labs." Fred emphasized wanting to avoid the "field trip"-like experience of taking students to the computer lab once in a while. Buying Felix laptops, Fred confirmed, was because that's what the robotics class needs. "They can’t use Chromebooks. So we are removing the desktops and putting in floors, so that [students] can run the robots.”

As a believer in project-based learning (PBL), Fred said that “robotics lends itself to authentic project-based work. Research says that if it's authentic, then it doesn't matter if they're working on the project for an entire year. Students build a lot more ‘neural pathways’ by researching authentic projects for even a year, than they would with new topics each week. So we really subscribed to the project-based learning model.” Robotics projects in Felix’s PLTW courses typically have a written component. Felix said that his more studious students are able to write their project reports, but students he categorizes as struggling have issues with written reports. Felix recalled how the lower achieving students would write only one sentence instead of a multi-page summary report for a class activity. “They were eager to just build stuff. They say, ‘I’m done! Can we build now?’” Faced with the challenge of motivating students to write more about their robotics experiences, Felix decided to ask his struggling writers to draw their ideas. “Some of them like to draw more than they like to write,” he said, “so at least they’re able to draw something.” Fred said that Felix would be receiving literacy training to help him with teaching the writing components of the PLTW curriculum.

Fred also acknowledged that a robust robotics program requires adequate funding, and in the economically strapped Falcon neighborhood it is not realistic to expect parents to finance their children’s robotics activities. To that end, he established a GoFundMe webpage with a goal of $5,000 to help pay for travel expenses for students to attend robotics competitions.

Implementation Supports and Resources

Although he said his robotics classes are approximately 60% male and 40% female students, Felix mentioned that the females are more likely to understand robotics concepts than the male students. In fact, Felix recalled how his female students complained about having to collaborate with some of the boys in the class. According to Felix, the females said the boys were more interested in playing around than doing serious work. However, Felix said he tried to maintain heterogeneous groups in his robotics class, despite the girls' requests to work only with other girls. He would tell them, "'This is part of learning. You have to work with different groups.'" At the same time, Felix pointed out that he liked to "switch up" the teams regularly, because "otherwise they start misbehaving after a while."

While Felix described his advanced robotics students as "obsessed", he described the students in his beginning robotics class as getting bored quickly. His advanced robotics students expressed their wish to stay in robotics class all day long. "[The advanced robotics students] are not like the smartest kids, but they are the ones who are willing to learn, because even if they fail they want to keep going." Another way Felix distinguished his beginning robotics students from his advanced robotics students was by the type of programming tasks he could assign each group. According to Felix, the beginning robotics students generally stop at programming virtual robots on their computer screens. Felix observed that the advanced students preferred to run their computer programs on physical robots that they had built.

Felix explained that in Year 1 he introduced students to text-based RobotC at the beginning of the course, because that is all he knew and "that's what they taught us in the training". Felix believed that his robotics students generally spent about 30-40% of class time programming in RobotC. However, Felix discovered that students struggled with their textual code. Due to the physical setup of his classroom, it was difficult for him to get to every student computer to help students debug their code. As an alternative to personally helping every student who was stuck, Felix decided to display snippets of the correct RobotC code on the front board.

He said, “The problem with the code on the board is that the students either cannot see it or they forget it’s there.”

Felix also recalled RobotC syntax errors slowing down his class. He remembered his students getting stuck because of a missing bracket or other RobotC syntax character that would prevent an entire program from running at all. His students would yell out, “You typed a name in wrong. You forgot to capitalize it… Oh, you forgot a bracket over there.” Felix found that these errors hindered his ability to motivate his students to want to do more programming. As a result of the frustrations caused by text-based RobotC with his novice student programmers, Felix decided to switch from text-based to graphical RobotC. After the switch to graphical, Felix saw more students nodding and saying, “Oh, forward. Oh, backward. Turn left, turn right.” He said, “They liked [graphical] because they could see it on the screen. ‘Oh, it’s doing it. Oh yeah!’”

Since the second semester of Year 1 of his PLTW robotics course, Felix has stuck to using RobotC graphical. Despite observing that block-based programming commands were easier for his students to use, Felix saw students continue to struggle with the logical order of a program. In fact, Felix witnessed students missing opportunities to kick-start their robot programs with RobotC starter files that provide students with the beginning blocks of a program. Even when Felix showed his students how he used RobotC starter code, his students would often forget to take advantage of the starter files themselves and would struggle with how to start a program. Refusing to give up on his students despite their difficulties with programming, Felix gave his students an opportunity to program the robots for autonomous challenges (i.e., without a joystick), as well as teaching them how to program the VEX joystick for more efficient robot movements. Students who learn to program their robot joystick can send a sequence of movements (i.e., a mini program) to their robot’s microcontroller with the press of one joystick button. Each button of the joystick can be programmed to execute different “mini programs”.

Summary

Over a period of three years, Falcon MS had an enthusiastic science teacher and principal eager to bring more CS opportunities to students. Teacher Felix received VEX EDR robotics training to teach Project Lead The Way automation and robotics courses. Felix and Fred have worked together to implement a robotics programs that continues to grow and provide computer science access to more and more students. With entry level and advanced robotics offerings, as well as incorporating key literacy and other academic skills into electives courses such as robotics, Felix and Fred are moving Falcon Middle School towards a complete project-based learning experience for all students.

CHAPTER 5: CROSS-SITE ANALYSIS

In Chapter Four, middle school administrators' and computer science teachers' provided a detailed description of the computer science education implementations at six different public middle schools in a large urban district with a 50% or higher minority and/or socioeconomically disadvantaged student population. In this chapter I answer the following research questions as I compare and contrast the data from all six sites (S1, S2, S3, S4, S5, and S6) in a cross-site analysis:

1. According to middle school administrators and computer science teachers in a district with a large number of socioeconomically disadvantaged (SED) and traditionally underrepresented minority students (URM), what are the essential elements (i.e., teacher, technology, funding, etc.) necessary to offer computer science courses to its middle school students?
    a. What are the essential elements necessary to sustain a computer science education program?
2. According to middle school administrators and computer science teachers in a district with a large number of SED and traditionally URM students, what are the challenges they faced while implementing a computer science curriculum in their schools?
    a. How did the challenges vary, if at all, by student population (i.e., URM, SED, special education students, female students, male students, etc.)?
    b. According to middle school administrators and computer science teachers, how has the school and/or district addressed the challenges?
3. What supports and resources do middle school administrators and computer science teachers say they need to help them integrate computer science into the middle school curriculum?

a. How do the supports and/or resources vary, if at all, by student population (i.e., URM, SED, special education students, female students, male students, etc.)?

Research Questions 1 and 1a: Analysis of Implementation and Sustaining Elements

When analyzing each of the six sites' implementation elements and timelines, participants were aligned with Rogers' innovation adopter categories (see Table 6). It also helped to look at the essential elements of implementation at each site through Rogers' five stages of diffusion (see Table 7).

Table 6

CS Implementation Initiators, CS Teacher Backgrounds, and Admin/Teacher Adopter Categories

Site Name	Initiator	CS Teacher Background	Admin/CS Teacher Adopter Categories
Arbor MS	Parent	Science	Late Majority/Early Majority
Bloom MS	Principal	Mathematics	Early Adopter/Early Majority
Crest MS	Principal	Science	Late Majority/Early Adopter
Delta MS	Principal	Science	Early Majority/Early Majority
Eagle MS	Principal	English	Early Adopter/Early Adopter
Falcon MS	Principal	Science	Innovator/Innovator

Table 7

Essential Elements of CS Implementations Aligned with Five Stages of Innovation (Rogers, 2003)

Site Name	Administrator	Teacher	Curriculum	Student-Centered Theories
Arbor MS	Agenda-Setting (1)	Matching (2)	Basic	1
Bloom MS	Matching (2)	Redefining/Restructuring (3)	Basic	2
Crest MS	Matching (2)	Redefining/Restructuring (3)	Intermediate	3
Delta MS	Matching (2)	Matching (2)	None	1
Eagle MS	Redefining/Restructuring (3)	Clarifying (4)	Advanced	6
Falcon MS	Clarifying (4)	Routinizing (5)	Advanced	6

Administrator Expectations, Vision, and CS Identity

Across the six sites, the administrators indicated they were aware of the importance and urgency of offering computer science instruction to middle school students. All six expected the CS teacher to teach students computer science concepts, and the administrators supported their CS teachers' efforts to teach CS. These elements of administrator behavior fell under the first stage of diffusion, or agenda-setting (see Table 7). Rogers (2003) defines the agenda-setting stage of diffusion as identifying "general organizational problems that may create perceived need for innovation (p. 422)." Aaron, the administrator at the largest of the six schools reached the first stage of diffusion, because his enrollment was dropping and he needed to introduce an innovation to compete with nearby non-public schools offering innovations such as CS courses. The social network that guided Aaron to the agenda-setting stage consisted of some of his teachers, including Alan, and several parent volunteers, including one parent who was a computer programmer. Only Brenda, Evelyn, and Fred stated that they had done extensive research about the need to teacher computer science in middle school, which is consistent with Rogers' (2003) findings that "[early adopters] are motivated to seek further information about the innovation in order to cope with the uncertainty that it creates" (p. xx). Aaron, Chuck, and Dan acted more like the majority of administrators in the district and across the country who give into the uncertainty of implementing computer science courses.

At the newest school, Bloom MS, and the two schools with the highest Hispanic student population, Crest MS and Delta MS, the administrators had reached the second stage of innovation or matching, defined as "fitting a problem from the organization's agenda with an innovation" (Rogers, 2003, p. 422). Brenda, the principal at the smallest and newest school, chose to send her high school and middle school CS teachers to the high-school level Exploring

Computer Science (ECS) training. The high school ECS training was not the right fit for what the middle school CS teacher would have to teach, but Brenda had no other training options for her middle school teacher at the time. This is why diffusion at Bloom did not go beyond stage 2. At Crest Middle School, with the highest Hispanic population, the principal matched one of his science teachers with the right training, which sent science teacher Carol to New Orleans to learn the Project GUTS curriculum. However, the collaboration between Project GUTS, Code.org, and SUSD changed, so the diffusion at Crest did not move beyond stage 2. At Delta Middle School, the site with the second highest Hispanic population, the principal introduced a new robotics platform into science teacher Doris's existing robotics classroom. Instead of offering training, the magnet coordinator at Delta Middle School sent a high school graduate with some robotics experience to help support Doris. At these first four sites, the CS implementations based on the administrators' descriptions did not get beyond the initiation phase of CS implementation, which consists of the first two stages of innovation—agenda-setting (identifying a need for innovation) and matching (fitting the need with an innovation).

Administrators stated two reasons for implementing CS at their sites. Aaron and Evelyn said they implemented CS to maintain their student enrollment after they had lost some students to nearby schools that were offering CS opportunities. Brenda, Chuck, Dan, and Fred said they implemented CS to prepare their students for future college majors and careers that required more CS knowledge. However, the CS identities of the six administrators interviewed for this study were different. Aaron, Chuck, and Evelyn had a general idea of CS instruction. They did not use the technical CS terminology that Brenda, Dan, and Fred used when describing the CS implementations at their respective sites. Aaron, Chuck, and Evelyn all relied on their mathematics or science backgrounds to promote the general idea of STEM education at their respective sites and lacked CS experience to be strong and targeted CS advocates. They had the most laissez faire attitudes towards their CS teachers. Dan, who had a technology background, was slightly more comfortable expressing his vision of the benefits of middle school CS instruction. Dan offered some specific CS support for Doris by matching her with a high school graduate with some VEX robotics experience.

On the other hand, Brenda and Fred had read specific research on CS implementation. They had each witnessed the impact of CS implementation on middle school students. They were the two administrators most comfortable with describing in CS terms what they observed in their CS teachers' classrooms. Brenda's description of watching Becky's students doing CS unplugged activities was richer in CS concepts than the more general descriptions used by Aaron, Chuck, Dan, and Evelyn to describe their observations of their respective CS teachers' lessons. Fred had the highest level of CS descriptiveness when he described his observations of students programming robots at a VEX Cortex competition. Fred was also the administrator who used his CS knowledge and experience to conclude that middle school students at Falcon Middle School needed to have multiple robotics platforms and that computer science could be taught across the curriculum. Fred observed that the VEX Cortex platform was too advanced for the majority of students at Falcon Middle School, so he concluded that he needed to purchase the more entry-level VEX IQ robotics kits to implement CS across the entire campus. Fred's vision aligned with research that supports teaching computer science in a cross-curricular setting as a way to help female and minority students see the real-life applications of computer science (Carruthers, Milford, Pelton, & Stege, 2010; Rodger et al., 2014). Research about Fred's cross-curricular and project-based learning views of implementing CS indicated that they led to greater student engagement in CS (Burke, 2012; Rodger et al., 2014).

Principals Emily and Felix were instructionally responsive to their CS teachers' needs. Offering literacy training and collaborative learning training was a response to the CS teacher's needs that could have hampered the CS implementation if not addressed. Only Evelyn and Fred spoke about the importance of continuing education for their CS teachers, because they were mindful of how CS curricula change from year to year.

Teacher Expectations, Opportunities and Recruitment to Teach CS, and CS Identity

In accordance with prior CS research, none of the six teacher participants in this study had formal computer science education, yet all six CS teachers came to their respective CS implementations with initial optimism about implementing CS (Margolis et al., 2008). Alan, Emily, and Felix volunteered to be CS teachers, while Becky, Carol, and Doris were recruited to teach CS. The three teachers who volunteered to take on CS at their sites were eager to continue teaching CS in the future. However, as a result of prolonged isolation in trying to implement CS without ongoing guidance or specific curriculum, Becky, Carol, and Doris were not as eager to continue teaching CS. They were more interested in returning to only teaching their core subjects of mathematics and science. In contrast, Alan, Emily, and Felix were eager to teach only CS. Alan was looking to focus only on technology courses. Emily was looking forward to not having to teach any English courses. Felix was already only teaching CS courses.

With respect to CS teacher identity, the six teachers spoke either as developing CS insiders or as beginning CS insiders. The developing CS insiders, Carol, Emily, and Felix, all spoke about their CS instruction with references to technical CS topics. Doris spoke as a beginning CS insider by virtue of her LEGO Mindstorms expertise. Becky also spoke as a beginning CS insider thanks to her ECS training. Alan did not speak about CS topics with the same comfort level or technical specificity. Alan's, Becky's, and Doris's beginning CS identities evolved over the course of their CS implementations. Alan generally spoke about his experience in teaching the Google CS First curriculum with confidence that indicated he would teach the Google CS First again in the future. Alan eased up on pressuring himself to stay ahead of his students and began to feel comfortable letting students help each other. Alan saw himself as a computing teacher, especially when looking ahead to the future, even though his principal spoke more highly of Alan as a science teacher. Becky also evolved in her CS practice by easing up on how much she controlled the flow of her Code.org Course 2 implementation. Like Alan, Becky transitioned to a more student-driven rather than a teacher-driven approach to helping students debug their code. However, Becky identified herself as primarily a mathematics teacher. Doris's CS identity was a little more evolved, but she did not have much room to grow her CS identity. One change in her CS identity perspective was how confident she was in her belief that she could not teach VEX IQ robotics without any training. Doris, like Becky, was ready to give did her CS identity when the absence of curriculum let her down. Carol's isolation and lack of encouragement kept her from developing her CS identity to the level of Emily and Felix.

During my conversations with Carol and Emily, each one shared that she thought she might have pursued an engineering or computer science major in college had she been exposed to CS in middle school. These feelings indicate that Carol and Emily wished they had come to CS sooner. The diffusion of the CS implementation went further for Carol, Emily, and Felix because they all expressed the social motivation they experienced from earning their CS students' support. What further fueled Emily and Felix was having the social partnership of their respective principals (Rogers, 2003). They also felt socially connected to CS by virtue of their CS training cohorts. Emily's and Felix's CS social networks, as compared to those of the other four teachers, were well-developed as a result of having to follow their magnet grants' multi-year training schedule.

All six teacher participants in this study confirmed what prior research has found. Sustaining a CS program is about sustaining the CS teacher (Ni & Guzdial, 2011). According to Ni and Guzdial (2011), obstacles to diffusion of CS curriculum innovation is related to teacher identity theory, which they posit consists of four elements: "their educational background and certification, CS curriculum and department hierarchy, availability of CS teacher community, and teachers' perceptions about the field of CS" (p. 5). This case study supports Ni and Guzdial's (2011) research that complexities surrounding CS course creation, CS teacher certification, and CS teacher isolation further compound the CS teacher identity issue.

CS Curricula

Arbor, Bloom, and Crest's CS implementations did not involve any physical computing, whereas Delta, Eagle, and Falcon middle schools implemented a robotics curriculum that required the physical programming of robots. At all six sites, there was at least one aspect of the chosen curriculum that the CS teacher was not able to access easily: for Alan, it was the mentors he requested on the Google CS First website who never materialized; for Becky, it was the inability to see her students' solutions to coding challenges; for Carol, it was more of an inability to maintain a channel of communication with the other Project GUTS trainees from her district; for Doris, it was a lack of access to a VEX IQ curriculum that she could implement without training like she had the LEGO Mindstorms curriculum; for Emily, it was the lack of comfort to toggle between text-based RobotC to graphical RobotC to accommodate students' different comfort levels with computer programming; and, for Felix, it was the inability to access the free CS2N website where students more interested in virtual robot programming could earn hundreds of motivational achievements and badges (Shoop, Flot, Friez, Schunn, & Witherspoon, 2016; Witherspoon, Schunn, Higashi, & Shoop, 2017). The majority of these obstacles are design flaws in the curricula that may eventually be ironed out. While overall the CS curriculum feedback was positive at Arbor, Bloom, Crest, Eagle, and Falcon, had these extra obstacles to accessing all the curricular resources been resolved early on, the CS teachers may have had more positive feedback to report about their CS implementations.

While the administrators themselves had not received training in the CS curricula at their respective sites, they all agreed that during observations of CS lessons students were engaged with their CS activities. None of the administrators expressed adverse opinions about CS implementation at their respective sites. On the contrary, all the administrators agreed that they needed more support in order to expand CS learning opportunities at their schools.

A notable achievement with all the curricular implementations was what the CS teachers were able to accomplish in terms of generalizable computational thinking (Witherspoon et al., 2017). Alan, Becky, Carol, Emily, and Felix shared evidence of students applying CS skills to non-CS topics. Alan noted how the Storytelling Google CS First module he taught connected to topics in English and mathematics with which students were already familiar. Becky's students were able to program electronic parts attached to a chair to make certain sounds when someone sat on the chair. Carol's students were able to model how epidemics behave using the StarLogo Nova CS environment. Finally, Emily and Felix were teaching introductory engineering principles as they guided their students through robotics building and programming challenges (Ozis, Newley, & Kaya, 2016). Doris had similar examples of generalizable computational thinking and engineering design concepts when she was teaching only LEGO robotics, but she had difficulty realizing the same achievements when feeling pressured to teach both LEGO and VEX robotics platforms.

All six CS teachers shared that they had to learn the CS material as they were teaching it. Even Evelyn and Felix, who had the most extensive CS training that covered all the units they would eventually teach, expressed that they were learning more about their VEX EDR curriculum as they were teaching it. These results are supported by prior research that confirms how CS curricula are designed to allow CS teachers to learn the material along with their students (Koch & Gorges, 2016; Yadav, Gretter, Hambrusch, & Sands, 2016).

Multiple teachers (Becky, Carol, Emily, and Felix) spoke of the need to make significant changes to the intended CS curriculum in order to fill an entire semester- or year-long elective course. This detail was confirmed by the administrators as well—namely, Brenda, Evelyn, and Fred. In terms of the absence of CS curriculum at Delta Middle School, what was unique about Doris was that she did not take one of the two VEX IQ kits and build a teacher robot for programming. The urge to build a teacher robot or other mechanical model was mentioned by both Emily and Felix. A VEX teacher model would have aided Doris in a variety of ways. First, it would have given Doris a chance to get acquainted with the VEX kit and compare and contrast it to the LEGO kits with which she was already familiar. Reading the VEX robot building instructions and gaining experience building her own VEX IQ robot would have given Doris enough knowledge to share with the few students who were assigned a VEX IQ kit to use. While it proved impossible to teach both LEGO and VEX IQ robotics in one class, when there were only two VEX IQ kits to use, a VEX IQ teacher-built robot would have given Doris's most advanced students an alternative platform with which to program and problem solve. Doris took on her two VEX IQ kits at a time when her work-life balance didn't allow her the time to spend hours developing her own VEX IQ lesson repertoire. The lack of time for novice CS teachers to develop their own CS lessons was confirmed by Alan and Becky.

Student Expectations and Student-Centered Theories of Engagement

Students' expectations in middle school CS courses were conveyed by administrators and CS teachers during interviews. Table 8 lists the CS literature pertaining to the student-centered theories that were brought up by administrators and CS teachers in the study.

Table 8

CS Research on Student-Centered Theories of Engagement

Theory No.	Studies	Student-Centered Theory of Engagement
1	Margolis, Ryoo, & Goode (2017); Tatoglu & Russell (2016)	Choice, Independence, Self-Learning
2	Margolis, Ryoo, & Goode (2017)	Collaboration Skills
3	Yadav et al. (2016)	Student Recognition
4	Margolis, Ryoo, & Goode (2017)	Positive Feelings About School, Teacher, or CS
5		Physical Environment/Classroom Space
6	Margolis, Ryoo, & Goode (2017)	Student Seating
7	Israel et al. (2015)	Scaffolding
8	Yadav et al. (2016)	Grading Practices
9		Motivational Practices
10	Witherspoon et al. (2017)	Competition
11	Israel et al. (2015)	Peer Mentoring

Choice, Independence, and Self-Learning

CS research supports that student engagement increases when students are given choice, independence, and are supported in self-learning (Margolis, Ryoo, & Goode, 2017; Tatoglu & Russell, 2016). The CS curricula used at the six sites offered students these elements. But, these are seen as the lowest levels of CS student engagement and learning because they do not involve peer-to-peer interaction. Research indicates that a rigorous, relevant, and engaging curriculum should be accompanied by teaching techniques that support "active student learning" (Margolis, Ryoo, & Goode, 2017). At Arbor Middle School, where the school had tried to shift from teacher-driven instruction to more student-driven instruction, the principal had seen more active learning in the temporary after-school robotics program than he had seen in Alan’s Google CS First classroom. In addition to seeing self-learning, Aaron wanted to hear more from the CS First students about what they were creating on their screens. Instead of using collaborative teams or pair programming, like teachers at the other sites used, Alan allowed his students to help each other when they were stuck and he encouraged his students to acknowledge the helpers.

Collaboration Skills and Grading Practices

A reliance on student collaboration skills in CS courses were brought up in interviews at Bloom, Crest, Delta, Eagle, and Falcon. At Arbor, the principal had expected to see more student collaboration, but instead he saw students engaged in the CS First curriculum individually. All the other site administrators and CS teachers brought up student collaboration as an essential part of what goes on in the CS classroom and at competitions.

One of the benefits CS teachers of using online CS curricula is that the online CS platform is designed to give students immediate feedback as to the correctness or incorrectness of their solutions to CS challenges. These regular instances of immediate feedback given by the online system means less student work that the CS teacher has to grade. However, a challenge for CS teachers is how to give students credit for their CS classwork efforts and completed tasks. Student grades become even more nebulous when students complete CS tasks as a team. Alan expressed that a benefit of teaching a CS curriculum was less grading as compared to his science classes. On the other hand, Alan said, “I had to give a lot of thought to the logistics of the class and how to grade student work.” Consistent with prior research, Becky and Carol both expressed how they relied on their observations of students doing the classwork as a way to assign student grades, especially when students worked collaboratively. Emily and Felix also relied on observations of students during team work (Yadav et al., 2016). Yadav et al. (2016) found the need for a free online repository of lesson and grading ideas, because currently grading rubrics and suggestions for CS First and Code.org courses are not easy to find. It is up to individual CS teachers to determine how to grade students for all their work in these two online curriculum platforms.

Scaffolding

The data are consistent with the literature regarding the importance of using students' "informal" knowledge to more gently introduce technical CS topics (Goode, 2010; Scott et al., 2010) and the need to scaffold CS instruction in order to provide multiple entry points for students to access technical CS knowledge (Israel et al., 2015; Sentance & Csizmadia, 2015). The idea of using students' prior knowledge to increase their engagement in their own learning also ties in with the improvement of teachers' CS identities. As teachers Alan, Becky, Carol, Emily, and Felix let go of controlling the CS learning experiences in their classrooms, the level of students' self-learning increased. Self-learning is considered an essential skill in classrooms that are trying to prepare students for future STEM majors and careers (Tatoglu & Russell, 2016). Furthermore, empowering students to feel successful with scaffolded CS instruction supports the findings of Shapiro, Williams, & Hambarchyan (2013) that warn against using successful CS role models that students do not believe they can reach.

Positive Feelings Towards School, Teacher, or CS

Teachers at all six sites said that they had some students who did not want to learn CS, a point that was not mentioned by all the administrators as they did not distinguish between students who may and students who may not want to learn CS. Alan and Becky taught strictly sixth grade CS classes, and they did not express a need to select their students. Carol noticed an improvement in behavior when her students were coding. Doris had positive experiences with students using LEGO robotics, but introducing two VEX IQ kits to the classroom prevented the LEGO work to continue and the VEX work to commence. Emily specifically mentioned that not being able to select the students in her robotics class was a challenge. Emily and Felix were the only teachers experienced enough in CS instruction to customize their PLTW courses to their diverse students' needs. Emily's ability to customize her CS course to her student population was her past experience with teaching socioeconomically disadvantaged students. Meanwhile, Felix's ability to raise his students' positive feelings about school and CS came from his background in physical science, his personal hobby of construction, and his past basketball coaching experience.

Summary of Essential and Sustaining Implementation Elements and Timelines

Data collected at six middle schools with large numbers of socioeconomically disadvantaged and traditionally underrepresented students indicate that the essential elements necessary to offer CS courses begin with an administrator's expectation and vision for a CS implementation. The data support the finding that a CS teacher's implementation expectations and vision should be aligned with the administrator's CS implementation expectations and vision, so that both administrator and CS teacher(s) are all working towards the same identified goal.

Lastly, student engagement techniques should be used even with an engaging curriculum in order to maintain student interest in CS. In this study, scaffolding, using challenges as motivation techniques, and other methods of lowering the ceiling or entry point to student CS engagement were examples of effective ways to retain students in middle school CS courses, especially socioeconomically disadvantaged and traditionally underrepresented minority students.

Another sustaining element is making sure that the CS teacher develops and maintains their CS network for ongoing CS support. Isolation or the wrong network support can lead to further disenchantment and isolation for the CS teacher. Alan, Becky, Carol, and Doris all lacked the network support that would have made their CS implementation more successful and sustainable. In terms of networking and developing a CS community, Emily and Felix benefited immensely early on in their implementation from their follow-up trainings with their magnet grant cohort. When they reached a saturation level with their follow-up trainings, then the benefits of their CS community diminished.

Research Questions 2, 2a, and 2b: Analysis of Implementation Challenges and Solutions

Analysis of the implementation challenges and solutions at the six sites highlighted two major types of constraints—namely training and budgetary constraints. They are summarized in Table 9 below.

Table 9

CS Implementation Challenges and Solutions

Site Name	CS Training	CS Training Solution(s)	CS Budget	CS Budget Solution(s)
Arbor MS	None	None	One-Time Expenditure	None
Bloom MS	Multiple Sessions; Wrong Grade Level	None	Grant-writing and Fundraising	Continued Grant-writing and Fundraising
Crest MS	One Session; Correct Grade Level	None	None	None
Delta MS	None	None	None	None
Eagle MS	Multiple Sessions; Correct Grade Level	Required Follow-Up Trainings	Large Grant	Continued Grant-writing and Fundraising
Falcon MS	Multiple Sessions; Correct Grade Level	Required Follow-Up Trainings	Large Grant	Continued Grant-writing and Fundraising

Training Constraints

The challenges that CS teachers and administrators faced from lack of any training or the lack of ongoing training manifested themselves in a broad set of issues. Two of the six teachers did not receive any CS training in the curricula they had to teach. Alan and Doris had to learn their respective curricula on their own. Alan felt comfortable following the Google CS First curriculum on his own, provided that he had more time to do so. Doris, however, said that she could not deliver any VEX IQ instruction without any VEX training. For Doris, having had LEGO Mindstorms training did not translate to knowing how to teach VEX IQ and RobotC.

Becky, Carol, Emily, and Felix all had face-to-face CS trainings to implement their specific curricula. Their training provided them with a certain amount of CS content knowledge, but they each had to modify the curriculum to suit students' needs. Becky expressed the most difficulty adjusting her training curriculum to her sixth graders' needs, because she received high school CS training that did not easily apply to sixth graders. Carol weaved the Project GUTS lessons into her science curriculum, but the lack of ongoing training in implementing Project GUTS limited what she was able to do with the curriculum. Emily and Felix had the most expensive and most comprehensive CS trainings, which resulted in their certification to teach VEX Cortex robotics with RobotC programming. Theirs was the only training that remained ongoing throughout the first and subsequent years of implementation. Emily and Felix both credited their ongoing face-to-face robotics trainings with being able to thrive as CS teachers.

Alan, Becky, Carol, and Doris expressed varying levels of burnout partly as a result of not receiving continuous CS training and partly due to their lack of CS background. Alan decided to give up trying to get ahead of the students, because he realized that students were outpacing him as they navigated the CS First lessons in class. Becky was ready to relinquish her CS instructor status citing that it was too much for her to learn to use all the CS resources at Bloom on her own. Carol's and Doris's inability to find Project GUTS and VEX IQ collaborators, respectively, resulted in their isolation and disenchantment with their CS implementations. Even though these four teachers represent a mix of early and late majority adopters, their disenchantment with the absence of CS collaborators makes discontinuance of the innovation more likely. According to past research, "… sustainability is less likely (and discontinuance more frequent) when the innovation is less compatible with the individual's beliefs and past experiences" (Rogers, 2003, p. 190).

Teachers' attitudes towards training, especially expensive training, was noted by Evelyn and Fred. Emily had more than average buy in, and she exuded commitment to the curriculum. She took her PLTW training very seriously, as evidenced by her understanding of the high cost of the training, by acing her training assessments with 100% scores to get certified, and that she spent over 14 hours per day to complete her work in the first three days of her face-to-face training. The need for ongoing CS training and establishing a teacher CS network or community is affirmed by Fisher, Lang, Craig, and Forgasz's (2015) research results.

Administrators also expressed frustration about the lack of CS training opportunities for their teachers, but they added that a CS teacher needs to have a certain way of interacting with students in addition to content knowledge. Evelyn and Fred both credited their CS teachers with having a special rapport and energy with their robotics students. These qualities appeared to elude Alan, Becky, Carol, and Doris, who all struggled to find their CS identities without ongoing support. Despite being novices at CS instruction, Alan, Becky, Carol, and Doris approached CS "insider" status, but without ongoing support they could not attain the same level of CS confidence that Emily and Felix possessed.

At Arbor MS and at Bloom MS, teachers did not express any serious concerns about students other than Alan saying he had mostly boys in his class and Becky saying that her all-girl CS class was always very loud due to all the unplugged team activities that were happening. Alan's students were all GATE students, as Arbor is a Gifted Magnet. Student engagement did not seem to be a concern with their curriculum, and their principals confirmed that they saw students engaged during their classroom visits.

On the other hand, when I asked teachers at Crest, Delta, Eagle, and Falcon about different outcomes for different students, they all expressed concerns about student behavior. Carol spoke about a couple of male students who were argumentative and refusing to do work, yet during the Project GUTS computer programming portions of class these boys were on task and cooperative. Doris spoke about discipline issues in general and that her students were not serious and focused about taking a robotics class, and according to her the outcome was that students broke important parts from the VEX IQ robotics kit. Emily and Felix both expressed having difficulty teaching to special education students who did not come to class with a designated aide. Emily explained that the autistic students came to her class with an aide, yet special day students who also needed extra support did not have an aide. In a class of 45 students, challenge of managing student behavior was exacerbated by the fact that the special day students did not know how to work in teams and work on projects that were time-sensitive. Felix also mentioned being perplexed about how a large number of special education students were mostly scheduled into one of his robotics courses. He was told by a counselor that it had been a scheduling error, yet he also did not get any special education aides to assist with reaching this special population of students. Felix also mentioned that female students preferred not to work with male students, but that he explained heterogeneous grouping was a part of real life and that they had to work in all kinds of teams. Felix had decided that changing student groups often in his robotics classes minimized behavior issues. None of their principals mentioned these issues when asked about different outcomes for different students. Dan, the magnet coordinator at Delta, mentioned that the type of students in Doris's robotics class was not ideal, so he was aware of the behavior issues she mentioned.

Budgetary Constraints

Among the budgetary constraints to implementing CS, administrators cited the high cost of quality CS and project-based learning training. Evelyn and Fred both spoke about the need to train their star CS teachers in pedagogical and project-based learning methods in addition to the content-centered trainings they received from PLTW. CS training on its own does not translate into high quality CS instruction without the added training in collaborative learning or literacy techniques.

Budgetary constraints at Arbor and Bloom Middle Schools are linked to a lack of training specific to the curriculum that Alan and Becky needed to deliver to sixth grade students. Alan and Becky talked about a lack of content support for their respective curricula. Throughout her year of implementing CS, Becky received a variety of hardware tools to use with her sixth graders but learning to use all the tools fell on her alone. She would have benefited as a CS teacher had some of the funding been spent on on-going content professional development. Had funds been allocated to provide Alan and Becky with content-level support, their CS social networks would have grown and increased the level of CS diffusion at Arbor and Bloom Middle Schools. A similar budget-induced CS isolation occurred at Crest, where funds were not allocated to help grow Carol's Project GUTS implementation. She had reached a higher level of CS adoption than Alan and Becky in that she felt comfortable customizing the GUTS curriculum to her science needs.

Principals Aaron and Chuck met the late majority adopter category, because they remained skeptical about allocating funds to develop a CS program at their sites. According to Rogers (2003), "their relatively scarce resources mean that most of the uncertainty about a new idea must be removed before the late majority feel that it is safe to adopt" (p. 283). As an early adopter with an extensive knowledge about the benefits of CS instruction, Brenda was less uncertain about spending money on developing CS instruction at Bloom Middle School. Her information-gathering, especially through her CS social network, and understanding about the misalignment of the ECS curriculum to middle school CS needs helped her decide to allocate funds to hire a CS expert to teach CS in the future.

Due to his background in technology and his eagerness to leverage his social ties with other magnet coordinators, Dan possessed some of the qualities of an early adopter. However, due to his short tenure at Delta, Dan's inability to secure funding for Doris to get VEX IQ training in a timely fashion puts Delta Middle School in the adopter category of laggards. Rogers (2003) warns against blaming a specific individual for the slowness of adoption inside an organization, because there could be cultural reasons embedded in the organization that prevent diffusion of innovation from happening more quickly.

Despite the abundance of funding for CS implementation at Eagle and Falcon Middle Schools, Evelyn and Fred continued their fundraising activities. These two principals, each an early adopter and innovator, respectively, saw how money spent on CS training and resources benefited their CS teachers. As a result, they used their social networks to raise more money for future CS expenditures.

Summary of Implementation Challenges and Solutions

All of the middle school administrators and CS teachers in this study agreed that either budgetary or training constraints were either direct or indirect challenges to CS implementation. Training that could meet CS teachers' specific needs was a challenge even for those teachers in the study who received the most expensive PLTW training. A lack of support staff for special education students in a CS or robotics course was a budgetary and training challenge. CS teachers who expressed difficulties with special education students mainstreamed in a robotics course did not have any training on how to accommodate or differentiate for these students.

The administrators at the sites that addressed budgetary challenges wrote grant proposals and actively raised funds to sustain ongoing CS training, to pay for field trips, and to purchase technology hardware. At Bloom MS, funding was allocated to hire a CS expert, which was a budgetary challenge that would address a more significant training challenge. At Arbor MS, where significant funds were allocated to implement a temporary program, lasting benefits of the budgetary allocation were not sought nor realized.

Research Questions 3 and 3a: Analysis of Implementation Supports and Resources

I analyzed how the six sites leveraged supports and resources during their CS implementations. Four categories emerged during the analysis of implementation supports and resources. The four categories, as seen in Table 10 below, were student engagement in learning CS, CS partnerships and collaboration, problem-solving mindset, and visions of CS in the future.

Table 10

CS Implementations Supports and Resources

Site Name	Student Engagement	Partnerships and Collaboration	Problem-Solving Mindset of Administrator	Problem-Solving Mindset of Teacher	Visions of CS in the Future
Arbor MS	Collaborative Beginners	1	Beginner	Beginner	Ambiguous
Bloom MS	Collaborative Beginners & Designers	Multiple	Intermediate	Beginner	Specific
Crest MS	Collaborative Designers	1	Beginner	Intermediate	Ambiguous
Delta MS	Collaborative Beginners	1	Beginner	Beginner	Ambiguous
Eagle MS	Collaborative Problem-Solvers	Multiple	Beginner	Intermediate	Specific
Falcon MS	Collaborative Problem-Solvers	Multiple	Intermediate	Advanced	Specific

Student Engagement in CS Learning

At Arbor and Delta Middle Schools, where the CS teachers did not receive any CS training, the CS implementation plan was to expose students to CS. CS research suggests that teachers without all the right technology expertise to teach CS were nonetheless able to provide their students with choice and individualization that reached students at their level of CS comfort (Israel et al., 2015).

At the four sites where the teachers received training (Bloom, Crest, Eagle, and Falcon), students received more than just exposure. According to their CS teachers, students at Bloom, Crest, Eagle, and Falcon were taught applications of CS computation thinking skills.

Emily witnessed the benefits of advancing in high school robotics for her own children, so she was a firm believer in the importance of the robotics curriculum for all her students. Emily also considered robotics to be fun, and she did not consider giving up on robotics as an option. She told her students that frustration is good, and that if they persevere through frustration, then they will feel like the smartest person in the world for having figured out a challenging problem.

Partnerships and Collaboration

A common characteristic among administrators at four of the six sites—predominantly at the middle school sites with the less rigorous computer science implementations—was the designation of issues concerning CS courses solely to the teacher. As one teacher put it, “They offer [computing courses] because it looks good on paper.” What was also true about these sites was that the teacher who was teaching programming was more focused on his or her core subject (i.e., science or mathematics). These administrators were completely hands-off with respect to what was going on in the CS classroom. In contrast, the administrator at Falcon Middle School was very connected with the goings on of the robotics classroom. At robotics competitions, he cheered on his winning Falcon students

There were four examples of how an investment in CS implementation did not gain enough momentum to grow due to an inability to leverage or prolong a partnership. While the after-school club at Arbor MS had been a success, according to Aaron and Alan, the money spent on the non-profit that ran the club ultimately became a large one-time expenditure that did not leave behind any robotics or coding resources for Alan to use with his start-up Google CS First class. Some might call this a *one and gone* situation—a short-term and costly implementation that disappears and takes its learning, curriculum, and resources with it. A similar misalignment in partnership occurred with the ECS training that Becky attended that did not fully meet her needs with sixth grade students. Carol had a similar experience with being identified as a potential national trainer for Code.org's Project GUTS curriculum, but ultimately not having any CS collaborators at all. The fact that SUSD had a partnership with Code.org, a non-profit organization offering professional development to increase the number of teachers able to teach computer science concepts in grades K-12, was a boon to Crest science teacher Carol. She believed that SUSD should have maintained their *CS in Science* relationship with Code.org. Lastly, Delta MS was fortunate to receive a small grant that funded two VEX IQ robotics kits. However, without a plan on how to integrate VEX IQ robotics with the existing LEGO Mindstorms implementation, the introduction of a new robotics platform was more of a disruption than a support to Doris's CS implementation.

Problem-Solving Mindset

Of the 12 interviewees in this study, seven regarded challenges through a problem-solving mindset: Alan, Brenda, Carol, Emily and Evelyn, and Felix and Fred. Israel et al. (2015) found that challenges to CS implementation were easier to overcome when teachers and administrators viewed challenges with a problem-solving mindset. Consistent with Israel et al.'s (2015) findings, Eagle and Falcon middle schools were further along in their CS implementations than the other four sites partly because both the CS teacher and administrator at the site had developed a problem-solving mindset when facing any challenges to their CS implementation.

A majority of the CS teachers also noted how they were able to sustain their early CS implementations by simplifying their CS practices. This is what Rogers (2003) calls re-invention or a “flexibility in the process of adopting an innovation [that] may reduce mistakes and encourage customization of the innovation to fit it more appropriately to local and/or changing conditions” (p. 184). Middle school CS teachers face critical decisions when trying to keep all their students engaged, especially their underrepresented female and minority students. Keeping the difficulty threshold at a level that maximizes student engagement is key (Manches & Plowman, 2105). Alan stuck to the most basic module he could find on the Google CS First website. He lowered the ceiling for student engagement partly because he was aware of his own limitations in CS content knowledge. Similarly, when Becky opted to teach Code.org’s Course 2 curriculum, she adapted parts of her ECS training to meet the age level of her sixth graders. Carol used fewer Project GUTS modules during her first year of implementation, even though she had planned to use all four available modules. Emily simplified the organization of all the VEX parts in her classroom in order to minimize student confusion during the robot building phases. A source of confusion for middle school students with VEX robotics kits is that the kits come with more parts than are necessary for building a given robot. Unless a teacher removes the unnecessary parts from the kits in advance, students could face building challenges that could hamper learning. Felix chose to use a graphical version of the RobotC software to help students minimize syntactical issues with programming their robots. These instances of teacher problem-solving using simplification are evidence of teachers wanting to keep students in CS. Just like Felix transitioned from text-based to graphical RobotC early in the implementation to prevent losing students to technical programming issues, Fred looked for ways to introduce Falcon students to the less intimidating VEX IQ robotics kits in addition to the more advanced VEX EDR kits currently in use.

CS teachers were limited in using self-teaching as a method to solve the problem of lack of content knowledge. None of the CS teachers in this study had a background in CS, but all of them expressed a desire to learn more CS given extra time. They all felt like they could have done more with CS if they had had more time to learn it on their own. However, according to Rogers (2003), “discontinuance happened less often because the re-invented innovations better fit a school’s circumstances, leading to sustainability” (p. 184).

Vision of CS in the Future

Administrators Brenda, Evelyn, and Fred were the most hopeful about continuing their CS implementation efforts. Brenda’s optimism was a result of her finding a CS expert to teach the sixth grade CS wheel. Evelyn’s enthusiasm about CS courses was linked to having a very enthusiastic CS teacher in Emily. Fred’s enthusiasm resulted from witnessing Felix’s CS successes and planning to bring more entry-level CS opportunities to his school.

The three administrators who stood out as not having a well-articulated vision of CS in the future were Aaron, Chuck, and Dan. Each had different reasons for not being completely certain about the future of CS at their sites. Aaron was more eager about the obsolete after-school program than he was about Alan’s CS First course. Chuck reiterated what has been documented in prior research about the reverse causation caused by the lack of CS professionals knocking on the doors of school such as Crest Middle School, where the student population is almost entirely minority students (Byars-Winston, 2014; Rosenbloom et al., 2007). Chuck felt that none of the influential entities in CS wanted to work with his student population, and that the

CS opportunities automatically fall with the more privileged non-minority students. This hopelessness about the future of CS at Crest Middle School was in direct contrast to the optimism displayed by Brenda at Bloom Middle School, who actively pursued the CS or STEM partnerships that would open doors for her female students. What was unique about Brenda's CS outreach efforts, which are supported by prior research, was that she sought culturally relevant opportunities for Bloom's female students to see themselves in STEM fields (i.e., field trip to see a screening of the film *Hidden Figures*) (Ryoo et al., 2012; Byars-Winston, 2014). Finally, Dan had a lot of creative ideas about how CS could be better implemented at Delta Middle School in the future, however he could not articulate the financial means of getting to those ends he envisioned. Altogether, these three administrators (Aaron, Chuck, and Dan) had a collective hopelessness about the feasibility of offering CS curriculum to middle school students, and they were all distracted by other priorities at their respective sites that overshadowed the need for CSE implementation. This confirmed what prior research on diffusion of innovation considers a not compelling enough innovation (Strang and Meyer, 1993). The data also align with research findings about the importance of raising the image of CS instruction among administrators (Black et al., 2013).

Of the six sites, the one that most closely resembled what prior CS research considers the gold standard in CS implementation was Falcon Middle School. As Margolis et al. (2008) discovered, CS education innovation is more likely to happen in informal educational settings due to the stubborn and historic systemic obstacles in traditional school site settings. The key ingredient in making middle school CS courses flourish is to replicate the same level of freedom and innovation present in informal after-school CS clubs inside the fledgling CS courses being offered by more and more middle schools—designing a sort of CS lab environment where students can design, problem-solve and create.

Summary of Implementation Supports and Resources

The supports and resources credited with helping middle school CS teachers and administrators sustain their CS implementations included maintaining successful partnerships and ongoing collaboration, developing a problem-solving mindset, and espousing a vision of how to continue offering CS in the future. Evidence of student engagement in CS courses included signs of collaboration, fun, and creativity. Sites where the training partnerships were ongoing were further along in their implementations and had a clearer vision of where their CS implementations were headed next. Lastly, at Eagle and Falcon, where administrators and CS teachers developed the strongest problem-solving mindsets, participants were more vested and entrenched in their CS implementations than participants at the other four sites. Evelyn, Emily, Fred, and Felix did not allow challenges to become permanent obstacles to their implementations. Meanwhile, sites where either the administrator or the CS teacher had a problem-solving mindset were not able to overcome their challenges.

The interplay of all the elements that worked to catapult the CS implementations at Eagle and Falcon middle schools to the forefront of the sites in this study is complex. There were enough elements working together to make their programs stronger than at the other four sites. However, subtracting all the money the committed teachers and the dedicated principal at Falcon, and the CS implementation story at Eagle and Falcon would be much different.

It is difficult to analyze what happened at Eagle and Falcon with their VEX EDR robotics programs without considering their multi-million dollar grant. Would these sites have come as far without all that money? Probably not, because they would not have been able to purchase as many VEX EDR robotics kits as they were able to with the money. Each kit costing over $1,000 and the sites spending almost $50,000 on building kits to be able to give more students a tangible robot to program. Grant writing teams at Eagle and Falcon had done research in writing the MSAP grant that informed them about what works and what doesn't work. So it was not a coincidence that they had the money and that they used the money to purchase the state of the art hardware and among the best teacher training. If the other sites had as much money as Eagle and Falcon but had not done their research, they may have spent the money on programs that may not have had lasting and sustainable impacts.

It is important to underscore the problem-solving mindsets that the teachers at Eagle and Falcon developed with their VEX EDR implementations. Given all the money, there were still teachers at each site that were not as resilient and successful at teaching VEX EDR robotics as Emily and Felix were. In addition to their commitment to the robotics implementations, they set goals for themselves, they maintained their curiosity and thirst for learning in the program (which was exciting enough to keep away burnout), and they had a way of talking with their students that was inspiring to students (to sustain engagement).

CHAPTER 6: DISCUSSION AND RECOMMENDATIONS

In the previous chapter, I conducted a cross-site analysis of the data in order to answer my three research questions. In this chapter, I summarize my findings and propose five recommendations for future CS implementation in middle schools.

Key Findings

The data suggest that a CS teacher's implementation vision needs to be aligned with the administrator's implementation vision. For this to happen it is necessary for both the teacher and the administrator to identify to some degree as CS insiders. Long-term middle school CS implementation can be sustained by recognizing a teacher with an insider-like CS identity and allowing that teacher to specialize in CS education, rather than diluting their CS focus with added curricular assignments. Alternatively, if CS specialization is not an option due to teacher shortages in the teacher's credentialed subject area, then administrators who encourage CS teachers to express their CS needs and meet their CS teachers' needs are more likely to sustain their CS implementations.

The majority of middle school administrators and CS teachers in this study agreed that budgetary and training constraints were their biggest challenges to CS implementation, regardless of student population. The administrators at the sites that addressed these challenges actively raised funds to sustain ongoing CS training.

The supports and resources credited with helping middle school CS teachers and administrators sustain their CS implementations included witnessing the student benefits of CS education, maintaining successful partnerships and ongoing collaboration, developing a problem-solving mindset, and espousing a vision of how to continue offering CS in the future.

Recommendations

Curriculum Selection (Diffusion Phases 1 & 2: Agenda-Setting & Matching)

Recommendation #1: A single CS curriculum that matches student needs should be chosen by the administrator and the CS teacher, and a realistic assessment should be made about what the teacher will need to invest to implement the chosen curriculum.

Prior to choosing a CS curriculum, middle school administrators need to gauge its "compatibility with the values, beliefs, and past experiences of individuals in the social system"—namely, the school (Rogers, 2003, p. 3). There are high-quality CS curricula that are free to implement. However, even with free curricula novice CS teachers need to attend face-to-face training. This is a great way for novice CS instructors to network and discuss implementation ideas with each other.

Administrators should be wary of adding too many curriculum platforms on a CS teacher's curriculum load and allowing the CS teacher to pick and choose which aspects of a curriculum to implement and to opt to only focus on one curriculum. Burnout for novice CS teachers learning CS for the first time can be very high. Administrators and teachers in charge of CS curriculum selection should also take into account how updates to the curriculum will be accessed and implemented by the teacher. Staying current with CS education updates is a constant process that requires ongoing funding and teacher time.

The Need for Ongoing CS Training (Diffusion Phase 3: Redefining/Restructuring)

Recommendation #2: CS teachers, especially those without a CS degree or background, need to continuously participate in CS trainings to meet the needs of their students and to maintain a CS social network from which they can keep learning.

CS training was one of the most significant challenges faced by the participants of this study. The most expensive type of CS training was also the most comprehensive. Nevertheless, CS training needs to be more customized. Gathering feedback from CS teachers about implementation successes and failures would inform trainers who care to adjust and improve their offerings for future participants.

CS training also needs to include administrators in order to avoid the disconnect between how administrators and teachers understand CS implementation. Administrators may be more likely to support the CS teacher if they understand the curriculum.

At the very least, CS expenditures should promote ongoing CS networks between CS teachers and support providers. When these networks are not enough, as with Emily and Felix, for example, then CS teachers would benefit from personalized networking such as with a CS listserv or CS Facebook group where they could read and post ideas.

CS Teacher Recruitment, Retention, and CS Identity Development (Diffusion Phase 3: Redefining/Restructuring)

Recommendation #3: CS teacher recruitment needs to be broad and ongoing and provide CS teachers with the opportunity to specialize in CS and to advocate for their growth of their own CS identity.

CS teacher recruitment should not be limited to asking science and mathematics teachers to teach CS courses. Administrators also recommended asking teachers hobby-related questions to identify hobbies that tie in naturally with teaching computer science.

After administrators identify a teacher who excels at teaching CS, they should make an effort to help the teacher certify and specialize in CS. CS certification can help motivate a CS teacher, as well as boost their CS confidence and help them identify as a CS insider.

Charting a pathway for CS teachers to become CS insiders will improve teacher confidence and minimize imposter syndrome for teachers who are eager to teach CS but lack a formal CS background. As long as administrators outsource their CS instruction to so-called CS experts outside their school, they will have difficulty finding willing CS teachers among their own faculty. Instead, administrators should give willing CS teachers the opportunity to develop their problem-solving mindset. A developed problem-solving mindset may minimize the intimidation that beginning CS insiders might feel by the technical aspects of the topic.

Administrators and CS teachers who co-develop their CS identities also help to sustain CS implementations. When the site administrator does not support the efforts of the CS teacher like an informed partner, the CS teacher often feels isolated and unsupported. Administrators should make an effort to participate in CS decisions with their CS teachers rather than assigning CS decisions solely to the teacher.

Student-Centered CS Decision-Making (All Five Stages of Diffusion)

Recommendation #4: CS instructional practices needs to align with what is best for students in terms of age, cultural and gender differences, and learning abilities.

The data suggest that CS implementation should be student-centered. Any consideration of diverse student populations should include specific populations such as special education students. CS instruction needs to be modified for them but expecting such modifications from novice CS teachers may be unrealistic.

It may not be as simple as determining the length of the class by the grade level. The amount of CS instruction a student receives should at least partly be related to the student's interests. There are also nuances to teaching a rigorous topic such as computer science that well-trained teachers must be aware of lest they fall into traps documented by research (Yadav et al., 2016). Computer science education training programs need to caution novice computer science teachers that a failure to adjust the difficulty level of computer science curricula to the needs of their students for a prolonged period of time could result in a portion of the students getting permanently turned off by the topic.

Another aspect of student-centered decision making is for teachers to have access to uniform and consistent grading policies for CS courses. These grading policies also need to be adjusted by grade level and student learning abilities. Special education students should not be graded the same way that non-special education are graded on CS tasks. Whenever possible, CS teachers should collaborate with special education teachers who may be able to frontload introductory CS activities in the special education classroom to help special education students be more prepared for collaborative team work in the CS classroom.

Partnerships and Collaboration: Making CS More Compelling (Diffusion Phases 4 & 5: Clarifying and Routinizing)

Recommendation #5: Middle school administrators and CS teachers should continuously seek out CS partnerships to sustain their CS programs and make CS more appealing to students.

Administrators and CS teachers need out-of-school partnership and collaboration opportunities to keep growing as CS educators and to sustain their site implementations. Funding for training poses a substantial hurdle to administrators who want to obtain training for teachers willing to teach CS. Partnerships that result in free or subsidized teacher training can prove to be beneficial. The district's partnership with Code.org made it possible for Carol to receive CS training she considered to be one of the best professional development experiences of her relatively fresh teaching career. Without partnerships and training subsidies, comprehensive CS training can cost over $3,000 per teacher. Both Evelyn and Fred saw training money go to waste because a trained teacher could not ultimately teach the CS course for which they were trained.

Limitations of the Study

While rich data emerged from twelve interviews of middle school administrators and CS teachers, one of the limitations of this study was its reliance on only interview data. This data relied on administrators' and teachers' recollections and own perspectives of how they implemented CS education. Such "verbal reports" may be prone to "bias, poor recall, and poor or inaccurate articulation" (Yin, 2014, p. 113). An additional interview with each of the 12 participants could have increased the depth of the findings.

Using only two people's perspectives from each site may have limited the data. In addition to interview data, future studies of middle school CS education implementations should include observation data as well. Important data about implementation elements, challenges, and solutions can be gleaned by observing a CS teacher in his or her CS classroom.

A third limitation to this study was that it took place at six sites in one large, urban, public school district, which could limit the findings' generalizability to other districts and middle schools. Future research on middle school CS education could shed more light on implementation phenomena by including more school sites.

Need for Future Research

As CS implementation research continues to focus on elementary and high school implementation, more large scale CS research is needed to fully understand middle school CS implementation. A large scale mixed methods study of an entire district and collects quantitative data to support a variety of qualitative data could provide more generalizable findings about middle school CS implementation. Answers to the following research questions would benefit the field of middle school computer science education: How does a middle school CS teacher transition from teaching a robotics course to building student teams for robotics competitions? What are the sustaining habits of successful middle school CS teachers that could help other teachers improve their CS instruction? What are the characteristics of a well-written middle school CS lesson plan that includes direct instruction, checking for understanding, independent and group practice, assessment, and a grading rubric. To summarize, there is a need for more research that will help bring consistency to middle school CS instructional practices. Bringing more consistency to this field will help administrators understand what to look for in a middle CS classroom and a CS teacher. Future research could also investigate CS instruction from middle school students' perspectives.

Conclusion

The findings of this six site case study of middle school CS education implementation indicate that the training and budgetary challenges faced by middle school administrators and CS teachers outweigh the supports and resources available to deal with the challenges. Unless middle schools and districts place a greater focus on overcoming the dual challenges of budget and training, novice implementers of CS education may not have enough resources to sustain their implementations. Despite increased expansion in computer science opportunities to middle school students, there is a great variety and disparity in what is actually being taught inside the middle school classrooms that have given into the pressure to offer computer science learning to middle school students. While there is agreement among middle school principals and CS teachers that offering computer science is important, few middle school CS programs go beyond the initiation stages of implementations to reach the more sustainable routinizing stage of implementation.

List of Appendices

Appendix A

Recruitment Email

From: **Verjinia Mayer**
Date: Sun, Apr 23, 2017 at 7:04 PM
Subject: Middle School Computer Science Education
To: <middle school principal's name>

Dear Middle School Principal,

I am a graduate student in the UCLA School of Education (GSE&IS), and my doctoral dissertation is about how middle school administrators and teachers in SUSD are implementing computer science (CS) in the middle school curriculum. Attached is my research approval from SUSD's Committee for External Research Review.

Considering how popular and important computer science education has become in K-12 schools recently, I believe that districts looking to implement CS curricula soon can learn a lot from early implementers like SUSD's middle schools.

If you have already implemented some form of computer science in your school (i.e., through technology, computing, and/or robotics courses, etc.), please email or call me if you would be willing to participate in a case study. All information about your school will be kept confidential, and your participation is voluntary. The case study consists of 1-2 interviews with the administrator(s) and 1-2 interviews with the computer science teacher(s) involved in the CS implementation. All participants will receive a $50 Amazon gift card following their telephone interview(s).

I have been a public school STEM educator for 19 years—13 years as a middle school and high school mathematics and technology/robotics teacher and six years as a teacher specialist in technology. This year I am teaching middle school computer science via two sections of robotics and two sections of applied technology, all of which involve coding for game design, programming robots, or creating mobile applications. I am eager to learn from the practices of other middle school computer science teachers and the implementation experiences of their administrators. I also look forward to sharing my research with SUSD in order to help inform educators about computer science education.

I look forward to hearing from you, and I thank you in advance for your time. Feel free to forward this email to anyone at your site who would be able to participate in the study.

Sincerely,

Verjinia Mayer, Graduate Student
UCLA Educational Leadership Program
Cohort 21

Appendix B

Computer Science Administrator Interview Protocol

Interviewer:____Verjinia Mayer______ Date:___________________
Interviewee (pseudonym):___________ Phone:__________________
Start Time:________________________ End Time:_______________

Hello! Thank you for taking time to have this telephone interview with me today. Thank you for taking the time to complete the informed consent form and agreeing to participate in my study. My name is Verjinia Mayer, and I am a graduate student at the UCLA School of Education. I have been studying computer science education for the past two years, specifically middle school computer science education. I will be asking you a series of questions that will help me understand how SUSD middle schools have implemented computer science. All of the study participants' identities will be kept confidential in order to obtain more authentic responses. I would like to record our conversation so that it can be transcribed and coded accurately. Is that okay? You are free to ask me to stop recording at any time during this interview. This interview will take about 30-40 minutes with mostly open-ended questions. Do you have any questions for me about this interview?

1. First, please tell me a little about your professional background:
    a. Years in education
    b. Years at SUSD
    c. Academic background
    d. Other professional experiences prior to education
    e. How are you involved with implementing computer science in SUSD?
2. Tell me about how your school decided to offer a computer programming course.
    a. What were your main goals?
    b. What input did you have in the curriculum selection?
    c. What were some of the challenges in getting the chosen CS curriculum to your school?
3. What were the potential benefits you foresaw prior to adopting the curriculum?
    a. To students?
    b. To teachers?
    c. To ____________?

4. Do you see other benefits after [*number of years*] of implementing CS?
    a. To students?
    b. To teachers? How many teachers are teaching CS?
    c. To ____________?
5. Have you observed the CS classes?
    a. What benefits have you noticed?
    b. What challenges have you noticed?
    c. What teaching techniques have you noticed?
6. What, if any, methods (i.e., mentorship, advocacy, etc.) do you use to encourage others to support computer science education at your school?
7. How did you prepare to offer CS?
8. What was your first year ever involved in offering CS?
9. Was there anything you particularly enjoyed or didn't enjoy about implementing CS in the beginning? Now?
10. How many sections of CS do you offer and how many students are in the classes?
11. What was the gender and ethnic makeup of your computer science courses?
    a. Did the way you implemented the CS curriculum in your class differ for any of the traditionally underrepresented student groups? (i.e., gender, race, socioeconomic status)
12. What were some of the challenges:
    a. in preparing to implement CS courses?
    b. in implementing CS courses?
    c. that the students faced?
13. Was there anything in your professional experience in education or other special skill that helped you persevere through the challenges?
14. What access to technology do you have at your school to teach CS?
15. Will you continue to offer the CS curriculum?

16. Do you plan to offer a different CS curriculum in the future? The same one or different?
17. What was the financial impact, if any, of choosing to offer this CS curriculum?
18. What partnerships, if any, have you formed to help implement computer science at your school?
    a. How were the CS partnerships involved with your CS classes?
    b. What are your primary sources for finding potential CS partnerships?
    c. What advice would you give a site or district about how to initiate CS partnerships?
19. Do you think this curriculum can easily be adopted by middle schools in other districts?
20. What, in your opinion, are the essential or key elements of a successful middle school CS curriculum implementation?
21. What do you see as the DISTRICT's role and what do you see as YOUR role in supporting the CS teacher(s)? Short-term? Long-term?
22. What recommendations would you have for other districts trying to implement this or similar curriculum?

Thank you for your time and your responses. This completes our interview. If you have any questions or need to contact me, my contact information is on the consent form. May I contact you if I need any additional information for the study?

Appendix C

Computer Science Teacher Interview Protocol

Interviewer: Verjinia Mayer
Interviewee (pseudonym):__________________
Start Time:__________________________
Date:______________________
Phone:_____________________
End Time:_________________

Hello! Thank you for taking time to have this telephone/Skype interview with me today. Thank you for taking the time to complete the informed consent form and agreeing to participate in my study. My name is Verjinia Mayer, and I am a graduate student at the UCLA School of Education. I have been studying computer science education for the past two years, specifically middle school computer science education. I will be asking you a series of questions that will help me understand how SUSD middle schools have implemented computer science. All of the study participants' identities will be kept confidential in order to obtain more authentic responses. I would like to record our conversation so that it can be transcribed and coded accurately. Is that okay? You are free to ask me to stop recording at any time during this interview. This interview will take about 30-40 minutes with mostly open-ended questions. Do you have any questions for me about this interview?

1. First, please tell me a little about your professional background:
    a. Years in education
    b. Years at SUSD
    c. Academic background
    d. Other professional experiences prior to education
    e. How are you involved with implementing computer science in SUSD?
    f. What experiences led you to believe that computer programming must be offered to middle school students to prepare them for the future?
2. Tell me about how your school decided to offer a computer programming course.
    d. What were your main goals?
    e. What input did you have in the curriculum selection?
    f. What were some of the challenges in getting the chosen CS curriculum to your school?

3. What were the potential benefits you foresaw prior to adopting the curriculum?
    d. To students?
    e. To teachers?
    f. To ____________?
4. Do you see other benefits after [*number of years*] of implementing CS?
    d. To students?
    e. To teachers?
    f. To ____________?
5. How did you prepare to teach CS?
6. What was your first year ever teaching CS?
7. Was there anything you particularly enjoyed or didn't enjoy about teaching CS in the beginning? Now?
8. How many sections of CS do you teach and how many students are in your classes?
9. What was the gender and ethnic makeup of your computer science courses?
    a. Did the way you implemented the CS curriculum in your class differ for any of the traditionally underrepresented student groups? (i.e., gender, race, socioeconomic status)
10. What were some of the challenges:
    d. in preparing to offer CS courses?
    e. in offering CS courses?
    f. that the students faced?
11. Was there anything in your professional experience in education or other special skill that helped you persevere through the challenges?
12. What access to technology do you have in your class (or at your school) to teach CS?
13. Will you continue to teach the CS curriculum?
14. Do you plan to offer a different CS curriculum in the future?
15. What was the financial impact, if any, of choosing to offer this CS curriculum?

16. What partnerships, if any, have you formed to help implement computer science at your school?

17. How were the CS partnerships involved with your CS classes?

18. Do you think this curriculum can easily be adopted by middle schools in other districts?

19. What, in your opinion, are the essential or key elements of a successful middle school CS curriculum implementation?

20. What recommendations would you have for other districts trying to implement this or similar curriculum?

21. Please share your CS course guidelines and/or syllabus. You may attach a PDF or .doc to your responses.

22. Please share a piece of student code or sample CS challenge solved by a student that you are particularly proud of and explain why you chose it. You may attach a file or a link to the code to your responses.

Thank you for your time and your responses. This completes our interview. If you have any questions or need to contact me, my contact information is on the consent form. May I contact you if I need any additional information for the study?

Appendix D

CSTA K-12 Computer Science Standards

Level I Standards (K-8)

Grades 6–8:

- Apply strategies for identifying and solving routine hardware and software problems that occur during everyday use.
- Demonstrate knowledge of current changes in information technologies and the effects those changes have on the workplace and society.
- Exhibit legal and ethical behaviors when using information and technology and discuss consequences of misuse.
- Use content-specific tools, software, and simulations (for example, environmental probes, graphing calculators, exploratory environments, Web tools) to support learning and research.
- Apply productivity/multimedia tools and peripherals to support personal productivity, group collaboration, and learning throughout the curriculum.
- Design, develop, publish, and present products (for example, Web pages, videotapes) using technology resources that demonstrate and communicate curriculum concepts to audiences inside and outside the classroom.
- Collaborate with peers, experts, and others using telecommunications tools to investigate educational problems, issues, and information, and to develop solutions for audiences inside and outside the classroom.
- Select appropriate tools and technology resources to accomplish a variety of tasks and solve problems.
- Demonstrate an understanding of concepts underlying hardware, software, algorithms, and their practical applications.
- Discover and evaluate the accuracy, relevance, appropriateness, comprehensiveness, and bias of electronic information sources concerning real-world problems.
- Understand the graph as a tool for representing problem states and solutions to complex problems.
- Understand the fundamental ideas of logic and its usefulness for solving real-world problems.

Appendix E

Excerpts from Every Student Succeeds Act (ESSA)

Public Law 114-95

PART A—SUPPORTING EFFECTIVE INSTRUCTION (p. 121)
SEC. 2101. FORMULA GRANTS TO STATES.
  (c) State Uses of Funds
    (4) State Activities
      (B) Types of State Activities
        (xvii) "Developing and providing professional development and other comprehensive systems of support for teachers, principals, or other school leaders to promote high-quality instruction and instructional leadership in science, technology, engineering, and mathematics subjects, including ***computer science***." [emphasis added]

SEC. 2103. LOCAL USES OF FUNDS. (p. 128)
  (b) Types of Activities
    (3) "may include, among other programs and activities—…
      (M) developing and providing professional development and other comprehensive systems of support for teachers, principals, or other school leaders to promote high-quality instruction and instructional leadership in science, technology, engineering, and mathematics subjects, including ***computer science***." [emphasis added]

SEC. 4102. DEFINITIONS. (p. 169)
  "In this subpart:…
    (8) STEM-FOCUSED SPECIALTY SCHOOL.—The term 'STEM-focused specialty school' means a school, or dedicated program within a school, that engages students in rigorous, relevant, and integrated learning experiences focused on science, technology, engineering, and mathematics, including ***computer science***, which include authentic schoolwide research." [emphasis added]

SEC. 4104. STATE USE OF FUNDS. (p. 171)
  (b) "STATE ACTIVITIES.—Each State that receives an allotment under section 4103 shall use the funds available under subsection (a)(3) for activities and programs designed to meet the purposes of this subpart, which may include—…
    (3) supporting local educational agencies in providing programs and activities that—
      (A) offer ***well-rounded educational experiences to all students***, as described in section 4107, including ***female students***, ***minority students***, English learners,

children with disabilities, and low-income students who are often ***underrepresented in critical and enriching subjects***, which may include—

- (i) ***increasing student access to and improving student engagement and achievement in***—
  - (I) ***high-quality courses*** in science, technology, engineering, and mathematics, including ***computer science***.” [emphasis added]

SEC. 4107. ACTIVITIES TO SUPPORT WELL-ROUNDED EDUCATIONAL OPPORTUNITIES. (p. 176)

- (a) “IN GENERAL.—Subject to section 4106(f), each local educational agency, or consortium of such agencies, that receives an allocation under section 4105(a) shall use a portion of such funds to develop and implement programs and activities that support ***access to a well-rounded education*** and that—…
  - (2) may be conducted in ***partnership*** with an institution of higher education, business, nonprofit organization, community-based organization, or other public or private entity with a ***demonstrated record of success in implementing activities*** under this section; and
  - (3) may include programs and activities, such as—
    - (C) programming and activities to ***improve instruction and student engagement in*** science, technology, engineering, and mathematics, including ***computer science***, (referred to in this section as ‘STEM subjects’) such as—
      - (ii) supporting the ***participation of low-income students*** in nonprofit competitions related to STEM subjects (such as robotics, science research, invention, mathematics, ***computer science***, and technology competitions);” [emphasis added]

SEC. 4109. ACTIVITIES TO SUPPORT THE EFFECTIVE USE OF TECHNOLOGY. (p. 181)

- (a) “USES OF FUNDS.—Subject to section 4106(f), each local educational agency, or consortium of such agencies, that receives an allocation under section 4105(a) shall use a portion of such funds to improve the use of technology to improve the academic achievement, academic growth, and digital literacy of all students, including by meeting the needs of such agency or consortium that are identified in the needs assessment conducted under section 4106(d) (if applicable), which may include—
  - (1) providing educators, school leaders, and administrators with the ***professional learning tools, devices, content, and resources*** to—…
  - (5) providing ***professional development in the use of technology*** (which may be provided through ***partnerships with outside organizations***) to enable teachers and instructional leaders to ***increase student achievement*** in the areas of science, technology, engineering, and mathematics, including ***computer science***; and”… [emphasis added]

SEC. 4205. LOCAL ACTIVITIES. (p. 191)

(a) “AUTHORIZED ACTIVITIES.—Each eligible entity that receives an award under section 4204 may use the award funds to carry out a broad array of activities that advance student academic achievement and support student success, including—

(13) programs that ***build skills*** in science, technology, engineering, and mathematics (referred to in this paragraph as ‘STEM’), including ***computer science***, and that ***foster innovation in learning*** by supporting ***nontraditional STEM education teaching methods***; and”… [emphasis added]

SEC. 8002. DEFINITIONS. (p. 298)

(52) WELL-ROUNDED EDUCATION.—The term ‘well-rounded education’ means courses, activities, and programming in subjects such as English, reading and language arts, writing, science, technology, engineering, mathematics, foreign language, civics and government, economics, arts, history, geography, ***computer science***, music career technical education, health, physical education, and any other subject, as determined by the State or local education agency, with the ***purpose of providing all students access to an enriched curriculum and educational experience***.” [emphasis added]

Appendix F

Sample Foundational Computer Science Curricula
Implemented by Middle Schools

Google CS First (with Scratch)

- Storytelling
    - Activity 1: Dialogue
    - Activity 2: Check It Out
    - Activity 3: Setting
    - Activity 4: Premise
    - Activity 5: Characterization
    - Activity 6: Interactive Storytelling
    - Activity 7: Personal Narrative
    - Activity 8: Your Innovation Story
- Fashion & Design
- Art
- Sports
- Music & Sound
- Game Design

Code.org: CS Fundamentals, Project GUTS and Bootstrap

- **CS Fundamentals** (formerly known as Courses 1 thru 4) – curriculum designed to introduce students ages 4+ to Computer Science. Each course has 10-15 lessons on the computer that students can do in their native language. In addition, each course includes “unplugged” English lesson plans that teachers can use to teach computer science concepts and practices without computers.
    - Course 4 (Grades 4-8)
        - Lesson 1: Unplugged: Tangrams
        - Lesson 2: Maze and Bee
        - Lesson 3: Artist
        - Lesson 4: Unplugged: Envelope Variables
        - Lesson 5: Unplugged: Madlibs
        - Lesson 6: Artist: Variables
        - Lesson 7: Play Lab: Variables
        - Lesson 8: Unplugged: For Loop Fun
        - Lesson 9: Bee: For Loops
        - Lesson 10: Artist: For Loops
        - Lesson 11: Play Lab: For Loops
        - Lesson 12: Artist: Functions
        - Lesson 13: Unplugged: Songwriting with Parameters
        - Lesson 14: Artist: Functions with Parameters
        - Lesson 15: Play Lab: Functions with Parameters

    - Lesson 16: Bee: Functions with Parameters
    - Lesson 17: Unplugged: Binary
    - Lesson 18: Artist Binary
    - Lesson 19: Super Challenge – Variables
    - Lesson 20: Super Challenge – For Loops
    - Lesson 21: Super Challenge – Functions and Parameters
    - Lesson 22: Extreme Challenge – Comprehensive

- ***CS in Science*** (Project GUTS) – a middle school science curriculum with four modules; "each consists of five or six lessons that augment the educational outcomes of traditional science instruction to include computational thinking within engaging activities of modeling and simulation."
  - Module 1: Introduction to Computer Modeling and Simulation
  - Module 2: Water as a Shared Resource
  - Module 3: Ecosystems as Complex Systems
  - Module 4: Chemical Reactions

- **CS in Algebra** (Bootstrap) – a middle school Common Core Mathematics Standards-aligned algebra curriculum with 20 lessons "designed to provide scaffolded support to both students and teachers who are new to computer science. The twenty lessons focus on concepts like order of operations, the Cartesian plane, function composition and definition, and solving word problems—all within the context of video game design.

  - Video Games and Coordinate Planes [U=unplugged]
  - Evaluation Blocks and Arithmetic Expressions
  - Strings and Images
  - Contracts, Domain, and Range [U]
  - Writing Contracts
  - Defining Variables and Substitution
  - The Big Game – Variables
  - Composite Functions
  - The Design Recipe [U]
  - Rocket Height
  - Solving Word Problems with the Design Recipe
  - The Big Game – Animation
  - Booleans and Logic [U]
  - Boolean Operators
  - Sam the Bat
  - The Big Game – Booleans
  - Conditionals and Piecewise Functions [U]
  - Conditionals
  - Collision Detection and the Pythagorean Theorem [U]
  - The Big Game – Collision Detection

Project Lead The Way (PLTW) Gateway

Middle school STEM curriculum "divided into independent, nine-week units, assuming a 45-minute class period."

- **Foundational Units**
  - Design & Modeling
  - Automation & Robotics (with VEX EDR robot kits)
    - Lesson 1: What is Automation and Robotics?
      - Positive and negative effects of automation and robotics:
        - Safety
        - Comfort
        - Choices
        - Attitudes
    - Lesson 2: Mechanical Systems
      - Study of gears and mechanisms used to change:
        - Speed
        - Torque
        - Force
        - Type of movement
        - Direction of movement
    - Lesson 3: Automated Systems
      - Flexible manufacturing systems
      - Programming of sensors, motors and building components
  - App Creators
  - Computer Science of Innovators and Makers
  - Energy and the Environment
  - Flight and Space
  - Green Architecture
  - Magic of Electrons
  - Medical Detectives
  - Enhanced Medical Detectives
  - Science of Technology

References

Aesaert, K., Vanderlinde, R., Tondeur, J., & van Braak, J. (2013). The content of educational technology curricula: A cross-curricular state of the art. *Educational Technology Research and Development, 61*, 131-151. doi: 10.1007/s11423-012-9279-9

American Association of University Women (AAUW). (2000). Tech-Savvy: educating girls in the new computer age. *AAUW Educational Foundation Commission on Technology, Gender, and Teacher Education.*

Andersen, M. L. (2000). *Thinking About Women: Sociological Perspectives on Sex and Gender* (5th Ed.), Allyn and Bacon, Boston, MA.

Anderson, N., Lankshear, C., Timms, C., & Courtney, L. (2008). ‘Because it’s boring, irrelevant and I don’t like computers’: Why high school girls avoid professionally-oriented ICT subjects. *Computers & Education, 50*(4), 1304-1318. doi: 10.1016/j.compedu.2006.12.003

Anthony, A. B., & Patravanich, S. (2014). The technology principal: To be or not to be? *Journal of Cases in Educational Leadership, 17*(2), 3-19.

Aronson, J., Fried, C. B., & Good, C. (2002). Reducing the effect of stereotype threat on African American college students by shaping theories of intelligence. *Journal of Experimental Social Psychology, 38*, 113-125.

Bamberger, Y. M. (2014). Encouraging girls into science and technology with feminine role model: Does this work? *Journal of Science Education and Technology, 23*(4), 549-561. doi: 10.1007/s10956-014-9487-7

Barker, L. J., & Aspray, W. (2006). The state of research on girls and IT. In J. M. Cohoon and W. Aspray (Eds.), *Women and information technology: Research on underrepresentation*, pp. 3-54. Cambridge, MA: MIT Press.

Barker, L. J., Snow, E., Garvin-Doxas, K., & Weston, T. (2006). Recruiting middle school girls into IT: Data on girls’ perceptions and experiences from a mixed-demographic group. In J. M. Cohoon and W. Aspray (Eds.), *Women and information technology: Research on underrepresentation*, pp. 115-136. Cambridge, MA: MIT Press.

Bartol, K. M., & Aspray, W. (2006). The transition of women from the academic world to the IT workplace: A review of the relevant research. In J. M. Cohoon and W. Aspray (Eds.), *Women and information technology: Research on underrepresentation*, pp. 377–419. Cambridge, MA: MIT Press.

Bernier, D., & Margolis, J. (2014). The revolving door: Computer science for all and the challenge of teacher retention. *Exploring Computer Science Working Papers, #3*, 1-12.

Betz, D. E., & Sekaquaptewa, D. (2012). My fair physicist? Feminine math and science role models demotivate young girls. *Social Psychological and Personality Science, 3*(6), 738-746.

Black, J., Brodie, J., Curzon, P., Myketiak, C., McOwan, P. W., & Meagher, L. R. (2013). Making computing interesting to school students: Teachers' perspectives. *Innovation and Technology in Computer Science Education (ITiCSE), July 1-3, 2013*, 255-260.

Brown, N. C. C., Kölling, M., Crick, T., Jones, S. P., Humphreys, S., & Sentance, S. (2013). Bringing computer science back into schools: Lesson from the UK. In *Proceedings of the 44th ACM technical symposium on Computer science education* (SIGCSE '13). ACM, New York, NY, 269-274. doi: 10.1145/2445196.2445277

Brown, N. C. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Transactions in Computing Education, 14*(2), 1-22. doi: 10.1145/2602484

Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education, 4*(2), 121-135.

Byars-Winston, A. (2014). Toward a framework for multicultural STEM-focused career intervention. *The Career Development Quarterly, 62*(4), 340-357. doi: 10.1002/j.2161-0045.2014.00087.x

Campbell, P. B., Jolly, E., & Perlman, L. (2004). Introducing the trilogy of success: Examining the role of engagement, capacity and continuity in women's STEM choices. *Women in Engineering ProActive Network (WEPAN) Conference, June 6-9, 2004*, Albuquerque, NM, 1-3.

Carbonaro, M., Szafron, D., Cutumisu, M., & Schaeffer, J. (2010). Computer-game construction: a gender-neutral attractor to computer science. *Computers & Education, 55*(3), 1098-1111. doi: 10.1016/j.compedu.2010.05.007

Carruthers, S., Milford, T., Pelton, T., & Stege, U. (2010). Moving K-7 education into the information age. In Proceedings of the Western Canadian Conference for Computing Education (WCCCE).

Cassell, J., & Jenkins, H. (Eds.). (2000). *From Barbie to Mortal Kombat: Gender and computer games.* Cambridge, MA: MIT Press.

Charles, M., & Bradley, K. (2006). A matter of degrees: female underrepresentation in computer science programs cross-nationally. In J. M. Cohoon and W. Aspray (Eds.), *Women and information technology: Research on underrepresentation*, pp. 183–203. Cambridge, MA: MIT Press.

Cheryan, S., Drury, B. J., & Vichayapai, M. (2012). Enduring influence of stereotypical computer science role models on women's academic aspirations. *Psychology of Women Quarterly, 37*(1), 72-79.

Cheryan, S., Plaut, V. C., Handron, C., & Hudson, C. (2013). The stereotypical computer scientist: gendered media representations as a barrier to inclusion for women. *Sex Roles, 69*(1-2), 58-71.

Clark, K., & Sheridan, K. (2010). Game Design Through Mentoring and Collaboration. *Journal of Educational Multimedia and Hypermedia, 19*(2), 5-22.

Cohoon, J. M., & Aspray, W. (2006). A critical review of the research on women's participation in postsecondary computing education. In J. M. Cohoon and W. Aspray (Eds.), *Women and information technology: Research on underrepresentation*, pp. 137-180. Cambridge, MA: MIT Press.

Cohoon, J., Cohoon, J. M., & Soffa, M. L. (2011). Focusing high school teachers on attracting diverse students to computer science and engineering. *Frontiers in Education Conference (FIE)*, F2H1-F2H-5.

College Board. (2007). AP Computer Science Teacher's Guide. New York, NY: The College Board. Retrieved February 21, 2015, from http://apcentral.collegeboard.com/apc/members/repository/ap07_compsci_teachersguide.pdf

College Board. (2010). Program summary report (pp. Number of students who took each advanced placement exam in 2010, by year in school and gender). New York, NY: The College Board. Retrieved February 15, 2015, from http://media.collegeboard.com/digitalServices/pdf/research/AP-Program-Summary-Report-2010.pdf

College Board. (2018). Number of schools offering AP exams from 2008 to 2018 (by subject). New York, NY: The College Board. Retrieved March 9, 2019, from https://secure-media.collegeboard.org/digitalServices/pdf/research/2018/2018-Exam-Volume-Change.pdf

College Board. (2018). Program summary report (pp. Number of students who took each advanced placement exam in 2018, by year in school and gender). New York, NY: The College Board. Retrieved March 9, 2019, from https://secure-media.collegeboard.org/digitalServices/misc/ap/national-summary-2018.xlsx

Cooper, J., & Weaver, K. D. (2003). *Gender and computers: Understanding the digital divide.* Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Computer Science Teachers Association (CSTA). (2006). *The new educational imperative: Improving high school computer science education using worldwide research and* *professional experience to improve US schools.* Final report of the CSTA Curriculum Improvement Task Force, February 2005. Retrieved from http://csta.acm.org/Communications/sub/DocsPresentationFiles/White_Paper07_06.pdf

Computer Science Teachers Association (CSTA). (2009). *A model curriculum for K-12 computer science level I objectives and outlines.* CSTA Curriculum Committee, 2009, 1-57. Retrieved from http://www.csta.acm.org/Curriculum/sub/CurrFiles/L1-Objectives-and-Outlines.pdf

Computer Science Teachers Association (CSTA). (2010). *Running on empty: The failure to teach K-12 computer science in the digital age.* The Association for Computing Machinery (ACM) and the Computer Science Teachers Association (CSTA). Retrieved from http://runningonempty.acm.org/fullreport2.pdf

Computer Science Teachers Association (CSTA). (2012a). *Computer Science K-8: Building a strong foundation.* Special report of the Association for Computing Machinery (ACM) and the Computer Science Teachers Association (CSTA). Retrieved from http://csta.acm.org/Curriculum/sub/CurrFiles/CS_K-8_Building_a_Foundation.pdf

Computer Science Teachers Association (CSTA). (2012b). *In need of repair: The state of K-12 computer science education in California.* The Alliance for California Computing Education for Students and Schools (ACCESS) and the Computer Science Teachers Association (CSTA). Retrieved from http://access-ca.org/wp-content/uploads/sites/4/2015/08/ACCESS-InNeedofRepair.pdf

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th ed.). Thousand Oaks, CA: Sage.

Cuny, J. (2012). Transforming high school computing: a call to action. *ACM Inroads, 3*(2), 32-36.

Denner, J. (2011). What predicts middle school girls' interest in computing? *International Journal of Gender, Science and Technology, 3*(1), 53-69.

Denner, J., & Werner, L. (2007). Computer programming in middle school: How pairs respond to challenges. *Journal of Educational Computing Research, 37*(2), 131-150.

Denner, J., Martinez, J., Thiry, H., & Adams, J. (2014). Computer science and fairness: Integrating a social justice perspective into an after-school program. *Science Education and Civic Engagement, 7*(2), 41-54.

Diekman, A. B., Brown, E. R., Johnston, A. M., & Clark, E. K. (2010). Seeking congruity between goals and roles: A new look at why women opt out of science, technology, engineering, and mathematics careers. *Psychological Science, 21*(8), 1051-1057.

Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? *Workshop in Primary and Secondary Computing Education (WiPSCE), November 5-7, 2014*, 60-69.

Dryburgh, H. (2000). Underrepresentation of girls and women in computer science: classification of 1990s research. *Journal of Educational Computing Research, 7*(2), 181-202.

Eccles, J. S. (1989). Bringing young women to math and science. In M. Crawford and M. Gentry (Eds.), *Gender and Thought: Psychological Perspectives*. New York: Springer-Verlag.

Eccles, J. S., Midgley, C., Wigfield, A., Buchanan, C. M., Reuman, D., Flanagan, C., & Iver, D. M. (1993). Development during adolescence: The impact of stage-environment fit on young adolescents' experiences in schools and in families. *American Psychologist, 48*(2), 90-101.

Eccles, M. P., & Mittman, B. S. (2006). Welcome to implementation science. *Implementation Science, 1*(1), 1-3. doi: 10.1186/1748-5908-1-1

Emmott, S., & Rison, S. (2006). Towards 2020 Science. The 2020 Science Group. Cambridge, UK: Microsoft Research.

Eubanks, V. (2009). Double-bound: Putting the power back into participatory research. *Frontiers, 30*(1), 107-138.

Eubanks, V. (2011). Digital dead end: Fighting for social justice in the information age. Cambridge, MA: The MIT Press.

Falkner, K., Vivian, R., & Falkner, N. (2014). The Australian digital technologies curriculum: Challenge and opportunity. *Conferences in Research and Practice in Information Technology, 148*, 3-12.

Finson, K. D. (2009). What drawings reveal about perceptions of scientists: visual data operationally defined. In Pedersen, J.E. and Finson, K.D. (Eds.), *Visual Data: Understanding and Applying Visual Data to Research in Education*, pp. 59-78. Rotterdam, The Netherlands: Sense Publishers.

Fisher, J., Lang, C., Craig, A., & Forgasz, H. (2015). If girls aren't interested in computers can we change their minds? *ECIS 2015 Completed Research Papers,* 45, 1-14.

Fixsen, D. L., Naoom, S. F., Blase, K. A., Friedman, R. M., & Wallace, F. (2005). *Implementation research: A synthesis of the literature.* National Implementation Research Network. (FMHI Publication #231), Tampa, FL: University of South Florida, Louis de la Parte Florida Mental Health Institute.

Frank, K. A., Zhao, Y., & Borman, K. Social capital and the diffusion of innovations within organizations: The case of computer technology in schools. *Sociology of Education, 77*, 148-171.

Frenkel, K. A. (1990). Women and computing. *Communications of the Association for Computing Machinery (ACM), 33*(11), 34-46. doi 10.1145/92755.92756

Glenn, J. (2000). Before it's too late. *A Report to the Nation from The National Commission on Mathematics and Science Teaching for the 21st Century.*

Goode, J. (2007). If you build teachers, will students come? the role of teachers in broadening computer science learning for urban youth. *Journal of Educational Computing Research, 36*(1), 65-88.

Goode, J. (2010). Connecting K-16 curriculum & policy: Making computer science engaging, accessible, and hospitable to underrepresented students. *Proceedings of the 41st ACM technical symposium on computer science education (SIGCSE '10)*, 22-26. doi: 10.1145/1734263.1734272

Goode, J., Margolis, J., & Chapman, G. (2014). Curriculum is not enough: The educational theory and research foundation of the Exploring Computer Science professional development model. *Journal of Educational Computing Research, 36*(1), 65-88.

Google. (2014). *Women who choose computer science—what really matters: The critical role of encouragement and exposure*. Retrieved from https://www.google.com/edu/resources/computerscience/research/

Google & Gallup. (2015a). *Searching for computer science: Access and barriers in U.S. K-12 education*. Retrieved from https://www.google.com/edu/resources/computerscience/research/

Google & Gallup. (2015b). *Images of computer science: Perceptions among students, parents, and educators in the U.S*. Retrieved from https://www.google.com/edu/resources/computerscience/research/

Guzdial, M., Ericson, B., McKlin, T., & Engelman, S. (2014). Georgia Computes! An intervention in a US state, with formal and informal education in a policy context. *ACM Transactions on Computing Education, 14*(2), 1-29.

Halpern, D. F., Aronson, J., Reimer, N., Simpkins, S., Star, J. R., & Wentzel, K. (2007). Encouraging girls in math and science [IES Practice Guide]. Washington, DC: Institute of Education Sciences, U.S. Department of Education. Retrieved from http://ies.ed.gov/ncee/wwc/practiceguide.aspx?sid=5

Hess, R. D., & Miura, I. T. (1985). Gender differences in enrollment in computer camps and classes. *Sex Roles, 13*(3-4), 193-203.

Hill, C., Corbett, C., & St. Rose, A. (2010). Why so few? women in science, technology, engineering, and mathematics. *American Association of University Women Report.*

Hilton, T. L., & Lee, V. E. (1988). Student interest and persistence in science: changes in the educational pipeline in the last decade. *The Journal of Higher Education,* (59), 510-526.

Holland, J. M., Major, D. A., & Orvis, K. A. (2012) Understanding how peer mentoring and capitalization link STEM students to their majors. *The Career Development Quarterly, 60*, 343-354.

Hong, L., & Page, S. E. (2004). Groups of diverse problem solvers can outperform groups of high-ability problem solvers. Proceedings of the National Academy of Sciences of the United States of America, 101, 16385–16389. doi: 10.1073/pnas.0403723101

Hubweiser, P., Armoni, M., Giannakos, M. N., & Mittermeir, R. T. (2014) Perspectives and visions of computer science education in primary and secondary (K-12) schools. *ACM Transactions on Computing Education, 14*(2), 7:1-7:9.

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education, 82*(4), 263-279. doi: 10.1016/j.compedu.2014.11.022

Kaczmarczyk, L., & Dopplick, R. (2014). *Rebooting the pathway to success: Preparing students for computing workforce needs in the United States*. New York, NY: Association for Computing Machinery.

Kafai, Y. B., Griffin, J., Burke, Q., Slattery, M., Fields, D. A., Powell, R. M., Grab, M., Davidson, S. B., & Sun, J. S. (2010). A cascading mentoring pedagogy in a CS service learning course to broaden participation and perceptions. In *Proceedings of the 44th ACM technical symposium on Computer science education* (SIGCSE '13). ACM, New York, NY, 101-106.

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming.* Cambridge, MA: MIT Press.

Karcher, M. J. (2008). The study of mentoring in the learning environment (SMILE): A randomized evaluation of the effectiveness of school-based mentoring. *Prevention Science, 9*(2), 99-113. doi: 10.1007/s11121-008-0083-z

Koch, M., & Gorges, T. (2016). Curricular influences on female afterschool facilitators' computer science interests and career choices. *Journal of Science Education & Technology, 25*(5), 782-794.

Kori, K., Pedaste, M., Leijen, A., & Tõnisson, E. (2014). The role of programming experience in ICT students' learning motivation and academic achievement. *International Journal of Information and Education Technology, 6*(5), 331-337.

Ladson-Billings, G. (2006). From the achievement gap to the education debt: Understanding achievement in U.S. Schools. *Educational Researcher, 35*(7), 3-12.

Lee, A. (2015). Determining the effects of computer science education at the secondary level on STEM major choices in postsecondary institutions in the United States. *Computers & Education, 88*(8), 241-55. doi: 10.1016/j.compedu.2015.04.019

Lehman, K. J., Sax, L. J., & Zimmerman, H. B. (2017). Women planning to major in computer science: Who are they and what makes them unique? *Computer Science Education, 26*(4), 277-298.

Lemons, M. A., & Parzinger, M. (2007). Gender schemas: a cognitive explanation of discrimination of women in technology. *Journal of Business and Psychology, 22*(1), 91-98.

Lightbody, P., Siann, G., Tait, L., & Walsh, D. (1997). A fulfilling career? factors which influence women's choice of profession. *Educational Studies, 23*(1), 25-37. doi: 10.1080/0305569970230102

Ma, Y. (2011). Gender differences in the paths leading to a STEM baccalaureate. *Social Science Quarterly, 92*(5), 1169-1190.

Manches, A., & Plowman, L. (2015). Computing education in children's early years: A call for debate. *British Journal of Educational Technology, 46*(6), 1-11.

Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse: Women in computing.* Cambridge, MA: MIT Press.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., and Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing.* Cambridge, MA: The MIT Press. Kindle Edition.

Margolis, J., Goode, J., & Binning, K. R. (2015). Exploring Computer Science: Active learning for broadening participation in computing. *Computing Research News, 27*(9), 16-19.

Margolis, J., Goode, J., & Chapman, G. (2015). An equity lens for scaling: A critical juncture for Exploring Computer Science. *ACM Inroads, 6*(3), 58-66. doi: 10.1145/2794294

Marshall, C., & Rossman, G. B. (2011). *Designing qualitative research* (5th ed.). Thousand Oaks, CA: Sage.

Maxwell, J. A. (2013). *Qualitative research design: An interactive approach.* Thousand Oaks, CA: Sage.

Melguizo,T., & Wolniak, G. C. (2012). The earnings benefits of majoring in STEM fields among high achieving minority students. *Research in Higher Education, 53*(4), 383-405. doi: 10.1007/s11162-011-9238-z

Menekse, M. (2015). Computer science teacher professional development in the United States: A review of studies published between 2004 and 2014. *Computer Science Education, 26*(1), 1-26. doi: 10.1080/08993408.2015.1111645

Merriam, S. B. (2009). *Qualitative research: A guide to design and implementation* (2nd ed.) San Francisco: Jossey-Bass.

Mertens, D. M. (2015). *Research and evaluation in education and psychology: Integrating diversity with quantitative, qualitative, and mixed methods* (4th ed.). Thousand Oaks, CA: Sage.

Millar, J., & Jagger, N. (2001). Women in ITEC Courses and Careers. *Report WIT 1, Department for Education and Skills*, November 2001. Retrieved from http://www.employment-studies.co.uk/system/files/resources/files/1416wit1.pdf

Minogue, J. (2010). What is the teacher doing? What are the students doing? An application of the Draw a-Science-Teacher test. *Journal of Science Teacher Education, 21*(7), 767-781.

Modekurty, S., Fong, J., & Cheng, H. H. (2014). C-STEM girls computing and robotics leadership camp. *121st ASEE Annual Conference & Exposition,* 1-14.

Molla, R., & Lightner, R. (2014, December 30). Diversity in tech. *Wall Street Journal*. Retrieved February 17, 2016, from http://graphics.wsj.com/diversity-in-tech-companies/

Mosatche, H. S., Matloff-Nieves, S., Kekelis, L., & Lawner, E. K. (2013). Effective STEM programs for adolescent girls: Three approaches and many lessons learned. *Afterschool Matters, 17*, 17-25.

Moses, M. S., Howe, K. R., & Niesz, T. (1999). The pipeline and student perceptions of schooling: Good news and bad news. *Educational Policy, 13*(4), 573-591.

National Coordination Office for High Performance Computing and Communication (NCO-HPCC). (1996). High Performance Computing and Communications: Foundation for America's Information Future (Supplement to the President's FY 1996 Budget). Bethesda, MD. Retrieved March 16, 2015, from https://www.nitrd.gov/PUBS/bluebooks/1996/index.aspx

National Science Foundation (NSF). (2006). New Formulas for America's Workforce 2: Girls in Science and Engineering. Program for Gender Diversity in STEM Education.

Ni, L. (2009). What makes CS teachers change? Factors influencing CS teachers' adoption of curriculum innovations. *ACM SIGCSE Bulletin, 41*(1), 544-548. doi: 10.1145/1539024.1509051

Nord, C., Roey, S., Perkins, R., Lyons, M., Lemanski, N., Brown, J., and Schuknecht, J. (2011). The Nation's Report Card: America's High School Graduates (NCES 2011-462). U.S.

Department of Education, National Center for Education Statistics. Washington, DC: U.S. Government Printing Office.

Olszewski-Kubilius, P. (2003). Do we change gifted children to fit gifted programs, or do we change gifted programs to fit gifted children? *Journal for the Education of the Gifted, 26*(4), 304-313.

Papastergiou, M. (2008). Are computer science and information technology still masculine fields? High school students' perceptions and career choices. *Computers & Education, 51*(2), 594-608. doi: 10.1016/j.compedu.2007.06.009

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer.* New York, NY: Basic Books.

Parker, M. C., & Guzdial, M. (2015). A critical research synthesis of privilege in computing education. *Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 1-5. doi: 10.1109/RESPECT.2015.7296502

Patton, M. Q. (2015). *Qualitative research & evaluation methods: Integrating theory and practice* (4th ed.). Thousand Oaks, CA: Sage.

Prensky, M. (2008, January 13). Programming is the new literacy. *Edutopia*. Retrieved from www.edutopia.org

Ravitch, D. (2001). *Left back: A century of battles over school reform*. New York: Simon & Schuster.

Reid, P. T., & Roberts, S. K. (2006). Gaining options: A mathematics program for potentially talented at-risk adolescent girls. *Merrill-Palmer Quarterly, 52*(2), 288-304. doi: 10.1353/mpq.2006.0019

Reinen, I. J., & Plomp, T. (1993). Some gender issues in educational computer use: Results of an international comparative survey. *Computers & Education, 20*(4), 353-365. doi: 10.1016/0360-1315(93)90014-A

Reinen, I. J., & Plomp, T. (1997). Information technology and gender equality: A contradiction in terminis. *Computers & Education, 28*(2), 65-78. doi: 10.1016/S0360-1315(97)00005-5

Rodger, S. H., Brown, D., Hoyle, M., MacDonald, D., Marion, M., Onstwedder, E., Onwumbiko, B., & Ward, E. (2014). Weaving computing into all middle school disciplines. In Proceedings of the 2014 conference on Innovation & technology in computer science education (ITiCSE '14) (pp. 207-212). New York, NY: Association for Computing Machinery (ACM) Publications. doi: 10.1145/2591708.2591754

Rogers, E. (2003). *Diffusion of innovations* (5th ed.). New York, NY: Free Press.

Rosenbloom, J. L., Ash, R. A., Dupont, B., & Coder, L. (2008). Why are there so few women in information technology? assessing the role of personality in career choices. *Journal of Economic Psychology, 29*, 543-554. doi: 10.1016/j.joep.2007.09.005

Rubin, H.J., & Rubin, I.S. (2012). *Qualitative interviewing: The art of hearing data* (3rd ed.). Thousand Oaks, CA: Sage.

Ryoo, J. J., Margolis, J., Lee, C. H., Sandoval, C. D. M., & Goode, J. (2012). Democratizing computer science knowledge: transforming the face of computer science through public high school education. *Learning, Media and Technology, 38*(2), 161-181.

Sanders, J. (2006). Gender and technology: What the research tells us. In C. Skelton, B. Francis, & L. Smulyan (Eds.), The Sage Handbook of Gender and Education (pp. 307-321). London, England: Sage Publications.

Sax, L. J., & Harper, C. E. (2005). Origins of the gender gap: Pre-college and college influences on differences between men and women. *Paper presented at the Annual Meeting of the Association for Institutional Research*.

Sax, L. J., Lehman, K. J., Jacobs, J. A., Kanny, M. A., Lim, G., Monje-Paulson, L., & Zimmerman, H. B. (2017). Anatomy of an enduring gender gap: The evolution of women's participation in computer science. *The Journal of Higher Education, 88*(2), 258-293.

Sentence, S., & Csizmadia, A. (2015). Teachers' perspectives on successful strategies for teaching computing in school. *Paper presented at IFIP TCS, June 2015*, 1-10.

Shapiro. J. R., Williams, A. M., & Hambarchyan, M. (2013). Are all interventions created equal? A multi-threat approach to tailoring stereotype threat intervention interventions. *Journal of Personality and Social Psychology, 104*(2), 277-88. doi: 10.1037/a0030461

Simard, C., Stephenson, C., & Kosaraju, D. (2010). *Addressing core equity issues in K-12 computer science education: Identifying barriers and sharing strategies*. Anita Borg Institute for Women and Technology.

Simard, C., & Gammal, D. L. (2012). *Solutions to recruit technical women.* Anita Borg Institute Solutions Series Report.

Stake, R. E. (2006). *Multiple Case Study Analysis* (1st Kindle ed.). New York, NY: The Guilford Press.

Stephens, M. C., Jelenac, P., & Noack, P. (2010). On the leaky math pipeline: comparing implicit math-gender stereotypes and math withdrawal in female and male children and adolescents. *Journal of Educational Psychology, 102*(4), 947-963.

Stout, J. G., Dasgupta, N., Hunsinger, M., & McManus, M. (2011). STEMing the tide: Using ingroup experts to inoculate women's self-concept in science, technology, engineering, and mathematics (STEM). *Journal of Personality and Social Psychology, 100*(2), 255-270. doi: 10.1037/a0021385

Strang, D., & Meyer, J. W. (1993). Institutional conditions for diffusion. *Theory and Society, 22*, 487-511.

Tan, E., Barton, A. C., Kang, H., & O'Neill, T. (2013). Desiring a career in STEM-related fields: how middle school girls articulate and negotiate identities-in-practice in science. *Journal of Research in Science Teaching, 50*(10), 1143-1179.

Trigg, L. J., & Perlman, D. (1976). Social influences on women's pursuit of a nontraditional career. *Psychology of Women Quarterly, 1*(2), 138-150. doi: 10.1111/j.1471-6402.1976.tb00814.x

U.S. Department of Labor, Bureau of Labor Statistics. (2015a, December 17). *Occupational outlook handbook, 2016-17 Edition*, Software Developers. Retrieved March 2, 2016, from http://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm#tab-6

U.S. Department of Labor, Bureau of Labor Statistics. (2015b, December 17). *Occupational outlook handbook, 2016-17 Edition*, Computer Network Architects. Retrieved March 2, 2016, from http://www.bls.gov/ooh/computer-and-information-technology/computer-network-architects.htm#tab-6

U.S. Department of Labor, Bureau of Labor Statistics. (2015c, December 17). *Occupational outlook handbook, 2016-17 Edition*, Information Security Analysts. Retrieved March 2, 2016, from http://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm#tab-6

Wang, M.-T., & Degol, J. (2013). Motivational pathways to STEM career choices: Using expectancy-value perspective to understand individual and gender differences in STEM fields. *Developmental Review, 33*, 304-340.

Ware, J., & Stein, S. (2013). From "mentor" to "role model": Scaling the involvement of STEM professionals through role model videos. *Journal of Educational Multimedia and Hypermedia, 22*(2), 209-223.

Watson, C., & Li, F. W. B. (2014). Failure rates in introductory programming revisited. *ITiCSE*, 39-44.

Werner, L., Denner, J., Campe, S., Ortiz, E., Delay, D., Hartl, A.C., & Laursen, B. (2010). Pair programming for middle school students: Does friendship influence academic outcomes? *ACM*.

Whitecraft, M. A., & Williams, W. M. (2010). Why aren't more women in computer science? In *Making Software: What Really Works, and Why We Believe It*, ed. A. Oram and G. Wilson, 221-238. Sebastopol, CA: O'Reilly Media, Inc.

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). Running on empty: The failure to teach K-12 computer science in the digital age. Report of the Association for Computing Machinery (ACM) & Computer Science Teachers Association (CSTA). Retrieved from http://runningonempty.acm.org/fullreport2.pdf

Wing, J. M. (2006). Computational thinking. *Communications of the Association for Computing Machinery (ACM), 49*(3), 33-35.

Yin, R. K. (2014). *Case study research: Design and methods* (5th ed.). Thousand Oaks, CA: Sage.

Zarrett, N. R., & Malanchuk, O. (2005). Who's computing? Gender and race differences in young adults' decisions to pursue an information technology career. *New Directions for Child and Adolescent Development,* (110), 65-84.

Zarrett, N. R., Malanchuk, O., Davis-Kean, P. E., & Eccles, J. (2006). Examining the gender gap in IT by race: Young adults' decisions to pursue an IT career. In J. M. Cohoon and W. Aspray (Eds.), *Women and information technology: Research on underrepresentation*, pp. 55-88. Cambridge, MA: MIT Press.

Zweben, S. (2014). Computing degree and enrollment trends: Undergraduate enrollment grows for sixth straight year and Ph.D. production reaches an all-time high (From the 2012-2013 CRA Taulbee Survey). Retrieved March 15, 2015, from http://cra.org/resources/

Zweben, S., & Bizot, B. (2014). The 2013 Taulbee survey: second consecutive year of record doctoral degree production; continued strong undergraduate CS enrollment. Retrieved March 15, 2015, from http://cra.org/resources/