**ABSTRACT**

Mark J. Samberg, PROBLEM SOLVING IN THE DIGITAL AGE: BRINGING DESIGN AND COMPUTATIONAL THINKING TO THE K-12 CLASSROOM (under the direction of Dr. Matthew Militello) Department of Educational Leadership, February 2018.

A focus on Science, Technology, Engineering, and Math (STEM) education has been an ongoing trend in the United States for most of the last decade. Recently, computer science has stood out as a focus within this movement. Supported by industry, non-profits, federal and local governments, the “CS4All” movement aims to provide every student the opportunity to learn to code. While many of these initiatives focus solely on coding, others are also advocating for students to learn skills required to structure problems so that they may be solved by a computer. As defined Jeanne Wing in 2006, *computational thinking* is part of a suite of problem solving tools in engineering, among design thinking (including human-centered design) and data literacy (the ability to collect, understand, use, and share data with others).

Computational thinking skills, combined with design thinking and data literacy (collectively called *digital-age problem solving)* blend core critical thinking concepts from both STEM education and the Humanities. This study focuses on preparing teachers to integrate digital-age problem solving into their instructional practice by immersing teachers in a Massive Open Online Course for Educators (MOOC-Ed) through the Friday Institute for Educational Innovation. The MOOC-Ed focuses on exposing teachers to digital-age problem solving concepts and supporting them in identifying examples in their current practice, and deepening integration in both their reflective practice and their work with students. Applications of digital-age problem solving are found both online and offline, and the MOOC-Ed focuses on helping educators identify and use these practices and skills in their daily practice. Through this MOOC-Ed, a model for digital-age problem solving was shared with practitioners from around the world.

The MOOC-Ed was a valuable tool for participating teachers, with 97% of all course completers prepared to make positive changes to their practice. The digital-age problem solving cycle demonstrated value in helping teachers develop language around problem solving, making changes to their reflective practice, and creating hands-on learning experiences for students. Digital-age problem solving was useful to teachers beyond STEM fields, with teachers from all disciplines reporting and demonstrating value in the model.

PROBLEM SOLVING IN THE DIGITAL AGE: BRINGING DESIGN AND COMPUTATIONAL THINKING TO THE K-12 CLASSROOM

A Dissertation

Presented to

The Faculty of the Department of Educational Leadership

East Carolina University

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Education in Educational Leadership

by

Mark Justin Samberg

February, 2018

PROBLEM SOLVING IN THE DIGITAL AGE: BRINGING DESIGN AND COMPUTATIONAL THINKING TO THE K-12 CLASSROOM

by

Mark Justin Samberg

APPROVED BY:

DIRECTOR OF DISSERTATION:_______________________________________

Matthew Militello, PhD

COMMITTEE MEMBER:_______________________________________

R. Martin Reardon, PhD

COMMITTEE MEMBER:_______________________________________

Karen Jones, PhD

COMMITTEE MEMBER:_______________________________________

Thomas Williams, EdD

COMMITTEE MEMBER:_______________________________________

Glenn Kleiman, PhD

INTERIM CHAIR OF THE DEPARTMENT OF EDUCATIONAL LEADERSHIP:

_______________________________________

Marjorie Ringler, EdD

DEAN OF THE GRADUATE SCHOOL:

_______________________________________

Paul Gemperline, PhD

## DEDICATION

This dissertation is dedicated to my future husband, Cody Hamrick, whose love and support made completing this dissertation possible; to my parents Lynn and Art Samberg; my brother Stuart Samberg and sister-in-law Brandy and to their kids, Noah, Ainsley, and Charlotte.

## ACKNOWLEDGEMENTS

This project would not have been possible without the support and encouragement of the fantastic team at the Friday Institute for Educational Innovation.

I would like to express my deepest gratitude to Dr. Glenn Kleiman for helping me develop the idea for this project, and for providing the time and resources to move it forward. I would also like to thank Dr. Dave Frye for his support and encouragement and for helping me arrange my workload to make this project possible.

The development of a MOOC is not a one-person show. Our small-but-scrappy MOOC team easily rivals any team out there: Alex Dreier for supporting the instructional design and helping refine the course content, Meghan Day for turning my last-minute deluge of Google Docs into an actual course, Lauren Acree for developing the micro-credential program, Alison Graham and Greg Garner for coordinating and scoring micro-credential submissions, Dr. Shaun Kellogg and Rob Maser for their work in developing the surveys used across all of our MOOC courses and for helping me collect the data I needed, Patrick Nichols for driving all over Raleigh to collect video from our expert panelists and combining it with my awkward stammering to create studio-class intro videos, and to Blythe Tyrone and Heather Bronson for their efforts in getting the word out about this new MOOC course. I would also like to thank the people who reviewed and critiqued the content: Greg Garner, Sam Morris, Ed Chase, Dr. Eric Wiebe, Dr. Glenn Kleiman, Dr. Dave Frye, Dr. Shaun Kellogg, Dr. Matt Militello, and Dr. Martin Reardon. Your input and feedback was critical to making this course a success.

Throughout the course units, we heard from practitioners in a variety of fields about how they integrated computational thinking and design in their field. Thank you to Dr. Avril Smart, Shreyas Bharadwaj, Jason Bosko, Bob Clark, Jaquasha Colon, Dr. Jen Sabourin, Blythe Tyrone, Bob Clark, Stuart Samberg, and Dr. Brian Wolf for your contributions and for sharing your experiences with our participants.

Thank you also to Dr. Tom Williams for encouraging me to apply and pursue a degree at East Carolina, for checking up on me for the last few years, and for returning to East Carolina to serve on my committee. Thank you also to Dr. Martin Reardon for reviewing so many early drafts of the first two chapters for serving on my committee along with Dr. Karen Jones. I am also extremely grateful to Dr. Matt Militello for supervising me through this process, for encouraging and supporting my work, and for reviewing each and every draft I threw at him.

And last, but certainly not least, I need to acknowledge and thank every educator and participant in the *Computational Thinking and Design* MOOC-Ed. Thank you for diving in to something new, coming back each week, trusting the process, participating fully and thoughtfully, trying something new in your classrooms, and for providing me with invaluable feedback.

## TABLE OF CONTENTS

| | Page |
|---|---|
| TITLE PAGE | i |
| COPYRIGHT | ii |
| SIGNATURE | iii |
| DEDICATION | iv |
| ACKNOWLEDGEMENTS | v |
| LIST OF TABLES | xi |
| LIST OF FIGURES | xii |
| CHAPTER 1: INTRODUCTION | 1 |
| Background | 1 |
| Digital Age Problem Solving – A Conceptual Framework | 3 |
| Purpose of Study | 5 |
| MOOCs and Micro-Credentials | 7 |
| Problem of Practice Project | 8 |
| Improvement Goal | 8 |
| Questions and Tasks | 10 |
| CHAPTER 2: REVIEW OF RELATED LITERATURE | 13 |
| Content Considerations of a Computational Design Thinking Course | 13 |
| Computational Thinking | 13 |
| Coding Courses | 16 |
| Pedagogical Approaches to Teaching Code | 19 |
| Design Thinking | 21 |

Data Literacy…………………………………………………………….. 25

Social Context……………………………………………………………... 25

Rise of STEM Education………………………………………………… 25

STEM Schools – Implications for Leadership…………………………… 30

Under-Represented Populations in Computer Science…………………... 31

Teacher Professional Learning…………………………………………….. 35

Effective Practices for Teacher Professional Learning…………………... 35

Rise of MOOCs…………………………………………………………... 36

Badging and Micro-Credentials………………………………………….. 38

CHAPTER 3: METHODOLOGY………………………………………………….. 40

MOOC Course Design……………………………………………………….. 40

Micro-Credentials and Completion Certificates……………………………... 45

Participant Population……………………………………………………….... 47

Research Agenda……………………………………………………………... 48

Quantitative Analysis………………………………………………………… 49

Qualitative Analysis………………………………………………………….. 49

CHAPTER 4: DATA COLLECTION……………………………………………… 51

MOOC Participants………………………………………………………….. 52

Recruitment……………………………………………………………… 52

Study Population…………………………………………………………. 57

Engagement…………………………………………………………………... 64

Resources………………………………………………………………… 65

Forum Participation……………………………………………………… 71

Summary of Course Engagement Data..... 78
Course Completers..... 78
In-Course Evaluations..... 86
Summative Evaluations..... 91
Summary of Survey Data..... 103
Engagement in Practice..... 105
Participant Interviews..... 105
Micro-Credentials..... 111
Summary..... 120
CHAPTER 5: DISCUSSION AND IMPLICATIONS..... 123
Summary of Findings..... 123
MOOC-Ed Effectiveness..... 124
Translation to Practice..... 125
Digital-Age Problem Solving as a Conceptual Framework..... 127
Implications..... 131
STEM Education..... 131
For Future Research..... 138
Revisions for Future Courses..... 139
Reflection..... 143
REFERENCES..... 145
APPENDIX A: IRB APPROVAL LETTER..... 161
APPENDIX B: MOOC COURSE DESCRIPTION..... 162
APPENDIX C: EXISTING MOOC-ED SURVEY QUESTIONS..... 163

APPENDIX D: NEW SURVEY QUESTIONS AND INTERVIEW QUESTIONS….. 169

APPENDIX E: MOOC-ED COURSE OUTLINE…………………………………….. 172

APPENDIX F: MOOC-ED COURSE CONTENT……………………………………. 176

## LIST OF TABLES

1. Logic Model........ 9

2. Research Questions and Data Sources........ 11

3. Study Timeline........ 12

4. Digital-Age Problem Solving Micro-Credential Stack........ 46

5. Research Questions and Analysis Plan........ 50

6. User-Reported Referrals to the MOOC-Ed Course........ 55

7. Most Frequently Accessed Course Resources........ 69

8. Least-Frequently Accessed Course Resources........ 70

9. View Counts for Course "Deep-Dive" Content and Videos........ 72

10. Transcript Analysis Framework Codes (Fahy et al., 2001) with MOOC-Ed Examples........ 74

11. Additional Learning Community-Focused Codes in MOOC-Ed........ 79

12. Course Competencies Explicitly Stated or Implied in MOOC-Ed Discussions...... 81

13. Percentage of Users Answering "Agree" or "Strongly Agree" in End-of-Unit Surveys........ 88

14. *End-of-Course Survey Responses: Effectiveness of MOOC Components* ........ 92

15. *End-of-Course Survey Responses: Improvement of Knowledge/Skills* ........ 93

16. Major Themes from End-of-Course Survey Question on Useful Course Elements 95

17. Major Themes from End-of-Course Survey Question on Application to Practice... 97

18. MOOC-Ed Impact Survey Results........ 99

19. Participant Response Themes in Application to Practice in Impact Survey........ 102

20. Major Themes from Implementation with Students Question in Impact Survey.... 104

21. Impact on Practice Aligned to Clarke and Hollingsworth's Interconnected Model of Professional Growth........ 107

## LIST OF FIGURES


1. Draft conceptual framework describing the relationship between computational thinking, design, and data literacy………………………………………………. 6

2. Components of Design Thinking (Ingle, 2013)……………………………………… 22

3. Computational Thinking and Design (Digital Age Problem Solving Cycle)……… 41

4. Screenshot from MOOC-Ed promotional video…………………………………... 53

5. Social media marketing slide for CTD MOOC-Ed………………………………... 54

6. Course participant demographic information……………………………………… 59

7. Number of participants by U.S. States (not pictured, Hawaii, 1 participant)……… 60

8. Number of participants by country………………………………………………... 61

9. Participant motivation for enrolling/self-assessment……………………………… 63

10. Producers and consumers by unit………………………………………………….. 66

11. Number of users viewing course resources by unit………………………………... 68

12. Registration data on enrollment motivations for course completers………………. 87

13. Interconnected Model of Professional Growth (Clarke & Hollingsworth, 2002)…. 106

14. Micro-credential artifact: Student root cause analysis…………………………….. 112

15. Teacher micro-credential artifacts: Student solutions……………………………... 114

16. Administrator micro-credential: Infographic submission…………………………. 117

17. Administrator micro-credential submission: Decomposition of a school master schedule…………………………………………………………………………. 118

18. Digital-age problem solving……………………………………………………….. 128

19. Digital-age problem solving cycle…………………………………………………… 129

20. STEM 5.0 Conceptual Framework………………………………………………... 134

# CHAPTER 1: INTRODUCTION

## Background

When people look at any computer application, website, or video game, they tend to assume that it was created by a single computer programmer (usually in a dark room, discarded remnants of caffeinated beverages strewn about). However, in most cases, that could not be further from the truth. Consider the last video game that you played. To create that game, there were definitely programmers involved. But there were also graphic artists creating imaginative virtual landscapes and gaming elements, video producers and animators bringing these creations to life, sound designers and musicians setting the scene with background music and effects, creative writers developing engaging and interesting plots, mathematicians and physicists making sure all of the movement in the game is realistic, and businesspeople keeping everything on time and on budget. This team of experts worked together to create something new and unique, and none of them could have done it alone. This approach to problem solving is becoming increasingly commonplace in many industries. How can schools teach students to use them?

Over the last few years, there has been seemingly unending publicity around the need for an increased focus on STEM education and computer science education in the schools. One program, known as the *Hour of Code* (“What's the impact of the Hour of Code?,” 2016) has been gaining popularity in the last few years, by encouraging schools to spend an hour (usually in the first week of December, during Computer Science Education Week) celebrating coding and having students engage in computer coding activities. In 2015, the *Hour of Code* boasted over 198,000 educators helping students to write over 11 billion lines of code (“What's the impact of the Hour of Code?,” 2016). Additionally, the U.S. Department of Education under President

Obama unveiled the *Computer Science for All* initiative ("FACT SHEET: President Obama Announces Computer Science For All Initiative," 2016). Among other things, this initiative directs the National Science Foundation and other granting agencies to provide funds to increase the availability of computer science programs within American schools. President Obama has called coding "a basic skill" (Obama, 2016). In 2017, President Trump continued this effort by issuing a memo directing the Secretary of Education to "establish a goal of devoting at least $200 million in grant funds per year to the promotion of high-quality STEM education, including Computer Science in particular (Trump, 2017)."

Since the earliest personal computers, students in school have learned computer programming using a variety of tools, largely starting with LOGO Turtle (Papert, 1993). The development of LOGO led to a collaboration with LEGO for the creation of computer controlled LEGO kits (Papert, 1993), eventually becoming the LEGO Mindstorms Robotics Kits. The FIRST Robotics Competition has engaged high schoolers in creating computer controlled robots for competition since the mid-90s, and now has competitions for students grades from grades pre-K through 12 (Chung, Cartwright, & Cole, 2014). Toys for children as young as three years old are now also teaching coding, as computer programming becomes more accessible, and computers become cheaper, more durable, and more portable. Devices such as Sphero, Blockly, Primo, Dash and Dot aim to teach coding skills using physical devices that can be manipulated like a puzzle or with tablets and smartphones (Olivares-Giles, 2015).

Despite the swell of policy and corporate support around increasing computer science program availability within schools, there are still many barriers to implementation. Many schools are working on implementing coding programs in their schools. However, there is an extreme shortage of skilled, qualified computer science teachers, and many states still have unclear or ever-changing certification requirements to teach computer science (Computer Science Teachers Association, 2013). Additionally, there is a strong need to engage girls and students of color in computer science initiatives. Only 18% of the computer science workforce is female, which is down from 37% in the mid-1980s (Google, 2014). Students of color are also severely under-represented in computer science courses, with only 13.2% of all AP Computer Science test takers in 2013 being black or Hispanic (College Board, 2014c). Several states are beginning to create standards and efforts to further computer science education and the integration of computational thinking.

According to the North Carolina Department of Public Instruction, in North Carolina, less than 300 teachers currently teach computer science courses with only 18 students graduating ready to teach computer science in 2015 (Frye, Samberg, Moris, & Keller, 2017). Additionally, only 22% of the 5,000 students taking a computer science course are female, and only 30% are African American or Hispanic (Frye et al., 2017).

**Digital Age Problem Solving – A Conceptual Framework**

Just as the grammar, syntax, and vocabulary only defines the structure of language and not the ability to create works of literature, so too is it with computer science – coding is only the language of computer science (Igoe, 2016). Many current computer science courses focus on basic computer operations or repair, along with coding, with minimal focus on the design and problem solving skills necessary to be a successful software engineer (Jones, Mccowan, & Stephenson, 2003). There is concern that this emphasis on code and other "low level skills" could potentially further the equity gap in computer science (Anderson, 2016). Additionally, while coding classes exist, and the *Hour of Code* is becoming increasingly popular, these activities are often integrated into programs like *Genius Hour* instead of as vehicles to engage students in core instruction (Davis et al., 2014).

Primarily beginning with Jeannette Wing's 2006 editorial ("Computational Thinking") in the Association of Computing Machinery's monthly magazine, a change has been brewing encouraging the teaching of computational thinking in addition to simply teaching programming. Wing argued that computational thinking, the ability to decompose problems into algorithms, create abstractions, and essentially "think like a computer scientist" or think in a way to easily prepare input for a computer is necessary for both maintaining interest and advancement in computer science, and to advance computer science as a tool to solve real-world problems. The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association have created a framework defining the component skills and mindsets that make up computational thinking computational thinking (ISTE & CSTA, 2011). Further need for teaching computational thinking is identified in the 2017 Horizon Report, a yearly report released by the New Media Consortium and the Consortium for School Networking, identifying the biggest trends and challenges in education. In 2017, Computational Thinking was identified as a "difficult challenge", a problem "that we understand but for which solutions remain elusive" (Freeman, Adams Becker, Cummins, Davis, & Hall Giesinger, 2017).

While the skills of computational thinking are defined, there is no context for teaching them, and very little material produced beyond the list of skills. Three of the skills in the ISTE & CSTA document are: data collection, data analysis, and data representation. These skills are fairly well developed in other areas as "data and information literacy" (Tyner, 1998). Additionally, as the focus on science, technology, engineering, and mathematics (STEM) increases in schools, many schools are turning to design thinking strategies to engage students in real-world STEM scenarios (Goldberg & Nemcsok, 2015). In the development of this document and based on other work being undertaken at the Friday Institute, it seems to me that these three areas (computational thinking, data literacy, and design thinking) have significant overlap and complement each other nicely. This model is also validated by the inclusion of computational thinking in the 2017 Horizon Report, which also identified STEAM learning as a short term trend, and the creation of authentic learning experiences as a "solvable challenge" (Freeman et al., 2017). This is documented in a draft conceptual framework, found in Figure 1. In this diagram, the dashed lines are functional skills that a student with mastery in the areas should possess. The intersection of the three major areas is something that I am referring to as *Digital Age Problem Solving* and defining as "the ability to use data, design thinking, and computational thinking to understand the ill-defined problems encountered in the digital age and to design and develop effective solutions."

## Purpose of Study

With increasing emphasis on teaching code, and STEM education, *Digital Age Problem Solving* provides an ideal vehicle for teaching problem solving and introducing both computer science and design thinking into K-12 classrooms. The purpose of this project is to refine the conceptual framework, and develop a program to train teachers on how to integrate digital age problem solving in to their instruction (with an emphasis on middle-grades education). Teachers were trained on this model via a specifically designed Massive Open Online Course (MOOC) though the MOOCs for Educators (MOOC-Ed) program at the Friday Institute for Educational Innovation at North Carolina State University. Participants in the MOOC-Ed also had the opportunity to demonstrate understanding by completing the course and implementation by earning a micro-credential. Through analytics data on the MOOC platform, evaluation of

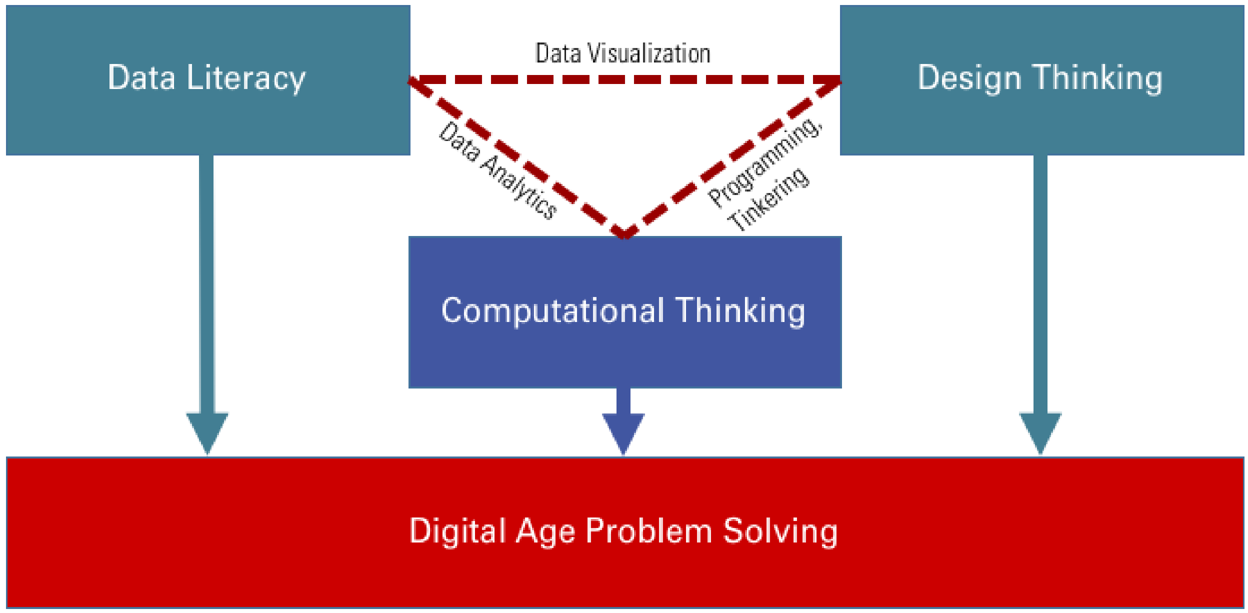



*Figure 1.* Draft conceptual framework describing the relationship between computational thinking, design, and data literacy.

materials submitted to the course, as well as by conducting interviews with course participants, I evaluated the impact on teacher practice in teaching problem-solving skills.

**MOOCs and Micro-Credentials**

In my role as the Technology Innovations Lead at the Friday Institute for Educational Innovation at North Carolina State University, I work as the developer of the platform and an instructional designer for the MOOCs for Educators (MOOC-Ed) project (www.mooc-ed.org). This program offers courses for educators focusing on digital learning, student learning differences, and instructional content. The program has run a dozen courses with over 25,000 participants. MOOC-Ed courses are delivered online, average 500-1,000 users, and are usually six to eight weeks in length. Units have a defined start date but are available asynchronously. MOOC-Ed courses are designed around four central design principles (Kleiman & Wolf, 2015): peer-supported learning; job-embedded activities; multiple perspectives in course content; and self-directed pathways.

In addition to the MOOC-Ed project, this platform also hosts a facility for rewarding competency-based learning through the issuance of micro-credentials. Micro-credentials are competency-based, personalized, on-demand, sharable indicators of skills in educator practice (Digital PromiseCenter for Teaching Quality, 2016). Instead of issuing traditional professional learning credit based on clock hours, educators who earn micro-credentials demonstrate mastery of a topic, regardless of where and how they learn it, and earn the micro-credential based on proficiency rather than attendance. Micro-credentials can also be aligned into a collection or progression of skills, known as a *stack*. The rubric and feedback cycles built in to earning micro-credentials allow teachers to reflect and get feedback which promotes retention (Gulumhussein, 2013) as well as provide teachers with multiple and varied opportunities to dig more deeply into content, collaborate with others, and connect back to practice which are characteristics of effective professional learning (Darling-Hammond & Wei, 2009).

**Problem of Practice Project**

In order to conduct this study, I created MOOC-Ed course, entitled *Computational Thinking and Design: Getting Started with Digital-Age Problem Solving*. A course description can be found in Appendix B. A micro-credential stack which aligns with course content is also available. The study used an exploratory-sequential mixed methods approach, where course analytics will be analyzed quantitatively, and analysis of participant responses within the course and interviews will be used to generate qualitative data to further explain the quantitative data. During the course run, analytics data and discussion forums were used to make real-time tweaks to the course design as needed. Usage logs and analytics data from the MOOC-Ed and micro-credentials were used to determine how participants engaged with the material and made changes to their practice. Participant interviews and survey data were used to further explore the analytics data, while a review of activities and end-of-unit surveys were used to drive changes for future runs of the course. A logic model describing the inputs and activities can be found in Table 1.

**Improvement Goal**

It's impossible to speculate as to how many people will complete a MOOC or a micro-credential (though previous MOOC-Ed courses average between five and eleven percent of enrollees). However, as a result of completing this project, the goal is that (a) at least 75% of MOOC-Ed participants who complete the end-of-course survey report that they have made changes in their instructional practice as a result of this course; (b) 90% of survey respondents will report that they found value in the course; (c) 100% of participants who complete at least one micro-credential will report that they found value in applying the course content in a real-

Table 1

*Logic Model*

| Resources/Inputs | Activities | Outputs | Outcomes/Impact |
|---|---|---|---|
| Friday Institute MOOC-Ed Program | Identify fundamental core competencies of digital age problem solving and computational thinking that are transferrable across the curriculum | At least 300 enrollees in the MOOC course, resulting in at least 30 completers and 5 MC attempts | Course participants implement coding core competencies within their normal instruction |
| Time | Create a MOOC course to instruct grades 3-12 teachers across all subject areas in these core competencies | Participant learner analytics data | Students have foundational knowledge for coding when starting coding courses |
| Marketing for course enrollment | Create a stack of Micro-credentials for teach core competency | Participant course participation data | |
| Assessors for micro-credentials | Advertise and enroll course | Participant micro-credential submissions | |
| Development assistance | Run course | Interview data with course participants | |
| Expert review of course content | Assess and award Micro-credentials | | |
| | Conduct interviews with MC submitters | | |

world scenario. It is my hope that these instructional practices will result in increased student efficacy with problem solving, and eventually, this will translate into higher student achievement measures.

## Questions and Tasks

The questions in this study revolve around the implementation of digital age problem solving skills across the curriculum. Specifically, is the conceptual framework valid, how can teachers be prepared to implement design thinking, computational thinking, and data literacy in their courses? Additionally, while computational thinking is largely portrayed as a coding activity, can teachers integrate the language and processes behind computational thinking into traditional classroom instruction without having a coding background, or integrating coding into their courses? A full list of research questions and data sources can be found in Table 2 and the study timeline can be found in Table 3.

As a result, the major tasks for this study included the creation of a MOOC for teachers of grades 3-12 containing instructional materials for teachers to integrate these competencies. Each unit provided activities for participants to engage in, along with discussions and opportunities for teachers to collaborate and create. Micro-credentials at the end of each unit provided educators the opportunity to submit artifacts and reflections demonstrating attainment of the competency in practice. Review of these artifacts is scored against a rubric for issuance of a micro-credential in that competency area. In the study phase of this project, I surveyed all active MOOC participants to determine how helpful the materials were in making changes to their practice, and what impact that may have on student learning. I also interviewed a sampling of course completers in order to determine if and how they continue to implement competencies into their instructional practice.

Table 2

*Research Questions and Data Sources*

| Research Question | Data Collection |
| --- | --- |
| How are educators able to integrate digital-age problem solving into their instructional practices? | Micro-credential submissions<br>Interview questions |
| To what extent is the conceptual framework a useful tool for teachers? | Interview questions<br>Discussion forum posts |
| How useful is the MOOC-Ed in strengthening participants' understanding of computational thinking? | End-of-course survey data<br>Discussion forum posts<br>Micro-credential |
| What elements of the MOOC were the most helpful for teachers? | End-of-course survey data<br>MOOC click-log and analytics data<br>Interview questions |

Table 3

*Study Timeline*

| Date | Event |
|---|---|
| August 1, 2016 – March 1, 2017 | Course Content Development, Review |
| November, 2016 | Preparation of pre- and post- course survey materials |
| October, 2016 | Submission of research plan to Institutional Review Board at East Carolina University |
| Mid-October, 2016 | Expert review and feedback on course content |
| November, 2016 | Expert review of micro-credential content |
| December, 2016 | Course marketing begins |
| December 15, 2016 | Expert review of course outline and preliminary course content |
| January, 2017 | Course enrollment and marketing begins<br>Submission of IRB paperwork to ECU |
| March 2017 – May 2017 | First run of MOOC-Ed and collection of micro-credential submission |
| May 2017 – October 2017 | Analysis of survey and course analytics data<br>Revision of course content<br>Analysis of Discussion Posts and Interviews and qualitative analysis |
| Early September, 2017 – Early November, 2017 | Second round of MOOC-Ed in progress |
| September, 2017 – January, 2018 | Completion of final report |

## CHAPTER 2: REVIEW OF RELATED LITERATURE

### Content Considerations of a Computational Design Thinking Course

Preparing content for the online course will be an essential component of the development of this project. In order for the content to be useful to course participants, efforts should be made to ensure that the course will be aligned to existing content standards and instructional frameworks for teaching computer science and design thinking.

### Computational Thinking

Computational thinking is defined as "taking an approach to solving problems, designing systems and understanding human behavior that draws on concepts fundamental to computing" (Wing, 2006). References to using computers to teach students processes of thinking date back as early as the early works of Seymour Papert (1972) though the term "computational thinking" largely came to the foreground after the Wing 2006 article. Wing (2006) argues that computational thinking is a fundamental set of skills that are applicable to solving problems across any discipline, and not a rote process. She emphasizes that computational thinking is distinct and different from programming, as programming is simply the language to make a computer execute instructions. In a follow-up paper Wing (2008) argues that the growth of technology and science in popular culture has driven an increased imperative to teach computational thinking to all students, and also proposes that while computational thinking is traditionally taught to college freshmen, young children have the capacity needed to understand the underlying concepts. This does differ from the way Papert (1993) describes computational thinking, where Papert argues more that computational thinking was the act of structuring input for the computer. However, Wing's definition seems to be the one that has stuck, and it has become much more prevalent in the current discussion.

The National Academy of Sciences in 2010 convened a working group to attempt to define a scope for computational thinking (National Research Council, 2010). There is debate throughout the report about the role and importance of computer programming in teaching computational thinking skills. But there is some agreement around the idea that teaching computational thinking is important to help students move into being able to create and define abstractions, as well as more generally, a useful set of cognitive skills. There are also concrete examples of research presented where students have been able to engage in computational thinking skills by drawing pictures or by verbalizing processes or sequences. Interestingly, they also include anti-examples of computational thinking, namely, the use and operation of computers, but there is also a significant argument that computational thinking is a new frame of thinking that has been enabled by access to technology tools.

While Wing notes systemic barriers to adopting computational thinking instruction at lower grades, there are perception issues as well. Yadav, Mayfield, Zhou, Hambrusch, and Korb (2014) conducted a study measuring teacher perceptions on what computational thinking is and how it can be used in their classrooms. They found that there were many misconceptions about the definition of computational thinking, and that teachers who were given training in computational thinking were able to see applications in problem solving across subject areas. They point to a need for computational thinking to be integrated across subject areas. Barr and Stephenson (2011) assembled a group of educators and computer scientists into a working group to attempt to operationalize the role of computer science in K-12. In the discussion framing, they noted "certainly, K-12 students already learn how to think and to problem solve, but computer scientists can help teachers understand these processes as algorithmic, and identify where actual computation and manipulation of data with a computer may fit in" (Barr & Stephenson, 2011, p.

49). They created a table referencing the components of computational thinking, which eventually was adopted by ISTE & CSTA (2011) across subject areas, but also note that "the computer scientists participating, in particular, noted that educational change was considerably more complex than they suspected and that working with educators from multiple diverse disciplines meant learning to 'disconnect computational thinking from computer science'" (Barr & Stephenson, 2011, p. 51).

Much of the current work on computational thinking is focused on the AP Computer Science Principles course. AP Computer Science Principles has a computational thinking "core" component, and CT principles are weaved throughout the AP Computer Science Principles framework (College Board, 2014a). Google, ISTE, and Code.org have also created portals for mainstreaming computational thinking instruction. However, despite the issue noted by Barr and Stephenson of separating computational thinking and computer programming, many computational thinking curriculum modules produced by Google, Code.org, and others, still include a very heavy emphasis on using code and computer programming to teach computational thinking skills. While the Problem Solving in the Digital Age MOOC-Ed will be designed with a focus on implementation without coding, most of the existing studies on computational thinking focus on integrating computational thinking skills using code. A study by Kim, Kim, and Kim (2013) noted a marginal increase in logical/computational thinking proficiency in non-CS majors who engaged in paper and pencil approaches to learning computer science versus using LOGO. They also noted a significant increase in student comfort with the course content and desire to engage in further CS education when using a paper and pencil approach.

Some states such as Massachusetts (Massachusetts Department of Elementary and Secondary Education, 2016) have designed digital literacy curricula that include computational thinking as a component of digital literacy. Though the MOOC, I've learned other states are getting started in this process as well. British Columbia, Canada has a new curriculum that includes skills from computational thinking, design thinking, and vocational courses integrated throughout K-12 (British Columbia, Province of, n.d.).

**Coding Courses**

Many computer science curricula, including the ACM's recommended curriculum (Jones et al., 2003; Seehorn et al., 2011)include both computational thinking and coding as a part of the curriculum. Aside from the Advanced Placement courses, this has largely been a patchwork implementation, with state and local school boards making the decision about what is taught and how (Computer Science Teachers Association, 2013). Teachers may be trained to deliver CS content through providers such as Code.org, Project Lead the Way, and University of California at Berkeley's Beauty and Joy of Computing (Garcia, Harvey, & Barnes, 2015). While these skills are considered "computer science", the rise of maker culture has also given way to knowing how to code, without necessarily receiving instruction in computational thinking or deep computer science knowledge (Prottsman, 2015). Code.org also encourages schools to engage in *Hour of Code* events during Computer Science Education Week (late fall) in order to expose students to programming in an effort to spark interest in coding (Wilk & Garcia, 2014). If coding is truly the language of computer science, it appears that several states are poised to treat it as such, making coding count as a foreign language credit (Hatter, 2016).

Many of the changes in the current coding curriculum are stemming from changes in the Advanced Placement (AP) computer science program. In 2008, the College Board announced that the *AP Computer Science AB* course would be replaced with new courses, citing low and declining participation, difficulty getting teachers, low pass rates, and significant racial and gender disparities (Goode, 2008). Goode also noted that many changes in the computer science curriculum happened at a much faster pace than other courses, and the subject material was highly technical. As a result, the College Board redoubled their efforts to reform *AP Computer Science A*, and develop the new *AP Computer Science Principles* course. The AP Computer Science A course is still highly technical, with an emphasis on programming, heavier computer science content, and software engineering (College Board, 2014b). The AP Computer Science Principles course, by contrast, focuses largely on computational thinking, design, human-computer interactions, and some of the "higher level" skills in computer science (College Board, 2014a). AP CS Principles is also notable because it doesn't require a computer science background to teach, opening it up to more schools. The CS Principles exam will be launching in the spring of 2017, so there is no data on the effectiveness and long-term impact of the course as of yet. There were five pilot courses at universities around the country, with generally positive results – diversity in these courses were notably different than standard CS courses, and while differences in outcomes between represented and under-represented students was statistically significant, it wasn't significant in practice (Snyder, Barnes, Garcia, Paul, & Simon, 2012). It's noteworthy that neither the CSTA report on computer science curriculum (Stephenson, Gal-Ezer, Haberman, & Verno, 2005) nor the Israeli computer science model which is cited several times (Hazzan, Gal-Ezer, & Blum, 2008) mention computational thinking or a CS Principles type approach and therefore seem outdated by the current trends in the field.

From these courses, an ecosystem of courses and content grew. Buffum et al. (2014) studied the integration of the AP Computer Science principles into middle school instruction around big data. The researchers aligned the core principles to Common Core Practices for Mathematics, and also selected Common Core standards that deal with data and CS principles.

They tested an approach that taught "Big Data" concepts to middle schoolers with generally positive results. Code.org is also working on revamping their middle school and elementary offerings as precursors to the AP CS Principles course (Code.org, 2016).

With the realignment of the AP Computer Science curriculum, there now seem to be two discrete tracks – the AP Computer Science A track, which focuses on "hard" computer science and programming versus the new AP Computer Science Principles track, which doesn't have the depth of coding as the AP Computer Science A, yet includes much more explicit computational thinking, human-computer interactions, and societal impacts of computing. There's an open question about what the right balance is. The White House's Computer Science for All initiative declares coding to be a basic skill ("FACT SHEET: President Obama Announces Computer Science For All Initiative," 2016). Vee (2013) argues that computer coding is a basic literacy, and much as how written information in English is the key to civic engagement in modern life, coding will be the tool for engagement in the future. Shein (2014) largely concurs, but also notes that "thinking like a programmer" (computational thinking) may be a more pertinent skill for students in the digital age. There are still detractors, like Barba (2016) who argue that Wing's definition of computational thinking is too watered-down, and includes thinking skills, but none of them are unique to computer science. Debugging and certain types of testing, for example, are skills which are unique to computer science, but are not part of the standard computational thinking models.

Additionally, in late 2016, a K-12 curricular pathway was released by a collaborative group of several organizations including CSTA and ISTE. The framework identified computational thinking in the context of solving real-world problems as one of the core standards across all CS courses (K-12 Computer Science Framework, 2016).

**Pedagogical Approaches to Teaching Code**

Computer science, by nature, would seem to lend itself to teaching in a project-based, artifact-rich environment. Brennan and Resnick (2012) completed a study identifying the relative value of different types of assessment approaches in computer science courses, resulting in suggestions for evaluating computational thinking within programming artifacts in Scratch. Hazzan et al. (2008) recommend a framework for designing courses, including recommendations for teacher licensure. Zendler and Klaudt (2015) examined the effect of different knowledge processes across different instructional strategies, and examined where direct instruction versus more hands-on approaches could be more beneficial. Fee and Holland-Minkleya (2010) provide a model for a more project-based approach in college computer science courses.

A National Research Council committee attempted to define pedagogical approaches for computational thinking (Report of a Workshop of Pedagogical Aspects of Computational Thinking, 2011). The report defines a need for a process for computational thinking (rather than a list of skills), but fails to define what it could or should be. The report also indicated the needs for teacher professional learning, alignment to content standards, and the need for a connection to jobs and careers for students.

While AP Computer Science A uses Java programming, AP Computer Science Principles (CS Principles) and programs from Code.org use block-based languages. Block-based languages use drag-and-drop blocks to represent computer code structures. Some applications, like Code.org's platform and Google's *Blockly* allow users to switch between blocks and JavaScript code. Because block-based languages are drag-and-drop and prompt the user for any required inputs, and because many languages such as Scratch and Google's *Blockly* language also scaffold the user using colors, shapes, and matching connectors, they are often used to teach the basics of computer programming without needing to get in to the specific and unique syntax of a programming language (Price & Barnes, 2015; Weintrop & Wilensky, 2015). Weintrop and Wilensky (2015) studied student perceptions of block-based programming, specifically focusing on ease of use and the difference between block programming and traditional programming languages. They studied several classes in a block language (*Snap!*) for five weeks, and for five weeks in Java. More than half of the students surveyed reported that learning programming in Snap! was easier than learning in Java. They specifically pointed to the ease of composition, block shapes and colors, and readability as advantages of Snap! and other block-based languages. They pointed towards a lack of authenticity, a lack of features and robustness, and the fact that coding in block languages requires more steps than Java as potential pitfalls but noted that these drawbacks don't detract from block-based languages being an effective entry point for students learning to code. Price and Barnes noted that students using a block interface completed activities in less time and had more time on task than traditional text-based interfaces. Students also reported higher confidence in being able to program when using block-based languages.

There is also much written about both game-based learning and physical computing in teaching computer science. Since LOGO in the 1970s (Papert, 1972; Solomon, 1978), students have been using games and challenges to learn how to manipulate computers. Games have a significant impact on student motivation – even simple games that don't have much graphical or technical sophistication (Papastergiou, 2009; Prensky, 2006). Physical computing, such as LEGO Mindstorms (Papert, 1993) are extremely popular with students who can control the LEGO robotics kits by programming them to complete certain tasks (Chung et al., 2014). Much of this has been folded in to what has become known as the "maker movement," with students being able to "tinker" and use connected devices to explore coding and computational thinking in greater depth (Halverson & Sheridan, 2014; Mohomed & Dutta, 2015; Olivares-Giles, 2015). Devices such as Arduino, Littlebits, and Sphero allow students to be able to explore both electronics, and computer science, while solving problems without extensive technical knowledge and also enables students who wouldn't be interested in an on-screen activity (or who have an interest in the hardware) to be able to engage in new ways (Bers, Flannery, Kazakoff, & Sullivan, 2014; Olivares-Giles, 2015; Przybylla & Romeike, 2014; Rubio, Romero-Zaliz, Manoso, & De Madrid, 2015). With many schools not having formal computer science education programs, many coding activities can be undertaken with physical devices during so-called "*Genius Hour*" where students get a chance to explore topics of interest to them (Davis et al., 2014).

**Design Thinking**

As a teacher in STEM schools, I completed many professional development sessions on design thinking, as design was a cornerstone of the STEM mindset for our school. Stanford's Design School defines design thinking as a "process first defines the problem and then implements the solutions, always with the needs of the user demographic at the core of concept development. This process focuses on needfinding, understanding, creating, thinking, and doing. At the core of this process is a bias towards action and creation: by creating and testing something, you can continue to learn and improve upon your initial ideas" (Stanford Design School, 2012). The process is broken down into a cycle consisting of five components (Figure 2): "empathize, define, ideate, prototype, test" (Ingle, 2013). Design thinking, at the core, focuses on bridging the gap between the design and the end user. Kolko (2015) discusses the need for design thinking as an emergent property of systems and products that have grown

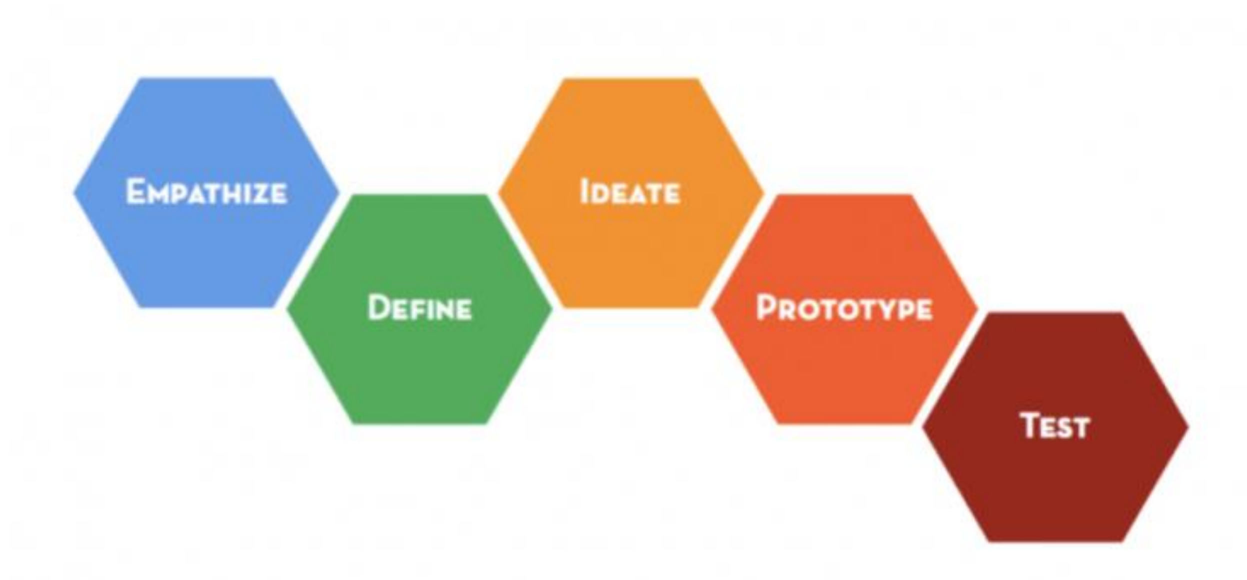



*Figure 2.* Components of Design Thinking (Ingle, 2013).

increasingly complex. Systems now integrate analog and digital processes, inputs from multiple users or a combination of user input plus additional data, and a series of complex dependencies. In building the system, design thinking keeps the end-user in mind in order to bring order into the system.

Engaging in a design-thinking process is very much about designing for user experience, the emotional responses a system will produce, and being very strategic about what a system will do and NOT do (and how). Both Buchanan (1992) and Rittel and Webber (1973) describe design thinking as a tool for solving “wicked problems”. Wicked problems are the problems that the authors argue are the ones that designers and engineers are most likely to encounter – problems that are themselves unique and ill-defined, with no correct or concrete solutions, no defined stopping point, no “right” answers, and no pleasing everyone. Further, these problems don’t fit neatly into boxes (sciences or arts or math), they are “fuzzy”, and the disciplines needed to address components may not even be clear at the outset. This, in many ways is antithetical to the “traditional” schooling approach – where subjects are isolated and problems are typically pre-defined to have definable steps, a known start, and a known solution.

Studies such as Gattie and Wicklein (2007) indicate the value of teaching engineering design for student engagement in math and science and Carroll et al. (2010) noted how design processes sparked student imagination and creativity. They take it a step further by identifying a set of critical mindsets for design thinking, and link these mindsets to student engagement, excitement, and resiliency. Resiliency, and learning through failure and integration is another benefit of implanting a design thinking environment in classrooms, as noted by Goldberg and Nemcsok (2015). Scheer, Noweski, and Meinel (2012) argued that design thinking was the operationalization of constructivism. They conducted a comparison case study with two groups:

one completing a project using a design thinking approach, the other with a more traditional approach. The students and teachers were surveyed about their perceptions about the experience (focusing mostly on student engagement). The results were significantly more positive in favor of a design thinking approach, though it's worth noting that there was no measure of actual learning employed in this study. Both Carroll et al. (2010) Doppelt, Mehalik, Schunn, Silk, and Krysinski (2008), among others, argue that engagement and creativity in a design-thinking context can measure learning, but that this learning may not be measured in the current system of standardized assessments. In fact, Doppelt's team found that lower-achieving students produced better evidence of scientific thinking and concept mastery than the high-achieving students studied, even though standardized testing measures didn't confirm these findings. Among all of the studies reviewed, there was a common theme about design thinking heightening skills in student creativity and critical thinking.

Much of design thinking is a human-centered outgrowth of similar problem-solving processes. The scientific method is a formula for iterative design, where an assumption is tested to prove or disprove a particular hypothesis. The Deming cycle (plan/do/study/act) is also a commonly-used tool that is reflected in design thinking.

However, while computational thinking practices could be aligned to a design thinking context, and I found many references to design thinking in computational thinking research, I was not able to locate a comprehensive "computational design thinking" framework. Polya (1957) created a problem-solving process commonly used in mathematics, which in many ways, could be a predecessor to the implementation of computational thinking habits of mind.

### Data Literacy

Both the K-12 CS framework and the Next Generation Science Standards include and stress the importance of data literacy as a key skill in the digital age (Committee on a Conceptual Framework for New K-12 Science Education Standards, 2012; Next Generation Science Standards, 2013; Sneider, Stephenson, Schafer, & Flick, 2014). Beyond doing statistics, the United Nations defines data literacy as the intersection between technical skills – the ability to crunch and display data, statistical skills – the knowledge of statistical techniques to analyze the data, and information literacy – the ability to collect data and understand what data is telling us (Independent Expert Advisory Group on a Data Revolution for Sustainable Development, 2014). More frequently, this is being referred to as "data science". Much of this work is being done in the journalism field, which has produced a field guide for data scientists which identifies the need for data in investigations and includes a field guide to statistical techniques and technical resources (Gray, Bounegru, & Chambers, 2012a). Google News Lab also provides resources for data journalism and data storytelling, key components of both computational thinking and design ("Google News Lab," n.d.).

## Social Context

### Rise of STEM Education

As an educator, it's impossible for me to go to any conference or workshop without hearing about the term "STEM" (Science, Technology, Engineering, Mathematics) in education. STEM education was originally known as SMET (or SMET-E) after federal agencies, most notably, the NSF, combined funding for Science, Mathematics, Engineering, and Technology education grant projects (National Science Foundation, 1996). A staffer at the NSF objected to the “SMET-E” acronym in 2001, as it was commonly pronounced “smutty”. She recommended the letters be reorganized as “STEM” instead (Petroski, 2014).

While the acronym itself is relatively new, STEM education is not. The birth of the modern STEM education movement is generally accepted to be the launch of the Sputnik satellite in 1957 (Garrett, 2008). The launch of Sputnik led to the creation of the National Defense Education Act (Flemming, 1960). Arthur Fleming, who was at the time was the Secretary of Health, Education, and Welfare wrote about the rationale and purpose of the law. Primarily, the purpose of the law was to invest federal funding in STEM education and foreign languages. Title I of the policy states that “The Congress hereby finds and declares that the security of the Nation requires the fullest development of the mental resources and technical skills of its young men and women” (via Flemming, 1960). For four years, $70 million was provided to strengthen STEM and foreign language education. The act also provided for testing for gifted and talented students, and for research and implementation of early forms of educational technology. Flatteau (2007) noted that Title III of the NDEA funded matching grants to schools for science and laboratory equipment. This infusion of money was actually overmatched by local spending by almost $11 million (with a federal contribution of $560 million). As a result, some states saw enrollments increase in foreign language courses by almost 95%, and in science and math courses by almost 50%. Title IV of the program also funded 45,829 graduate fellows from 1959-1973 for the purposes of increasing college faculty. The results of this program were generally positive for persons earning doctoral degrees, though outcomes were much more favorable for men than for women (Harmon, 1977). Harmon’s report did also report that a significant number of engineers got jobs in the private sector, and many were published in peer-reviewed journals. Flatteau also noted that this program had measurable impacts on the number of teachers in K-12 schools with advanced degrees.

Additionally, many new curricular programs, funded largely by the National Science Foundation (NSF), the Carnegie Corporation of New York, and the Rockefeller Brothers fund emerged during this time, partially through the birth of new organizations to lead curriculum reform efforts such as the Lawrence Hall of Science and the Education Development Center (EDC) (Bybee, 2013). Bybee also noted that by 1976, 60% of school districts were using federally funded science curricula. While many depictions of the Sputnik area tend to paint the time with an almost uniform nationalism, there were certainly detractors. Bybee noted that the federally funded math programs did not fare as well, with many educators supporting a return to traditional curricula as worries surfaced from mathematicians that concepts were too abstract, teachers didn't have the content knowledge to teach the new curricula, and parents worried that the curriculum lacked significant focus on computational skills. This "new math" has since passed into the vernacular as something of a joke describing math that doesn't quite "add up", a description of a failed program, or a fad that doesn't last. Dow (1991) chronicles the rise and fall of the MACOS (Man: A Course of Study). MACOS was a social studies curriculum largely focusing on humanism and human development, and became a flashpoint for right-wing outrage over curriculum reform, as it was believed by many to be too secularist and too anti-religion to be appropriate for American schools. Many people referred to MACOS as "communist indoctrination" and "a threat to democracy." Bybee points to the controversy over MACOS in 1976 as the end of the Sputnik-era curriculum reforms.

Further pushes for STEM education came around with the 1983 report: *A Nation at Risk* (National Commission on Excellence in Education, 1983). This report cited lackluster requirements for graduating students in science and mathematics, as well as a significant teacher shortage in STEM fields. The report goes on to recommend that all students take at least a half-year of computer science as one of the "five new basics", and to understand how computers work and their role in society. The report recommended that Science focuses on inquiry, application, and social implications of science and that Math follow a similar track with a focus on the application of math content to everyday problems. This report, among others, led to STEM grants and presidential awards for math and science in the Education for Economic Security Act of 1984 (Gonzalez & Kuenzi, 2012).

In the 1990s, we saw the aforementioned consolidation of federal funding into math and science grant programs. The 1996 science standards (National Committee on Science Education Standards and Assessment, 1996) were unique in that they focused solely on scientific inquiry, scientific thinking, and citizen science skills and didn't include any recommendations on science content. The National Council of Teachers of Mathematics released their first standards in 1989, followed by teacher guides and assessment guides in the 90s (Hekimoglu & Sloan, 2005). The mathematics standards focused on problem solving and developing conceptual understanding over computation and suffered many of the same criticisms as the ill-fated new math. The 1989 standards increased the use of calculators and had a goal to provide mathematical power and equity for all students. An emphasis on basic computational standards was added in the 2000 revision of the standards, and they saw much wider adoption.

Much as Sputnik and the Cold War influenced educational reforms in the 60s and 80s, the attacks of September 11, 2001 have impacted the time since. The No Child Left Behind Act of 2001 created the Math Science Partnership grant program, administered by the National Science Foundation, with the goal of bringing schools, museums, companies, and universities together to create innovative programs to improve STEM education. This program is among the largest STEM education programs at the NSF (Kuenzi, 2008). The report *Rising Above the Gathering Storm* (National Academy of Sciences, 2005) stated that student proficiency and participation in STEM fields was falling behind other countries. Additionally, many viewed Thomas Friedman's 2005 book *The World is Flat* as a call to action to increase STEM education and the way American education prepares students to be globally-competitive (Sanders, 2009). The America COMPETES Act of 2007 (Kuenzi, 2008) was passed, and was a bill targeted at STEM education in the United States. President Obama's 2009 Educate to Innovate program (The White House, 2016) was launched with the goal of increasing STEM performance in US schools. In addition to bolstering corporate partnerships, and creating 100,000 new STEM teachers, and launching the White House Science Fair, and ties in to the STEM Education Act of 2015.

Among the policy implications, the recent STEM revolution has seen a new set of curriculum reforms. Common Core Math started with the goal of reducing emphasis on rote memorization, in favor of critical thinking (Garland, 2014), which is in many ways similar to the "new math" and 1989 NCTM standards. Common Core math has faced similar controversies as previous math curricula, and has an uncertain future. The Next Generation Science Standards (NGSS) have also emerged in the past few years. While these standards focus more on science principles, engineering design, and critical thinking, they have faced backlash for including content such as evolution and global warming, and for what is perceived to be a lack of depth in certain content areas (Asif, 2013). The NGSS includes computational thinking skills as an essential element, and aligns to both computer programming and computational thinking competencies (Sneider et al., 2014).

**STEM Schools – Implications for Leadership**

Bybee (2013) points out that STEM Education is different than other types of school reform measures. Specifically, Bybee argues that STEM education done well is a whole school transformation and a transformation in the way instruction is done (versus "one-off" programs, or things done at the classroom level). Bybee defines four "versions" of STEM: STEM 1.0 where subjects are all taught separately, STEM 2.0 where any two STEM disciplines are integrated together, STEM 3.0 where three disciplines are integrated together, and STEM 4.0 where all four disciplines are integrated. Bybee points to the fact that STEM conversions in schools require careful planning and coordination to implement, and that consensus must be built for STEM initiatives to succeed.

The Friday Institute at NC State University has developed a set of survey instruments to measure student and teacher attitudes and efficacy towards STEM education (Wiebe et al., 2013). The State of North Carolina has developed and validated rubrics and guides for creating a STEM school.

If Bybee's definition of STEM 4.0 is the integration of STEM subjects together, I would argue that STEM 5.0 is the blending of STEM and the Humanities. This is commonly referred to as STEAM (STEM+Arts), and is growing in popularity. Land (2013) points to the importance of the arts in maintaining student interest in STEM fields, and the importance of basic literacy in any advanced STEM career. Land argues that while STEM skills are critical in filling career requirements in the United States, the arts and creativity skills that STEAM brings are necessary to actually take engineering advances to market.

Many schools are moving towards a STEM focus, and there is an opportunity to include computational thinking and data science in such a transformation.

**Under-Represented Populations in Computer Science**

A 2011 report by the Department of Commerce (Beede et al., 2011) highlights the gender gap in STEM fields. Specifically, as of 2009, women made up only 24% of the STEM workforce in the United States, earned about 50% fewer STEM degrees than men, and experienced about a 14% wage gap. Women were more prevalent in biological and life sciences, but still a significantly smaller percentage of the workforce than men. The tech industry is working on diversity in a more public way, as a majority of the tech industry employees are white men and this diversity gap is becoming a more prominent national focus (Martin, 2015).

While women are a minority in computer science now, the field owes much of its existence to women. Some of the pioneers of computer science were women, including Ada Lovelace and Grace Hopper, the inventor of the first computer language (Sydell, 2014). Sydell's report with NPR notes that many women in the 1930s and 40s got degrees in mathematics, and while many went on to be teachers, others went on to work as mathematicians for the engineers working on early electronics and space programs. In 1984, almost 37% of computer science graduates were women, and that has plummeted to 18% by 2014 (Google, 2014).

Both Sydell and Margolis and Fisher (2002) note that much of the drop in computer science majors in the 1980s could be attributed to the release of the personal computer, and the marketing of personal computers to males. The early computers for the home were treated more as toys and vehicles for games, and were marketed almost exclusively to boys. Margolis and Fisher (2002) conducted interviews with 263 students at Carnegie Mellon University to understand the effects of this growing gender gap. Many computer games are targeted towards young boys. As computer labs proliferated through schools in the 90s, and as boys computer use increased, boys often retreated to the computer labs in their schools as a safe haven from the lunchroom and the other spaces where boys who were labeled “nerdy” would try to avoid (Margolis & Fisher, 2002). The researchers noted that because these boys often knew more about the computers than the school staff, they became troubleshooters, and found a sense of place and belonging at a time where most students struggle to find that. As a personal aside, this resonated pretty heavily with me, because I point to this exact experience as the one that propelled me towards a career in computer science education. However, the computer lab being the territory of young boys did further the perception that computers were for boys, and girls who weren’t as interested in computers or weren’t as skilled didn’t belong in the space.

As computers became more prolific (and as computer science became more “geeky”), this became an issue for girls, as there was an expectation of prerequisite knowledge that girls may not have had if they didn’t take computer science classes in high school, or have the same intensity of interest as the boys. Margolis and Fisher (2002) discuss the predominance of the emergent “geek culture” of the 1990s and 2000s as a discourager for girls entering the field. Many girls who entered as computer science majors felt isolated and “behind” (even though they weren’t), and many changed majors.

The research prompted changes in the computer science program at Carnegie Mellon University, and enrollment in the CS program had increased from 7% female in 1995 to 42% in 2000, and persistence of female students to graduate had matched that of men. Margolis and Fisher found that a student’s prior experience with computers before college wasn’t an accurate predictor of college success, so CMU removed that from admissions requirements. They created a course that looks a lot like the AP Computer Science course to talk about computers and their role in society without delving into programming, with the hope of capturing a more broad-based interest prior to getting in to the “heavy” content. They renewed their attention on pedagogy and attempted to make their CS-program more applicable to real life situations. They created targeted groups for girls and worked to create a more inclusive culture in the school as a whole. They did much more targeted outreach to girls in high school and middle school, trying to get kids involved early. Google's 2014 report noted that "encouragement, exposure, self-perception, and career perception" were four important factors to consider when attempting to get girls more engaged in computer science.

While gender diversity is a significant issue in computer science, the gap in race is much more significant. Google's workforce is 59% white, and only 2% black (Google, 2016). Facebook is 55% white, but only 2% black (Williams, 2015). In both companies, Asians are the second largest ethnic group. This significant gap between black and white has garnered national attention, and is one of the rationales behind the Computer Science for All initiative. There's debate about whether this is a pipeline problem, an issue with companies being unable to recruit or retain black people, or a matter of people simply hiring those that look like them (Martin, 2015). Jane Margolis along with a new group of collaborators completed another study of computer science involvement, this time targeting differences in race (Margolis, Estrella, Goode, Holme, & Nao, 2008). The authors point out that while technology was once billed as the "great equalizer", it has in fact furthered gaps for students. The authors look at three schools in Los Angeles. They noted that while one school was awash in technology, there were no programming courses in the school. There was at one point, but they had trouble getting students to enroll, and the problems presented were not interesting to the students. They cite barriers such as scheduling and overcrowding, an over-emphasis on testing and accountability, and a lack of access to computer science curricula as other parries. They cited issues of lack of teacher training, lack of equipment in other classes that used computers (video production, for example), and a lack of district investment in making computer science a priority. The authors cite a lack of role models and possibly an over-emphasis on sports for black kids as a potential barrier to increasing interest in computer science. Kids of color and poor kids are much less likely to have computers in the home than a white student or a wealthier counterpart. Additionally, many portrayals of computer users in marketing and in the media are white males. As a result, many of the issues regarding access and marketing and perception of computers as a "white male" domain manifests for minority students the same as for girls. Additionally, because of the aforementioned idea of "claimed spaces" in computer labs, black kids often felt excluded the same as girls did. Also, since many schools traditionally only offered computer science at the AP level, with heavy math prerequisites, many students who were lower performing or who are less likely to take AP classes in the first place were excluded from being able to participate in whatever computer science program a school may offer. The authors point to the fact that computer science is an elective, and that many schools either don't care what electives a student takes, don't know student interests well enough to help them make informed decisions, or make assumptions on behalf of the student, rather than helping them to cultivate their interests.

As a result of their investigation, the authors created an AP Computer Science teachers institute for teachers in the Los Angeles school district. They noted one school where the summer institute inspired a teacher to create a computer science course. They did find the principal, along with other school and district leadership, was the lynchpin that could make or break a CS initiative. They were instrumental in getting the program set up, driving students into the program (the authors noted that after two years, enrollment in the course had doubled, latinx enrollment had quadrupled and female enrollment had tripled. However, they were instrumental in killing the program, when student performance gaps in mathematics required schools to shift teachers from teaching computer science to teaching math. The gains in enrollment were largely attributed to a support program that was set up at UCLA, where students would be bussed to the campus on Saturdays and supported in the AP Computer Science curriculum. They found that being on a college campus, and that the attention from actual students in the field, was instrumental in student success.

Google (2015) conducted another study indicating that students who are poor or who come from less-educated households are much less likely to have computers in the home, and less likely to have an adult in the home that works with computers. They noted that students who are black or Hispanic are less likely to attend a school that offers a computer science course. Also, parents with high incomes were much more likely to believe that students should be required to learn computer science. However, the study finds that it's simply not a priority for school districts among the other priorities they have.

This is one of the values of computational thinking, in that it can begin to introduce conversations about computer science, yet requires very little overhead from districts and teachers. Other programs like the aforementioned FIRST Robotics and programs targeting minorities with mentorship and exposure opportunities are successfully getting students involved in these fields.

## Teacher Professional Learning

### Effective Practices for Teacher Professional Learning

Primarily, the MOOC-Ed being developed is designed to (1) expose teachers to a framework and process for digital age problem solving and (2) encourage them to use these processes with their students. The decision to encourage teachers to go through the process as a learner is influenced by Clarke & Hollingsworth's Interconnected Model of Professional Growth

(Clarke & Hollingsworth, 2002). Specifically, before a teacher can experience outcomes with students, they will need to be exposed to the information and learn more ("personal domain") and have a chance to experiment and try something new "domain of practice". Additionally, by scaffolding one new skill set per week, along with opportunities to try, fail, and discuss with peers, we are creating support spaces and spaces for modeling and support (Gulumhussein, 2013). Gulumhussein identifies five key principles for effective professional learning: enough time for participants to demonstrate mastery, support during implementation (which will be accomplished through use of micro-credentials), active exposure to new content, modeling, and content-specific.

**Rise of MOOCs**

At the turn of the century, the Massachusetts Institute of Technology (MIT) launched the OpenCourseWare project. The eventual goal of OpenCourseWare was to make all of the content of the undergraduate and graduate courses at MIT available to the world, for free (Abelson, 2008). This was the first time that such an endeavor had been undertaken, and it took off, with multiple universities getting involved in 2005 and forming the OpenCourseware Consortium. This eventually led to the idea of running courses, open to all on the Internet, with the first significant reference to such an idea being credited to George Siemens and Stephen Downes (McAuley, Stewart, Siemens, & Cormier, 2010). MOOCs as defined here are what later became known as cMOOCs (Daniel, 2012). Based on connectivist principles, cMOOCs are online courses that were made available, where students self-organized and proceeded in learning communities and these self-organized groups proceeded to work together, sometimes independently of the central course site to learn the course material (McAuley et al., 2010). By contrast, a xMOOC is based more on  traditional courses with larger numbers of students (Bates,

2014) and include the types of materials and formats you would traditionally find in an online course, including a centralized Learning Management System. Coursera, Udacity, and edX are the largest commercial and non-profit MOOC providers. Their courses are examples of xMOOCs (Daniel, 2012).

While MOOCs have been heralded as the “future of education” in many circles, the evidence is less than conclusive. Completion rates for MOOCs has traditionally been extremely low, hovering below 10%  (Clow, 2013; Daniel, 2012). Having participated in several MOOCs (while completing none), I am not sure completion is a good metric to use. There have been several MOOCs where I have learned things that are useful, but I did not have the time or inclination to complete the entire course. The course was still valuable to me and my learning, even if I was unable to complete it. For others, the stresses of daily life prevented me from returning to the course. I would question if completion is a valid metric in a context where credit is not being awarded, and a user is opting in to participation with no stakes (MOOCs are typically free, so there is no financial incentive to complete). DeBoer et al. (2014) recommends a shift away from completion metrics, to allowing participants to define their criteria for success, and measuring that. They recommend looking at a user’s initial intention to complete and attempting to measure why drop-offs in that subpopulation occur, and note that participant patterns and behaviors in MOOCs (and the fact that interactions are recorded granularly) can provide better insight into their intentions and motivations.

While adaptations of MOOCs into university courses have been less than successful (Kolowich, 2013), MOOCs have been seen as being useful for professional development (Kleiman & Wolf, 2015; Milligan & Littlejohn, 2014; Vivian, Falkner, & Falkner, 2014). Milligan and Littlejohn (2014) note that many teachers didn’t fully exploit the potential value of the MOOCs they studied, but Vivian et al. (2014) did note the value of MOOCs for teachers in rural areas who do not have ready access to professional learning. Friday Institute MOOCs, as studied by Kleiman and Wolf (2015), Avineri (2016), and Kellogg (2014) all noted positive results for teacher participants, specifically with respect to mathematical concept knowledge (Avineri, 2016), and peer-supported learning (Kellogg, 2014).

Discussion forums in online course environments remains a challenge. Many participants tend to engage in lower-level discussions, either “shouting into the void” and sharing their responses without engaging in high-quality dialogue, or participants tend to engage in superficial responses such as “I agree” or “good job” (Gunawardena, Lowe, & Anderson, 1997; Kellogg, Booth, & Oliver, 2014; Stump, DeBoer, Whittinghill, & Breslow, 2013).

**Badging and Micro-Credentials**

With the availability of information on the Internet, and the immediate access to information, it is now more possible than ever before to learn new information and skills outside of formal structures. Participants can learn from sites such as MIT Open Courseware, as well as through MOOCs, even if the user doesn’t earn a certificate. Additionally, in technology contexts, the possession of specific skills can be much more highly valued over a degree or formal certification. The Mozilla Open Badge architecture has aimed to solve this problem by creating digital badges as a way to represent skills. Mozilla has created a technical architecture and a framework for this (Mozilla, 2011). The Open Badge architecture has become the de facto standard for digital badging and has been adopted into several Learning Management Systems, including Moodle. Open Badges can be displayed on a LinkedIn profile. The badge ecosystem, outlined by Mozilla (2011) depends on cooperation among a series of major actors. Mozilla has developed a JSON-based metadata standard for badges, which are in effect, an image and the

JSON-based metadata (Open Badges Alliance, 2016). The ecosystem of badging depends on *issuers* (organizations which will issue badges) and *assessors* (individuals and organizations who will assess whether a person meets the criteria for a badge), *developers* who develop the badge and metadata, as well as the assessments necessary for issuance; *endorsers* such as a school district or employer who will lend additional value and acknowledgement to a badge. Ideally, badges can be assessed and issued independently of the developer, though I have seen very few examples of this in practice. However, theoretically, a badge could be developed by one party, issued by another, and assessed by a third or by peer review. The key value proposition for badges is that they can demonstrate mastery of specific skills, and the method used to learn these skills is irrelevant – the badge assesses mastery, not the learning process.

In an education context, badges are being used by several school districts in North Carolina as a way to acknowledge teacher professional learning – the districts providing badges are asking teachers to use a technology tool and submit evidence that they have integrated the tool into their instruction to earn the badge. Whether they learn how to use the tool by experimentation, using an online tutorial, or from a peer is irrelevant, the quality of the product is what is assessed. Digital Promise is currently rolling out a national model of micro-credentials for teachers (Digital PromiseCenter for Teaching Quality, 2016), though as of right now, there are very few endorsers of these micro-credentials.

# CHAPTER 3: METHODOLOGY

## MOOC Course Design

The MOOC-Ed Course at the center of this study is titled *Computational Thinking and Design: Getting Started with Digital-Age Problem Solving*. The course is introductory, as it is assumed that many teachers may not have deep pre-existing knowledge of this content. The course introduces computational thinking based on the ISTE/CSTA (2011) definition of computational thinking, along with a modified version Stanford Design School definition of design thinking (Hasso Plattner Institute of Design, 2013). Data literacy is also emphasized, though there is significant overlap with both design thinking and computational thinking, with the growing fields of data science and data journalism (Gray, Bounegru, & Chambers, 2012b) providing relevant examples. I have worked to select strategies of computational thinking that align to each phase of the design process, as illustrated in Figure 3. The strategies are not exclusive to that phase of the process, but as the place where it appeared most likely a participant would engage with those skills.

This course is developed and hosted as a part of the MOOC-Ed program at the William and Ida Friday Institute for Educational Innovation at North Carolina State University. The course will be hosted in a platform called The PLACE (Professional Learning and Collaboration Environment). The PLACE is an implementation of the Moodle open source learning management system, which has been customized in the course of my work with the Friday Institute. All modifications are open-sourced and documented on the GitHub platform. The application is hosted using Amazon Web Services (AWS). The database is hosted using Amazon's Relational Database Service (RDS) with access restricted to application components within the local subnet on AWS.


*Figure 3.* Computational Thinking and Design (Digital Age Problem Solving Cycle).

I also have write access to the database, and Friday Institute researchers are able to access the database using an openSSH tunnel secured with a password-protected private key along with a mobile push-notification-based second-factor authentication. The application front-end is hosted on Amazon Elastic Compute Cloud (EC2) machine instances, which are connected to Amazon's Elastic Load Balancer to ensure high availability and low latency access to the application. Course assets are hosted using Amazon's Simple Storage Service, and course videos are stored using private videos on a Google Apps for Education channel on YouTube. Participant-generated assets are stored using Amazon's Elastic File System, along with file-level caching. In-memory caching is accomplished using Amazon's Elasticache service running a Memcached server.

The MOOC-Ed program is arranged around four major design principles (Kleiman & Wolf, 2015): peer-supported learning, job-embedded learning, self-directed learning, and multiple voices on course content. In this course, job-embedded learning is demonstrated by participants engaging with course content using a set of simulations and in application to practice via micro-credentials, and brainstorming how to connect the course content with their students or in their contexts. Peer-supported learning is manifested by users posting and sharing their work in the discussion forums. Because users can choose to complete the course or not, engage with only certain parts of the course or the entire course, and they can choose to attempt micro-credentials or not, users can create self-directed pathways through the course. Additionally, course videos will feature educators, industry professionals, and experts using the skills presented, bringing in multiple voices and perspectives.

The *Computational Thinking and Design* MOOC-Ed is the sixth in a series of courses made possible with funding from the William and Flora Hewlett Foundation. The first three courses were titled *Fraction Foundations, Disciplinary Literacy for Deeper Learning* and *Teaching Statistics Through Data Investigations.* These three courses launched in the fall of 2014. *Teaching Statistics Through Data Investigations* has run seven additional times since the additional launch, and *Disciplinary Literacy* has run one additional time. The fractions course was developed by Theresa Gibson, Dr. Shaun Kellogg, Dr. Sherry Freeman, and Dr. Glenn Kleiman. The statistics course was developed by Dr. Hollylynne Lee and Theresa Gibson and the disciplinary literacy course was developed by Dr. Hiller Spires and Erin Lyjak. The remaining three courses from the Hewlett Foundation funding are launching in the fall of 2016 and the spring of 2017. Aside from this course, the other two are called *Teaching Mathematics with Technology* (developed by Dr. Karen Hollebrands, Theresa Gibson, and Dr. Gemma Mojica) and *Teaching Statistics Through Inferential Reasoning* (developed by Dr. Hollylynne Lee and Dr. Gemma Mojica). The very first MOOC-Ed course was developed by Dr. Glenn Kleiman and Dr. Mary Ann Wolf, and was entitled *Leading the Digital Learning Transition*. This course was designed for school leadership teams embarking on transitions to digitally-enabled learning environments. Other courses that have been developed by the Friday Institute including *Learning Differences* (created by Lauren Acree, Alex Dreier, Dr. Lisa Hervey, Brittany Miller, Mark Samberg, and Dr. Mary Ann Wolf) and *Coaching Digital Learning* (created by Dr. Lisa Hervey , Brittany Miller, and Jaclyn Stevens). Supplemental funding for this MOOC-Ed is provided by NC State University's Game-Changing Research Incentive Program (GRIP), with some funding for future runs of the course funded as a part of the iCS4All grant – a grant awarded to East Carolina University by the National Science Foundation for the purposes of bringing computational thinking into the arts classroom.

Over 25,000 people have registered for MOOC-Ed courses since the first one launched in 2013. All 50 states, and over 90 countries are represented. While completion for most courses is around 7-10%, nearly all of the participants surveyed in all of the courses indicate that they have found value in the course experience.

Since the working assumption was that most course participants have limited exposure to computational thinking or design thinking before joining the course, it would be helpful for them to engage in the process as a learner before transferring to their classrooms. Therefore, the MOOC-Ed course provided participants to work through the design process and engage in computational thinking skills using a set of activities.

Each element of the design cycle is a unit within the course, along with an overview introductory unit. Each unit lasts one or two weeks, however, participants may choose to engage with a unit for longer as needed.

Each unit consists of the following components:

- Introduction: The introduction features an overview and definition of the elements to be introduced within the unit. Introductions are presented as both text and video, and will also feature engineers, architects, teachers, and other practitioners who utilize the elements in their work.
- Dig Deeper: Text page that presents the relevant elements of design thinking, computational thinking, and data literacy while describing how they interact and explains how they could be implemented in real life.
- Activity or simulation: An activity to help participants engage with the elements of the unit. This includes having participants engage with the skills presented by

reviewing a scenario (usually unrelated to education) and discussing a prompt related to unit content.

- Resources: Websites, articles, and further readings to help participants dig deeper in to the instructional content if they are inclined to do so.
- Classroom Application: Participants discuss the applications of unit content to their instructional practice. Peers can provide feedback or critiques.
- Micro-credentials: Credentials will provide teachers with an opportunity to engage with course content in their real contexts

**Micro-Credentials and Completion Certificates**

As a part of the course, participants have the opportunity to earn both micro-credentials and a certificate of completion. In order to earn a certificate of completion, displaying ten contact hours, that can be used in many school districts towards Continuing Education Units of credit (CEU), participants must have completed the discussion forums in each unit and certify that they have spent at least ten hours on the course.

A micro-credential is a competency-based measure of learning. Participants earn a credential that indicates that they have a specific skill. Multiple micro-credentials that build upon each other are referred to as a “stack”. This course guided participants towards earning micro-credentials in a new stack entitled *Digital-Age Problem Solving*. The stack follows the design process, with one credential per phase of the process. Micro-credentials issued by the Friday Institute include an estimated number of hours so that earners can earn continuing education credit in their districts. Because of the nature of micro-credentials, it may take some earners much more time than the listed amount, and others much less time. A list of the micro-credentials can be found in Table 4.

Table 4

*Digital-Age Problem Solving Micro-Credential Stack*

| Micro-Credential Title | Estimated Hours (CEU Equivalency) | Objective, as Provided to Earners |
|---|---|---|
| Understanding the Context | 2.5 | Earners of this micro-credential will be able to identify a problem of practice for them or their students and engage in a process of identifying the people, systems, and structures that impact this problem. |
| Telling Stories With Data | 5 | Earners of this micro-credential will engage in a data collection process and will be able to parse a data-set, identify the key points, and be able to present the data in a way that is easily understood by others. |
| Defining the Problem | 5 | Earners of this micro-credential will be able to take a large, ill-defined problem and break it down into its component parts. |
| Creating Solutions | 5 | Earners of this micro-credential will work towards solving problems identified in earlier micro-credential submissions and engage in the process of creating algorithms to express their solutions. |
| Testing and Evaluating Solutions | 5 | Earners of this micro-credential will be able to identify how their proposed solutions will impact the problems they have identified. |

It is expected that participants will continue to earn micro-credentials after the course has concluded. A participant interested in earning a micro-credential is provided with a list of requirements for the credential as well as a scoring rubric that indicates how their submissions will be scored. Rubrics are scored on a two-point scale: “yes” or “not yet”. The first 50 submissions are scored by myself and another member of the Friday Institute staff using a pre-established process to validate reliability and quality of the scoring rubric. A third member of the staff serves as a reviewer in the event that there are disagreements between the other two reviewers. Participants are provided their rubric results, any feedback, and if earned, a micro-credential compatible with the Mozilla Open Badges metadata standards.

## Participant Population

The participant population was self-selected. MOOC-Ed courses are publicly available, and anyone may sign up. In general, is made clear on the website that the courses are designed for educators of students in grades 3-12, but access was not restricted (and in fact, we had people from education and non-education spaces, working with infants through adults). Because this course was running in parallel with five other MOOC-Ed courses, it was marketed through existing channels including social media, by email announcement, and by direct marketing to interested professional organizations. The Friday Institute Communications Team handles marketing efforts for all MOOC-Ed courses, including social media, listing on blogs, and email marketing. There were no fees for any participants to take the course or earn a certificate of completion or micro-credential.

MOOC-Ed courses traditionally enroll between 500 and 1,000 participants. Courses typically hit this range, though some have been as many as 2,000. This course fell just short of these projections with 498 participants enrolling. Based on past MOOC-Ed course analytics, 50% of people who register will never engage with the course content. 25% of registrants will engage with the first half of the course material, and only 7-10% of course participants will earn a certificate of completion or complete a micro-credential.

Because computational thinking and computer science tend to be related, it is possible that people viewed the content of this course as something that they may not feel comfortable pursuing. The marketing materials for the course made it clear that there is no coding experience required for the course (we also limited the use of the term "computational thinking" to audiences that would understand it). Additionally, I was curious as to how many of the participants in the courses are in STEM schools, so a question about this was added to the registration survey, located in Appendix D.

## Research Agenda

As a part of the larger MOOC-Ed program, and this course, funded by a grant from the William and Flora Hewlett Foundation, there is an existing research agenda that parallels this work and will provide additional data sources for use within this study. When registering for an account on the PLACE platform, users are asked a series of demographic survey questions. When enrolling in a course, the user is asked a series of short survey questions. The process is intentionally two-step to reduce the need to collect duplicate information.

Additionally, all courses contain a unit feedback survey found at the end of each weekly unit along with an end of course survey. All pre-existing surveys are listed in Appendix C. The end-of-course survey includes questions about how participants have made changes in their practice as a result of participating in this course. I have added a question asking users for consent to follow-up with interview questions. Participation in the demographic and registration surveys are mandatory for participation in the MOOC-Ed. Other surveys are optional. Additional questions to the registration survey for this study, which are listed in Appendix D.

Data from existing survey instruments will be combined with new data for this study. A list of research questions and data sources can be found in Table 5.

## Quantitative Analysis

The survey data and course analytics provide the basis for the quantitative analysis of the course. I analyzed survey data to determine the potential impact that this course has on practice by generating descriptive statistics. Course analytics, including page views, forum posts, and engagement with the resources were analyzed to provide insight into how participants engaged with the course (Greller & Drachsler, 2012; Long & Siemens, 2011).

## Qualitative Analysis

Qualitative data from the course, including user forum posts, and micro-credential submissions were loaded into NVIVO and coded from a grounded theory perspective (Corbin & Strauss, 1998; Miles, Huberman, & Saldana, 2013), specifically looking at the development of community, evidence of use of the course content, and changes in practice. Posts were also coded to examine the types of interactions using the Transcript Analysis Framework (Gao, Wang, & Sun, 2009). Additionally, participants who completed the final survey or submitted a micro-credential were randomly selected and invited to participate in an interview. Interview questions can be found in Appendix D and focused on how practice has been impacted by the course content. Interview sessions were conducted in August of 2017 since many teachers were out of school for the summer soon after the course ended.

Table 5

*Research Questions and Analysis Plan*

| Research Question | Data Collection | Outputs | Timeline |
|---|---|---|---|
| How are teachers able to integrate digital-age problem solving into their instructional practices? | Micro-credential submissions<br>Interview questions | Coding of micro-credential submissions and feedback given to participants, along with interview questions. | Micro-credentials will be analyzed on a rolling basis as submitted. Interviews conducted in August 2017 |
| To what extent is the conceptual framework a useful tool for teachers? | Interview questions<br>Discussion forum posts | Coding of focus group interview transcripts and forum posts | Discussion forums will be analyzed at the conclusion of each course. Interviews conducted in August 2017. |
| How useful is the MOOC-Ed in strengthening participants' understanding of computational thinking? | End-of-course survey data<br>Discussion forum posts<br>Micro-credential | Descriptive statistics generated from end-of-course survey questions<br><br>Coding of discussion data | Ongoing after course ends |
| What elements of the MOOC were the most helpful for teachers? | End-of-course survey data<br>MOOC click-log and analytics data<br>Interview questions | Descriptive statistics | Immediately after the course closes |

# CHAPTER 4: DATA COLLECTION

Development of the *Computational Thinking and Design MOOC-Ed* began in November of 2016. I built an outline for the course, refined the conceptual framework, created the week-by-week outline for the MOOC, and identified an early set of resources and goals for each unit, with input from professional development colleagues and subject-matter experts at the Friday Institute. In December, a group of faculty and staff from NC State and ECU, along with representatives from the computer programming industry came to the Friday Institute to review the course design and offer feedback. The course went live in early January, titled *Problem Solving in the Digital Age: Getting Started with Computational Thinking and Design.* Based on input from Friday Institute leadership, the course name was changed to *Computational Thinking and Design: Getting Started with Digital Age Problem Solving* in an effort to capture the momentum around computational thinking in the larger education space. From January until March, while participants were registering for the course, I was working on developing the content. The course started on March 1, 2017 and ended eight weeks later, though participants could still complete requirements through May 30, 2017. Units were launched every one or two weeks (see Appendix E) and were available for the duration of the course.

This chapter will examine the interactions within the course and the resulting impacts on participant practice. The first section will identify how participants were recruited into the course. The second section will examine the process for discarding non-participants to develop an analysis population. The remainder of the chapter will review how these participants interacted with the course and will examine both their perceptions of the course and how their practice may have been impacted. At the conclusion of the chapter, the data will be summarized to address the four research questions:

- RQ1: How are educators able to integrate digital-age problem solving into their instructional practices?
- RQ2: To what extent is the digital-age problem solving conceptual framework a useful tool for teachers?
- RQ3: How useful is the MOOC-Ed in strengthening participants' understanding of computational thinking?
- RQ4: What elements of the MOOC were the most helpful for teachers?

## MOOC Participants

### Recruitment

Recruitment for participants in the MOOC-Ed began in January of 2017. The Friday Institute Communications Team created a promotional video for the MOOC-Ed Course (Friday Institute for Educational Innovation, n.d.) to be featured on the course homepage and various social media platforms (see Figure 4). I created a series of slides which could be used as images in Facebook and Twitter posts (see Figure 5). Using the iContact platform, the Communications Specialist sent promotional emails to all past enrollees of MOOC-Ed courses, as well as to several partner organizations. The Friday Institute also purchased electronic advertisements which ran on Facebook, Twitter, Google, and the National Council of Teachers of Mathematics (NCTM) website. Friday Institute staff members also promoted the course on their personal social media accounts. As a result of all marketing efforts, 400 participants enrolled in the MOOC-Ed. Based on self-reported data collected enrollment (see Table 6), approximately 30% of referrals came from the email campaign, while another 25% came from referrals from peers or supervisors. The survey does not differentiate how the peers or supervisors referred them to the course, so it is possible that these referrals are from the email campaign or from social media.


*Figure 4.* Screenshot from MOOC-Ed promotional video.


*Figure 5.* Social media marketing slide for CTD MOOC-Ed.

Table 6

*User-Reported Referrals to the MOOC-Ed Course*

| Source | Referral Count |
|---|---|
| Friday Institute, e-mail | 121 |
| Colleague/peer | 69 |
| Search Engine (Google, Bing, etc.) | 45 |
| Friday Institute, social media | 32 |
| Administrator/supervisor | 27 |
| Professional organization | 25 |
| Google ad | 16 |
| Other | 10 |
| State Department of Education | 8 |
| NCTM | 7 |
| Conference | 5 |
| Twitter | 5 |
| Code.org | 4 |
| ISTE | 4 |
| NCCTM | 3 |
| Other Email | 3 |
| Class central | 2 |
| Future Ready campaign | 2 |
| ISTE | 2 |
| NBPTS | 2 |

Table 6 (continued)

| Source | Referral Count |
|---|---|
| Digital Promise | 1 |
| Open Culture | 1 |
| reddit | 1 |
| Regional Education Service Agency (RESA) | 1 |

Social media accounted for approximately 9% of referrals, while paid advertisements accounted for less than 5% of referrals. Additionally, of the 400 users enrolled, 198 had enrolled in at least one other MOOC-Ed course since the summer of 2015. Comparison data from Summer 2013-Summer 2015 was not available because of a systems architecture change.

While computational thinking and design thinking are common terms in certain circles, it is not clear to what extent these terms (and their importance) have reached teachers. Therefore, marketing for this course was a challenge – crafting a message to people who were possibly unfamiliar with computational thinking, or did not think it applied in their work. The marketing, therefore, focused on defining and explaining, as well as marketing the "problem solving" angle of the course content.

**Study Population**

Course registration opened on January 14, 2017 and was available through the end of the course on May 30, 2017. During this time, 400 users enrolled. Typically, approximately half of the people who register for a MOOC do not return post-registration (Jordan, 2015), and this holds true for the *Computational Thinking and Design MOOC-Ed*. It is necessary to discard users who did not return to the course after registration so that participant demographics and surveys are reflective of participants who actually engaged in the course. Users who did not return to the course after enrollment, and any user who did not interact with more than five course elements were removed from this analysis. This threshold was determined by manual review of all users with at least one interaction. All users with five or fewer interactions did not create any discussion forum posts, nor did they return to the course after a single visit. It is assumed that these users registered, accessed a few of the course elements, and decided against proceeding.

An *interaction* is defined as a click on any of the pages within the course, playing any of the videos embedded in the course, clicking a link to any outside resource linked within the course, accessing the forums or forum posts, and posting or replying to a forum post.

Engagement in the course varies widely, with some users only logging only a few interactions, and others logging over 1,000. Some users were very active in the course, accessing each page, reviewing the resources, and interacting deeply in the forums. Others were "lurkers" who reviewed the course content and discussion posts, but never interacted with other participants in the forums. Some users logged interactions in each unit in the course, others jumped around or interacted only with certain units. All of these interactions are considered valid measures of engagement, as they can still meet their learning goals according to the MOOC-Ed design principles as defined by Kleiman and Wolf (2015).

Based on the criteria specified above, 195 users were included in this analysis. When users created their account, they were required to provide answers to several demographic questions, found in Figure 6. 64.6% of participants identified as female, and 34.9% users identified as male. One user declined to identify. A majority of users (116) reported having earned a Master's degree, though all levels of educational attainment were reported from high school degree through college diploma.

When creating accounts, users were also asked to provide their location (city, state, country), which can be found in Figure 7 (United States) and Figure 8 (worldwide). In total, participants in the course hailed from 33 states in the United States, and 33 countries from across the world. The United States was the most frequently-represented, which is expected since marketing targeted US-centric organizations and because the course is only available in English.

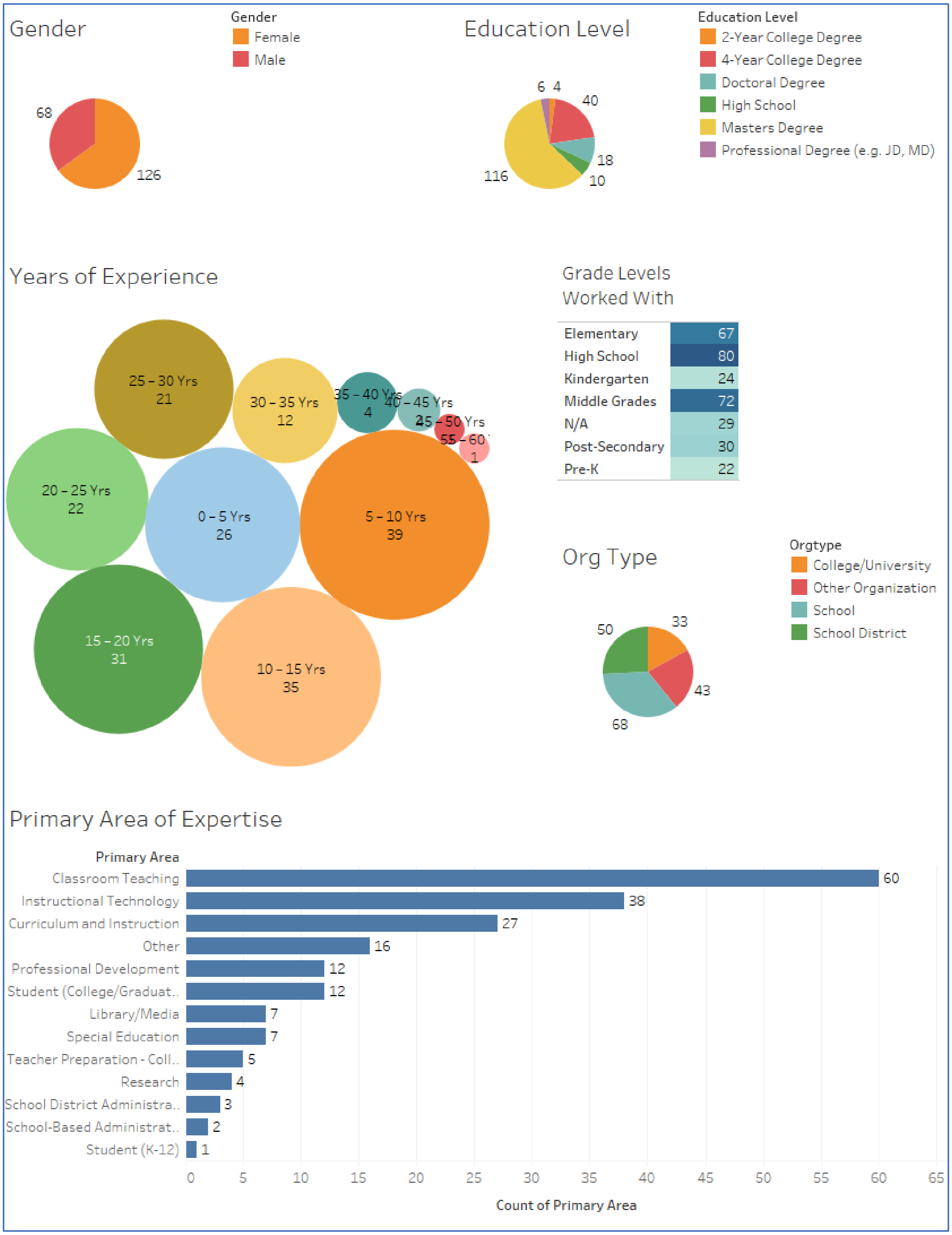



*Figure 6.* Course participant demographic information.

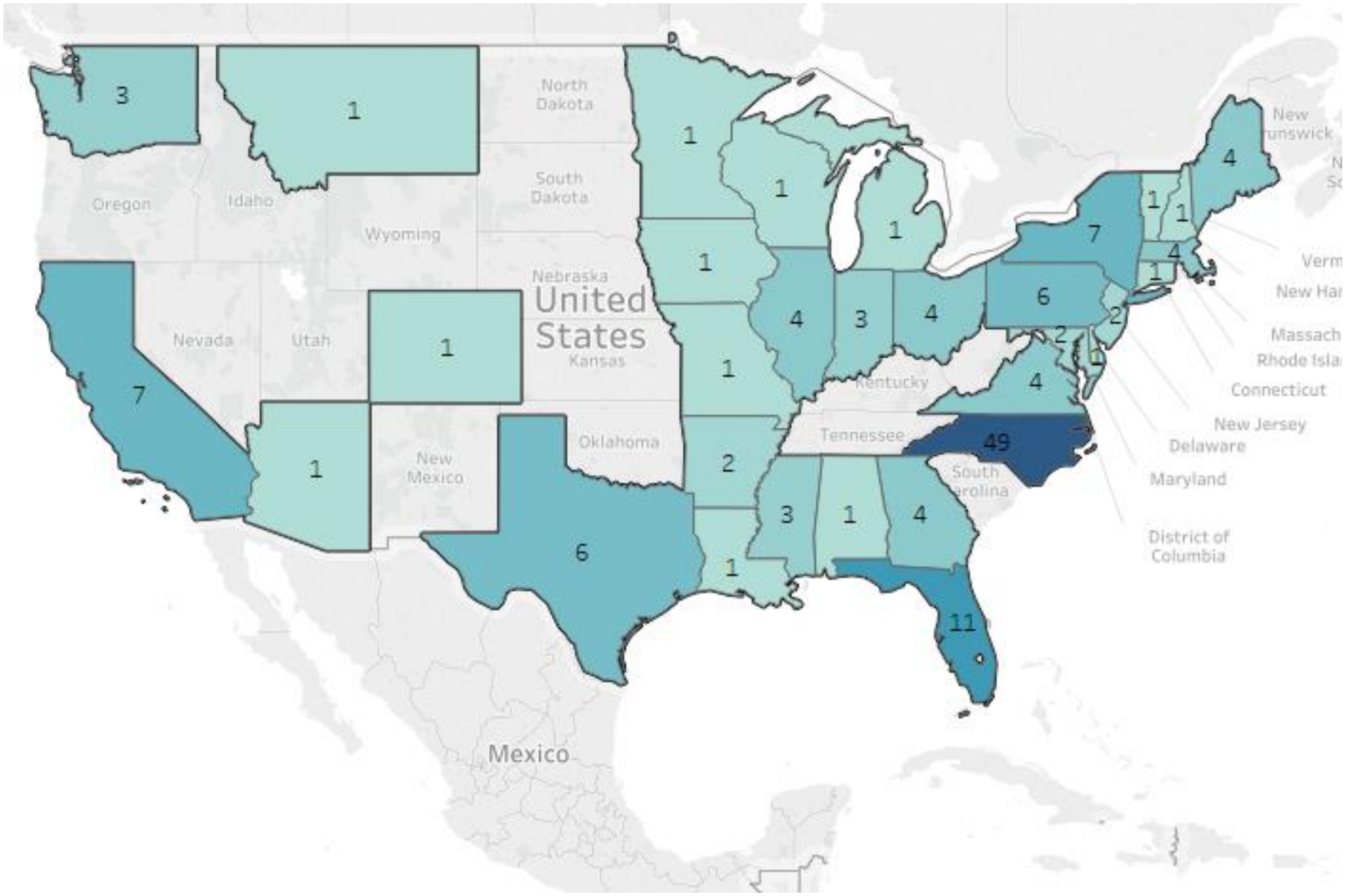



*Figure 7.* Number of participants by U.S. States (not pictured, Hawaii, 1 participant).

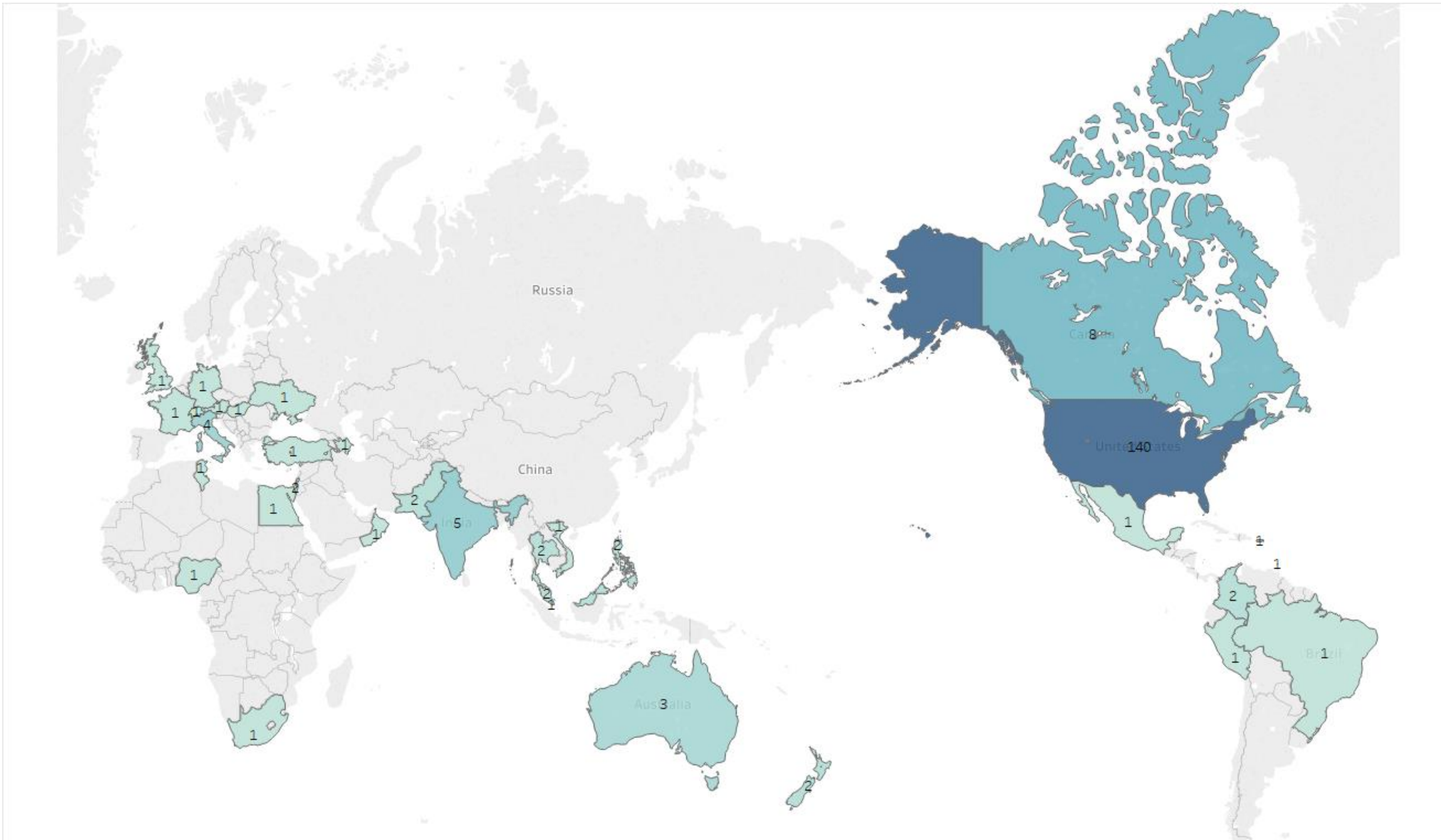



*Figure 8.* Number of participants by country.

Of the 195 participants in the sample population, 140 were from the United States. Within the United States, North Carolina was the most frequently-represented state in the course, with 49 participants. The Friday Institute, being located in North Carolina, has an advantage in attracting participants in North Carolina, due to name recognition and marketing reach. The difference in the number of North Carolina teachers relative to other locations is significant – Florida, New York, and California followed behind in terms of number of participants, with eleven, seven, and seven, respectively.  Worldwide, the next largest groups of participants were from Canada, India, and Italy with eight, five, and four participants respectively. The remainder were from countries scattered around the world. From the course discussion forums, I learned that British Columbia, Australia, and a handful of US States (Massachusetts, Ohio and Arkansas) have the elements of digital-age problem solving in their curricula already, which may drive adoption in future courses.

Enrollment is a two-step process by design. Basic demographic questions are only required to be provided once, regardless of how many courses in which a user wishes to enroll. A second survey, administered once a user has created an account but after they click the button to enroll in a course, is specific to each course. This survey asks questions specific to the user's motivations and goals for taking the course. This process limits duplicate information provided by users enrolling in multiple courses, though many questions are similar across courses for evaluation purposes. A summary of the results of these data for the sample population can be found in Figure 9. Both LeBar (2014) and Kleiman and Wolf (2015) argue that MOOC completion is an invalid metric to use to evaluate a MOOC – participants may gain knowledge from a MOOC and change their practice even if they do not complete it or engage intermittently.

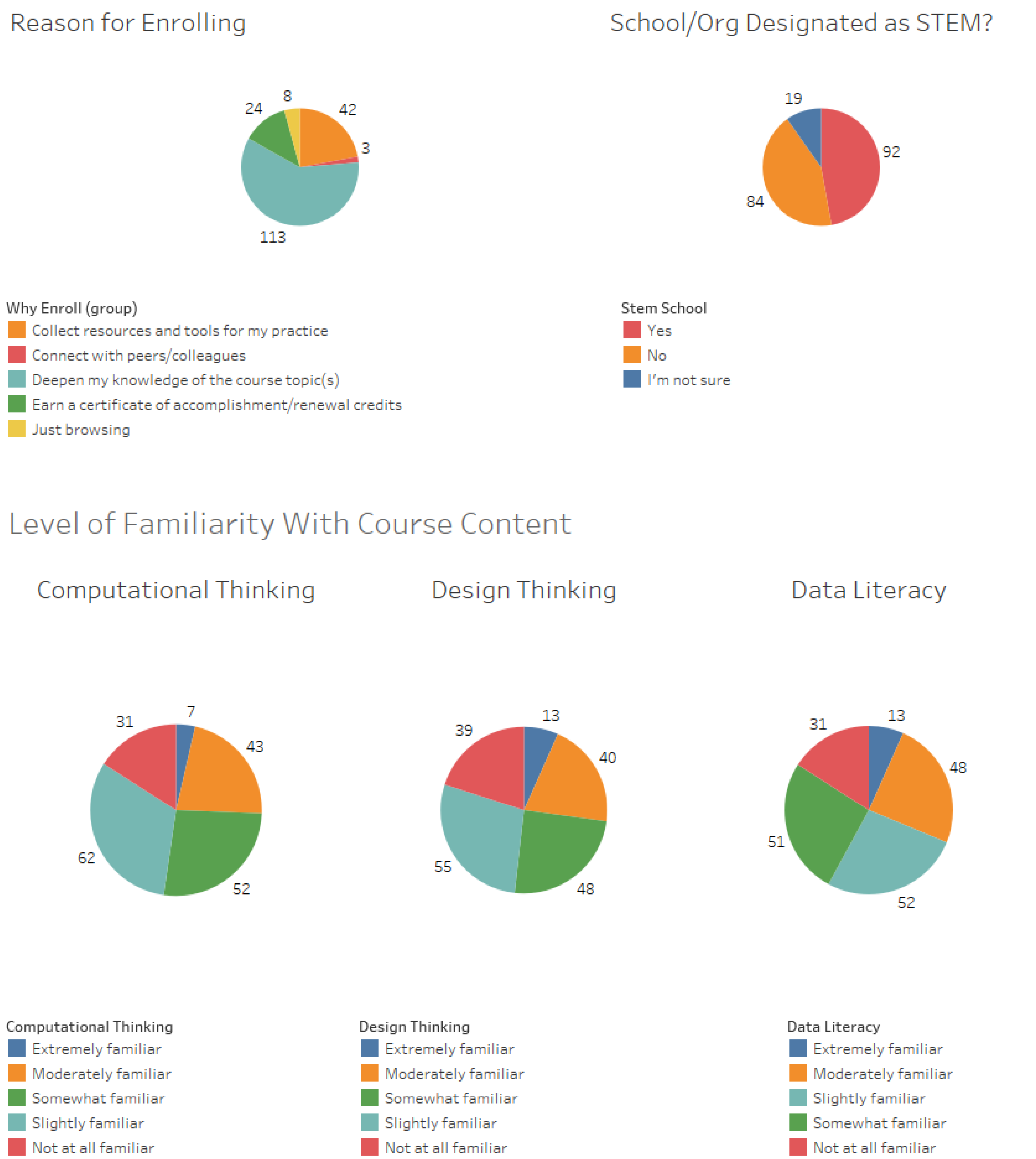



*Figure 9*. Participant motivation for enrolling/self-assessment.

In this MOOC, only 24 of 195 participants (12%) indicated that they were enrolling in the course in order to earn a certificate of completion. A majority of the sample population (57.9%) indicated that they were enrolling in the course in order to deepen their understanding of the course topics, with another 21.5% indicating that they were enrolling to collect resources to use in practice. The survey also asked users if they were in a school or organization with a STEM focus. This survey was essentially evenly split (47% responding "yes", the remainder as "no" or "I'm not sure").

In the course enrollment survey, users were also asked to self-assess their current familiarity with design thinking, computational thinking, and data literacy. While there are some minor differences, participant familiarity was relatively consistent between the three concepts. With respect to computational thinking, only 50 of 195 participants (25.6%) reported as being "extremely familiar" or "moderately familiar" with the content. Thirty-one (15.9%) reported as being "not at all familiar" with the course content. The remainder responded that they were "somewhat familiar" or "slightly familiar." With respect to design thinking, 53 of 195 participants (27.1%) reported as being "extremely familiar" or "moderately familiar" with the content, with 39 (20%) reported as being "not at all familiar" with the course content. The remainder responded that they were "somewhat familiar" or "slightly familiar". A larger number of participants (61, or 31.3%) reported being "extremely familiar" or "moderately familiar" with the concept of data literacy, with 31 (15.9%) responding as "not at all familiar.

**Engagement**

Earning a certificate of completion was not a primary motivator for many course participants. Engagement needed to be defined beyond earning a certificate to encapsulate both the stated motivations of participants and the fact that non-completion may still impact practice.

Interactions can primarily be grouped into two categories: reviewing the course resources and instructional materials or engaging in the course forums. A *consumer* is defined a user who has consumed course content – they viewed either a resource or a discussion board posting.

By extension, a *producer* consumes course content while also generating new discussion board posts. Any producer must also be a consumer, but for clarity, are only counted as *producers.* Adding total number of producers and consumers is the number of active users in each unit.

Figure 10 indicates the number of producers and the number of consumers in each unit of the course, restricted to only the identified study population. At least 25% of all of the active users were considered consumers within a unit. Unit One had the highest percentage of producers relative to the other units since a majority of the users in Unit One engaged in the "Introduce Yourself" forum where users were able to post a brief biography and network with other course participants. In Unit Three, there were slightly more consumers than producers. Units Two, Four, and Five interactions were approximately 60% producers/40% consumers. The cause of the anomaly in Unit Three is unclear, though one possible explanation is that an issue with email notifications in the platform means that some users weren't receiving weekly course emails until Unit Three was in progress. As a result, some enrollees had forgotten about the course and their initial logins were in Unit Three. It is possible that some users logged in that week, looked around, and decided against proceeding. Because of the requirements to complete the course, all course completers are required to be producers in each unit.

**Resources**

Each unit contained a curated list of external resources focusing on the topics presented in the unit. Participants were encouraged to review the resources most applicable to them in their

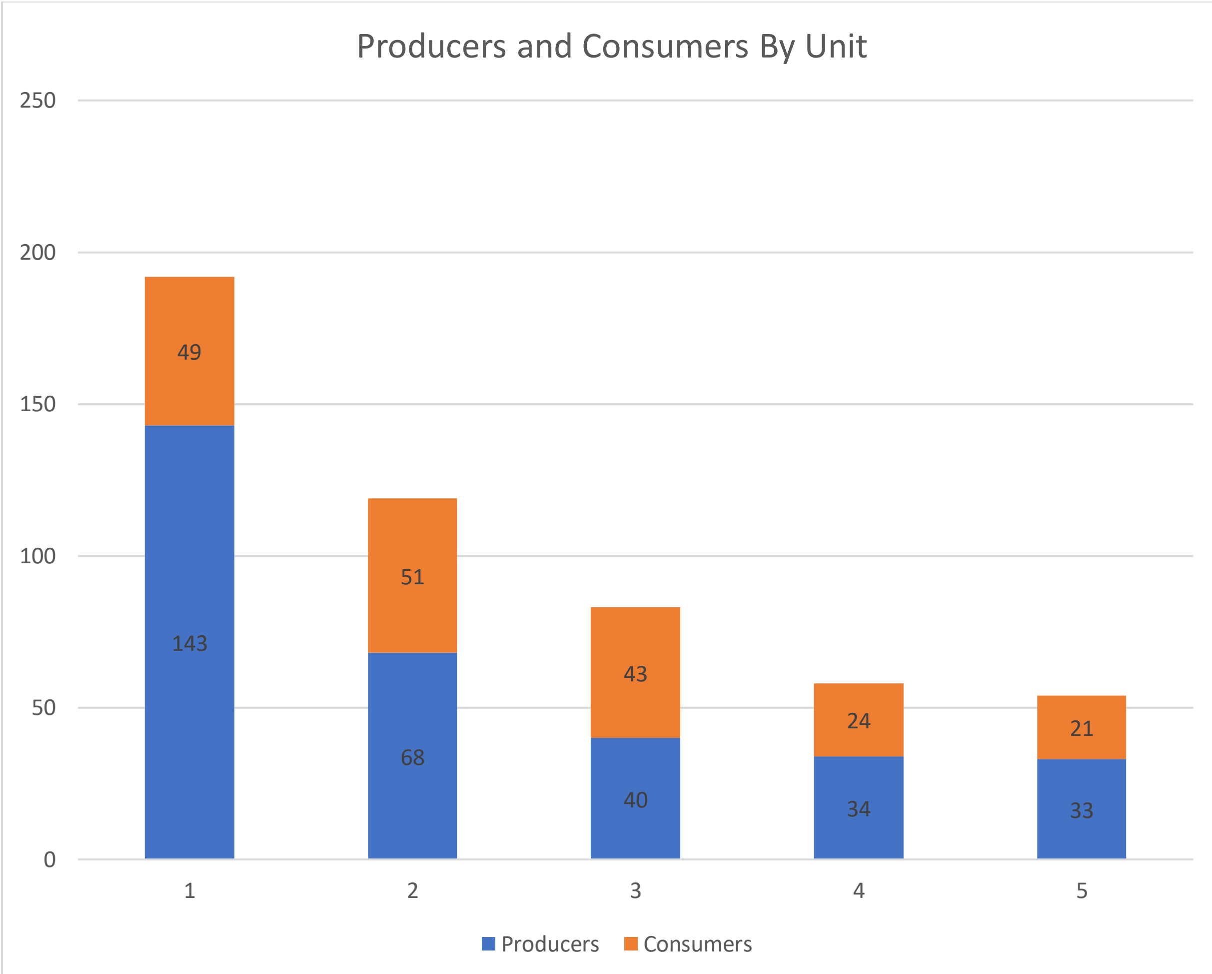



*Figure 10.* Producers and consumers by unit.

context. Each time a user clicked on a resource hyperlink, the platform recorded this activity in the database.

Figure 11 shows the number of users in the study population who clicked on at least one course resource in each unit. Resources are external websites that are cataloged and summarized for users wishing to dive deeper into course content.

In all units, between 53% and 62% of users accessed at least one of the course resources. Clicks were only logged if participants clicked on the link within the window. Users who bookmark links directly using certain plugins or browser settings may not be recorded. It's impossible to know how many users accessed the resources this way, so these data should be considered an underestimate using the best available data. Users also could rate resources on a 1-star to 5-star scale (there were no labels attached to the stars). Very few users took advantage of this feature.

Table 7 and Table 8 display the most- and least-frequently accessed resources in the course by users in the study population. Since users are able to access resources after the course has ended, click data was analyzed from the beginning of the course until September 1, 2017. The most-frequently accessed resources are contained on the first resources page in Unit 1, specifically the ISTE and Google resources on computational thinking as well as the Wing (2006) research paper where the term "computational thinking" is first defined. With a higher number of resource viewers in unit 1, it makes sense that the majority of resource views would come from Unit 1. The least-frequently accessed resources are all from Unit 2, specifically open-data sources intended for teachers to use in their classrooms to create data visualizations such as the NOAA and Word Bank open data websites. The Google Civic Information API had the fewest number of accesses. These resources are fairly technical and would only appeal to users

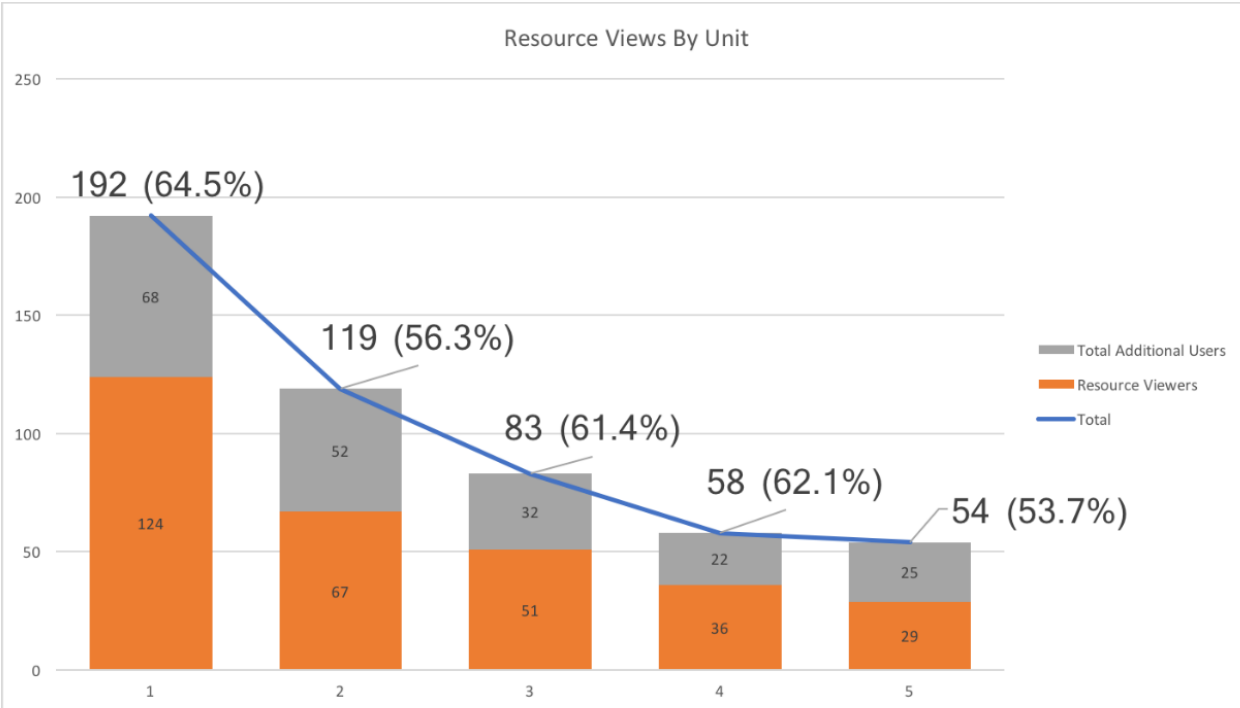



*Figure 11.* Number of users viewing course resources by unit.

Table 7

*Most Frequently Accessed Course Resources*

| Resource Title and Description (from course) | Click Count |
| --- | --- |
| *Computational Thinking for All*<br>Author: Carolyn Sykora<br>Source: ISTE<br>In the 2016 revision of the ISTE skills for students, Computational Thinking was included as one of the new ISTE standards. This resource hub contains definitions, getting started guides and resources for teachers, and school/district leadership. Free registration is required. | 126 |
| *Computational Thinking*<br>Author: Jeannette M. Wing<br>This opinion piece, from the Association of Computing Machinery, is the article that started the modern discussion on Computational Thinking. In it, Wing begins to lay out why Computational Thinking is an essential skill for everyone. | 110 |
| *Solving Problems at Google Using Computational Thinking*<br>Source: Google for Education<br>This video shares real-world examples of computational thinking components in use every day at Google, using Google Maps and Google Earth as an example. | 102 |
| *Bringing Computational Thinking to K-12: What is the Role of the Computer Science Community?*<br>Author: Valerie Barr and Chris Stephenson<br>Source: ACM Inroads<br>This seminal research study attempts to clarify what the role of computational thinking is in K-12, define core skills, and connect these skills to what teachers are doing in the classroom. | 79 |

Table 8

*Least Frequently Accessed Course Resources*

| Resource Title and Description (from course) | Click Count |
|---|---|
| *NOAA Open Data*<br>Climatological data for the United States, including temperature, rainfall, water temperatures, and forecasting model data. | 6 |
| *United Nations Open Data Initiative*<br>Repository of datasets maintained by the United Nations. Start by browsing the "Databases" box. Data for countries around the world can be found in the "Country Data Services" tab. | 6 |
| *U.S. Census Bureau*<br>The Census Bureau contains downloadable data from the last census, aggregated by census tract. They also have tools on the site to analyze, visualize, and sort the data. | 6 |
| *World Bank Open Data*<br>Datasets from the world bank regarding worldwide money and monetary policies. | 4 |
| *Google Civic Information API*<br>For programmers, this RESTful web service provides information about ballots, polling locations, and elected officials for a specified address. | 4 |

who had a pre-identified use case. Because the presentation of Digital-Age Problem Solving as a unified framework is unique to this MOOC-Ed, I developed a *primer* for each unit synthesizing the relevant elements of design, computational thinking, and data literacy into a single instructional unit. These *deep dives* are found in each unit and provide participants with a primer of the skills and mindsets relevant to the unit. Practitioners from a variety of fields also lent their voices and experience to a series of video introductions in each unit. Table 9 contains the number of accesses to each of these pages by unit and the number of video plays of the introductory video, from the start of the course through September 1, 2017. Video plays are captured if a user watches the video, accesses the transcript, or downloads the audio track. These numbers are significantly higher than the number of participants in each unit, which indicates that participants return to these resources multiple times and suggesting that these resources were valuable and useful in the course.

**Forum Participation**

With exception of Units 1 and 5, each unit of the MOOC had two discussion forums – one that encouraged the participants to deeply engage with the course material, and a second to encourage them to brainstorm applications into their classroom/educational practice. Unit 1 had an introduction forum for participants to introduce themselves to the other course participants, along with the classroom application forum. Unit 5 had a forum asking participants to reflect on their growth through the course experience as a whole. With exception of the introduction posts, all forum posts were analyzed (966 posts) against the Transcript Analysis Tool (Fahy, Crawford, & Ally, 2001) . Posts were also coded for users seeking and offering help and resources (Stump et al., 2013), for misconceptions, and explicitly stated new understandings and changes in

Table 9

*View Counts for Course "Deep-Dive" Content and Videos*

| Unit | Unit Introduction Access Count | "Deep-Dive" Access Count | Unit Introduction Video Play Count |
|---|---|---|---|
| 1 | 450 | 434 | 644 |
| 2 | 223 | 192 | 251 |
| 3 | 132 | 111 | 247 |
| 4 | 88 | 96 | 133 |
| 5 | 71 | 85 | 98 |

practice, and current high-fidelity applications practice. The elements of design thinking and computational thinking were also noted when stated accurately or demonstrated.

A majority of the posts (560 or 58%) were posts that present new information that does not build on other posts (Type 2A and Type 3 statements on the Transcript Analysis Tool). These posts were either at the start of a discussion thread or reply to a thread without building on any of the content that is already in the thread. Of these 560 posts, 295 contain some element of synthesis or reflection on course content (Type 3), while 353 simply restate course content without deep reflection or synthesis (Type 2A). A small number of these 353 posts were irrelevant, off topic, or unclear. While over half of the posts were responses to one or more posts in the forum, many of these interactions were superficial in nature and did not contribute to construction of new knowledge. Two hundred ninety-eight of the forum posts contained some type of interpersonal interaction between participants, often along the lines of “I agree!” or “good point!” (Type 4) while 207 posts engaged and expanded directly on the comments of another participant (Type 3). One hundred five posts referenced outside content directly or indirectly (Type 5). Descriptions of each of the types of posts in the Transcript Analysis Tool, along with sample quotes can be found in education, so I coded for these references as well. However, there were only 16 references to this movement. Example posts for each of these codes can be found in Table 10.

One hundred thirty-one posts pointed directly to a new understanding or to a change in practice, while many other posts pointed to work that teachers were already doing which they were able to identify as computational thinking or design thinking. There were 24 instances of participants either seeking help from other participants or providing help or resources. Thirty posts contained a critique of either the course content, the relevance of a particular strategy, their

Table 10

*Transcript Analysis Framework Codes (Fahy et al., 2001) with MOOC-Ed Examples*

| Code | Description (Fahy et al., 2001) | Example Quote(s) |
|---|---|---|
| Type 1A: Questioning | Includes vertical questions, which assume a "correct" answer exists, and the question can be answered if the right answer can be found. | *Family code night sounds pretty awesome! How many families typically attend?*<br><br>*I would love to see the template as well! What grade level did you use this with? Are there any suggestions you have for folks who want to give it a try? Anything that you will change as you use it again in the future? Any modifications you make for special populations?* |
| Type 1B: Questioning | Horizontal questions: there may not be one right answer, and others are invited to help provide a plausible or alternate "answer," or to help shed light on the question. | *Based on what you've read so far, do you see potential points of connection in your instruction?*<br><br>*We are looking at how students' access to the ADST curriculum and coding activities can be supported if they have more complex needs (e.g. physical access and communication challenges). Doesn't seem to be much out there written or shared in this area - do you have anything in your resource basket that highlights successful programs or which coding tools are more accessible?* |
| Type 2A: Statements | Statements contain little self-revelation and usually do not invite response or dialogue. The main intent is to impart facts or information. | *I'm not teaching in a classroom too at the moment, although I am a teacher: what I've seen here in Italy is that students are struggling, too, as they do not grasp the utility of what they are learning: from the materials I've been through into the dig deeper section, I'm confident there will be many things yet to learn, by us teachers, in order to make them grasp the "why" and not "how" and not only the "what" of the subjects they are studying. I hope to be able to implement them.*<br><br>*I use flow charts in my math classes to help students learn the steps to math techniques that have traditionally been difficult for them to follow.* |

Table 10 (continued)

| Code | Description (Fahy et al., 2001) | Example Quote(s) |
|---|---|---|
| T2B: Statements | Direct answers to questions, or comments referring to specific preceding statements. | *These are all excellent resources to use. SAS Curriculum Pathways has datasets that are pre-cleaned and ready to use. Some of the data tools we'll discuss in unit 2 may be helpful as well. In English classes, it's also very possible to create your own datasets to use. Network diagrams (maps that show relationships between characters and events), word clouds (to see if themes emerge or to compare two primary source documents -- word clouds for President Bush's inaugural address and President Obama's are virtually identical), and building maps and timelines are all potential ways to turn textual material into data that can be analyzed. At the younger grades, some of these visualizations may need to be pre-generated, but students can still do the interpretation and analysis.* |
| T3: Reflections | The speaker expresses thoughts, judgments, opinions or information which are personal and are usually guarded or private. The speaker may also reveal personal values, beliefs, doubts, convictions, and ideas acknowledged as personal. | *For many years I tried to design the "perfect" materials for my students only to realize that once in a while they didn't meet the needs of my students because those materials didn't take student feedback into account. So I started to do pilots. Before I engaged on a long process of writing material unfit for my students I presented them with a sample of what I was trying to come up with. Depending on their reaction I could then follow through or adapt my approach. In a next step I asked myself how do I let the students come up with part of the learning material themselves by providing them with tools which of course had to be tested over and over again before they could be applied on a bigger scale. Now, my students occasionally (not always) produce their own teaching material in the form of presentations and short quizzes. What I want to do next is to teach them a general habit of testing their materials on others in an early stage. I think that is exactly the point where this course has been giving me some valuable hints how to help my students.* |

Table 10 (continued)

| Code | Description (Fahy et al., 2001) | Example Quote(s) |
|---|---|---|
| T4: Scaffolding and Engaging | Intended to initiate, continue or acknowledge interpersonal interaction, and to "warm" and personalize the discussion by greeting or welcoming. | *Thank you for sharing! This is the first time I've seen the video and plan to share it with other teachers!*<br><br>*Wow, Carol! These suggestions are fantastic. I'd love to try some of them with the students! This is a bulletin board in one of our hallways, but it would be great to have a group focus on it and do more with it. Thank you so much for the ideas. I'm really excited! (Now I'm going to check out Ivan's flow chart -- thanks again!)* |
| T5A: References | References to, and quotations or (fairly direct) paraphrases of other sources. | *For middle-school students, I've found Alice 3, Scratch and robotics activities with Lego Mindstorms EV3 (or for older students, M-bot and Ranger-bot) as accessible ways to develop students' design and computational thinking, and the robotics activities for developing ability to deal with data. Barr and Stevenson's proposals for bringing computational thinking to K-12 students are a great prompt to integrate these methods with a range of other disciplines in the classroom.*<br><br>*I am reminded of Atul Gawande's "Checklist Manifesto." It does seem like this would be a "routine" list of items to check and verify.* |

Table 10 (continued)

| Code | Description (Fahy et al., 2001) | Example Quote(s) |
|---|---|---|
| T5B: References | Citations or attributions of quotations or paraphrases. | *I have created a few professional learning opportunities both face to face and virtual on design thinking. In CMS, we have our own design thinking process that we utilize that you can find here: https://goo.gl/8sde3x. During the PD our goal is to develop an understanding of the Design Thinking Process while also allowing the educators to create and implement a design thinking challenge in their classroom. We do this by having them go thought the design thinking process with our design thinking template. I would like to create professional learning opportunities around computational thinking and data literacy. I also need to do a better job of integrating data literacy and computational thinking into the professional learnings that we offer already so educators can see it seamless integrated like we want it to be in the classroom.* |

ability to implement given competing educational priorities, or representing disagreement with another participant. Twenty-four posts are explicit misunderstandings or misinterpretations of course content. I was also curious about references to the Maker movement and informal education, so I coded for these references as well. However, there were only 16 references to this movement. Example posts for each of these codes can be found in Tab;e 11/.

Specific to course content, data literacy and iterative design/productive failure were the most commonly mentioned design thinking topics, while interdisciplinary instruction was mentioned the least. Decomposition was the most commonly mentioned computational thinking skill, while abstraction was mentioned the least. Example quotes for each of the computational thinking/design thinking habits of mind can be found in Table 12..

**Summary of Course Engagement Data**

While forum posts are the primary vehicle for earning credit for the MOOC-Ed, there are very few meaningful exchanges (and virtually no multi-post exchanges) between participants. That does not mean the forums are not useful, as many of the ideas shared were deep reflections and good examples of application to practice that demonstrated mastery of the course content. Engagement patterns varied widely between users, with significant numbers of users only consuming content, and never creating any content. The supplemental resources in each course were accessed by approximately 50% of users. The unit introduction and deep dive content was accessed much more frequently and were the most commonly-accessed elements of the course.

## Course Completers

Thirty-seven participants in the MOOC-Ed earned a certificate of completion (approximately 18.97% of the study population). Other participants may have completed most (or all) of the course requirements, but are not recorded since they did not access their certificate.

Table 11

*Additional Learning Community-Focused Codes in MOOC-Ed*

| Description | Example Quote(s) | Number of Occurrences |
|---|---|---|
| Explicit references to changes in practice | *We use a lot of data in science experiments, but I love the use of data in real contexts to solve problems and design solutions to meet the needs of people. In Environmental Science, I see great applications of utilizing qualitative and quantitative data to design solutions to environmental problems, with a real emphasis on designing with people in mind. I also enjoyed all the data storytelling. Oftentimes, students will include graphs/data tables, but aren't really given the freedom to create some amazing graphics and present those to the class to tell a story with their data. I plan to use more infographics in the class and really pull in empathy in the design process.* | 54 |
| Critiques of user posts or course content | *I totally agree. I have to say that it is a bit scary to implement though! So much emphasis is on test scores and standards. I want to make sure I'm teaching my kids the skills and the concepts so that they feel successful on all ends.* | 30 |
| Seeking help from other course participants | *I am also no longer in the classroom, and am in a professional development, curriculum, coaching, and co-teaching role. It sounds like you are doing lots of great things with DT and CT. I'm curious what types of training your teachers are receiving on those topics? I'm looking at designing some professional learning for my teachers, and would love to hear what you are doing!* | 21 |
| Providing help to other course participants | *Thanks, Alex. So far no formal brick walls have presented. Getting the bus on-the-road ready will cost us about $400. I could eke that out of my budget. Town Accountant doesn't think insurance will be much as the bus is already insured as part of the school fleet. Still waiting on details. Glad you liked the comics! They're not available for commercial use but I think "fair use" dictates we can use them as long as they are part of a larger instructional package.* | 13 |

Table 11 (continued)

| Description | Example Quote(s) | Number of Occurrences |
|---|---|---|
| References to the Maker movement | *I have been helping out with our school's competitive Robotics club this year, and it's clear that these students are deeply involved with ideation, algorithms, design thinking, and more. For example, students are required to identify and come up with a solution for a problem that involves humans and animals (ideation). This problem-solving happens alongside their efforts to have their robots complete animal-related challenges (parallel processing and algorithms). In early club meetings, students identified a problem and then "Threw things against the wall" until they were able to come up with a solution the whole team could agree upon. They then divided up research topics so that they could use parallelization in writing up their problem/solution paper. When they work with their robots, the students are writing algorithms to program the robots through a series of tasks.* | 16 |
| Misconceptions in understanding of course content | *I think also abstraction is necessary there since a student may wish to take electives given a specific grade level and while they may test the schedule for one student they need to take what they have realized from that test and abstract it to many students who may also want to take an elective. We actually have the reverse problem of being very small so decomposition into grade level is key but we have to also decompose electives against each other since we only run each elective once a day.* | 24 |
| Explicit references to new understandings or realizations from engagement in course content | *I see myself using this content in teaching science. It helped give me a slightly different mindset when it comes to designing experiments and allowing students more freedom to explore, discover, fail, and learn on their own from the process, instead of it being so formulated for them much like a recipe to follow.* | 77 |

Table 12

*Course Competencies Explicitly Stated or Implied in MOOC-Ed Discussions*

| Competency | Example Quote(s) | Number of Occurrences |
|---|---|---|
| Data Literacy | *These are all excellent resources to use. SAS Curriculum Pathways has datasets that are pre-cleaned and ready to use. Some of the data tools we'll discuss in unit 2 may be helpful as well. In English classes, it's also very possible to create your own datasets to use. Network diagrams (maps that show relationships between characters and events), word clouds (to see if themes emerge or to compare two primary source documents -- word clouds for President Bush's inaugural address and President Obama's are virtually identical), and building maps and timelines are all potential ways to turn textual material into data that can be analyzed. At the younger grades, some of these visualizations may need to be pre-generated, but students can still do the interpretation and analysis.* | 120 |
| Empathy | *I read that work as well and we ended up using it for professional development for our entire staff. The Math reluctance is especially prevalent in some families and cultures with antiquated gender roles or academic expectations. As a mentor for struggling students, the hardest barrier to break is that fear of failure and its impact on their perception by their peers. They have built up so many strategies to protect themselves from failing that are often unable to even try.* | 88 |
| Identifying Problems | *My students use data from their pre-assessments to set learning goals for themselves for the units. They find areas where improvement is needed, based on the data from the assessment, and set a growth target that they hope to achieve by the post-assessment.* | 90 |

Table 12 (continued)

| Competency | Example Quote(s) | Number of Occurrences |
|---|---|---|
| Interdisciplinary Instruction | *I have used Design Thinking in my biology class by having students think of all the barriers that are in the way of a cell undergoing division and then offer a possible way for the process to occur. I also had them propose a redesign of a local green space when we visited to do water testing. They were to analyze the health of the watershed and then propose what can be changed in the area, focusing on 1) how people use the space currently, 2) how else the space can be used, and 3) changes that would not affect the current flora and fauna. I have only used computational thinking once, though I know in a Biology class we have used the skills elsewhere. Students had to design an algorithm (procedure) in order to prove that fruits and vegetables were made of cells. We had not even used a microscope at that point. They are so used to being given procedures that it was interesting to see them struggle. We use data analysis often and when it is used in analyzing what biomolecules are in foods (we do food testing), they analyze every student's results and use that information in many of the other activities that they do in the unit including the creation of a school menu that is balanced and based on science.* | 24 |
| Iterative Design, Productive Failure, Rapid Prototyping | *I agree with you that students are to wrap up in if they get the answer right. I teach Engineering and Design at the middle school level and in my class students learn to be problem solvers by learning and practicing the Engineering Design Process. We focus on identifying a problem, researching the problem, brainstorm possible solution, choosing possible solution, designing and prototyping, testing and then evaluating for improvement. But even at the end of a design challenge when something goes wrong with a design someone will ask me if this means they will get an F. I have to remind them that I am looking how they worked the process to arrive their solution. If they can demonstrate an understanding of how the process helps them solve the problems they are in good standing.  Even professional don't get it right the first time.* | 130 |

Table 12 (continued)

| Competency | Example Quote(s) | Number of Occurrences |
| --- | --- | --- |
| Real-world examples and connections | *During an inquiry unit that we do during science, our 4th graders learn about erosion, and then need to design a way to curtail the erosion from happening. First they gather data during an erosion simulation, then they study other ways that erosion problems have been addressed in other areas. Finally, they begin the design process. Eventually they get to build, test, and evaluate the effectiveness of their designs. It is one of the most engaging experiences of our year.* | 111 |
| Testing and Failure Analysis | *It sounds like a "failure of requirements." Perhaps the mandated timeline did not allow for sufficient time to get all of the data that would be needed. What will be site be used for? How many people will need to use it?  How will we test our iterations? These are all questions that needed to be thought about before the process began. Conversations with experts in the health care/insurance field would be imperative.* | 86 |
| Unconstrained Thinking | *Almost every activity in a person's daily life might be described with an algorithm, so it's very easy to find some (it would be sufficient to name any activity I might do during the day). What I'd like to observe is the difference is whether we are able to recognize them and utilize them to go through our activities at best. For example: if I'm baking a cake and I'm aware that is an algorithm, I can apply the procedures in order to spend less time and maximize the effectiveness of the time I'm spending making the cake.* | 29 |

Table 12 (continued)

| Competency | Example Quote(s) | Number of Occurrences |
|---|---|---|
| Abstraction | *In my work with new instructional coaches, I have opportunities to support them as they work with teachers in grade levels and PLCs in their schools. One challenge they often face is identifying the specific areas where they need to focus their work. Through the decomposition process, we are able to break down the focus area(s). Abstraction allows us to eliminate the extraneous factors and identify commonalities. The lens of "sphere of control, sphere of influence, and sphere of concern" provides an additional way to think about the challenges they face.* | 27 |
| Algorithms | *Thinking to my classroom, there are many math lessons that have to be sequential. For instance, I teach students linear equations before I tackle quadratics. I think students what a function is because we look at the characteristics of a function. From year to year, however, sometimes I change up the order. For instance, should I teach the Quadratic Formula first or the completing the square method first. The quadratic formula is derived from completing the square, so there is the argument of doing completing the square first. The steps for the quadratic formula are easier for the students to carry out than the steps for completing the square, so there is an argument for teaching the quadratic formula first. In my school, teachers are encouraged to do more personalized learning (Letting students work at their pace rather than the teachers). With this idea, I've tried to make lessons that are more parallel in nature so that students can pick and choose what topics are study first. It is, however, very important to go back at the end and help students see the connections to all the methods or types of problems they learned. We certainly don't want students to view math skills in isolation!* | 99 |

Table 12 (continued)

| Competency | Example Quote(s) | Number of Occurrences |
|---|---|---|
| Decomposition | *In both my Algebra 2 and Geometry class, students use decomposition and abstraction to solve problems. Especially in word problems in Alg2 and formal proofs in Geometry. I always recommend students to decompose the problem, list skills (formulas, theorems, etc), come up with a plan, before they start solving or proving problems. It seems to help to break down to small pieces.* | 116 |
| Pattern Recognition | *"I feel like we want students to see the patterns in language for themselves, that pointing out those patterns to the students undermines their opportunity to learn how to identify patterns themselves." Pattern recognition is a key computational thinking skill. I think that having students identify these patterns, whether explicitly or not, is computational thinking. Knowing how colleges can be sometimes, I wonder if there's a possibility for you to infuse digital-age problem solving not by creating new assignments, but by changing some of the language you use around the assignments you're already doing. Is this realistic?* | 74 |

Figure 12 includes course-specific registration survey data of all of the course completers. Of the 37 people who earned a certificate, only 8 indicated that earning a certificate was a goal for signing up for the course. This means that 16 participants who selected this goal initially did not end up completing the course. However, a majority of the completers did indicate that they were either “not at all familiar” or “slightly familiar” with design thinking, data literacy, and computational thinking when they registered for the course, and nearly half indicated that they enrolled to deepen their knowledge of the course content.

**In-Course Evaluations**

At the conclusion of each unit (1-4), participants were asked to complete a short survey about the unit they just completed. Unit 5 did not have a survey because participants were presented with an end-of-course survey instead. Participants were asked two Likert-scale questions on a six-point scale and starting with the prompt “to what extent do you agree with the following statements.” The two survey items were “this unit deepened my understanding of the topic addressed” and “the unit supported application of course content to my professional practice.” The number of users who responded “agree” or “strongly agree” for each unit can be found in Table 13. For each unit, more than 90% of participants responded that the unit did deepen their understanding of the topic, while over 80% responded that the unit supported application of the course into their practice (the lowest score on this question was in Unit 1, which was largely introductory).

Each unit also asked participants open-ended questions about the most valuable aspect of the course and recommendation for improvement

A few sample responses for the “most-valuable aspect” question in each unit follows. The primary themes across all of the responses centered on the course resources and the

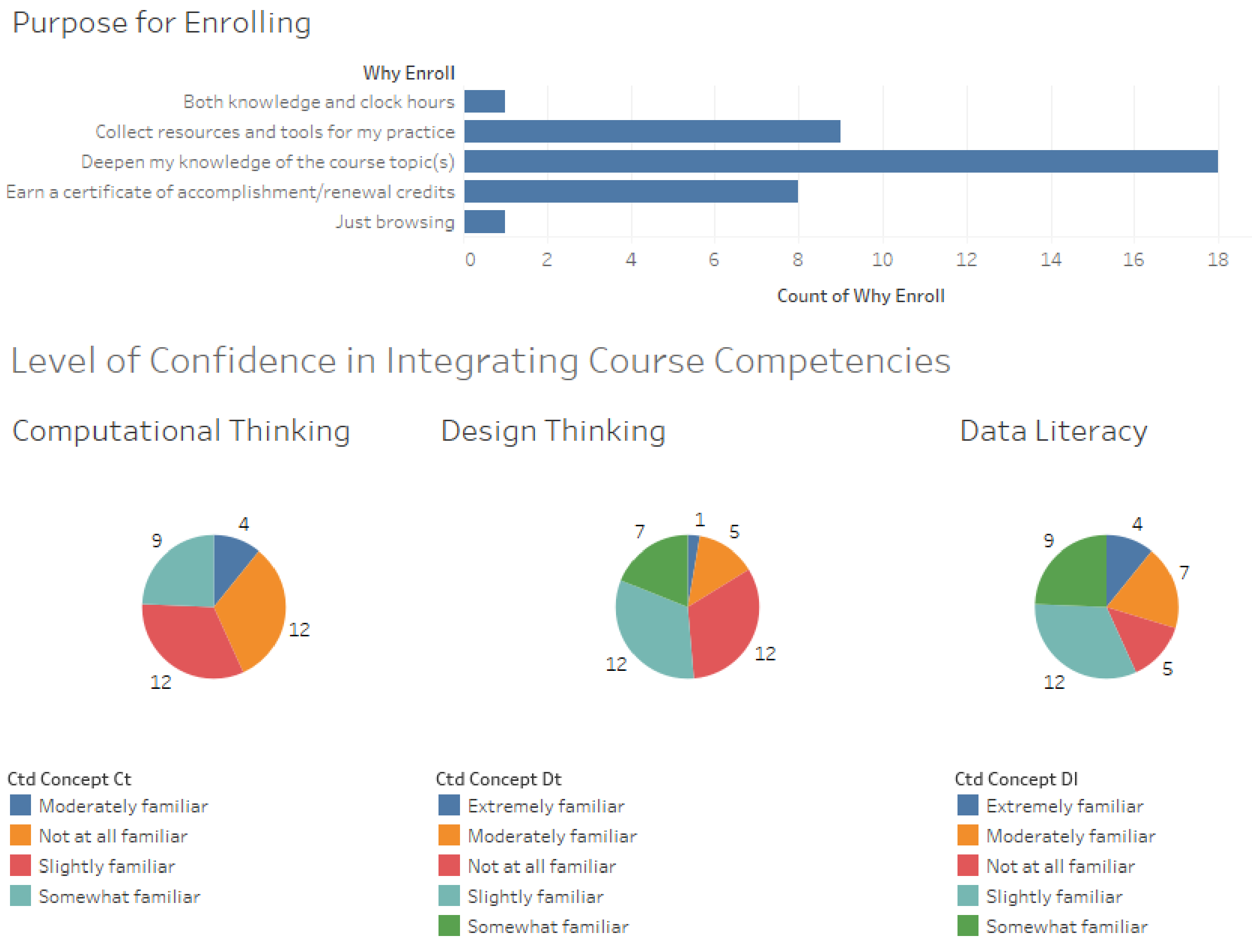



*Figure 12.* Registration data on enrollment motivations for course completers

Table 13

*Percentage of Users Answering "Agree" or "Strongly Agree" in End-of-Unit Surveys*

| Question | U1 | U2 | U3 | U4 |
|---|---|---|---|---|
| This unit deepened my understanding of the topic(s) addressed. | 77/81 (95%) | 51/55 (93%) | 39/41 (95%) | 38/39 (97%) |
| This unit supported the application of course content to my professional practice. | 67/81 (83%) | 52/55 (95%) | 38/41 (93%) | 36/39 (92%) |

digging deeper pages, followed by references to specific content, followed by the discussion forums. “The unit delivered very straightforward explanations of the different types of thinking involved in Digital Age Problem Solving. In terms of an introductory unit, this portion of the course did well to define the foundations necessary for understanding the rest of the course.”

“The resources are exceptional. There is a great variety and I anticipate coming back to them as my needs/interests change. It is an excellent collection and I've found things I want to immediately use in my classroom.”

“Seth Godin's video. I wasn't familiar with him until I saw the link in this unit. I also liked Dr. Manchanda's video quite a bit. I shared this with other teachers and our school's counselor. Very inspirational and I was able to make a lot of connections in this unit. Loved it.”

A few responses for the question about improving the course experience follows. The primary themes here were either for participants to restate things that they liked about the course or suggesting additional resources they would like to see added. Other participants mentioned the balance of reading to videos in the course content and the amount of reading in general. Participants also suggested that more interactivity was needed in the course and wished the platform had a more modern visual layout. A few participants mentioned specific technical issues as well.

“I'd like to see more examples that are practical, classroom problems. A lot so this is for upper grades. Examples for the very lower range would specifically support me and probably give others easy launch projects.”

"I would say less is more for the reading.  Perhaps choose 1-2 reading and 1-2 videos. The other teachers that take these MOOCs from the Friday Institute with me often say the same thing-- that there's an overload on reading. I had to download the resources and save them for later viewing."

"Maybe it's because I'm not far enough along in my own planning yet but the core resources in this section didn't seem as helpful as some of the other sections."

"I think the concepts of this unit are more difficult to understand and are not as well explained in the intro video as they are in many of the core resources, I would include more information in the intro video that transitioned better into the core resources."

Unit 3 also asked for examples of how participants were implementing the course content in their practice. Almost all of the responses provided here were anticipatory – things participants were thinking about or planning to do in their practice. While there were no concrete examples of application to practice in this section, there was evidence that participants were beginning to think about possible ways they could begin to apply to their practice. A few sample quotes follow:

"I am going into a role as a digital coach next year.  I definitely see these skills helping out when I help teachers plan their units about technology integration into their curriculum maps. I plan to do this with questions that function in a backward design process of decomposition. Step by step, what is the end goal teachers have for their students and how can they break those milestones down into smaller more manageable steps?"

“I think I will utilize the "Is it Broken" thoughts from the core resource and look at some of lessons to see if they are "broken" from either my vantage point or the vantage point of my students.”

“When faced with a problem to solve, I need to make sure that I can commit the necessary time upfront to adequately define and thoroughly decompose/abstract before looking for a solution.”

## Summative Evaluations

At the conclusion of the course, all remaining participants were surveyed and asked to provide feedback on the course and the impact on their practice. A similar survey was sent via email to all of the enrollees who signed up for the course.

Forty participants responded to the end-of-course survey. The survey included a series of Likert scale questions (see Tables 14 and 15), and several yes/no questions along with spaces for more detailed feedback. Of the 40 respondents, 38 reported the course overall being “Effective” or “Very Effective” in supporting their professional learning goals. With respect to the individual course elements, the course “digging deeper” pages were the most useful to course participants (4.68 numerical equivalent average), with video resources and readings also being highly rated with numerical averages of 4.58 and 4.55 respectively. The course introduction page was the only one that received any rating of “ineffective” or “very ineffective” (one participant rated the pages ineffective). The application to practice discussion forum received the lowest numerical average with 4.23. Related to this, 8% of course participants responded “neither agree nor disagree” (no participants responded “disagree” or “strongly disagree”) when asked if the course improved their knowledge or skills relative to integrating course content into their instructional practice, though 97% responded in the affirmative that the MOOC-Ed was effective

Table 14

*End-of-Course Survey Responses: Effectiveness of MOOC Components*

| Question | Very Ineffective | Ineffective | Neutral | Effective | Very Effective |
|---|---|---|---|---|---|
| As a whole, how effective was this MOOC-Ed in supporting your personal and/or professional learning goals? | 0<br>0% | 0<br>0% | 2<br>5% | 14<br>36% | 23<br>59% |
| How effective was the *Course Introduction* in supporting your professional learning? | 0%<br>0% | 1<br>3% | 2<br>5% | 23<br>58% | 14<br>35% |
| How effective were the *course "digging deeper" readings* in supporting your professional learning? | 0<br>0% | 0<br>0% | 1<br>3% | 11<br>28% | 28<br>70% |
| How effective were the *video resources* in supporting your professional learning? | 0<br>0% | 0<br>0% | 2<br>5% | 13<br>33% | 25<br>63% |
| How effective were the *articles/course readings* in supporting your professional learning? | 0<br>0% | 0<br>0% | 1<br>3% | 16<br>40% | 23<br>58% |
| How effective was the *Discussion Forum: Course Activity/Real-World Application* in supporting your professional learning? | 0<br>0% | 0<br>0% | 4<br>10% | 23<br>58% | 13<br>33% |
| How effective was the *Discussion Forum: In My Classroom* in supporting your professional learning? | 0<br>0% | 0<br>0% | 3<br>8% | 24<br>60% | 13<br>33% |
| Overall, how effective do you feel this MOOC-Ed was in preparing you to make positive changes in your professional practice? | 0<br>0% | 0<br>0% | 1<br>3% | 17<br>43% | 22<br>55% |

*Note.* (n=40).

Table 15

*End-of-Course Survey Responses: Improvement of Knowledge/Skills*

| Question | Strongly Disagree | Disagree | Neither Agree Nor Disagree | Agree | Strongly Agree |
|---|---|---|---|---|---|
| As a result of my participation in this MOOC-Ed, I have improved my knowledge and/or skills related to the design thinking process. | 0<br>0% | 0<br>0% | 1<br>3% | 16<br>40% | 23<br>58% |
| As a result of my participation in this MOOC-Ed, I have improved my knowledge and/or skills related to core components of computational thinking. | 0<br>0% | 0<br>0% | 2<br>5% | 16<br>40% | 22<br>55% |
| As a result of my participation in this MOOC-Ed, I have improved my knowledge and/or skills related to the role of empathy and data collection in the design process | 0<br>0% | 0<br>0% | 5<br>13% | 17<br>43% | 18<br>45% |
| As a result of my participation in this MOOC-Ed, I have improved my knowledge and/or skills related to integration of digital-age problem solving strategies into my instructional practice | 0<br>0% | 0<br>0% | 3<br>8% | 18<br>45% | 19<br>48% |

*Note.* (n=40).

in preparing participants to make changes in their professional practice. 95-97% of participants responded that they did improve their skills relative to computational thinking and design thinking and 87% reported the same for data literacy. Eighty percent of respondents reported that they were able to complete all of the activities that they wanted to in this course.

The data from the free-response data was coded according to the major themes that emerged. The first question, “What was the most valuable aspect of this MOOC-Ed in supporting your personal or professional learning goals?” received 32 responses. These responses were coded based on the major themes that emerged and frequencies for these codes can be found in Table 16. Course resources and course content were mentioned 26 times, representing a majority of responses. The course layout and collaborative elements were mentioned but were mentioned the fewest number of times. Here are a few example quotes:

> “As always I appreciated the summaries of complex topics in the videos provided and the collection of useful and engaging materials throughout the course.”
>
> “The additional resources were a lot to look through, but I have found some very helpful articles and videos that will help me with some of my professional learning goals.”
>
> “I really enjoyed reading the various articles. I was pleasantly surprised at how to non classroom examples (sic) still helped me to see value in the design process.”
>
> “I loved seeing the real-world applications of the content discussed in this course. It was so helpful to hear these concepts discussed by other professionals in a variety of fields and to hear how others in school settings were applying them.”
>
> “The "Digging Deeper" readings and the opportunities to collaborate with other participants.”

Table 16

*Major Themes from End-of-Course Survey Question on Useful Course Elements*

| Theme | Frequency |
|---|---|
| Course Videos (type not explicitly stated - can include introduction videos and videos in the resource library) | 4 |
| Course Readings (type not explicitly stated – can include course content or external resources) | 4 |
| Course Resources (external) | 14 |
| Forums and Collaboration | 3 |
| "Digging Deeper" course content | 5 |
| Diversity of Materials Presented | 4 |
| Content/New Learning (specific mention of the content overall and translation into new learning) | 12 |
| Non-education focus (specific mention of the value of having resources from other fields beyond education) | 5 |
| Course Structure and Layout | 2 |

"The course is a practical guide to design planning with readings, strategies, questions for thought. I liked the practical tips and strategies, the encouragement to apply meta-cognition in reflecting on how Design Planning and Computational Thinking come into play in the work that I do."

"I haven't submitted my entire micro-credential project yet, but I really liked implementing what I learned with my kids. It was like my own test run. They felt so important, and I've gotten such great feedback from the staff about their enthusiasm to problem solve. Other than the project, I also really enjoyed the videos and connection to resources."

The second free-response question in the end-of-course survey was "please describe any changes you have made to your practice, including how you have applied the knowledge, skills, and/or resources you gained in this course." Thirty-two responses were analyzed and coded inductively based on major themes which were mentioned in the response. The themes presented, with frequencies can be found in Table 17. A significant number of teachers (13) did indicate that they have made changes in their lessons and in their interactions with students as a result of this MOOC.

Eight teachers mentioned that they are using the course content as a tool for reflection upon their practice, while 7 mentioned that they are using the content in work with their peers. A sample of representative responses follows:

"I have become more intentional about sharing the computational thinking process with my students during problem solving."

"I use the 5 step method to develop and maintain effective data collection and design in the ec program...I now have a better understanding of how to gather and use the data I

Table 17

*Major Themes from End-of-Course Survey Question on Application to Practice*

| Major Theme | Frequency |
|---|---|
| Use of course content in interaction with peers | 7 |
| Use of course content in lessons and interactions with students | 13 |
| Use of course content in reflective practice | 8 |
| Use of course content in online learning contexts | 1 |
| Specific application or mention of computational thinking | 5 |
| Specific application or mention of design thinking | 7 |
| Specific application or mention of data literacy | 6 |
| Use or mention of "decomposition" as a core skill | 3 |
| Use or mention of testing as a core skill | 1 |
| Use or mention of empathy as a core skill | 7 |
| Use or mention of iteration and productive failure as a core skill | 3 |

collect to make changes, if needed, or improve upon what I do for my students."

"Thinking about what I've read and the course discussions I've had, I've approached my admin colleagues to talk about possible projects we could implement. One involved data assessment and planning based on that assessment. The second involved thinking about coming up with a plan for turning an unused yellow school bus into an after-school resource with Internet access."

"I am encouraging innovation by taking the fear out of failure. I am showing my co-workers the nuts and bolts of decomposition and abstraction."

"Slight change in assessments and setting up intervention based on these. Also am encouraging students to design our classroom for more effective learning and to encourage empathy in understanding what people want or need (students)".

A second survey (dubbed "Impact Survey") was sent to course participants in July, and again in September 2017 (see Table 18). This survey was part of the evaluation for the larger MOOC-Ed initiative at the Friday Institute. This was sent to all course enrollees, whether they completed the course or not. A total of 49 users responded, though it is impossible to know how many of these users also took the end-of-course survey. The survey consisted of several "Yes/No" and "check all that apply" questions, with open-ended follow-up. Of the 38 participants who did not complete the course, 29 pointed to a lack of time for completing the course while 9 pointed to either that they were "just browsing" or changed their mind about taking the course (users could select multiple options). 32 out of 46 respondents indicated that they acquired new skills or resources that could be applied to their professional practice and 23 out of those 32 indicated that they have applied these skills in their professional practice, with 14 of 21 from this subset indicating that these practices have directly impacted students.

Table 18

*MOOC-Ed Impact Survey Results*

| | Yes | No | Total |
|---|---|---|---|
| Did you complete the Computational Thinking and Design MOOC-Ed course? | 11<br>22.45% | 38<br>77.55% | 49 |
| I did not complete this course because *I had less time than I anticipated.* | 29<br>56.86% | - | 51 |
| I did not complete this course because *I was just browsing.* | 5<br>9.80% | - | 51 |
| I did not complete this course because *I changed my mind about taking this course.* | 4<br>7.84% | - | 51 |
| I did not complete this course because *the course required more time than I anticipated.* | 4<br>7.84% | - | 51 |
| I did not complete this course for *another reason*. | 4<br>7.84% | - | 51 |
| I did not complete this course *because I did not want to earn a certificate of completion.* | 2<br>3.82% | - | 51 |
| I did not complete this course because *the content was different than what I expected.* | 1<br>1.96% | - | 51 |
| I did not complete this course *because I accomplished my learning goals.* | 1<br>1.96% | - | 51 |
| I did not complete this course because *I was already familiar with the information presented.* | 0 | - | 51 |
| I did not complete this course because *the course material was not of high quality.* | 0 | - | 51 |
| I did not complete this course because *the course was difficult to navigate or I had technical problems with this MOOC-Ed.* | 0 | - | 51 |

Table 18 (continued)

| | Yes | No | Total |
|---|---|---|---|
| As a result of your participation in the Computational Thinking and Design MOOC-Ed, did you acquire any knowledge, skills, and/or resources applicable to your professional practice? | 32<br>69.57% | 7<br>15.22% | 46 |
| Have you applied any knowledge, skills, and/or resources acquired through your participation in the MOOC-Ed to your professional practice? | 23<br>71.88% | 7<br>21.88% | 32 |
| As a result of your participation in the Computational Thinking and Design MOOC-Ed, have you made any change(s) in your professional practice that have directly affected students (e.g. use of new instructional strategy, integration of technology, changes to lesson plan, etc.)? | 14<br>66.67% | 2<br>9.52% | 21 |
| Did this MOOC-Ed offer opportunities to support you in engaging students in *critical thinking and problem solving*? | 12<br>85.71% | - | 14 |
| Did this MOOC-Ed offer opportunities to support you in engaging students in *collaboration*? | 10<br>71.43% | - | 14 |
| Did this MOOC-Ed offer opportunities to support you in engaging students in *self-directed learning*? | 10<br>71.43% | - | 14 |
| Did this MOOC-Ed offer opportunities to support you in engaging students in *developing an academic mindset*? | 10<br>71.43% | - | 14 |
| Did this MOOC-Ed offer opportunities to support you in engaging students in *communicating effectively*? | 9<br>64.29% | - | 14 |
| Did this MOOC-Ed offer opportunities to support you in engaging students in *core academic content*? | 4<br>28.57% | - | 14 |
| Did this MOOC-Ed offer opportunities to support you in engaging students in *other areas not listed in this survey*? | 2<br>14.29% | - | 14 |
| Were there any specific activities, resources, or supports that were critical in helping you apply what you learned to your practice? | 6<br>28.57% | 8<br>38.10% | 21 |

Within the impact survey, participants who answered yes to the question "have you applied any knowledge, skills, and/or resources acquired through your participation in the MOOC-Ed to your professional practice" were asked to provide additional detail. All of the participants who indicated "No" pointed to either a lack of time, or the fact that they didn't complete the course as reasons. Fifteen respondents who answered "yes" provided additional feedback. The feedback was categorized by major theme, which can be found with frequencies in Table 19.

Six responses mentioned deeper integration of technology into instruction, while a few others directly mentioned course content that was integrated. A few selected quotes follow below:

> "In helping teachers, I am able to reach more students. Helping teachers plan more engaging and student created technology changes the learning and thinking."
>
> "I plan to include preparing infographics/videographics as a student project in my stats course."
>
> "I don't think I have at this time but my semester just ended. I do plan to apply the concepts when I analyze what topics for which I need to create additional instruction for students."

In the impact survey, participants who indicated that they have applied course skills in their practice were asked a follow-up question about whether or not they had applied these skills with students. All of the participants answering "No" cited a lack of time, school schedules, and the fact that they don't work with students as the reason they have not been able to implement. Nine participants who answered "Yes" provided follow-up responses. These responses generally grouped into the "4-Cs" (collaboration, communication, critical thinking, and creativity) plus

Table 19

*Participant Response Themes in Application to Practice in Impact Survey*

| Major Theme | Frequency |
|---|---|
| Student or teacher engagement in learning | 3 |
| Focus on content creation | 1 |
| Integration of technology into the instructional program | 6 |
| Integration of general problem solving into the curriculum | 2 |
| Integration of data visualization into practice | 2 |
| Integration of computational thinking into practice | 2 |
| Plan to integrate in the future | 2 |

exposure to the course concepts. The responses were categorized according to major theme, found in Table 20. Selected quotes are as follows:

> "Changing the way you question students to lead them into deeper thinking, giving students time and space for collaboration, and giving students voice and choice for their learning." "Currently I have 7th graders working on STEM projects as a project-based learning activity, and although not all the students are showing the components I clicked, the goal is for them to begin to see these components, and to embrace them."
>
> "In an effort to help students see the personal relevance of the subject matter I teach, as a result of this course, I am trying to redesign my courses to involve more of the deeper how/why type questions rather than the generally more superficial who/what/where/when questions. That should help foster deeper critical thinking about the subject matter."

## Summary of Survey Data

The impact survey indicates that a lack of time is the primary reason that people fail to complete the course, not the course quality or content. From both the impact survey and the end-of-course survey, many participants indicated the course was of high quality and was effective in helping them accomplish their learning goals. A majority of participants gained new skills and indicated that they have made some type of change in their professional practice as a result of their participation in the MOOC-Ed. The course resources emerged from the survey as a useful course component along with more general course content (resources and deep dive content). Teachers indicated that they were using the content with students in the classroom, as well as in their own reflective practice. There were also multiple mentions of empathy and data collection as content that emerged as particularly useful.

Table 20

*Major Themes from Implementation with Students Question in Impact Survey*

| Major Theme | Frequency |
|---|---|
| Changes in classroom questioning techniques | 1 |
| Change in problem solving practices in the classroom | 1 |
| Cross-cultural and global learning | 1 |
| Exposure to course concepts | 3 |
| Communication and Collaboration | 4 |
| Critical Thinking | 3 |
| Learner Agency | 3 |

## Engagement in Practice

At the end of Unit 5 was a discussion forum asking course participants to reflect on their growth throughout the course. Participants were also asked to evaluate changes to their practice in the “end-of-course” survey, the “impact survey”, and the “end-of-unit” survey in Unit 3. All of these items were coded against Clarke and Hollingsworth’s (2002) Interconnected Model of Professional Growth (see Figure 13). Eighty-four (84) items were coded in total. Based on the model and the wording of the prompts, the *Personal Domain* was notated for every post where a participant mentioned something new that they learned. Participants who responded that they were planning to try something new (or were developing something new) were coded as *Domain of Practice.* Finally, participants indicating that they made changes in their practice with students (or adult learners) were coded as *Domain of Consequence*. Because responses were free-response, a response could mention multiple domains. The responses were relatively evenly split between the Domain of Practice and the Domain of Consequence. Half of responses were in the Personal Domain, indicating that the participant did increase their own skills and knowledge relative to the course content. However, 68 (or 80%) of the responses existed in either the Domain of Practice (35) or the Domain of Consequence (33), indicating that participants responding in one of these posts did make actual changes to their practice, if only to try something new (see Table 21).

### Participant Interviews

As a part of follow-up evaluations for the MOOC-Ed, participants who earned a certificate of completion and indicated that they were willing to be contacted were invited to participate in a follow-up interview about the MOOC and their application of the content into their professional practice. Three participants volunteered.

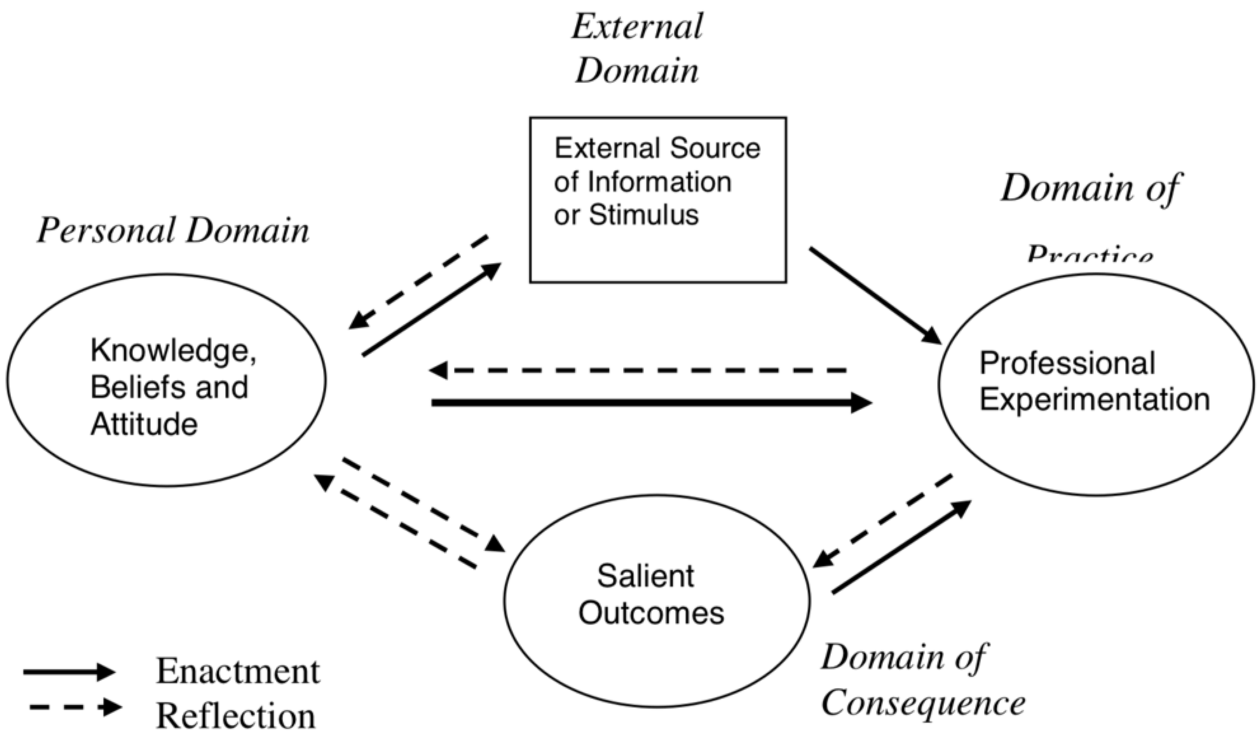



*Figure 13.* Interconnected Model of Professional Growth (Clarke & Hollingsworth, 2002).

Table 21

*Impact on Practice Aligned to Clarke and Hollingsworth's Interconnected Model of Professional Growth*

| Domain | Mentions (n=84) |
|---|---|
| Personal Domain | 42 |
| Domain of Practice | 35 |
| Domain of Consequence | 33 |

The first interviewee is a reading and math intervention specialist at an all-girls charter school in an urban town in North Carolina, recently moving from a technology facilitator position for a rural school district. They were attracted to the MOOC because they had an interest in the course contact and they have been involved with other professional learning programs run by the Friday Institute. They took the Friday Institute brand as an indication of the course quality and usefulness to their work. The course was useful but “not necessarily in the ways that I intended.” They began the course “thinking it was going to be more lesson application. But it was, I think for me, almost a mind shift on change that I found to be a much more beneficial opportunity than I had probably intended when I signed up for the course. Just kind of providing a different lens to view thinking and problem-solving approaches.” This participant did not choose to pursue micro-credentials, citing both time and the fact that she needed to have a group of teachers to work with in her previous role that weren’t necessarily available to her. However, in this role, there are STEM clubs and activities. The interviewee pointed to the resources and materials in the course “the resources, the videos that you assigned, the links and resources, the articles, things like that, are always great for me professionally because there's such a diverse range of materials offered. So it appeals to all sorts of learners. But if I'm having a day where I just need to watch a good inspirational video, I can usually find something smart. Or if I need to be able to print something out and take some notes to use it to do professional development, there's something in there for that as well. So I really do appreciate being able to bookmark all of those sites. The least helpful to me probably would be the micro-credential offerings, just because in my situation I wasn't able to take part of that. It would have been great if I could, but it just didn't work for my people.” As for additional feedback, the interviewee responded:

"I'm hoping to continue to take part in these MOOC-Ed opportunities. But this one to me was unique from some of the others, just because I don't have an engineering background. So I feel like when people talk about STEM, and you don't really know how the E kind of fits into it, just being able to explain that it's a way of thinking. And I go back to the idea of coding. We keep saying we need-- there's this huge emphasis on teaching coding to students. And I don't necessarily think it's that we need to teach students to code. I think we need to teach students to think computationally. And I think that people-- that's what you do when you're coding, but I think people are getting kind of caught up in the idea that we have to have the robot, or we have to do Minecraft, or we have to do one of these things, and get students coding, rather than taking a step back and looking at it in terms of the thinking or the skill that they're using in order to do that. So I think that's what's the most exciting for me."

The second interviewee is also from North Carolina, residing in a large city. They are the instructional coordinator at a magnet middle school, with a new focus on computational thinking and project-based learning. The school has been engaged with the Friday Institute for on-site professional learning services, and a few teachers also took the MOOC to go deeper into the content. She indicated that:

"The course was useful to me because it just gave me a little bit more confidence when talking about computational thinking and just some-- it gave me some examples of how computational thinking is being used out in the community and in different types of businesses, and even things like just concrete examples of different types of thinking, data literacy, things like that, that I could use when talking to other teachers. And one thing that was kind of neat is right about the time I started taking this class, we started doing some things in school with design thinking. Some of our teachers were doing that and one of our business partners, IBM, had us come out and do a design thinking workshop, and it just tied in really nicely with that aspect of your course."

The non-education examples in the introduction videos were also mentioned as useful. "I remember you had one participant who talked about traffic patterns and how they would have to gather data on where traffic was going at a certain intersection before they tried to solve a problem and tried to decide how to approach the problem there. And I remember giving that example to one of the teachers I work with and talking about that as matching up with something that they were doing with their students."

The interviewee did not complete the micro-credentials, as completing the micro-credentials is a part of the school's professional learning plan for the 2017-2018 school year. The school will also be integrating design thinking, and continue to work on integrating computational thinking in the 2017-2018 school year. As this is a focus for this school, the interviewee noted that students and teachers alike are becoming more familiar with the language of computational thinking, and are increasingly using appropriate language in the classroom.

The third interview subject is a technology consultant in the Ohio Department of Education, currently working on writing standards for computer science and digital-age problem solving. In addition to computer science course standards, they are currently working on integrating design thinking, computational thinking, and data literacy into courses from kindergarten through grade 12. Asked about the most useful components of the course, the interviewee responded that the course was valuable in helping validate their own knowledge and solidify understanding around a few concepts. They also pointed to the resources as useful to read, come back to, and share with others. While this participant did not attempt micro-credentials, they did review them, and actually contacted and met with the project lead on micro-credentials from the Friday Institute to explore methods of implementation in the state of Ohio. While the interviewee pointed to the content and resources as useful, the most valuable component for them was examining the approach to online learning in the MOOC, and figuring out how that could be adapted for their context.

**Micro-Credentials**

There were only two participants who earned the Digital-Age Problem Solving micro-credential stack, one teacher and one administrator. The teacher works at a small rural school in Hawaii and teaches sixth grade while the administrator works in a suburban high school in North Carolina.

The teacher chose to engage her students in the process of solving a problem within their school. The students involved were members of the school "Strategy Club", a club that "focuses on building problem-solving skills through games". The problem selected was "at our school, the middle school academy kids wait outside the cafeteria for their homeroom class to be called in for lunch. It is often very noisy and disorganized outside the cafeteria. Lunch monitors are yelling at kids to pay attention." The students began by creating a fishbone diagram attempting to identify all of the possible systems in play in keeping order in the cafeteria (see Figure 13). The teacher then recorded video of student activity in the lunchroom and had students collect data on the number of students in the lunchroom, the noise level, and the number of times students had to be redirected. They analyzed and coded the video as qualitative data, and supplemented it with the quantitative data collected. The students also interviewed other students, administrators, other teachers, and lunchroom monitors to gather their perspectives on the difficulties they have in the lunchroom. Through all of this data collection, students initially had hypothesized that the

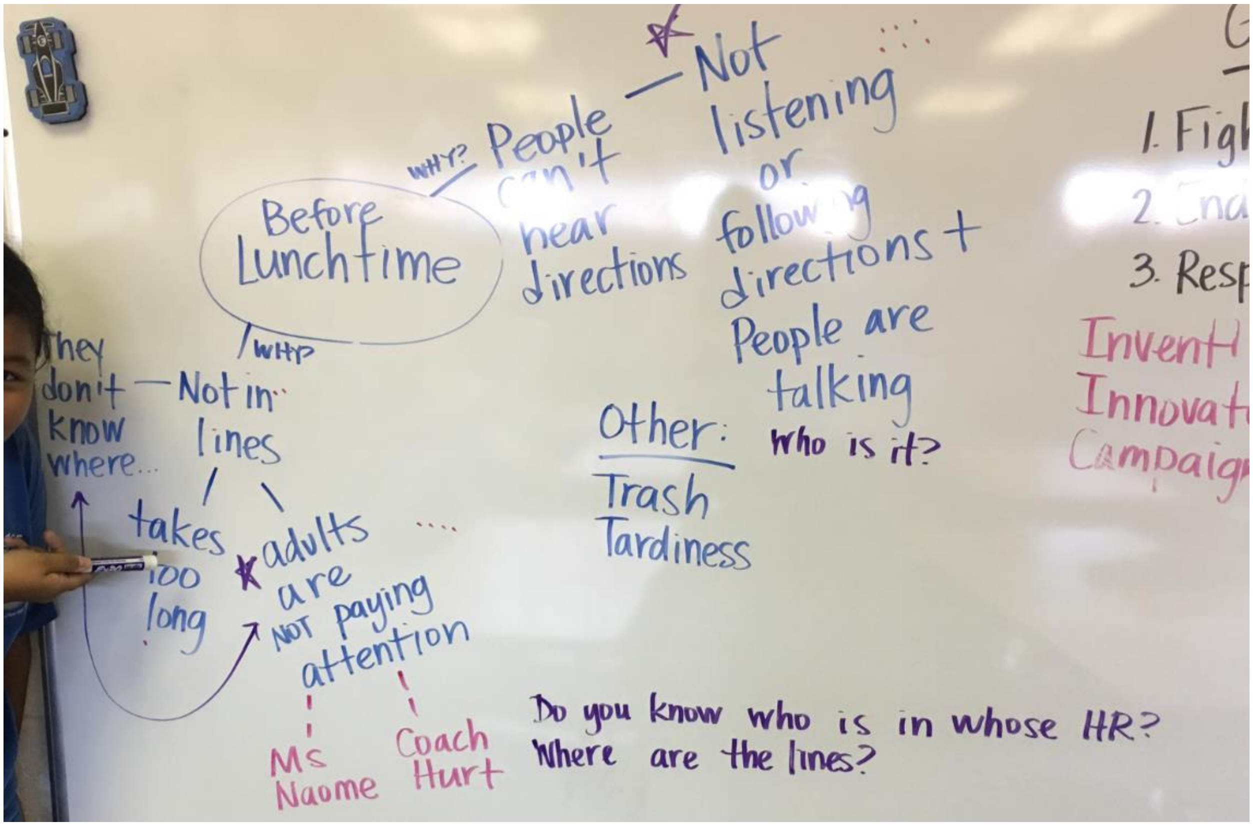



*Figure 14.* Micro-credential artifact: Student root cause analysis.

lunchroom behavior was due to simply to student misbehavior. However, the data collected was able to convince school administration that "there is no system to the lunch lines, which then causes the students (stakeholders) to either get frustrated with the process or take advantage of the lack of structure to achieve their own goals". This also started a conversation among the school staff because "the students were very quick to blame their peers. I believe this is because we as adults, unfortunately, often give behavior consequences rather than have these conversations with kids."

Once these problems have been identified, students began the process of decomposing the problem. They reviewed all of the areas where problems existed, and selected the ones they wanted to address. They iterated and brainstormed different solutions (see Figure 15) before selecting ones to pilot:

> Based on their prototype models, the students came up with the following ways to organize the cones and tape (in addition to giving the lunch monitor a list of homeroom names). In the following days, they simulated each model and collected data and observations.

1. Only set up cones where students are expected to stand behind the cones
2. Cones and tape, where students stand behind the cones and the tape separates the different lines like barriers
3. Cones and tape, where students stand behind the cones and the tape (directly behind the cones) signifies the direction of the line.
4. Just tape (no cones) where the students just stand on the tape.

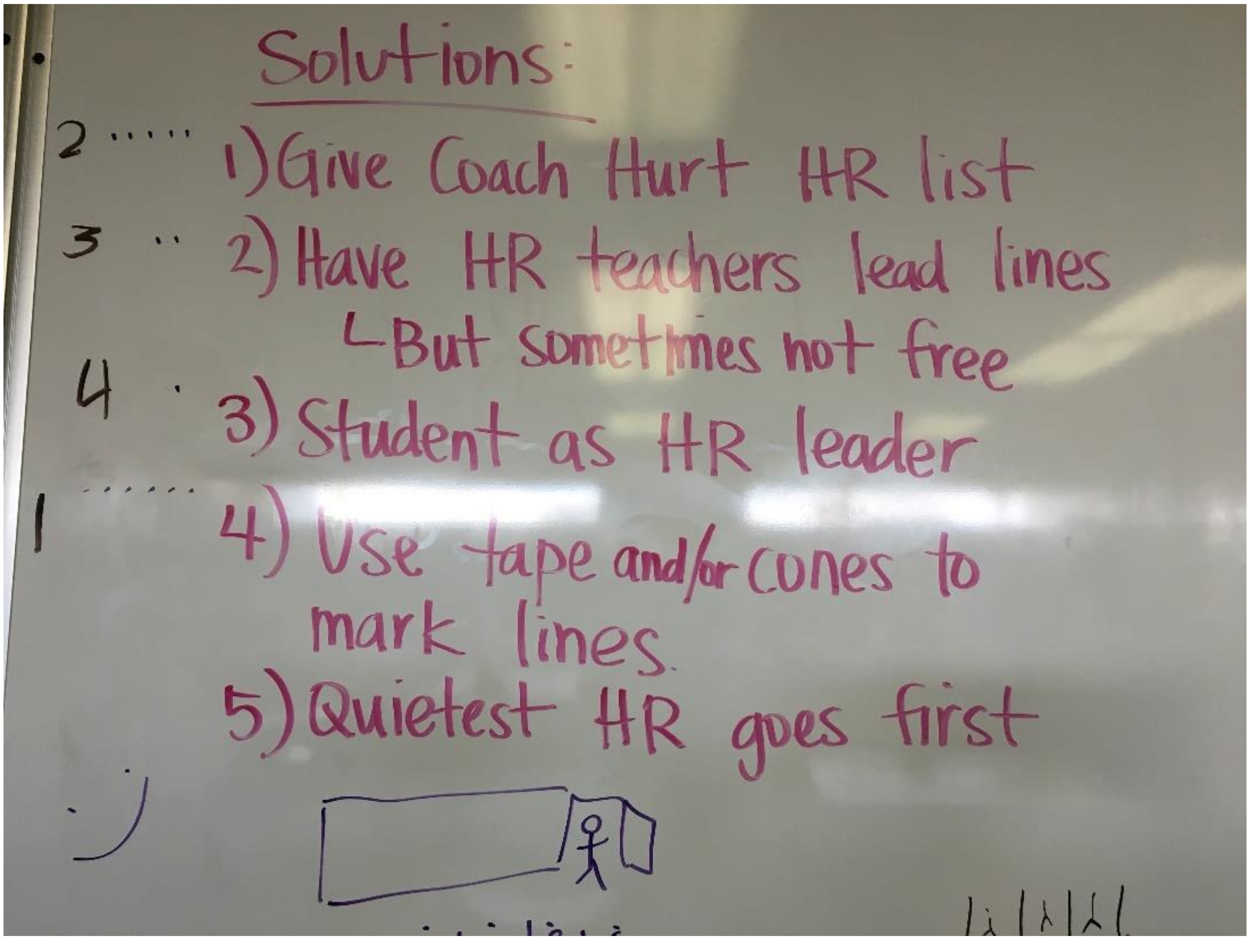



*Figure 15*. Teacher micro-credential artifacts: Student solutions.

5. Cones and tape, where the cones are placed in the mid-sections of the tape.

The first prototype the students tried was #1 since it followed the 2nd requirement that the solution should require the least amount of energy and time. The students decided to use 4 safety cones from the Junior Police Officers Club. This did not work out since the students in the front lined up behind the cones but it did not transfer to the students in the back. The cones only served as markers for the start of the line. It failed the 3rd requirement of being clear to the students.

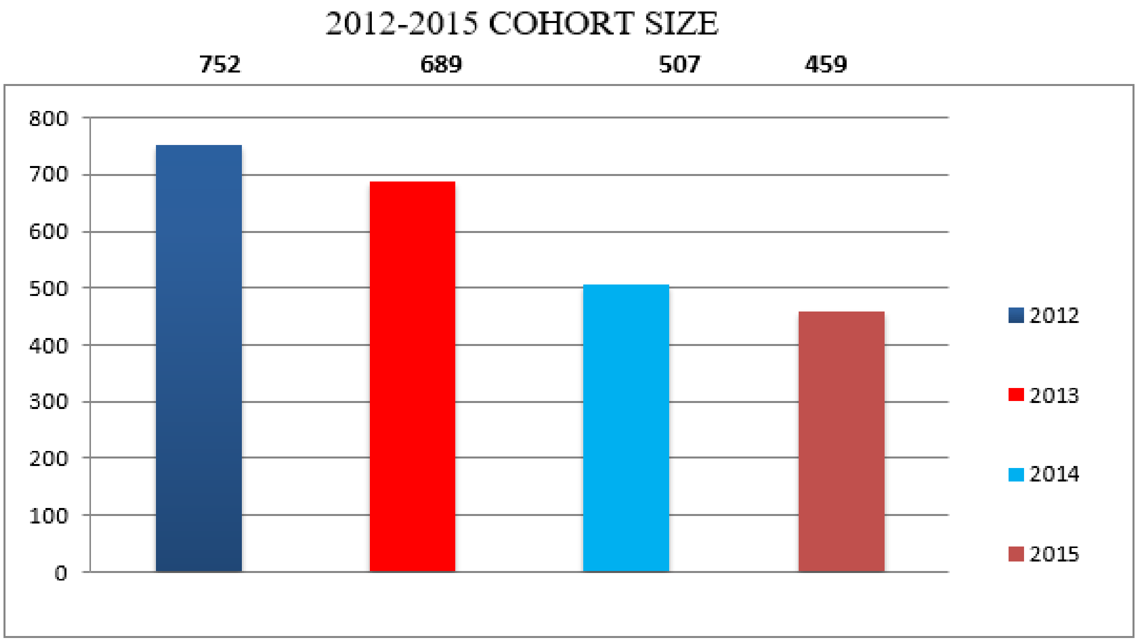
The second prototype followed, because the tape acted as barriers for students to see where NOT to stand. Unfortunately, either the students did not notice them or they did not understand what the tape was meant to do. This model also failed the 3rd requirement.

The third prototype was then chosen to clarify the tape's purpose. This model worked much better. We noticed the amazing effect of lines on the human mind. For some reason, the combination of the cone and tape clicked for the students. Now, the tape signified "line up here", and we did indeed see clearer lines. They were not completely straight lines as the students had hoped, but they were more defined. This is when we discussed whether straight lines were actually necessary. Which is more important: the clarity of the lines or the straightness of the lines? They finally decided that the clarity of where the lines were and which homeroom they were for was more aligned with solving the bigger problem.

The fourth prototype occurred by accident. One day, the students forgot to set the cones. We still made a video to record the data. It worked just as well as the third prototype. Students already knew where to stand based on previously models, and the other students stood right behind right on top of the tape. We noticed throughout this process, too, that since the lines were clearer, there was less line cutting. We noticed that certain students who would cut the lines before, now, raced to get a better spot at the front of the lines. So at the end, the solution that worked the best did require the least amount of time and energy since the tape could be set down at the beginning of the school year.

The fifth prototype was nixed earlier in the iteration process, because it was seen as too confusing. One difficulty was in measuring the 1st requirement. Our classes are not released at the same times, so it was a struggle on when to begin timing the whole lining up process. The students did notice that there was a significant decrease in staff reprimanding students. However, there were still several students who would continue to play basketball and hang out in different non-lunch-designated waiting areas. This indicated to the students that there are other areas of the lunch process that lacked overall structure.

Administrator addressed two separate problems in their micro-credential submission. In the Telling Stories with Data micro-credential, they attempted to identify the problem of "$9^{th}$ Grade Shock", where $9^{th}$ graders make up a disproportionate number of discipline referrals, but also, accounting for withdrawals, aren't graduating. The administrator developed an infographic to demonstrate his point (see Figure 16). For the remainder of the micro-credential submission, the administrator used work from an existing task, creating a school master schedule. He decomposed the problem by identifying all of the variables involved and creating a process to get stakeholder input and to build and test the schedule (see Figure 17).




TOTAL CHANGE = 39%
WITHDRAWALS = 15%
ATTRITION = 24%

*Figure 16.* Administrator micro-credential: Infographic submission.

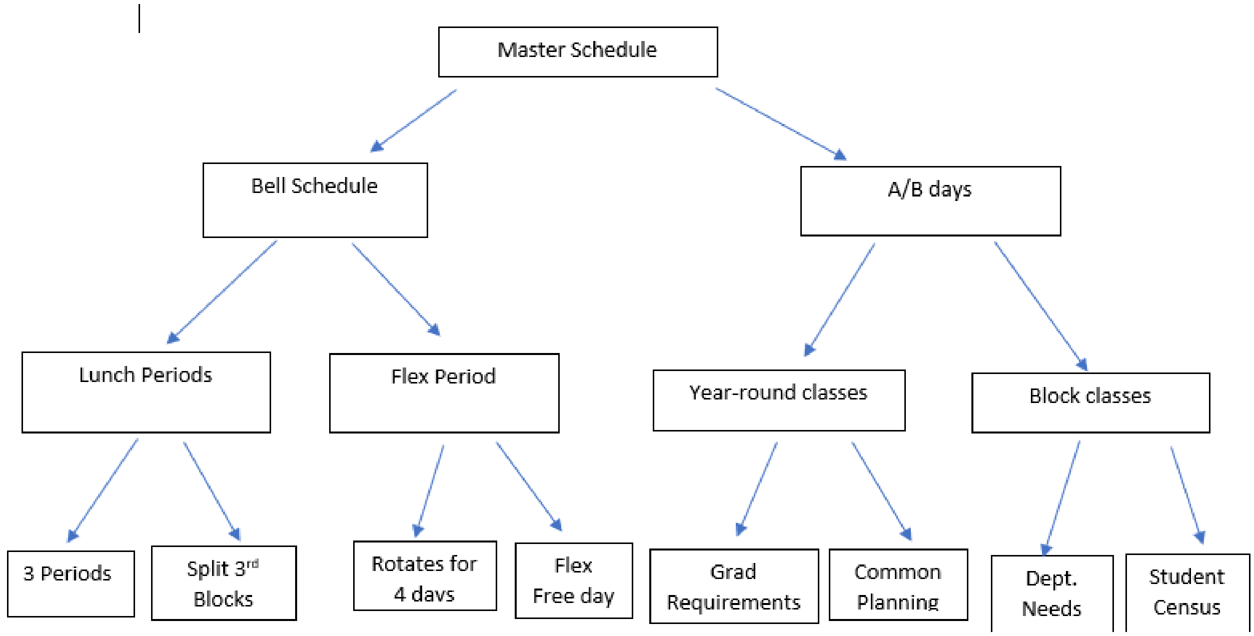



*Figure 17.* Administrator Micro-credential Submission: Decomposition of a school master Schedule.

Gather the departmental schedule analysis team. It consists of: one AP, One teacher, The department chair, a counselor, and the Dean of Students. Follow the Requirements testing plan for each individual departments' schedule as a discrete entity. When the best-case schedule is chosen by the committee, it will be analyzed again for conflicts. It will then be presented as a component for integration testing/review.

Integration Testing Plan:

Combine the individual departmental schedules into a comprehensive master schedule. Distribute the 5 iterations of the complete master schedule to the analysis team. They will follow the protocol set up for requirements testing to develop the most efficient prototype master schedule possible. They will review each of the iterations for the requirements and decide if conflicts arise and what they entail. After several cycles of review, the team will combine the best aspects of each schedule into a prototype master schedule.

The final prototype will be distributed to the administrative team for extensive, iterative requirements testing.

Test Cases:

1. Increase time of transitions to give students more time to socialize and get to class on time.
2. Split 3rd Block to incorporate lunches to ensure each lunch period is uniform and distribute the interruptions.
3. Move the lunch periods so they do not impact 3rd block or any other Blocks.
4. Increase class time to give more instructional time and improve student success.
5. Static Flex period to reduce changes in the routine.

## Summary

From all of the data, there were several themes that emerged. The MOOC was generally viewed as useful for all participants who engaged in it for any length of time, regardless of whether they completed the course or not. While the discussions were useful and participants found the activities with forums engaging, the resources, readings, and explanatory content was viewed as the most useful by participants and are consistently the things that were mentioned in feedback as being the most useful. Many of the participants in the forums were able to identify connections from the content to their current practice, but many also indicated that they made changes to their practice based on their work in the MOOC. Very few people attempted the micro-credentials, most citing a lack of time as the reason.

Research Question 2 considers the usefulness of the conceptual framework as a tool for teachers, and will be addressed in Chapter Five.

The goal of Research Question 1 was to ascertain how teachers are able to use digital-age problem solving in their classroom. The course feedback data produced 84 examples of integration in the field, in addition to the examples shared in the discussion forums along with the micro-credential submissions.

There were no instances where a participant indicated that they could not apply the content to their practice, though a few indicated that they did not yet due to time constraints. Many participants who participated in the unit discussion forums were able to identify points of connection to their existing practice and were able to reframe existing practices through a Digital-Age Problem Solving lens. Many participants indicated that there was value to them in helping them understand and reframe their work this way. Others experimented and made small tweaks within their contexts. There were a few participants who denoted significant wholesale changes to their practice. Some used the digital-age problem solving framework as a reflective tool on their own practice. Many also mentioned including more infographics, data storytelling, and empathy mapping throughout their instruction as well as changes in questioning techniques.

Research Questions 3 and 4 were more focused on the MOOC-Ed course itself. Both the survey data and the course interviews indicate that the MOOC-Ed was helpful in helping participants improve their understanding of computational thinking, design thinking, and data literacy (RQ3). 85% or more (based on specific concepts) of the course participants in the end-of-course survey reported that they improved their own knowledge or skills related to the various habits of mind addressed within the course. 95% of course completers reported that the course was effective or very effective in helping them meet their learning goals. The Impact Survey, which included non-completers also indicated nearly 70% of participants acquiring new knowledge or skills in their professional practice.

There were not any elements of the course that stood out as ineffective based on user feedback (RQ4). The course introduction pages, while still rated highly by participants, were the lowest rated of all of the elements. Additionally, several participants mentioned that the lack of education-specific examples in the introductory videos and course activities made them hard to relate to (while several others mentioned specifically that they found the non-education examples particularly valuable). The course resources were very highly rated and mentioned frequently in the open-text feedback. While resource accesses were low, the data are incomplete as to how many people specifically used them. The "deep dive" activities were also very well received and specifically cited as a useful "quick reference" sheet. Participation is the only requirement to earn credit in the course. While many people participated, the forum comments are mostly unidirectional in nature, people sharing out versus engaging in productive discussion. However, there were many great classroom examples shared in the classroom application forum, which were called out explicitly in the feedback and the interviews. The forums were also good indicators of content mastery and did help validate participant understanding of the course content.

# CHAPTER 5: DISCUSSION AND IMPLICATIONS

This chapter aligns the findings in the previous chapter to the four research questions, and re-examines the original frameworks proposed in the first two chapters based on the findings from the study. The chapter then examines consistencies with the literature and culminates with implications for practice, policy, and research.

The data for this study was gathered through the offering of a Massive Open Online Course (MOOC) for Educators through the MOOC-Ed program at the Friday Institute for Educational Innovation at NC State University. The MOOC-Ed, titled *Computational Thinking and Design* focused on integrating the design thinking framework (Hasso Plattner Institute of Design, 2013), computational thinking (ISTE & Computer Science Teachers Association, 2011), and data literacy (Gray, Bounegru, & Chambers, 2012a) to help teachers understand how to identify and solve ill-defined problems in the digital age. Spanning eight weeks in the spring of 2017, nearly 500 participants enrolled in the course while approximately 200 actively participated. The previous chapter analyzed the demographics of active course participants and followed their path through the course to determine which course elements were most useful. The chapter also looked at forum posts from course participants along with end-of-course survey results to identify professional growth and changes in practice. Additionally, analysis of interviews along with micro-credential submissions were used to further analyze the impact of the MOOC-Ed on participant practice.

## Summary of Findings

The purpose of this study was to evaluate the following four research questions:

- RQ1: How are educators able to integrate digital-age problem solving into their instructional practices?

- RQ2: To what extent is the digital-age problem solving conceptual framework a useful tool for teachers?
- RQ3: How useful is the MOOC-Ed in strengthening participants' understanding of computational thinking?
- RQ4: What elements of the MOOC-Ed were the most helpful for teachers?

For clarity of presentation, this chapter will review the MOOC-Ed as a delivery mechanism (RQ3 and RQ4), before reviewing the outcomes of the MOOC-Ed content (RQ1 and RQ3), and finally will examine how the application to practice reflects the digital-age problem solving conceptual framework (RQ2).

**MOOC-Ed Effectiveness**

The surveys administered to MOOC-Ed participants along with the interviews conducted suggest that the MOOC-Ed was valuable in developing understanding of digital-age problem solving and bringing elements of digital-age problem solving into instructional and professional practice (RQ4). Participants reported the resources and the *digging deeper* activities to be the most useful, while the forums were reported as the least useful. The *digging deeper* activity summarized the key concepts across design thinking, computational thinking, and data literacy, and synthesized them in to a single guide. Each unit had two forums, one to present a scenario for discussion, and the other to discuss application to classroom practice. It should be noted, however, that "least-useful" in this context still means that participants did find the forums highly useful, just slightly less useful than the other things – at least 85% of the participants surveyed found the various elements of the course useful.

Qualitative analysis of participant interactions, survey data, and course interviews suggests that the digging deeper pages and the course resource pages were helpful in supporting understanding of new content an in presenting a unified framework. The discussion forums did not always feature high levels of interaction among the participants, but was consistent with other MOOC-Ed courses and similar discussions (Gunawardena et al., 1997; Kellogg et al., 2014). Participant posts were useful in helping them process their existing practice through the lens of digital-age problem solving while helping them brainstorm ways to adjust and update their practice.

Overall, 95% of participants completing the end-of-course survey (n=40) responded that the course was effective in supporting their professional learning goals, while 97% responded that they are prepared to make positive changes in their professional practice. On average if 95% of participants completing the end-of-course survey responded that they have gained new knowledge from this course. With respect to computational thinking (RQ3), 95% of participants responded "agree or "Strongly agree" when asked "As a result of my participation in this MOOC-Ed, I have improved my knowledge and/or skills related to computational thinking." With respect to other course competencies, 97% responded in the affirmative with respect to design thinking, 88% affirmative on "the role of empathy and data collection in the design process", and 92% affirmative on "the integration of digital-age problem solving strategies into my instructional practice."

**Translation to Practice**

Micro-credential submissions, online forum posts and course evaluations indicate how course participants were able to integrate course content (or were already integrating content) into their instructional practice (RQ1). While many course participants shared how they were integrating computational thinking, design thinking, and data literacy into their instructional practice, there were also frequent mentions of these skills being used for reflective practice or for changing how teachers interacted with their own students (using design thinking-based approaches in lesson planning and collecting feedback from students). As digital-age problem solving is a collection of skills and mindsets that are applicable across industries, it is logical that teachers would be able to assimilate or identify these strategies in their own practice as well as teach them to students. When applications to practice from the final course forum and the end-of-course surveys were coded against Clarke and Hollingsworth's (2002) Interconnected Model of Professional Growth, half of the responses analyzed specifically pointed to a change in participants' reflective practice (personal domain). Of the 84 responses collected, 41.7% mentioned changes that have occurred directly in the classroom (domain of practice), while 39.2% reported sharing their learning from the MOOC-Ed with colleagues and peers (domain of consequence).

From the discussion forums, many of the applications of computational thinking suggested by the course participants mirrored the types of applications suggested by Barr and Stephenson (2011) and by ISTE's operational definitions for computational thinking (ISTE & CSTA, 2011). Much of the discussion around engaging students in the design thinking process referred to both productive failure (Goldberg & Nemcsok, 2015) and supporting the development of empathy as a vehicle to better understand how to design solutions that were human-centered (Carroll et al., 2010). Teachers noted that they could change their approaches to data collection and employ purposeful empathy to better collect and understand student learning. Both examples of helping students understand and solve real-world scenarios and applying digital-age problem solving to school administration were also validated by the interviews and micro-credential submissions. While data literacy is heavily discussed in the literature, and there are case studies on the creation of lessons where students are involved in the design thinking process, the literature on educator's use of design thinking individually or in professional learning teams to create lessons tends to be more anecdotal in nature (blog posts and small case studies).

## Digital-Age Problem Solving as a Conceptual Framework

RQ 2 focuses on whether or not the conceptual framework of digital-age problem solving (see Figure 18) and the digital-age problem solving cycle (see Figure 19) are useful tools for educators. The digital-age problem solving cycle (see Figure 19) combines the computational thinking elements from the ISTE definition of computational thinking (ISTE & Computer Science Teachers Association, 2011), the Stanford d-School design process (Hasso Plattner Institute of Design, 2013), and the scientific inquiry and engineering design process from the next generation science standards (Committee on a Conceptual Framework for New K-12 Science Education Standards, 2012). While computational thinking, design thinking, and data literacy are all separate concepts within the STEM fields, many elements overlap or support each other. Design thinking in particular also has roots outside of STEM in the social sciences (Rittel & Webber, 1973) as well as in arts and architecture (Kolko, 2015).

The creation of the MOOC-Ed *Digging Deeper* pages did indicate that there is a good fit between the design thinking, computational thinking, and data literacy concepts in each unit, and that the combination of these three frameworks can be merged in a cohesive way. The evaluation of the course indicates that the digging deeper pages were useful for course participants and were helpful in understanding the course material.

In reviewing feedback from course participants, they were able to integrate computational thinking, design thinking, and data literacy seamlessly in their posts. For example, this post from the final forum in the course weaves together the importance of the design process, testing techniques (a computational thinking skill), and data literacy:

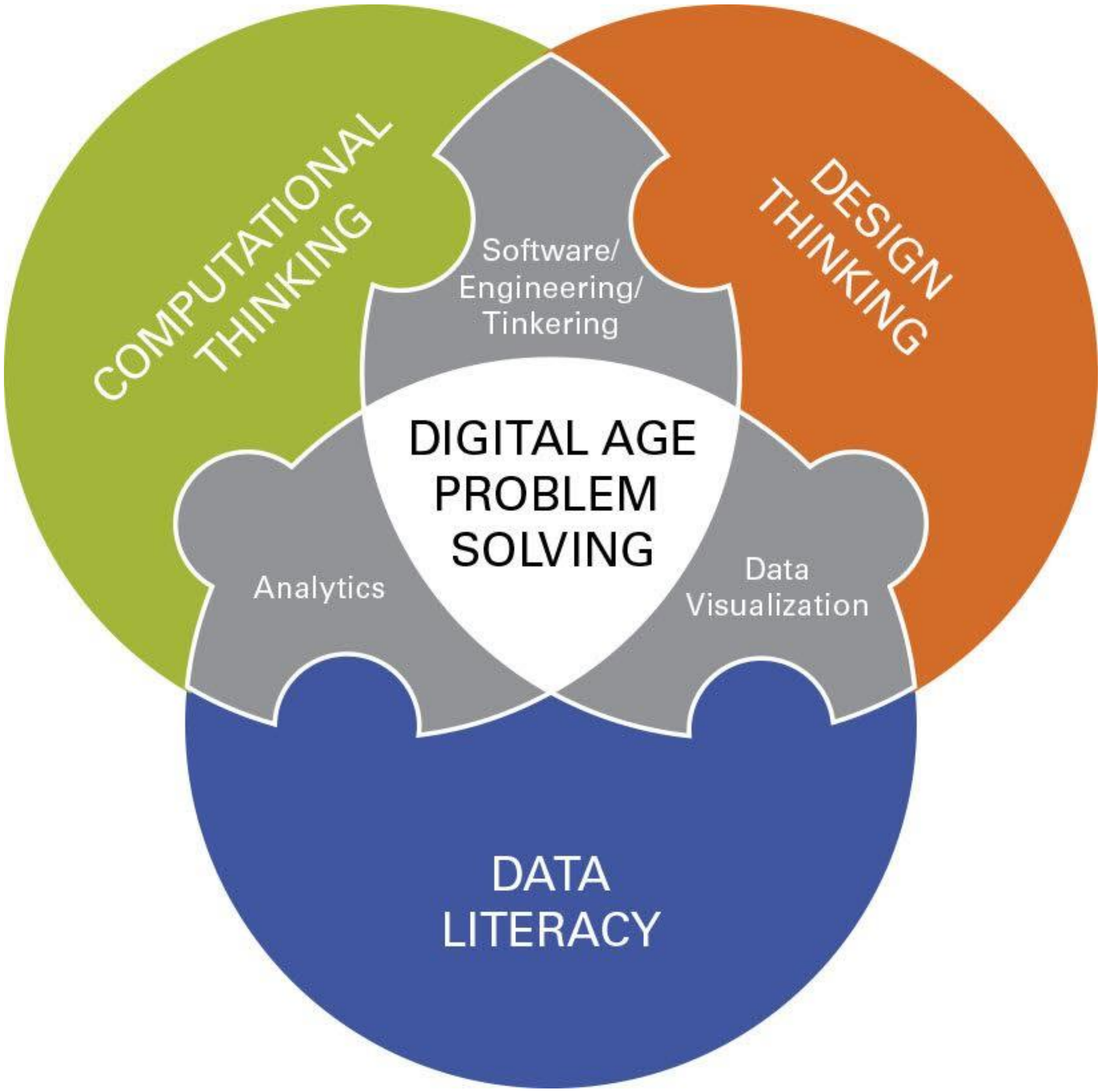



*Figure 18.* Digital-age problem solving.

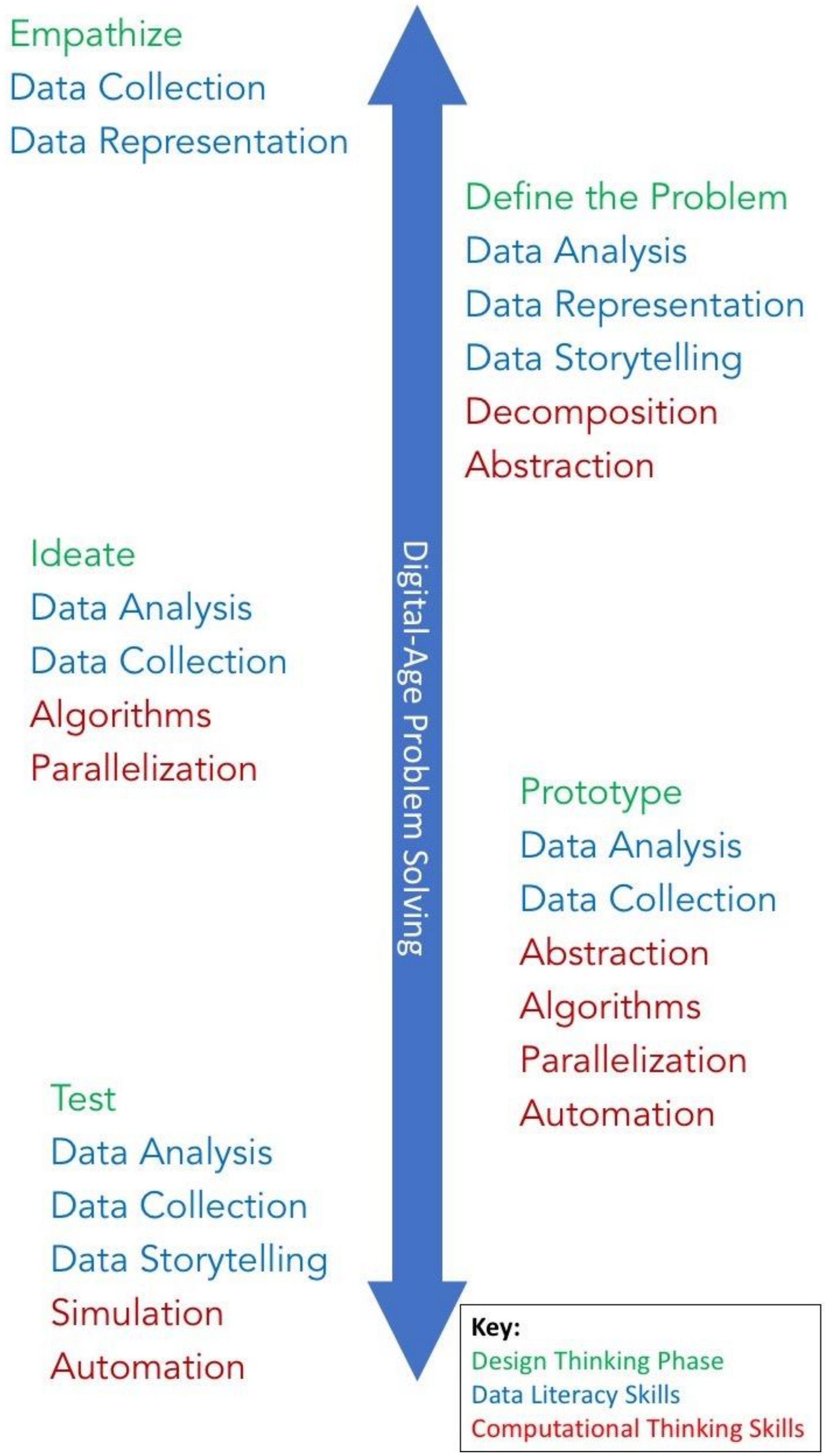



*Figure 19.* Digital-age problem solving cycle.

The creation of the MOOC-Ed *Digging Deeper* pages did indicate that there is a good fit between the design thinking, computational thinking, and data literacy concepts in each unit, and that the combination of these three frameworks can be merged in a cohesive way. The evaluation of the course indicates that the digging deeper pages were useful for course participants and were helpful in understanding the course material.

In reviewing feedback from course participants, they were able to integrate computational thinking, design thinking, and data literacy seamlessly in their posts. For example, this post from the final forum in the course weaves together the importance of the design process, testing techniques (a computational thinking skill), and data literacy:

> It's interesting how a course in Computational Thinking isn't really about math, it's about design and looking at all the parts of the design process from conception to implementation. That process is a complex one (not complex in terms of difficult to understand, but rather complex in terms of many parts). Key takeaway for me included the "Building a Testing Plan" comprising 5 different types of testing and revision required before final implementation. One particular quote really showed the contrast between design thinking and our educational high stakes testing: "in design thinking…designers use testing to refine their ideas and products and create better outcomes. Testing helps us identify and improve our systems — a "failure" does not indicate a deficiency or inadequacy of the system, but rather an opportunity for making the system better. Additionally, in all digital-age problem-solving contexts, testing and evaluation is formative — it should be done early, often, and consistently." With high stakes testing, there's no feedback till months later. How are students supposed to

improve when the feedback is so late in coming and makes little difference vis a vis individual student performance?

I also appreciated the unit on "Identifying Problems." One important idea shared in this unit is that problem solving requires that we look at "context" as well as symptoms...."that we understand the people and the systems and the stakeholders involved and their needs from whatever solution we're going to create and their understanding of the problem as it exists for them." In education, there are no quick fixes. The data we collect to determine next steps should be both quantitative and qualitative. If you only crunch the numbers and neglect to talk to the people involved, you capture only a partial picture.

## Implications

### STEM Education

Bybee (2013) articulated a STEM evolution, from STEM 1.0 where subject areas exist in isolation, to STEM 2.0 which integrates two STEM subject areas together, STEM 3.0 which integrates three subject areas together and which STEM 4.0 integrates all four subject areas together. With many schools deciding to transform STEM into STEAM (Land, 2013) to include the arts, digital-age problem solving can provide a foundation for STEM 5.0, where STEM skills and the Humanities are blended. Google has spent many years analyzing data on effective teams and management structures within their organizations. Their conclusions point to a need for a STEM 5.0 evolution. Despite the swell in schools around STEM education, Google found that:

> "Project Oxygen shocked everyone by concluding that, among the eight most important qualities of Google's top employees, STEM expertise comes in dead last. The seven top characteristics of success at Google are all soft skills: being a good coach;

communicating and listening well; possessing insights into others (including others different values and points of view); having empathy toward and being supportive of one's colleagues; being a good critical thinker and problem solver; and being able to make connections across complex ideas" (Strauss, 2017).

Many of these skills were identified at the very beginning of the STEM education movement, and became part of the Partnership for 21st Century Skills Framework for 21st Century Learning (Parnership for 21st Century Learning, 2015). The P21 framework, along with resources such as the National Academy of Engineering's grand challenges (National Academy of Engineering, 2008) are increasingly recognizing the importance of both soft skills and humanities in a well-rounded STEM education. Digital-age problem solving could be part of a larger conceptual framework for STEM 5.0 – where STEM content, Humanities, and the so-called "soft skills" come together to form a new way of looking at the world. (see Figure 20).

Moreover, while much of the literature on computational thinking focuses on preparing students to code, approaching computational thinking a problem solving context allows it to be a useful tool for teachers, students, and professionals to understand and solve problems whether coding is required or not. Many future problems will require a knowledge of coding. However, many more will not. Part of digital-age problem solving will be about knowing the difference and identifying when and if a computer can do something faster or more efficiently than a human (and following, how to make the computer do said task). In the MOOC-Ed, these ideas were put into practice and digital-age problem solving techniques were useful across disciplines, including humanities and social sciences.

Consider these three examples from the forums, the first one from a therapist in Singapore, who primarily works with adults:

“My adult learners typically come in because something or somebody is causing pain or confusion. When emotions are strong and overwhelming, making the problem go away might seem more urgent than solving it.

So I mostly start by engaging emotionally in priority to engaging intellectually. They are paying for the session, so mostly they need no encouragement to talk about what is happening. As they talk, potential issues and problems and solutions emerge.

Without interrupting, I might start writing these on the board, just bullet points, maybe a mindmap sort of format. I try to avoid making a list, so that new links or insights are easier to make.

When they feel calmer, when they finish telling their story, we often find that we are half-way to having a root cause analysis. Up to that point, I have mostly nodded and grunted and encourage them to keep going.

Now I start to clarify, check my understanding, ask questions, like "what if", "what makes the difference?" I get them thinking about what has happened, what patterns they see, and how it could be different.

So decomposition and abstraction go hand in hand when we follow that sort of process.

But some clients want to rush into action, fix it fast, get rid of it, not talk about it. If that is the scenario, then I might join them by stepping back from the immediate problem and looking at the context, the "upstream." As before, I write on the board a lot, maybe create a flow chart by getting them to run me through things step by step. I ask questions and clarify and check a lot more actively. And then we are typically at a similar stage as the more emotional clients.

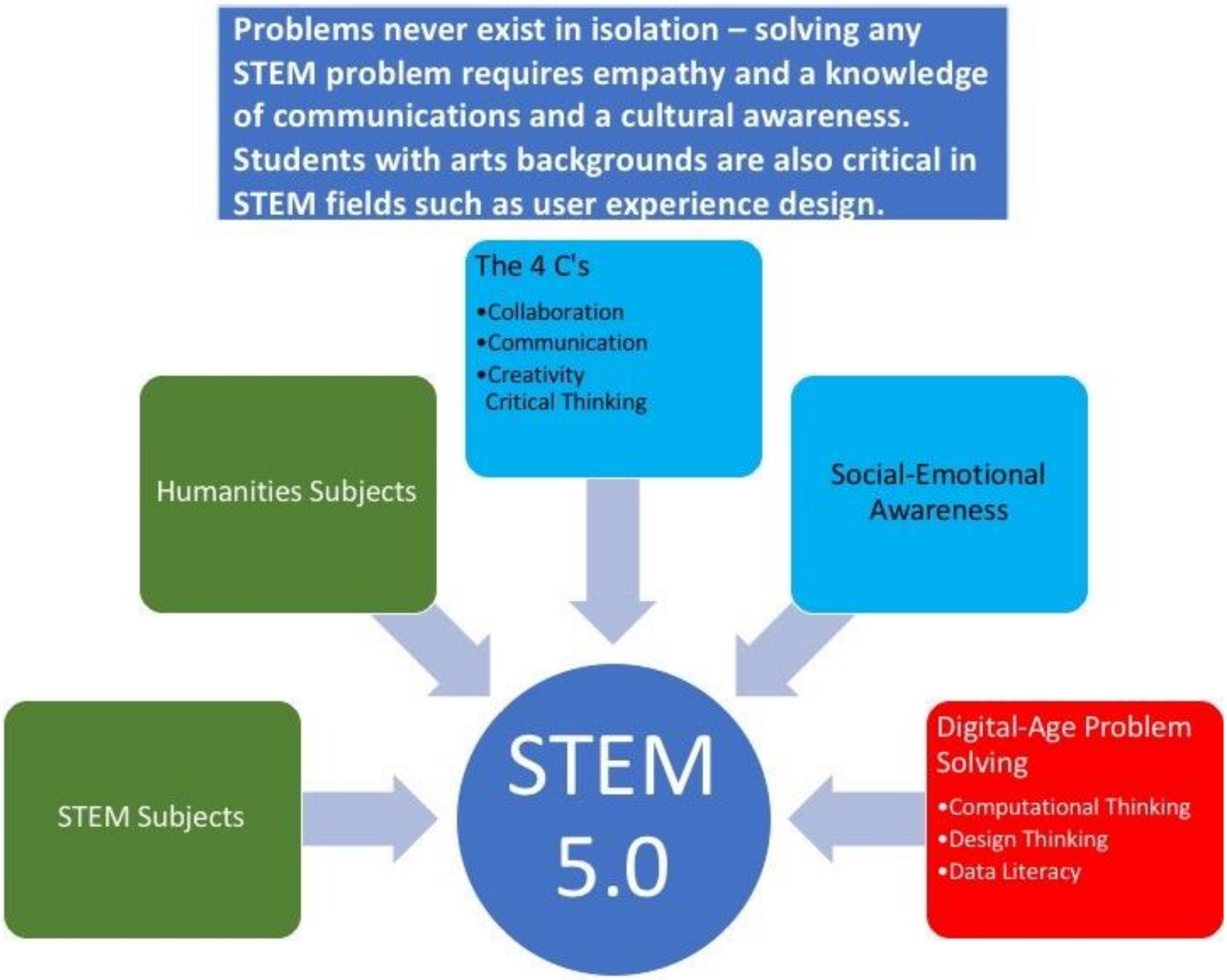



*Figure 20.* STEM 5.0 Conceptual Framework.

Sometimes we seem to have done a good job of identifying the problems. Sometimes we find ourselves going deeper inside, or back into the past, or needing more data. Mostly an action plan has emerged. If it hasn't, I'm not shy to say I don't know what they should do, but help them to figure it out. Often it's reassuring for them to discover that they are not stupid for not figuring it out!

Anyway, action is always their initiative and responsibility. So everything that happens in a session is oriented to the client and their own process. Not having a plan is also okay. Our discussion has almost always increased awareness, so their experience until we meet again will be different. If they come back, they'll be a different person, and we'll take another step."

Consider this second example from an elementary school teacher:

"We started the year with how-to/procedural writing with my 2nd grade students. As I was looking at the Core Resources, I read the Tynker blog about how to talk to children about algorithms. I immediately realized that I wish I owned a magical time machine to go back in time and use the language properly with students, to make these strong connections between their writing and skills that are most often explicitly taught in computer sciences/math/science. I wish that I had used the computational language of algorithm, repetition, sequencing, and conditional logic, as suggested in the article. I think that there might be a way to revisit this learning to teach this vocabulary and to make visible the design thinking inherent in their writing, but if I don't see a natural way to revisit, it's probably worth me going back and having them do a short writing piece to help them see the language between the work they've done and this important language."

And this third example from a technology director:

“The concept of distinguishing "what is needed" vs "what is wanted" was introduced in regard to defining and decomposing problems. An example that I can share was from a few years ago. A district that I was working with was trying to solve a complex issue of parent funding both in terms of payments for student lunches, to buying school related products, as well as paying for school fees. As the Technology Director I was asked to chair the process so I began by organizing a problem solving team that consisted of all stakeholders: students, parents, teachers, secretaries, administrators, cafeteria staff, technology staff.

Each team member had their own stake in the problem:

Teachers were tired of managing the collection of lunch money, field trip payments, book sales in the classroom, etc.

Principals disliked how financially focused tasks were impeding on instructional time

Secretaries hated the task of having to contact parents to collect fees

Cafeteria staff were faced with the dilemma of providing food to students who didn't have $$

Business Administrators struggled with collecting past debt, completing required state and federal reports, and assisting parents who needed support

Students who didn't have the money didn't like being left out

Parents found the multiple systems in place difficult to navigate and hassle to use

And tech staff definitely were looking for a way to streamline the systems they needed to manage and maintain.

With this many voices, each with their own perspective, decomposing the problem took some time.  Abstracting the non-essential issues was also critical to the process as I am

> sure you can imagine the numerous non-essential topics that could distract us from our problem solving goals (students who didn't like the food, teachers who didn't like the class schedule, principals needed more staff, parents thought the fees were too high, and so on). After many meetings and lots of conversation the group was able to outline a set of specifications that met the needs of all the stakeholders, and surprisingly was able to find a solution that could fulfill those needs. Thankfully the vendor (the district ended up selecting PushCoin, Inc) was adept at using the strategy of "separating needs from wants" so that changes were implemented in a timely, streamlined fashion.
>
> Although it took almost 3/4 of a school year to work through the process, the decomposing of the problem and involving all those impacted was critical to the success."

Each of the three examples presented provides a valid and productive use of the three core areas of digital-age problem solving in a context that isn't directly STEM related. For school leaders, STEM 5.0 and digital-age problem solving is already found in many schools through the use of project-based learning. For schools, much as the PBL core tenants are defined through the literature and by various supporting organizations, and many schools use common standards for writing, research, digital-age problem solving could provide schools with a common approach to problem solving, especially if schools are already heavily invested in a STEM focus or in PBL. Creating a unified framework for STEM 5.0 that includes digital-age problem solving will support teachers in creating STEM 5.0 experiences for all students without burdening teachers with "one more thing". For STEM schools, it also helps support the importance of STEM without diminishing (and in fact elevating) the role of the arts and Humanities. The emphasis on empathy in design thinking is also an important component of social-emotional awareness which further supports critical thinking (Arslan & Demirtas, 2016) and the experiential and empathy components of the framework have roots in Indigenous Ways of Knowing (Barnhardt & Kawagley, 2005)

One recommendation in this area is to translate the digital-age problem solving framework into a more student-friendly terms, similar to how The Launch Cycle (Spencer & Juliani, 2016) does for the design thinking process. Additionally, referencing the different types of testing common to computer science (integration testing, unit testing, regression testing, requirements testing, functional testing) generated good discussion in unit 5 of the MOOC, and may deserve a more prominent place in the framework.

**For Future Research**

While the survey data and analysis of the discussion forums indicated that the course content was useful in helping teachers understand concepts of digital-age problem solving and integrating them into their practice, further study is still needed. A series of classroom observations, over time, would be helpful in understanding how these changes unfold over time and how student thinking, problem-solving ability, persistence, and empathy change over time. Research and development is also needed to develop, refine, and support schools in the implementation of a STEM 5.0 framework. Moreover, a major focus of the MOOC was to help teachers reframe some existing practices using the language of Digital-Age Problem Solving while simultaneously encouraging teachers to reframe how students approach problem-solving. However, misconceptions may exist in participant understanding of how these concepts are operationalized in practice, or a teacher may use terms in their classroom without making any substantive changes to practice. Further study is needed in this area.

While MOOCs are a proven tool for professional development, further experimentation is needed to determine how to generate more sustained discussions in courses. While discussion posts were high quality, most of them were unidirectional – participants “shouting into the void.” While some changes were made to the course to force participants to interact, the transient nature of the MOOC meant that participants oftentimes did not return back to old posts, so the quality of discussion was limited. Additionally, while micro-credentials were considered useful by the people who completed them, very few participants engaged with the micro-credentials, even though the projects and activities described in the forums would probably have earned one. Further research should be considered on encouraging teachers to earn professional development credit using micro-credentials beyond requiring them for course completion, as other Friday Institute MOOC-Ed courses have done.

Additionally, while computational thinking has largely been the purview of the computer science course, further research is needed to determine if wide scale adoption of computational thinking principles can impact student self-efficacy in computer science courses, and if a focus on digital-age problem solving with underserved populations can encourage these students to pursue careers in the STEM field.

**Revisions for Future Courses**

The second run of the MOOC-Ed started on October 2, 2017. The course remained largely the same from the first round to the second. The primary change between the first run and the second is the unit extension activities. While the discussions in the extension activities were very good, I tried to restructure the activities to promote better discussions. The current content of the course can be found in Appendix F.

In Unit 2, I originally presented the course participants with a series of infographics and gave them an open discussion space to share their reactions to the visual appeal of the infographic and how well the infographic told a story with data. The discussions were disjointed because each thread contained reactions to three or four infographics which other participants may not have seen. In my opinion, it was hard for the discussions to evolve and for participants to develop an understanding of why each infographic was selected. In the fall run, I created a discussion thread for each infographic, and participants all contributed to that thread about a single graphic. Overall, I think this was a positive change. There was more engagement and co-ideation on the various qualities of the infographic. There was one in particular, focusing on a land bank. Many participants found the infographic hard to follow. However, one participant had a background in the topic, and with her comment, was able to generate a great discussion about the importance of context and audience for infographics:

> "I found this infographic to be interesting. I also reviewed the posts others have put up, and think that this may be of interest to me because I am involved with a local Community Supported Agriculture farm, and I have an understanding of the topic they are discussing. The land bank grows food, and the types of foods it grows are shared. Each year the percentages may change a bit, but in general there is land for forage, cereal, legumes, potatoes and vegetables.
>
> This is a large farm, so they also share more specific information on how it is used. Along with the number of sheep and cows, they share the hectares (a hectare is the equivalent of 100 acres, or 10,0000 meters) are used to grow vegetables and herbs, are used for feed, and are used for crops and legumes. This portion also shares how much is used for buildings related to farming, as well as the number of machines they have.

Clearly this is more information, but I will stop here. I think that this works for me because of my background, and I like that it uses images with very few words."

The next change was in unit 3. I really liked the original activity, but it didn't resonate with participants the way I wanted it to. The activity focused on a space shuttle disaster. While the question was worded to have participants focus on the importance of gathering detailed requirements up front, much of the discussion focused on systems testing. Testing was covered in unit 5, which had a failure analysis of the original Healthcare.gov website. Given the recent political upheaval with the Affordable Care Act, I wanted to remove this activity from the course to prevent a politically-charged discussion. Therefore, I removed this activity and replaced it with the space shuttle activity. In unit 3, I created a new activity on voting machines, and on all of the audiences who would need to be satisfied by electronic voting machines. This discussion was much higher quality, though I did struggle to get participants to explore the requirements of the voters themselves (further revision may happen for the Spring of 2018). However, there were a lot of very good discussions in this post, and several participants indicated that it was helpful in getting them to more deeply understand the content.

"I find myself standing back from the task, just to look at the context. I think this is a brilliant project, which could be a complete education all in itself. The context involves history, politics, psychology, sociology, philosophy, technology, science, security, geography, time management.

Being charged with designing a better voting machine is a guaranteed nightmare.

The user experience is primarily emotional. Hope, fear, anger, sadness, and joy are all part of the voting process.

Because the voting process is inclusive, the typical technology is lowest common denominator, making it as simple and clear as possible, however boring that may be for some. An inclusive system like voting cannot afford to be confusing for any. (Unless one starts out by wanting to rig the results.)

So the more people I can talk to the better. Ideally, I would want to talk to everybody, because everybody is involved. That is impossible at anything about village level, so it becomes a technical issue. And that means we miss the wonderful opportunity to debate the context and meaning. I feel tempted to give up in despair.

But in class, we have that opportunity. So I would work with my students to identify what voting is, what we could vote for, how we would vote, how to design the voting machines. A rich process. It might not appeal to everybody. Some students might want to shut down discussion and just get on with it ... I might feel the pressure of timetable or deadlines ... let us talk about that. Potentially a series of good life lessons.

Please, Mark, let the politics and the history be a part of the design process. I think it is important on so many levels. Brilliant."

In February of 2018, the course will launch again for the third time. I will be more deeply exploring data from the fall run, which concluded on December 22, 2017, to explore any additional course changes that may be needed. Early exploration of the survey results looks very similar to the first run of the course. The primary change for the spring will be a name change for the course. *Computational Thinking and Design: Getting Started with Digital-Age Problem Solving* will be renamed to *Digital-Age Problem Solving: Getting Started with Computational Thinking and Design*. The reason for this change is for marketing purposes – while computational thinking is an educational trend, the term has stuck mostly at the policy and administrative levels. Many teachers still do not know about computational thinking and may not immediately see the benefit to their instructional practice. Digital-age problem solving will hopefully erase those barriers and will attract a wider audience of teachers with less marketing effort. Registration numbers from the first two weeks of registration for Spring 2018 are nearly double that of previous semesters, and early observational data indicate that move was successful.

The second change will be that the course has transitioned to self-paced instead of week-by-week. There is a firm start date and a firm close date, and participants are being with a recommendation for pacing as they move through the course. However, participants can move ahead at their own discretion. The hope is to capture participants and help them engage in the whole course early while their attention is focused on the course. Since lack of time is often cited as a reason participant engagement wanes, the goal is to enable them to finish the course before life gets in the way.

The third change is the development of PLC guides. A PLC guide guides adapt the course content into a series of facilitated activities for face-to-face delivery. Therefore, an educator who participates in the MOOC-Ed has the resources that they need to begin to spread the course content within their schools. PLC guides have three main goals: to enable teachers who are not participating in the MOOC-Ed to gain an understanding of digital-age problem solving, to build a support network of teachers within a school who are integrating these practices into their instruction, and to provide concreate examples of what digital-age problem looks like to support high-fidelity implementation.

## Reflection

Over the past few years, the Hour of Code has become increasingly popular, and state after state has rushed to pass laws mandating computer science education. While I am a computer scientist by background, I worry that we run the risk of turning coding into "one more thing", or we are asking coding to fight for attention over some other subject area in an extremely crowded curriculum. Moreover, I worry that coding without teaching the problem solving inherent in computer science will create a situation where coders are commodified. As we look at how to get the best value-add from computer science education and integrate computer science across the curriculum, digital-age problem solving was an incredibly useful lens for me to understand the value that computer science can bring to other subject areas without a single line of code, and how we can take the best of computer science across multiple disciplines. Adding in design thinking and data literacy has helped me cement, through creating this course, one model for what problem solving could look like in the STEM world. Even after this dissertation is complete, I look forward to continuing to develop and refine this framework, and to look for opportunities to support educators in testing, refinement, and implementation.

# REFERENCES

Abelson, H. (2008). The creation of OpenCourseWare at MIT. *Journal of Science Education and Technology*, *17*(2), 164–174. http://doi.org/10.1007/s10956-007-9060-8

Anderson, M. D. (2016). Will the push for coding lead to "technical ghettos?" *The Atlantic*.

Asif, A. (2013). States are slow to adopt controversial new science standards. Retrieved from http://hechingered.org/content/states-are-slow-to-adopt-controversial-new-science-standards_6355/

Arslan, S., & Demirtas, Z. (2016). Social emotional learning and critical thinking disposition. Studia Psychologica, 58(4), 276-285. http://dx.doi.org.prox.lib.ncsu.edu/10.21909/sp.2016.04.723

Avineri, T. A. (2016). Effectiveness of a mathematics education massive open online course as a professional development opportunity for educators. http://doi.org/http://www.lib.ncsu.edu/resolver/1840.16/11025

Barba, L. (2016). Computational thinking: I do not think it means what you think it means. Retrieved from https://bids.berkeley.edu/news/computational-thinking-i-do-not-think-it-means-what-you-think-it-means

Barnhardt, R., & Kawagley, A. O. (2005). Indigenous Knowledge Systems and Alaska Native Ways of Knowing.Anthropology and Education Quarterly, 36(1), pp. 8-23.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community ? *ACM Inroads*, *2*(1), 48–54. http://doi.org/10.1145/1929887.1929905

Bates, T. (2014). Comparing xMOOCs and cMOOCs: Philosophy and practice. Retrieved from http://www.tonybates.ca/2014/10/13/comparing-xmoocs-and-cmoocs-philosophy-and-practice/

Beede, D. N., Julian, T. A., Langdon, D., McKittrick, G., Khan, B., & Doms, M. E. (2011). Women in STEM: A gender gap to innovation. Retrieved from http://doi.org/10.2139/ssrn.1964782

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145–157. Retrieved from http://doi.org/10.1016/j.compedu.2013.10.020

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Annual American Educational Research Association Meeting, Vancouver, BC, Canada*, 1–25.

British Columbia, Province of. (n.d.). *Applied design, skills, and technologies*. Retrieved from https://curriculum.gov.bc.ca/curriculum/adst

Buchanan, R. (1992). Wicked problems in design thinking. *Design Issues*, *8*(2), 5. Retrieved from http://doi.org/10.2307/1511637

Buffum, P. S., Martinez-Arocho, A. G., Frankosky, M. H., Rodriguez, F. J., Wiebe, E. N., & Boyer, K. E. (2014). CS principles goes to middle school (pp. 151–156). Presented at the Proceedings of the 45th ACM technical symposium on Computer science education - SIGCSE '14, ACM Press. Retrieved from http://doi.org/10.1145/2538862.2538949

Bybee, R. W. (2013). The case for STEM education: Challenges and opportunities. National Science Teachers Association.

Carroll, M., Goldman, S., Britos, L., Koh, J., Royalty, A., & Hornstein, M. (2010). Destination, imagination and the fires within: Design thinking in a middle school classroom. *International Journal of Art & Design Education*, *29*(1), 37–53. Retrieved from http://doi.org/10.1111/j.1476-8070.2010.01632.x

Chung, C. J. C., Cartwright, C., & Cole, M. (2014). Assessing the impact of an autonomous robotics competition for STEM education. *Journal of STEM Education: Innovations & Research*, *15*(2), 24–34.

Clarke, D., & Hollingsworth, H. (2002). Elaborating a model of teacher professional growth. *Teaching and Teacher Education*, *18*(8), 947–967. Retrieved from http://doi.org/10.1016/S0742-051X(02)00053-7

Clow, D. (2013). MOOCs and the funnel of participation. *Proceedings of the Third International Conference on Learning Analytics and Knowledge - LAK '13*, 185. Retrieved from http://doi.org/10.1145/2460296.2460332

Code.org. (2016). Computer science discoveries. Retrieved from https://code.org/educate/csd

College Board. (2014a). AP Computer science principles curriculum framework, 1–48.

College Board. (2014b). *Computer science: A course description*.

College Board. (2014c). The 10th Annual AP Report to the Nation: AP Computer Science A.

Committee on a Conceptual Framework for New K-12 Science Education Standards. (2012). A Framework for K-12 Science Education. National Academies Press. Retrieved from http://doi.org/10.17226/13165

Computer Science Teachers Association. (2013). *Bugs in the system: Computer science teacher certification in the US.*

Corbin, J., & Strauss, A. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publications, Inc.

Daniel, J. (2012). Making sense of MOOCs: Musings in a maze of myth, paradox and possibility. *Journal of Interactive Media in Education*, *2012*(3), 18. Retrieved from http://doi.org/10.5334/2012-18

Darling-Hammond, L., & Wei, R. C. (2009). Professional learning in the learning profession: A status report on teacher development in the United States and abroad. *National Staff Development Council*, 1–32. Retrieved from http://doi.org/10.1006/jfbi.2002.2063

Davis, V., Klein, E., Lirenman, K., Maiers, A., Martinez, S., & Wettrick, D. (2014). Genius hour – ISTE 2014.

DeBoer, J., Ho, A. D., Stump, G. S., & Breslow, L. (2014). Changing “course”: *Reconceptualizing Educational Variables for Massive Open Online Courses*, *43*, 74–84. Retrieved from http://doi.org/10.3102/0013189X14523038

Digital Promise, Center for Teaching Quality. (2016). *Micro-credentials: Driving teacher learning & leadership & leadership* (p. 20).

Doppelt, Y., Mehalik, M. M., Schunn, C. D., Silk, E., & Krysinski, D. (2008). Engagement and Achievements: A Case Study of Design-Based Learning in a Science Context. *Journal of Technology Education*, *19*(2), 22–39.

Dow, P. B. (1991). Schoolhouse politics. Harvard University Press.

FACT SHEET: President Obama Announces Computer Science For All Initiative. (2016). FACT SHEET: President Obama Announces Computer Science For All Initiative.

Fahy, P. J., Crawford, G., & Ally, M. (2001). Patterns of interaction in a computer conference transcript. *The International Review of Research in Open and Distributed Learning*, *2*(1). Retrieved from http://doi.org/10.19173/irrodl.v2i1.36

Fee, S. B., & Holland-Minkleya, A. M. (2010). Teaching computer science through problems, not solutions. *Computer Science Education*, *20*, 129–144.

Flattau, P. E., Bracken, J., Van Atta, R., Bandeh-Ahmadi, A., la Cruz, de, R., & Sullivan, K. (2007). *The National Defense Education Act of 1958: Selected Outcomes*.

Flemming, A. S. (1960). The philosophy and objectives of The National Defense Education Act. *The ANNALS of the American Academy of Political and Social Science*, *327*(1), 132–138. Retrieved from http://doi.org/10.1177/000271626032700116

Freeman, A., Adams Becker, S., Cummins, M., Davis, A., & Hall Giesinger, C. (2017). *NMC/CoSN Horizon Report: 2017 K–12 Edition* (pp. 1–64). Austin, Texas.

Friday Institute for Educational Innovation. (n.d.). *Computational Thinking and Design MOOC-Ed*. (M. Samberg & P. Nichols, Trans.). Raleigh.

Frye, D., Samberg, M., Moris, S., & Keller, L. (2017). *NC Computing Education Summit: Preliminary Landscape Report* (pp. 1–17). Raleigh: Friday Institute for Educational Innovation.

Gao, F., Wang, C. X., & Sun, Y. (2009). A new model of productive online discussion and its implications for research and instruction. *Journal of Educational Technology Development and Exchange*, *2*(1). Retrieved from http://doi.org/10.18785/jetde.0201.05

Garcia, D., Harvey, B., & Barnes, T. (2015). The beauty and joy of computing. *ACM Inroads*, *6*(4), 71–79. Retrieved from http://doi.org/10.1145/2835184

Garland, S. (2014). The man behind common core math. Retrieved from http://www.npr.org/sections/ed/2014/12/29/371918272/the-man-behind-common-core-math

Garrett, L. (2008). STEM: The 21st Century Sputnik. *Kappa Delta Pi Record*, 152–154.

Gattie, D. K., & Wicklein, R. C. (2007). Curricular value and instructional needs for infusing engineering design into K-12 technology education. *Journal of Technology Education*, *19*(1). Retrieved from http://doi.org/10.21061/jte.v19i1.a.1

Goldberg, A., & Nemcsok, S. (2015). Try again: Teaching teachers and students how to fail with design thinking (pp. 192–202). Presented at the Designing for Innovation Selected Proceedings. Paper presented at IDEAS 2016: Designing for Innovation.

Gonzalez, H. B., & Kuenzi, J. (2012). What is STEM Education and why is it important? *Congressional Research Service*, 1–15.

Goode, J. (2008). EducationReprogramming college preparatory computer science. *Communications of the ACM, 51*(11), 31. Retrieved from http://doi.org/10.1145/1400214.1400225

Google. (2014). Women who choose Computer Science — What really matters -- The critical role and exposure, 1–8. Retrieved from http://doi.org/https://docs.google.com/file/d/0B-E2rcvhnlQ_a1Q4VUxWQ2dtTHM/edit

Google. (2016). Diversity - Google. Retrieved from https://blog.google/topics/diversity/focusing-on-diversity30/

Google News Lab. (n.d.). Google News Lab. Retrieved from https://newslab.withgoogle.com/

Google, Gallup. (2015). *Searching for Computer Science: Access and barriers in U.S. K-12 Education* (p. 28).

Gray, J., Bounegru, L., & Chambers, L. (2012a). The data journalism handbook. “O'Reilly Media, Inc.”

Gray, J., Bounegru, L., & Chambers, L. (2012b). The data journalism handbook.

Greller, W., & Drachsler, H. (2012). Translating learning into numbers: A generic framework for learning analytics author contact details. *Educational Technology & Society*, *15*(3). Retrieved from http://doi.org/http://hdl.handle.net/1820/4506

Gulumhussein, A. (2013). Teaching the teachers: Effective professional development in an era of high stakes accountability. *Center of Public Education*, 40. Retrieved from http://doi.org/10.1108/eb016093

Gunawardena, C. N., Lowe, C. A., & Anderson, T. (1997). Analysis of a global online debate and the development of an interaction analysis model for examining social construction of knowledge in computer conferencing. *Journal of Educational Computing Research*, *17*(4), 397. Retrieved from http://doi.org/10.2190/7mqv-x9uj-c7q3-nrag

Halverson, E. R., & Sheridan, K. M. (2014). The maker movement in education. *Harvard Educational Review*, *84*(4), 495–504, 563, 565. Retrieved from http://doi.org/10.17763/haer.84.4.34j1g68140382063

Harmon, L. (1977). Career achievements of the National Defense Education Act (Title IV) Fellows of 1959-1973. National Academies Press.

Hasso Plattner Institute of Design. (2013). An introduction to Design Thinking, 1–15. Retrieved from http://doi.org/10.1007/978-1-4302-6182-7_1

Hatter, L. (2016). French, Spanish, German ... Java? Making coding count as a foreign language.

Hazzan, O., Gal-Ezer, J., & Blum, L. (2008). A model for high school computer science education: The four key elements that make it! *ACM SIGCSE Bulletin*, 281–285. Retrieved from http://doi.org/10.1145/1352322.1352233

Hekimoglu, S., & Sloan, M. (2005). A compendium of views on the NCTM Standards. *Mathematics Educator*, *15*(1), 35–43.

Igoe, T. (2016). Stop teaching programming, start teaching computational thinking. *Makezine*.

Independent Expert Advisory Group on a Data Revolution for Sustainable Development. (2014). *A World that Counts* (pp. 1–32). United Nations.

Ingle, B. R. (2013). I*ntroduction to design thinking* (pp. 1–15). Berkeley, CA: Apress. Retrieved from http://doi.org/10.1007/978-1-4302-6182-7_1

ISTE, Computer Science Teachers Association. (2011). CT Vocabulary and Progression Chart, 8–9.

ISTE, CSTA. (2011). Operational definition of computational thinking. *Report*.

Jones, J., Mccowan, D., & Stephenson, C. (2003). A model curriculum for K–12 computer science: Final report of the ACM K-12 Task Force Curriculum Committee. *Computer*.

Jordan, K. (2015). Massive open online course completion rates revisited: Assessment, length and attrition. *The International Review of Research in Open and Distributed Learning*, *16*(3). Retrieved from http://doi.org/10.19173/irrodl.v16i3.2112

K-12 Computer Science Framework. (2016). *K-12 Computer Science Framework* (pp. 1–307). Retrieved from https://k12cs.org/

Kellogg, S. (2014). *Patterns of peer interaction and mechanisms governing social network structure in two massively open online courses for educators*.

Kellogg, S., Booth, S., & Oliver, K. (2014). A social network perspective on peer supported learning in MOOCs for educators. *The International Review of Research in Open and Distributed Learning*, *15*(5), 1–27. Retrieved from http://doi.org/10.19173/irrodl.v15i5.1852

Kim, B., Kim, T., & Kim, J. (2013). Paper-and-pencil programming strategy toward computational thinking for non-majors: Design your solution. *Journal of Educational Computing Research*, *49*(4), 437–459. Retrieved from http://doi.org/10.2190/EC.49.4.b

Kleiman, G., & Wolf, M. A. (2015). Going to scale with online professional development: The Friday Institute MOOCs for Educators (MOOC-Ed) Initiative.

Kolko, J. (2015). Design thinking comes of age. *Harvard Business Review*.

Kolowich, S. (2013). San Jose State U. puts MOOC project with Udacity on hold. *Chronicle of Higher Education*.

Kuenzi, J. (2008). Science, Technology, Engineering, and Mathematics (STEM) Education: Background, federal policy, and legislative action. *CRS Report for Congress*, 1–18.

Land, M. H. (2013). Full STEAM ahead: The benefits of integrating the arts into STEM. *Procedia Computer Science*, *20*, 547–552. Retrieved from http://doi.org/10.1016/j.procs.2013.09.317

LeBar, M. (2014). MOOCs–Completion is not important. Retrieved from https://www.forbes.com/sites/ccap/2014/09/16/moocs-finishing-is-not-the-important-part/#11b50e0578f6

Long, P., & Siemens, G. (2011). Penetrating the fog: Analytics in learning and education, *46*.

Margolis, J., & Fisher, A. (2002). Unlocking the clubhouse: Women in computing. Massachusetts Institute of Technology.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2008). Stuck in the shallow end. Cambridge, MA: The MIT Press. Retrieved from http://www.worldcat.org/title/stuck-in-the-shallow-end-education-race-and-computing/oclc/904735486

Martin, R. (2015). More than a pipeline problem: In search of diversity in Silicon Valley. Retrieved from https://www.npr.org/sections/alltechconsidered/2015/07/26/426364306/more-than-a-pipeline-problem-in-search-of-diversity-in-silicon-valley

Massachusetts Department of Elementary and Secondary Education. (2016). Retrieved from https://curriculum.gov.bc.ca/curriculum/adst, 1–49.

McAuley, A., Stewart, B., Siemens, G., & Cormier, D. (2010). The MOOC model for digital practice. *Massive Open Online Courses: Digital Ways of Knowing and Learning*, 1–64. Retrieved from http://doi.org/10.1016/j.im.2011.09.007

Miles, M. B., Huberman, A. M., & Saldana, J. (2013). *Qualitative data analysis: A methods sourcebook.* Sage Publications, Inc.

Milligan, C., & Littlejohn, A. (2014). Supporting professional learning in a massive open online course. *International Review of Research in Open and Distance Learning*, *15*(5), 197–213.

Mohomed, I., & Dutta, P. (2015). THE Age of DIY and Dawn of the Maker Movement. *ACM SIGMOBILE Mobile Computing and Communications Review*, *18*(4), 41–43. Retrieved from http://doi.org/10.1145/2721914.2721929

Mozilla. (2011). Open badges for lifelong learning. *White Paper*, 1–14.

National Academy of Engineering. (2008). *NAE Grand Challenges for Engineering*. Retrieved from http://engineeringchallenges.org/challenges.aspx

National Academy of Sciences. (2005). Rising above the gathering storm. National Academies Press. Retrieved from http://doi.org/10.17226/11463

National Commission on Excellence in Education. (1983). A nation at risk: The imperative for educational reform. Retrieved from http://doi.org/10.1145/358150.358154

National Committee on Science Education Standards and Assessment. (1996). National Science Education Standards. *Science Education*. National Academies Press. Retrieved from http://doi.org/10.17226/4962

National Research Council. (2010). Report of a workshop on the scope and nature of computational thinking (Vol. 1, pp. 1–114). Washington, DC: National Academies Press. Retrieved from http://doi.org/10.17226/12840

National Science Foundation. (1996). *1996 Biennial Report to Congress: Committee on Equal Opportunities in Science and Engineering* (p. 15).

Next Generation Science Standards. (2013). Next Generation Science Standards.

Obama, B. (2016). Weekly address: Giving every student an opportunity to learn through computer science for all. Retrieved from https://www.youtube.com/watch?v=8sthaV8ddJ4&feature=youtu.be

Olivares-Giles, N. (2015). Toys that teach the basics of coding. *Wall Street Journal*.

Open Badges Alliance. (2016). Open badges technical specification. Retrieved from https://openbadgespec.org

Papastergiou, M. (2009). Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Computers & Education*, *52*(1), 1–12. Retrieved from http://doi.org/10.1016/j.compedu.2008.06.004

Papert, S. (1972). Teaching children thinking. *Innovations in Education & Training International*, *9*(5), 245–255. Retrieved from http://doi.org/10.1080/1355800720090503

Papert, S. (1993). Mindstorms: Children, computers, and powerful ideas. Basic Books.

Parnership for 21st Century Learning. (2015). *P21 Framework for 21st Century Learning* (pp. 1–9).

Petroski, H. (2014). Digging the roots of STEM. Retrieved from http://www.asee-prism.org/refractions-3/

Pólya, G. (1957). *How to solve it* (2nd ed.). Garden City, NY: Doubleday.

Prensky, M. (2006). *Don't bother me mom--I'm learning!* Paragon House.

Price, T. W., & Barnes, T. (2015). Comparing textual and block interfaces in a novice programming environment (pp. 91–99). Presented at the Proceedings of the eleventh annual International Conference on International Computing Education Research - ICER '15, ACM Press. Retrieved from http://doi.org/10.1145/2787622.2787712

Prottsman, K. (2015). Coding vs. programming — Battle of the terms! Retrieved from http://www.huffingtonpost.com/kiki-prottsman/coding-vs-programming-bat_b_7042816.html

Przybylla, M., & Romeike, R. (2014). Physical computing and its scope – Towards a constructionist computer science curriculum with physical computing, *13*, 241–254. Retrieved from http://doi.org/10.15388/infedu.2014.05

*Report of a Workshop of Pedagogical Aspects of Computational Thinking*. (2011). *Report of a Workshop of Pedagogical Aspects of Computational Thinking* (p. 177).

Rittel, H. W. J., & Webber, M. M. (1973). Dilemmas in a general theory of planning dilemmas in a general theory of planning. *Policy Sciences*, *4*(2), 155–169.

Rubio, M. A., Romero-Zaliz, R., Manoso, C., & De Madrid, A. P. (2015). Enhancing an introductory programming course with physical computing modules. *Proceedings - Frontiers in Education Conference, FIE*. Retrieved from http://doi.org/10.1109/FIE.2014.7044153

Sanders, M. (2009). STEM, STEM Education, STEMAnia. *Education*, *68*(4), 20–27.

Scheer, A., Noweski, C., & Meinel, C. (2012). Transforming constructivist learning into action: Design thinking in education. *Design and Technology Education*, *17*, 8–19.

Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cuniff, D., et al. (2011). CSTA K-12 Computer Science Standards.

Shein, E. (2014). Should everybody learn to code?, *57*, 16–18. Retrieved from http://doi.org/10.1145/2557447

Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Exploring the science framework and NGSS: Computational thinking in the science classroom. *Science Scope*, *38*(3), 10–15.

Snyder, L., Barnes, T., Garcia, D., Paul, J., & Simon, B. (2012). The first five computer science principles pilots. *ACM Inroads*, *3*(2), 54. Retrieved from http://doi.org/10.1145/2189835.2189852

Solomon, C. J. (1978). Teaching young children to program in a LOGO turtle computer culture. *ACM SIGCUE Outlook*, *12*, 20–29. Retrieved from http://doi.org/10.1145/964041.964045

Spencer, J., & Juliani, A. (2016). Launch. Dave Burgess Consulting, Incorporated.

Stanford Design School. (2012). The design thinking process. Retrieved from https://dschool-old.stanford.edu/sandbox/groups/designresources/wiki/36873/attachments/74b3d/ModeGuideBOOTCAMP2010L.pdf

Stephenson, C., Gal-Ezer, J., Haberman, B., & Verno, A. (2005). The new educational imperative: Improving high school computer science education. The New Educational Imperative. *Final Report of the CSTA Curriculum Improvement Task Force*.

Strauss, V. (2017, December 20). The surprising thing Google learned about its employees — and what it means for today's students. *Washington Post*. Washington, DC. Retrieved from https://www.washingtonpost.com/news/answer-sheet/wp/2017/12/20/the-surprising-thing-google-learned-about-its-employees-and-what-it-means-for-todays-students/?utm_term=.6ffcbe52c800

Stump, G. S., DeBoer, J., Whittinghill, J., & Breslow, L. (2013). Development of a framework to classify MOOC discussion forum posts: Methodology and challenges.

Sydell, L. (2014). The forgotten female programmers who created modern tech. *NPR News*.

The White House. (2016). Educate to innovate. Retrieved from https://www.whitehouse.gov/issues/education/k-12/educate-innovate

Trump, D. (2017). Presidential memorandum for the secretary of education | The White House. Retrieved from https://www.whitehouse.gov/presidential-actions/presidential-memorandum-secretary-education/

Tyner, K. R. (1998). Literacy in a digital world: Teaching and learning in the age of information. Erlbaum Associates.

Vee, A. (2013). Understanding computer programming as a literacy. *Literacy in Composition Studies*, *1*(2), 42–64.

Vivian, R., Falkner, K., & Falkner, N. (2014). Addressing the challenges of a new digital technologies curriculum: MOOCs as a scalable solution for teacher professional development. *Research in Learning Technology*, *22*(1063519). Retrieved from http://doi.org/10.3402/rlt.v22.24691

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question (Vol. 149, pp. 199–208). Presented at the Proceedings of the 14th International Conference on Interaction Design and Children - IDC '15, ACM Press. Retrieved from http://doi.org/10.1145/2771839.2771860

What's the impact of the Hour of Code? (2016). What's the impact of the Hour of Code? Retrieved from https://code.org/about/evaluation/hourofcode

Wiebe, E. N., Faber, M., Corn, J., Collins, T. L., Unfried, A., & Townsend, L. (2013). A large-scale survey of K-12 students about STEM: Implications for engineering curriculum development and outreach efforts (Research to Practice). *American Society for Engineering Education*, 1–9.

Wilk, B. P., & Garcia, B. K. (2014). CS Ed Week 2013: *The Hour of Code*, *46*(1), 2–4.

Williams, M. (2015). Driving diversity at Facebook. Retrieved from https://newsroom.fb.com/news/2015/06/driving-diversity-at-facebook/

Wing, J. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33. Retrieved from http://doi.org/10.1145/1118178.1118215

Wing, J. (2008). Computational thinking and thinking about computing. *IPDPS Miami 2008 - Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium, Program and CD-ROM*, 3717–3725. Retrieved from http://doi.org/10.1109/IPDPS.2008.4536091

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, *14*(1), 1–16. Retrieved from http://doi.org/10.1145/2576872

Zendler, A., & Klaudt, D. (2015). Instructional methods to computer science education as investigated by computer science teachers. *Journal of Computer Science*, *11*(8), 915–927. Retrieved from http://doi.org/10.3844/jcssp.2015.915.927

## APPENDIX A: IRB APPROVAL LETTER

**EAST CAROLINA UNIVERSITY**
**University & Medical Center Institutional Review Board Office**
4N-70 Brody Medical Sciences Building· Mail Stop 682
600 Moye Boulevard · Greenville, NC 27834
Office **252-744-2914** · Fax **252-744-2284** · www.ecu.edu/irb

### Notification of Initial Approval: Expedited

From: Social/Behavioral IRB
To: Mark Samberg
CC:
Matthew Militello
Date: 3/1/2017
Re: UMCIRB 16-002461
Problem Solving in the Digital Age

I am pleased to inform you that your Expedited Application was approved. Approval of the study and any consent form(s) is for the period of 2/28/2017 to 2/27/2018. The research study is eligible for review under expedited category #6, 7. The Chairperson (or designee) deemed this study no more than minimal risk.

Changes to this approved research may not be initiated without UMCIRB review except when necessary to eliminate an apparent immediate hazard to the participant. All unanticipated problems involving risks to participants and others must be promptly reported to the UMCIRB. The investigator must submit a continuing review/closure application to the UMCIRB prior to the date of study expiration. The Investigator must adhere to all reporting requirements for this study.

Approved consent documents with the IRB approval date stamped on the document should be used to consent participants (consent documents with the IRB approval date stamp are found under the Documents tab in the study workspace).

The approval includes the following items:

| Name | Description |
| --- | --- |
| Consent Form | Consent Forms |
| Dissertation Working Copy.docx | Study Protocol or Grant Application |
| Interview Protocol | Interview/Focus Group Scripts/Questions |
| IRB Authorization NC State | Dataset Use Approval/Permission |

The Chairperson (or designee) does not have a potential for conflict of interest on this study.

IRB00000705 East Carolina U IRB #1 (Biomedical) IORG0000418
IRB00003781 East Carolina U IRB #2 (Behavioral/SS) IORG0000418

# APPENDIX B: MOOC COURSE DESCRIPTION

In the Information Age, problems look different. Information comes at us faster than ever before, and our ability to solve problems depends on us being able to make sense of and synthesize this information. We must also design new solutions using all available technology and tools.

Digital-age problem solving combines three key skills and concepts essential to understanding and solving problems in the information age: data literacy, design thinking, and computational thinking. Data literacy is the ability to analyze, interpret, and tell stories using complex sets of data. Design thinking is the ability to understand problems and develop creative solutions. Computational thinking is the process of expressing solutions so that humans and computers can understand them.

Throughout this MOOC-Ed, you'll have the opportunity to dig into digital-age problem solving, engage with its component skills and concepts, and learn how to integrate them into your instructional practice. This course will not be heavy on coding, and you won't need to know any code going in - it will focus on how to integrate digital-age problem solving in a practical way into your classroom.

## Course Objectives

- Understand the components of digital-age problem solving: design thinking, computational thinking, and data literacy;
- Connect digital-age problem solving to existing content and problem-solving processes;
- Engage in the digital-age problem solving process through simulated activities;
- Apply digital-age problem solving in a real-world context;
- View digital-age problem solving in a variety of careers and subject areas;
- Explore connections to computer science, coding, and making.

## APPENDIX C: EXISTING MOOC-ED SURVEY QUESTIONS

Table 22: User Survey Questions at Account Creation

| Question Name | Input Type | Response Options (If Any) |
|---|---|---|
| Username | Free text | |
| Password | Free text (must contain at least 8 characters, at least 1 digits, at least 1 lower case letters) | |
| Email address | Free text | |
| First Name | Free text | |
| Last Name | Free text | |
| Country | Select from list | Countries with name displayed and ISO-3166-1 code stored |
| City | Free text | |
| State | Select from a list (disabled if country is not "US") | States with name displayed and ISO-3166-2 code stored |
| Gender | Select from list | Male<br>Female<br>I do not identify |
| Level of Education | Select from list | High school<br>2-year degree<br>4-year degree<br>Master's Degree<br>Doctoral Degree<br>Professional Degree (e.g. JD, MD) |
| Primary Area of Responsibility | Select from list | Classroom Teaching<br>Curriculum and Instruction<br>Professional Development<br>Instructional Technology<br>School Counselor<br>Special Education<br>Mentor<br>School-Based Administration<br>School District Administration<br>Teacher Preparation – College/University<br>Student (College/Graduate)<br>Student (K-12)<br>Research<br>Other |

| | | |
|---|---|---|
| Years of Experience in Education | Free text | |
| I specialize in the following grade levels: | Select multiple | Pre-K<br>Kindergarten<br>Elementary<br>Middle Grades<br>High School<br>Post-Secondary<br>N/A |
| Organization Type | Select from list | School<br>School District<br>College/University<br>Other |
| School District/Organization Name | Free text | |
| School Name | Free text | |

Table 13: User Survey Questions at Course Enrollment

| Question | Input Type | Response Options (If Any) |
|---|---|---|
| Do you plan to participate with a peer group outside of this MOOC-Ed? (e.g. a school-based PLC or informal group of colleagues) | Radio Buttons | Yes<br>No |
| Please select up to three goals for your participation in this MOOC-Ed | Select from list (3 lists, one is mandatory, all options are the same) | Course specific options (See Appendix C)<br>Engage in fun and inspiring activities<br>Exchange ideas and experiences with other educators<br>Collaborate on joint projects<br>Collect new resources or tools<br>Experience learning in a MOOC-Ed<br>Make changes to my professional practice<br>Earn a certificate of completion<br>Other |
| Were you familiar with the concept of micro-credentials or badges prior to this MOOC-Ed? | Radio Buttons | Yes<br>No<br>I'm not sure |
| Have you earned a micro-credential or badge prior to this MOOC-Ed? | Radio Buttons | Yes<br>No<br>I'm not sure |
| Do you intend to pursue a micro-credential for this MOOC-Ed? | Select from list | Definitely Yes<br>Probably Yes<br>Unsure<br>Probably Not<br>Definitely Not |

Table 24: End-Of-Unit Survey Questions

| Question Name | Input Type | Response Options (If Any) |
|---|---|---|
| To what extent do you agree with the following statements?<br>This Unit…<br>a. deepened my understanding of the topic(s) addressed.<br>b. supported the application of course content to my professional practice.<br>c. helped me progress towards my personal learning goals. | Matrix w/ Scale | Strongly Disagree<br>Disagree<br>Neutral<br>Agree<br>Strongly Agree |
| What changes, if any, have you made (or anticipate making) in your professional practice as a result of your participation in this MOOC-Ed so far? (E.g., Application of new knowledge, skills, or course resources) | Free text (Unit 3 only) | |
| What recommendations, if any, do you have for improving the user experience in this unit (e.g., navigation, visual design, unit organization, etc.) | Free text (Unit 1 only) | |
| Approximately how many hours did you spend on this unit's activities? | Select from a list | 1-2 hours<br>3-4 hours<br>5-6 hours<br>7-8 hours<br>more than 8 hours |

Table 25: End-Of-Course Survey Questions

| Question Name | Input Type | Response Options (If Any) |
|---|---|---|
| As a whole, how effective was this MOOC-Ed in supporting your personal and/or professional learning goals? | Select from a list | Very ineffective<br>Ineffective<br>Neutral<br>Effective<br>Very Effective |
| What was the most valuable aspect of this MOOC-Ed in supporting your personal or professional learning goals? | Free text | |
| Overall, how effective do you feel this MOOC-Ed was in preparing you to make positive changes in your professional practice? | Select from a list | Very ineffective<br>Ineffective<br>Neutral<br>Effective<br>Very Effective |
| Did you attempt to earn a micro-credential for this MOOC-Ed? | Select from a list | Yes<br>No<br>Not Sure |
| Why did you choose to pursue a micro-credential for this course? | Free text | |
| In what ways, if any, did the micro-credentialing process impact your professional practice? | Free text | |
| Why did you choose not to pursue a micro-credential for this course? | Free text | |
| To what extent do you agree with the following statements?<br>MOOC-Ed Micro-credentials are a valuable tool for...<br>a. Engaging in professional learning with an increased level of rigor.<br>b. Promoting significant changes to my instructional practice.<br>c. Communicating my professional competencies with others.<br>d. Personalizing my professional learning experience.<br>e. Facilitating collaboration and communication with other educators.<br>f. Motivating me to pursue additional learning opportunities within or beyond the MOOC-Ed | Matrix | Strongly Disagree<br>Disagree<br>Neutral<br>Agree<br>Strongly Agree |
| What recommendations do you have for making this course more valuable to future participants? (e.g., other resources, additional features, activities, etc.) Please explain. | Free text | |
| Were you able to complete all of the activities that you wanted to complete in this course? | Yes<br>No | |
| If no, please explain | Free text | |

| | | |
|---|---|---|
| Approximately how many hours per unit did you spend on MOOC-Ed activities? | Select from a list | 1-2 hours per unit<br>3-4 hours per unit<br>5-6 hours per unit<br>7-8 hours per unit<br>more than 8 hours |

## APPENDIX D: NEW SURVEY QUESTIONS AND INTERVIEW QUESTIONS

Table 26: User Survey Questions at Course Enrollment

| Question | Input Type | Response Options (If Any) |
|---|---|---|
| Do you currently work at a school or organization that has a STEM focus? | Radio buttons | Yes<br>No<br>I'm not sure |
| While coding is not a part of this course, it often tends to be grouped with Computational Thinking. Which of the following describes your comfort with computer coding/programming? | Select from a list | I've have limited or no coding experience<br>I've done small coding activities (e.g. Hour of Code)<br>I'm comfortable with coding, but don't do much with it<br>I'm comfortable with coding, and code for personal projects<br>I'm comfortable with coding, and teach it for students/work activities |
| How familiar are you with the following:<br>a) Computational Thinking<br>b) Design Thinking<br>c) Data Literacy | Select from a list | Not at all familiar<br>Slightly familiar<br>Somewhat familiar<br>Moderately familiar<br>Extremely familiar |
| This course is being developed as a part of a study on the effectiveness of teaching Digital Age Problem Solving. Are you willing to participate in an interview via telephone after the course has concluded? | Select from a list | Yes<br>No |

Table 27: End-Of-Course Survey Questions

| Question | Input Type | Response Options (If Any) |
|---|---|---|
| To what extent do you agree with the following statements.<br>I have improved my knowledge and/or skills related to…<br>a. problem solving methods to use in my classroom<br>b. the design thinking process<br>c. collecting and analyzing data<br>d. interpreting and visualizing data<br>e. decomposing and abstracting problems<br>f. creating processes to test solutions | Matrix (Likert Scale) | Strongly Disagree<br>Disagree<br>Neither Agree nor Disagree<br>Agree<br>Strongly Agree |

Interviews will use a random sampling of users who have completed at least one end-of-unit or course survey. The questions asked will be the following, and follow-up questions may be asked in the interview, as appropriate:

- Was the course useful to you? Why or why not?
- Was the process of completing the micro-credentials useful to you? Why or why not?
- Have you used any of the skills from this course in your classroom/context?
    - If not, why not?
    - If yes, please tell me how.
        - How have your students responded?
- Have you shared any of the course content with your peers?
    - If yes, please explain.
- Do you have any recommendations for future iterations of this course?

# APPENDIX E: MOOC-ED COURSE OUTLINE

The Problem Solving in the Digital Age MOOC will include five course units, spread over eight weeks, reflecting the Computational Design Thinking process identified in

Figure *3*. Table 30 shows the timeline of the course run. Table 28 lists the units of the course, and Table 29 lists the activities found within each unit.

Table 28: Course Units

| Unit Number | Unit Name | Unit Goals |
|---|---|---|
| 1 | What is Digital Age Problem Solving? | This unit will introduce the course design and course requirements to participants, as well as allow participants to meet each other. This unit will also provide an overview of the design process and computational thinking skills. |
| 2 | Identifying Problems | This unit will introduce the "Understand the Context" phase of the design thinking process, focusing on the computational thinking skills of data collection, data analysis, data representation. |
| 3 | Making Sense of Problems | This unit will transition to the "Define the Problem" phase of the design thinking process, with a specific focus on the computational thinking skills of problem decomposition, abstraction, and parallelization. |
| 4 | Creating Solutions | This unit will transition to the "Create Solutions" phase of the design thinking process, with a focus on the computational thinking skills of parallelization, algorithm development, automation, and simulation. |
| 5 | Assessing Solutions | This unit will transition to the "Evaluate, Reflect, Revise" phase of the design process, with a focus on the skills of simulation and automation. It will also serve as a capstone for the course, allowing users to reflect on what they have learned, and connect back to the beginning of the design cycle. |

Table 29: Course Activities

| Activity Name | Activity Description |
|---|---|
| "Introduce Yourself" (Unit 1 Only) | Introductory forum. A chance for course participants to meet each other and share a little about themselves. |
| Unit Introduction | Provides an overview of the phase of the design process and the computational thinking skills used in the unit. Includes a video of practitioners in various fields using these skills in their professional practice. In unit 1, this is an overview of the three elements of the digital-age problem solving process. |
| Digging Deeper | Toolkit of resources on the topics introduced in each unit for participants who want to learn more or review background research. Users can view and rate the resources in the library. |
| Simulation Activity | Provides an activity for the teachers to engage with the skills introduced in the unit as a learner and reflect on the process. |
| Brainstorming Classroom Applications | Discussion forum for teachers to identify how they could use the skills introduced in each unit in their classroom/professional practice. This forum serves as a chance for teachers to refine their ideas before heading in to the micro-credentials. |
| Micro-credentials | Allows teachers to demonstrate evidence of application of the skills in the unit into their professional practice. |
| Unit Survey/End of Course Survey | |

Table 30: Course Timeline

| Week | Unit 1 | Unit 2 | Unit 3 | Unit 4 | Unit 5 |
|---|---|---|---|---|---|
| 1 | Current Unit | Unit Opens | | | |
| 2 | | Current Unit | | | |
| 3 | | Current Unit | Unit Opens | | |
| 4 | | | Current Unit | | |
| 5 | | | Current Unit | Unit Opens | |
| 6 | | | | Current Unit | |
| 7 | | | | Current Unit | Unit Opens |
| 8 | | | | | Current Unit |

The current unit is the one that will be highlighted and actively facilitated for the week. A unit may open earlier to enable people who wish to move faster to do so. All units will remain open for an additional 6 weeks after the end of the course to enable stragglers to finish. Registration will close after week 6. After the 14th week, the course will become read-only so that users still have access to the resources, but cannot generate any new material.

## APPENDIX F: MOOC-ED COURSE CONTENT

This appendix contains the introductory and “digging deeper” content from each unit.

FRIDAY INSTITUTE Computational Thinking and Design - Spring 2017

Introduce Yourself!
Introduction
What Is Digital-Age Problem Solving?
Dig Deeper
Discuss: In My Classroom
Unit Feedback
Course Home

# Introduction

[ Listen to a Podcast | Read a Transcript ]
Save To My Bookmarks

Rate this Video:

Average of ratings: 4.4 (23 Submitted)

| | |
|---|---|
| "An architect is tasked with creating a new town square that is reflective of the town's history and can serve as a gathering place for the citizens of the town. What should this look like?" | "As factories continue to modernize, the role of people working on the factory floor changes. You are called in to a factory to design machines that can perforr repetitive and precise tasks while shifting the employees around to complete specialized tasks that machines can't do. What does this modernized factory loo like?" |
| "You are tasked with collecting data about the spread of the flu virus and helping keep communities safe. What do you need to know, and how do you share this information with the public?" | "Creating prosthetics and assistive devices for children is extremely time-consuming and expensive. Also, kids don't get to use them for long because they grow too quickly. Is there a way to provide children with the devices they need?" |

While these scenarios themselves are not new, new technologies and mindsets now allow us to approach them in very different ways:

- 3D modeling allows the architect to interview constituents and make refinements to their design in real time. The resulting product can be more responsive to the needs and concerns of the townspeople.
- Increasingly precise and complex robotics allow machines to complete tasks that until recently could have been only accomplished by a human being. As a result, employees on factory floors are required to have an increasingly complex and technical skillset.
- Electronic health records and the rise of Big Data allow us to map the spread of the flu in near real time and predict how it wil continue to spread.
- 3D printing allows the creation of plastic prosthetics cheaply, quickly, and by anyone. In addition, 3D scanning technology allows for a high degree of accuracy in determining fit.

In the last 15 to 20 years, nearly all fields have been radically altered by new technologies. More fundamentally, our *approach to problems* has been changed by these technologies. We are now able to make much greater use of data and technology, and, as a result, we are now able to iterate on problems much faster.

This course will examine how problem solving has changed in the digital age. During this course, we'll discuss some of the strategies and tools that are key to digital-age problem solving. You will have the chance to engage in activities and brainstorm classroom applications, and see how real-world practitioners are tackling these challenges in their daily work.

Each unit contains some background information, interviews with practitioners across a variety of careers, and opportunities for y to dig deeper, learn more, and engage with your colleagues. You should feel free to engage with the material in the way that work best for you. If you're familiar with a topic, feel free to skip it. If you want to learn more, feel free to spend time in the **Dig Deeper** sections in each unit.

## Course Format

Unit 1 of the course focuses on background and "big picture" ideas to set the stage for the course. Units 2 through 5 will focus on digging much more deeply into the content. Each unit will consist of three major components:

- **Explore Concepts** - This is your opportunity to learn about digital-age problem solving skills and explore the course concepts more depth.
- **Engage as a Learner** - Each unit will contain an activity in which you will be able to engage with the concepts of digital-age problem solving directly by them in simple activities.
- **Connect to the Classroom** - In the forums, we will have the opportunity to take everything we have learned and brainstorm how we can bring these concepts back to the classroom.

## Course Requirements

You can earn a certificate for **10 professional learning hours** by completing the activity and participating in the two discussion foru in each unit. An opportunity to earn **an additional 20 hours** of credit exists through the completion of micro-credentials. Micro-credentials are a form of assessment that let you earn recognition by demonstrating competency in a particular area. In this cours you'll have the opportunity to complete up to six micro-credentials on the course content. Each micro-credential will ask you to tal a component of the course material, apply it in your classroom/professional context, and reflect.

## Unit 1 Goals

In this unit, you will:

- Define "Digital-Age Problem Solving" and identify its core components.
- Learn more about design thinking, computational thinking, and data literacy.
- Introduce yourself to other course participants.
- Begin to brainstorm how you can use digital-age problem solving in your classroom/context.

FRIDAY INSTITUTE Computational Thinking and Design - Spring 2017

Introduce Yourself!

Introduction

**What Is Digital-Age Problem Solving?**

Dig Deeper

Discuss: In My Classroom

Unit Feedback

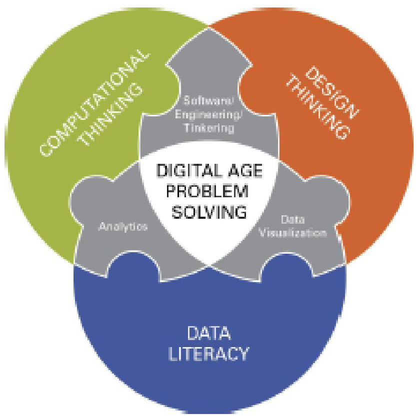
Course Home

# What Is Digital-Age Problem Solving?

Throughout this course, we will be using the term "digital-age problem solving". While this course is among the first to use this term, it's probably something you've heard of or experienced before.

**Digital-age problem solving is a term we're using to describe the collection of the skills and strategies required for students to be able to identify, frame, and solve problems in the Information Age.** Alternatively, you can think of this in another way: Using *technology* and *data* to help *solve problems for people.*

There are three essential skills and strategies that make up digital-age problem solving, and we will explore them in depth throughout the course: 1) design thinking, 2) computational thinking, and 3) data literacy. To help you begin to understand the relationship between these skills and strategies, take a look at the graphic below. The gray areas represent some of the practical skills that students will need to demonstrate at the intersection of the core skills and strategies.




Click on the tabs below to learn more about Design Thinking, Computational Thinking, and Data Literacy. You can find more information in the **Dig Deeper** section of this unit. And, of course, we'll spend the rest of the course digging much deeper into all o these concepts!

Design Thinking | Computational Thinking | Data Literacy

## Design Thinking

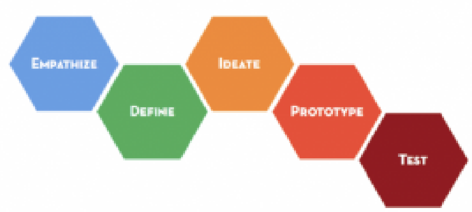



Image: Stanford University d.school

Design Thinking is a strategy to solve complex problems and develop human-centered solutions.

Design thinking differs from traditional problem solving practices in key ways:

- Engaging in design thinking is about *focusing on the solution* being developed and not the problem or the process.
- In traditional design, problems are solved *for* people, whereas in design thinking, problems are solved *with* people.

- Design thinking encourages design without constraints and boundaries, and encourages failure as mechanism for improving and iterating on the final product.

Many problems in design thinking are ill-defined, or not immediately clear. Consider the example of the town square from the **Introduction** page. A traditional approach would be to examine how successful town squares have been created in other places, ar to then simply replicate that design in this town. A design thinking approach, on the other hand, would involve going into the towr and interviewing citizens. You would seek to learn what they would want in a gathering place, how it could be useful to them, how could benefit the community, and what it should look like. While the architect may research other successful town squares, they would use them as a reference point, and not as the guiding model for the work. They may also collect other forms of data — including statistics, measurements, and survey feedback — to inform the design.

Once a designer has a clear understanding of what's needed, only then do they begin to define the requirements and brainstorm solutions. They should use all of the data that has been collected so far and brainstorm potential solutions. Unlike traditional desig which is very linear from start to finish, design thinking is much more iterative. Once the designer has created a few potential solutions, they'll take it back to stakeholders for review, feedback, and refinement. After several rounds of this, it may be time to develop and evaluate a larger prototype in the real world. The designer may undergo several rounds of iteration before arriving at final product.

**Key Points:**

- Design thinking focuses more on developing solutions for people than on implementing generic solutions.
- Design thinking goes through phases of collecting feedback and data, using the data to identify the problems to be solved, developing prototypes, and testing solutions. Feedback from key stakeholders should be solicited frequently.
- Phases of the design process are iterative and cyclical — revision after feedback is critical. In design thinking failure isn't an end, it's an opportunity to refine and create something better.
- Design thinking focuses on divergent thinking — generating and exploring as many ideas as possible before narrowing down to a solution.

## Computational Thinking

Computational Thinking describes the thought processes involved in expressing a problem and a solution in ways a machine (eith human or computer) can understand and implement.

In doing so, it's essential to be as specific and attentive to detail as possible — skipping any step will result in unexpected results.

During this course we will explore the following core computational thinking skills:

- **Decomposition**: Breaking down a problem into its component parts.
- **Abstraction:** Removing extraneous/irrelevant details from a problem to define the elements of a solution that are consistent.
- **Pattern Recognition**: Looking for common elements among different cases of a problem to help us define the rules that we ca use to solve it.
- **Creating Algorithms**: The detailed, step-by-step rules we use to solve a problem in a consistent and replicable way.
- **Evaluation:** Determining the effectiveness and efficiency of a solution and whether the solution accurately and precisely solve the problem.

Consider this in the context of our factory example. In order to determine what processes can be automated, you would first need examine the factory floor and learn about the manufacturing process. The first step would be to understand, in detail, each step in the process from start to finish. You would keep breaking down steps in the process until you couldn't go any further. This is an example of **decomposition**.

Once you have identified each of the steps, you would next determine the "general cases" for each step. For example, if your facto makes picture frames, you would need to know that the hanging wire needs to be attached at a certain place based on the size of t frame. This is where you are engaging in **abstraction** and **pattern recognition**.

Finally, it is time to put everything back together by creating **algorithms**. You would define at each station, step by step, what eithe the person or the machine would do. Algorithms must be incredibly detailed — an algorithm is much less instructing a machine to attach a wire to the frame, and much more choreographing every single movement that the machine makes.

Once your machines and your people are completing their algorithms, you would need to **evaluate** your results — test for errors a revise as needed (hopefully with a minimum of broken glass).

## Data Literacy

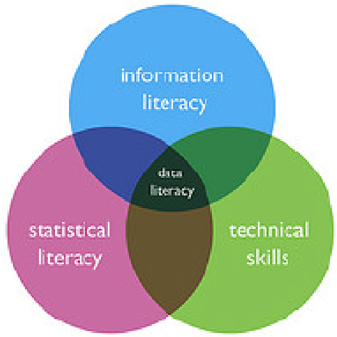



Image: United Nations Data Revolution Group

Data Literacy is much more than being able to "do statistics". Data Literacy describes:

- **Collection:** Being able to collect qualitative and quantitative data. Much of the design process involves interacting and designing solutions for people. Therefore it's critical to be able to turn conversation and observation into useful data, along with the numbers.
- **Critical evaluation:** Thanks to the Internet, data and information is incredibly easy to come by. However, much of this data is incomplete, biased, or flat-out false. Data Literacy involves being able to critically analyze data for accuracy and impartiality, and to be able to triangulate conclusions using multiple data sources.
- **Technical ability:** The ability to analyze and interpret data, both quantitative and qualitative. This includes both "small data" such as interviews and surveys and extremely large datasets commonly known as "big data".
- **Display and shareability:** The ability to summarize and reformat data to tell a story and share data in ways that are both objective and easy to understand. This may include the creation of graphics to represent data.

This is especially important for students on the Internet. As all of us increasingly get our news from social media, the stories we se tend to the the ones that validate our own world views. Being able to go to the data and determine what is true and what is not is one of the advantages of having access to the Internet. Students will need to know how to be able to determine if the information that they are reading is accurate or biased by going triangulating facts using multiple data sources.

Additionally, consider our flu virus example. Thanks to electronic health records, we can map each new case of the flu in near real- time. Additionally, data shared by merchants allows you to identify where there are unexpected spikes in medications to treat flu-like symptoms. Finally, by analyzing Google Trends data, you can see where people are searching for various symptoms of illness By correlating these data sets, you can see where the flu is prevalent, and where it may strike next. Data literacy is about being abl to retrieve and understand this data, synthesize the data from the various sources, and use the data to make an informed hypotheses. Additionally, someone who is data literate would also be able to synthesize all of these data in such a way that it can l easily shared and understood by others, for example, on the evening news. While data literacy is important by itself, it's also a vita element of both design and computational thinking, because making informed decisions means collecting good data.

FRIDAY INSTITUTE Computational Thinking and Design - Spring 2017

Introduce Yourself!

Introduction

What Is Digital-Age Problem Solving?

Dig Deeper

Discuss: In My Classroom

Unit Feedback

**Course Home**

# Dig Deeper

Learn more about the components of digital age problem-solving by exploring the resources below. You may explore as few or as many resources as you would like.

**Click on the three tabs below for resources related to Computational Thinking, Design Thinking, and Data Literacy.**

Computational Thinking | Design Thinking | Data Literacy

## Computational Thinking

Computational Thinking
**Author:** Jeannette M. Wing
This opinion piece, from the Association of Computing Machinery, is the article that started the modern discussion on Computational Thinking. In it, Wing begins to lay out why Computational Thinking is an essential skill for everyone.
Rate this resource: (Average: 4.3 / 12 Submitted) Bookmark this Resource Share |

Computational Thinking for All
**Author:** Carolyn Sykora
**Source:** ISTE
In the 2016 revision of the ISTE skills for students, Computational Thinking was included as one of the new ISTE standards. This resource hub contains definitions, getting started guides and resources for teachers, and school/district leadership. Free registratic is required.
Rate this resource: (Average: 4.5 / 14 Submitted) Bookmark this Resource Share |

Solving Problems at Google Using Computational Thinking
**Source:** Google for Education
This video shares real-world examples of computational thinking components in use every day at Google, using Google Maps and Google Earth as an example.
Rate this resource: (Average: 4.7 / 15 Submitted) Bookmark this Resource Share |

Exploring Computational Thinking
**Source:** Google
Google's resource hub on computational thinking, complete with lesson plans, example problems, and instructional materials.
Rate this resource: (Average: 4.6 / 13 Submitted) Bookmark this Resource Share |

BBC Bitesize: Introduction to Computational Thinking
**Source:** BBC
The BBC has developed a computer science curriculum for K-12 students in Great Britain. This section focuses on computational thinking skills.
Rate this resource: (Average: 4.3 / 14 Submitted) Bookmark this Resource Share |

Bringing Computational Thinking to K-12: What is the Role of the Computer Science Community?
**Author:** Valerie Barr and Chris Stephenson
**Source:** ACM Inroads
This seminal research study attempts to clarify what the role of computational thinking is in K-12, define core skills, and connect these skills to what teachers are doing in the classroom.
Rate this resource: (Average: 4.3 / 10 Submitted) Bookmark this Resource Share |

## Design Thinking

What Is Design Thinking?

**Source:** Sean VanGenderen

This brief animation provides an overview of what Design Thinking is and how it differs from traditional design.

Rate this resource: (Average: 4.4 / 7 Submitted) Bookmark this Resource Share |

Philips Healthcare Consulting and Design Thinking

**Source:** Philips Healthcare

This video, produced for Phillips Healthcare Consulting, provides a brief overview of their experience with a Florida clinic. Pay attention to how the values of the clinic are visible in the new space, and how voices which may otherwise be ignored are elevatec through this process.

Rate this resource: (Average: - Bookmark this Resource Share |

Design Thinking Comes of Age

**Author:** Jon Kolko

**Source:** Harvard Business Review

In this article, John Kolko discusses how design thinking is transforming the workplace.

Rate this resource: (Average: 4.3 / 3 Submitted) Bookmark this Resource Share |

Making Failure More Productive

**Source:** Harvard Business Review

Anjali Sastry and Kara Penn, authors of *Fail Better*, offer a three-step approach for harnessing failure.

Rate this resource: (Average: 4.5 / 4 Submitted) Bookmark this Resource Share |

d.school K12 Lab Network

**Source:** Stanford Design School

This resource, from the Stanford Design School contains sample projects and facilitator guides, as well as opportunities to engage with K-12 educators around the world engaging in Design Thinking.

Rate this resource: (Average: 4.5 / 2 Submitted) Bookmark this Resource Share |

Design Thinking for Educators

Review this website and download the toolkit in the upper right corner of this website. The toolkit contains graphic organizers, lesson guides, and resources to help teachers get started bringing design thinking into the classroom.

Rate this resource: (Average: 4.8 / 5 Submitted) Bookmark this Resource Share |

How to Apply Design Thinking in Class, Step by Step

**Author:** Anne Stevens

**Source:** MindShift, KQED

This blog post contains a set of "frequently asked questions" for teachers nervous about engaging in design thinking in their classrooms.

Rate this resource: (Average: 4.7 / 7 Submitted) Bookmark this Resource Share |

Design Thinking with Elementary Students (First Grade)

**Author:** David Lee

This video chronicles a design challenge for first graders called "Define the Ultimate Animal".

Rate this resource: (Average: 4.9 / 8 Submitted) Bookmark this Resource Share |

If students designed their own schools...

**Author:** Charles Tsai

The best small town in America experiments with self-directed learning at its public high school. A group of students gets to creat their own school-within-a-school and they learn only what they want to learn. Does it work? Charles Tsai finds out by spending a week with the Independent Project.

Rate this resource: (Average: 4.7 / 3 Submitted) Bookmark this Resource Share |

Destination, Imagination and the Fires Within: Design Thinking in a Middle School Classroom

**Author**: Maureen Carroll, Shelley Goldman, Leticia Britos, Jaime Koh, Adam Royalty, and Michael Hornstein

This 2010 study by researchers at the Stanford University Design School examines the benefits of design thinking in a middle scho context, specifically on student creativity and perseverance.

Rate this resource: (Average: 5 / 2 Submitted) Bookmark this Resource Share |

## Data Literacy

What Is Data Literacy?

**Source**: Data-Pop Alliance

This animation explains one definition of data literacy created by the Data-Pop alliance, a collaboration of several universities exploring use of "Big Data" to determine ways to promote social good.

Rate this resource: (Average: 3.8 / 6 Submitted) Bookmark this Resource Share |

The Age of Insight: Telling Stories With Data

**Source**: Google News Lab

As data has become more prominent, it has changed the face of journalism. In this video, pioneers in data journalism speak about the role and importance of using data in reporting, walk through some examples of their work, and share their thoughts on where the industry is headed.

Rate this resource: (Average: 4.6 / 5 Submitted) Bookmark this Resource Share |

The Age of Data Literacy

**Author**: Uldis Leiterts

**Source**: TEDx Talks

How does our brain process stories and does it like data? What about the combination of both? Uldis Leiterts elaborates on the question of bringing the information — and the stories to the people while capturing their attention on content with the help of infographics.

Rate this resource: (Average: 3.7 / 3 Submitted) Bookmark this Resource Share |

Data Journalism Handbook

**Author**: Jonathan Gray, Liliana Bounegru, Lucy Chambers (Editors)

In many ways, data literacy and data journalism are complimentary. This handbook is designed for data journalists who are new tc the field and contains background, tips and tricks, and practical examples.

Rate this resource: (Average: 4.5 / 2 Submitted) Bookmark this Resource Share |

Big Data Revolution

**Source**: NPR TED Radio Hour

Once invisible details of our lives can now be tracked and turned into data. Will this make life easier or more complicated? How wi big data reshape the world?

Rate this resource: (Average: 5 / 3 Submitted) Bookmark this Resource Share |

FRIDAY INSTITUTE Computational Thinking and Design - Spring 2017

Introduction
Learning from Data
Activity: Telling Stories with Data
Core Resources
Discuss: Data in the Classroom
Earn Micro-credentials
Unit Feedback
Course Home

# Introduction

[ Listen to a Podcast | Read a Transcript ]
Save To My Bookmarks

Rate this Video:

Average of ratings: 4.5 (10 Submitted)

*"It seems pretty straightforward. Food is sitting in the window upwards of ten minutes before it's being served. Obviously, the people working here aren't any good and need to be fired."*

While this might be the right conclusion to draw (particularly on an empty stomach), it is likely that there is more going on than meets the eye. While it's human nature to immediately jump to a problem's solutions based solely on its symptoms, it's also possible that the symptoms might be the result of a variety of problems which may or may not be immediately clear. Yes, the restaurant may have issues with staff performance. But since no one comes to work wanting to do a poor job, we should dig a littl bit deeper into the problem...

It's possible that the restaurant may be understaffed, or the staff may not be properly trained. It's also possible that they have one server covering all of the larger tables, or that they have table assignments that require the server to run all the way across the restaurant. The food window might not be visible to the serving staff, or it might not be clear that an order requires their attention. There may even be a combination of these possibilities happening all at once. Based on this, it becomes clear that before you can begin to identify a problem, you have to understand the context.

You can think of the context as the world in which a problem resides. Problems arise from the interactions among people, systems structures. Designing solutions to these problems requires a thorough understanding of all of the people, systems, and environme that make up the problem, and as well as an understanding of how these interactions can cause (or expose) different problems.

Some design processes refer to this phase of the process as the "Empathy" stage. Empathy is an essential element of good design and it must remain central to the design process in order to fully understand the context of a problem. You can't design solutions i the abstract -- you have engage problems first-hand, experience their richness, and develop a relationship with the people involve In this course, we include empathy as part of "Understanding the Context" because we not only need to understand the concerns and needs of the people involved, but also how these people interact with other aspects of the problem, and how this interaction will inform potential solutions.

## Unit Goals

In this unit, you will:

> Explore the data collection process.
> Understand the need to explore the context of a problem before trying to solve it.
> Discuss ways to present data to stakehollders.
> Apply data collection strategies in the classroom.

Computational Thinking and Design - Spring 2017

Introduction
**Learning from Data**
Activity: Telling Stories with Data
Core Resources
Discuss: Data in the Classroom
Earn Micro-credentials
Unit Feedback
Course Home

# Learning from Data

As educators, it's likely that you've heard the phrase "data rich and information poor" at some point in your life. This phrase captures the very real risk of collecting a large quantity of data without generating an actual understanding of the problem. In this section, we will walk through a data investigation cycle step-by-step using a real problem as an example.

Imagine that you are asked to build a garden for your school. The garden will grow fruits and vegetables that can be sold at cost to community citizens who need access to inexpensive fresh food. Though hypothetical in this case, this actual scenario is playing out in schools all over the country, and is a great example of digital-age problem solving in action.

## Collection

The first step in the process is to collect data. In traditional research, you start with your question(s) in mind, and attempt to find or collect data that answers your question(s). In a digital-age problem solving context, however, you collect data for the purpose of developing your questions. In fact, you will be collecting additional data throughout the entire design cycle, letting this data help guide any modifications to the questions you are asking. Additionally, designers mix qualitative and quantitative data, and use a variety of datasets to ensure that they can find multiple data points to validate a conclusion or indicate a need.

- Quantitative data are data in the forms of numbers and statistics. Quantitative data may come in the form of data collected from surveys, research, frequency counts, expenses, locations, etc.
- Qualitative data are data that are not typically analyzed numerically, but rather through trends and patterns that emerge in the collection. Qualitative data may come in the form of interviews, observations, written documents, etc.

In our garden example, you might want to collect data about soil and sunlight conditions around the school. You might also seek out public data sources about the average temperature and rainfall at different times during the school year, what crops could be grown given the conditions, characteristics of an ideal garden, and what supplies you might need. However, because designers are also focusing on designing for real people, a designer would want to go out into the community, talk to the people who are actually the ones who will purchase the fruits and vegetables, and find out what they need. This may include asking about their current diet and finding out what fruits and vegetables they know how to cook with. It will also likely involve finding out if this is a service that they would actually need and if there are any potential obstacles to them using it. There are no "wrong questions", and it is always possible to collect more data as new questions arise.

## Analysis

All of these data together would tell you what crops you can grow, what crops you should grow, and what is needed to get these crops out into the community. At this point the second phase of the data collection process begins: data analysis.

In this phase of the process, you process all of the data that you collected. You begin by reviewing all of the qualitative data that you collected and look for themes in the data (this is known as grounded theory). In the process, you would also use a variety of statistical techniques to organize and review your quantitative data. While this can involve more advanced statistics, you can often get what you need from looking at a simple correlation between two datasets. You can learn more about these terms and ideas in the Core Resources section of the unit.

The most challenging part of this process is often in identifying how to link two unrelated data sets (e.g., average temperature and average rainfall) to a third dataset (e.g., ideal crops for a specific location). This will be done differently depending on the data collected, but the simplest process is typically the most ideal. The result will likely be a data set that is incomplete, and you'll find yourself needing to collect more data to incorporate into your analysis before you proceed further.

## Interpretation

After your data have been analyzed, you will turn to the interpretation phase of the data collection process. In this phase, you are looking at your dataset in order to try to figure out what the data are telling you.

In the interpretation phase, it is important to be aware of biases. In traditional statistics, bias is something that we try very hard to avoid. While it's important to not let our own biases influence our interpretation of the data, the opinions, biases and perceptions of our stakeholders are an important and vital source of data that should be included as a part of our data collection and should influence our interpretation of it. When infrastructure projects are built, many people who are in favor of a project don't want the project to disrupt their commute or create too much noise at their house. It's important to hear and consider these concerns in the final design.

In order to generate buy-in from stakeholders in the process, it's important to share interpretations of the data once they have bee collected. These representations may take the form of summary reports, infographics, narrative stories, or through making data available in databases or electronic data structures. And while you may share raw data with your stakeholders, it's more likely that they will be focused on what you've learned from the data and what you've identified as a resulting need. In design, these summaries would be tailored to the specific stakeholder group, and used to both validate your interpretation and to collect additional data.

In our school garden example, it would make sense to share a list of the crops that could be grown compared to the crops that potential clients might want to buy. If food insecurity is an issue in the area, you would want to show data to validate the garden a a need. Additionally, your design could be further informed by other trends that you happened to identify in the process of collecti and analyzing the data (i.e., community health statistics, dietary concerns, etc.).

Because correlation does not equal causation, we would conduct a root-cause analysis at this point to determine if the needs we identify are actually the needs that exist. Oftentimes, in school gardens, community members are reluctant to purchase fresh fruits and vegetables not because they don't need them, but because they don't know how to cook with them. It's critical for a designer t understand these root causes when identifying problems, which will be covered in Unit 3 of our course.

## Action

Once the need has been identified, it is time to use the data to take action. This is the process of using the data to help inform solutions and to evaluate results, which will be covered in much greater detail in Units 4 and 5. From a data collection standpoint, these are additional places to collect, analyze, and interpret data. Once you build your garden, for example, which crops actually grow well and which ones don't? What do people buy? What other needs have you addressed by creating your garden?

FRIDAY INSTITUTE Computational Thinking and Design - Spring 2017

Introduction

Dig in!

Activity: Miles Apart and Kilometers to Go...

Core Resources

Earn Micro-credentials

Discuss: In My Classroom

Unit Feedback

Course Home

# Introduction

*"During my doctor's appointment, the doctor asked me to describe my symptoms, followed by a whole bunch of questions. I the had to go back for lots of tests and speak with a few specialists. Finally, after weeks of waiting, I was able to get a diagnosis."*

This process, though frustrating, is Digital-Age Problem Solving in action. Before rendering a diagnosis, the doctor had to engage a process of data collection, gathering qualitative data from speaking with the patient along with quantitative data from the lab results. Once the doctor had enough data to triangulate possible causes, only then were they able to render a diagnosis.

[ Listen to a Podcast | Read a Transcript ]
Save To My Bookmarks

Rate this Video:
Average of ratings: 4.3 (6 Submitted)

It may seem strange that we're halfway through a course on problem solving and yet we have yet to actually solve anything. It ma also seem counterintuitive that we've broken down collecting data and defining problems into two separate units. The reasons for doing so should become clearer as this unit (and course) progresses. In the last unit, we focused on collecting data along with the importance of being attentive to the many potential sources of data that exist. Now we turn our attention to the essential process c actually translating these data into action. By breaking these out into two steps (and corresponding units), we help ensure that we are collecting data without having to make it fit into our preconceived notions of the situation. However, as always with design thinking, these steps do not exist in isolation and aren't intended to be moved through sequentially. For example, it's likely that at some point in the process you will determine that you need additional data.

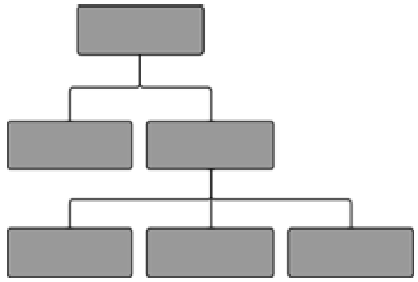
Consider any of the cases we have discussed thus far. Once we have collected data we are going to be much more capable of identifying the actual problem. You may find that you identify one or more "high leverage" problems, which is a problem that can address many or all of the concerns raised by the data. It is also possible that you will identify several smaller problems, with each needing to be addressed individually (or with some link between them). It's *also* likely that the problem is something unexpected c is really just a symptom of a larger problem. In the digital age, problems are often complex and multi-faceted, and require lots of work to identify with precision. And even once you have identified problems, additional data collection will probably be necessary

## Computational Thinking Strategies

Computational thinking provides us with a few strategies to help identify problems:

### Decomposition

Decomposition is the process of breaking down of problems into their smallest parts. Sometimes it's impossible to break a proble down completely, but the general goal is to have each component of the problem addressed separately while also preserving the relationships among different elements. This may take different forms, but may look something like this tree diagram:



### Abstraction

Abstraction is the process of removing extraneous or irrelevant details from a problem in order to make it simpler and more easily understood. Alternatively, abstraction is the act of creating general cases by identifying common features across a variety of situations. For example, consider a new distribution line at a company like Amazon. At most modern distribution facilities, machin deal with containers that hold objects, whereas humans deal with the specific items to prepare for delivery. For the machine, the type of item is irrelevant - it only needs to know the location in which the item is stored. For the human, the the only relevant facto is the size and/or the weight of the item. The specific item - whether it's an HD TV or a 4K TV, for example - doesn't matter for the purposes of packing and shipping. This idea connects directly to the doctor from Unit 1. A person who comes to a doctor's office may have a variety of symptoms that are irrelevant or secondary to an illness. The doctor needs to learn how to discard these fron consideration when making their diagnosis.

## Unit Goals

In this unit, you will:

- Explore the process for defining a problem based on data.
- Conduct a root cause analysis.
- Explore the computational thinking skills of decomposition and abstraction.
- Apply strategies in the classroom.

FRIDAY INSTITUTE Computational Thinking and Design - Spring 2017

Introduction

Dig in!

Activity: Miles Apart and Kilometers to Go...

Core Resources

Earn Micro-credentials

Discuss: In My Classroom

Unit Feedback

Course Home

# Dig in!

*"If I were given one hour to save the planet, I would spend 59 minutes defining the problem and one minute resolving it."* -Alber Einstein (possibly, accounts differ)

In Unit 2, we discussed the process of collecting and analyzing data in order to understand the people, environment, and structure impacting a problem. We have now reached the phase of the process where we actually define the problem(s) we want to address While there are many ways to do this, we have outlined one process for you here.

## Step 1: Conduct a Root Cause Analysis

At the conclusion of the data analysis, hopefully some problems will emerge as the primary ones. The first step, however, is to ensure that the problems that you have identified are the **root problems**. For example, sometimes students perform poorly on test not because of academic ability, but because of environmental factors such as a lack of food or sleep. Thus, academic tutoring ma yield minimal academic gains for the student, as it does not address their actual needs (i.e., the root problem).

A fishbone diagram is a useful tool to help illustrate this process. Below is a partially-completed example. At the "head" of the fish the problem statement that's been defined (i.e., "Bad coffee"). In each block across the top and bottom are the people, environmental factors, and structures at play. They can be general (e.g., people, procedures, etc.) or specific (e.g., Johnny, purchasing department, etc.), and are represented by long diagonal lines with arrows pointing to the center. The straight horizonta arrows pointing towards these lines (e.g., the "bones") are either the symptoms of the larger problem or other problems that are components of the larger problem. The smaller diagonal lines break these components down further. Based on your data collectio continue this process going along until you can stop answering "why" (you may need to still collect additional data).

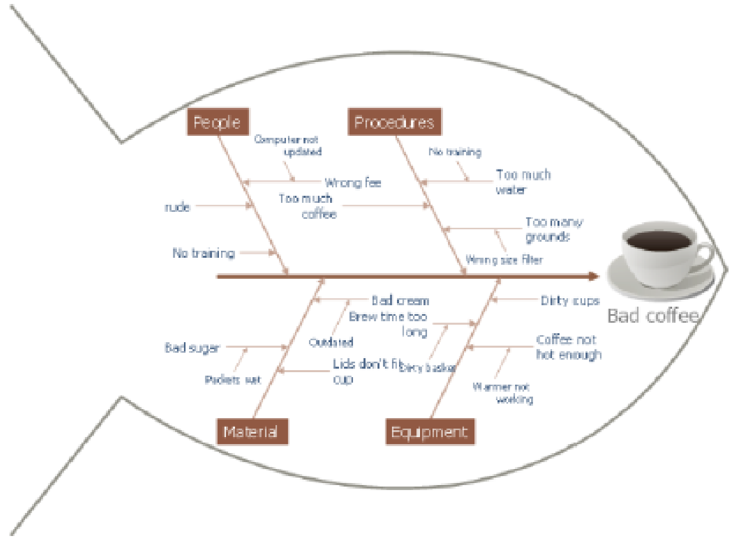



Image via Wikipedia

Although it's counterintuitive, the problems that designers and programmers address are very rarely the ones at the head of the fish. They're the ones along the smallest level of arrows. Some problems are called "high-leverage" because solving a few of them could potentially impact the entire system.

## Step 2: Decide Your Scope

When you begin to identify problems, it becomes increasingly tempting to want to fix all of them. While this might be possible at times, it's likely that you'll have to prioritize those that are most important. There are four key considerations:

- What's the simplest problem to address? Sometimes it can be best to tackle the easiest problem first.
- What problems are the highest leverage? In other words, what problems can be addressed that will have the largest impact o the system as a whole?
- What are your essentials? There are certain elements of a system that must be addressed in order to see any progress in the overall process.

The fourth consideration is a little more complicated. Design thinking is, by its very nature, intended to be unconstrained. Given th designers are encouraged to think of possibilities irrespective of the system as it currently exists. That said, there are inevitably going to be constraints that can't be ignored. With this in mind, It is helpful to define the problems based on whether or not they a in your "sphere of concern," "sphere of influence," or "sphere of control":

- Problems in your "sphere of concern" are problems that impact you but are beyond your control. For example, many designe in schools get bogged down in the issue of insufficient funding. If you were to define this problem as there "not being enoug money," you might quickly find that this problem exceeds your ability to influence it. That doesn't mean that you shouldn't tr but simply that you would want to consider defining your problem in a way that works within these constraints.
- Problems in your "sphere of influence" are outside of your direct control, but can be addressed by people in your network. These problems are usually very easy to address by ensuring all of your stakeholders are at the table.
- Problems in your "sphere of control" are problems where you have immediate access to everything required to begin workin on a solution. This makes solving problems a little easier, but makes it much more tempting (and makes the process move much faster) to forgo stakeholder input, potentially resulting in a lack of buy-in from those who might feel excluded from the process.

In general, most problems will have components across all three of these spheres. As a result, it's important to consider how you might bring the right people into the design process at the most advantageous times.

## Step 3: Write it Down!

Once you have decided on the scope, you can begin to write it down. In this stage, the process itself is more important than the product itself. It's not important for the problem to represented in a refined manner -- oftentimes at this stage a drawing or doodle will completely suffice. The important thing is to summarize your scope along with the elements of the fishbone that you plan to address. Sketchnoting is a popular method in this process, but an outline will work just fine for those who might be less artisticall inclined.

## Step 4: Create User Stories

In Unit 2, we discussed the topic of "empathy," and one way of empathizing in the design process is to create user stories. These stories can help guide our problems to ensure that the personal perspective isn't loss in the process. User stories can be specific (a named individual) or more general. The typical format is *"I am [person] and [problem they're having] because [symptom]. Therefore, I need _______."* For example, "I am a high school student and I have low test scores because I do not have adequate nutrition at home. Therefore, I need access to healthy food on a daily basis." When solutions are developed, they should address your user stories. Multiple user stories will overlap, both in terms of the problems they face, and the needs they have -- that's to b expected and is validation that you're covering a problem from all perspectives.

## Step 5: Decompose and Abstract

Looking across all of your planning documents, the next step is to create work streams to begin to create solutions. This process begins by grouping everything created so far by their common features or characteristics. From this point, they can be broken down. In computer science, this is known as functional decomposition, as problems are being broken down by their component functions. A flow-chart is usually the best representation for this process:

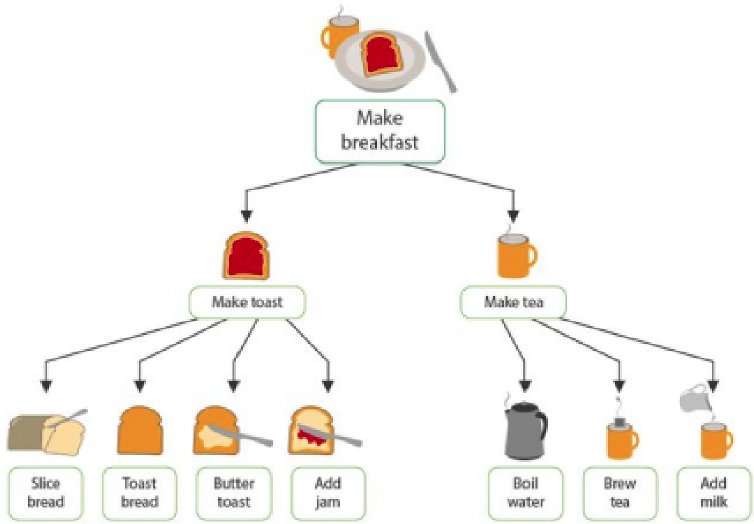



Image via Barefoot Computing

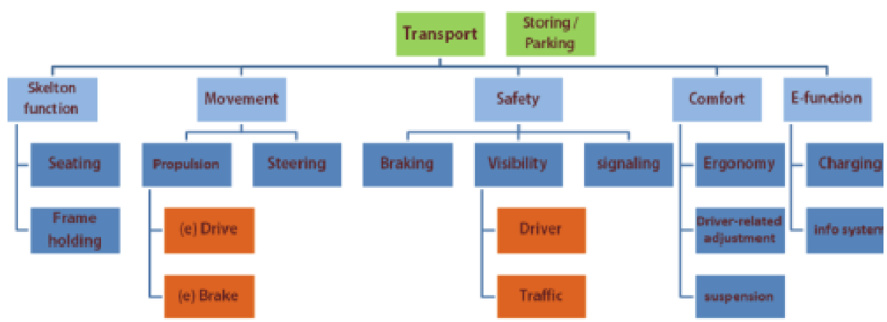



Image via Manish Abraham

In these examples, a functional team could be formed to address some of the elements in these charts (more will be discussed on this in Unit 4).

Part of this process is "Abstraction." In abstracting a problem, irrelevant details are excluded and commonalities between problems are identified (and what type of exceptions exist). For example, if our student is struggling in school because they don't have adequate access to food, we need to create a case for how to ensure that all students have enough to eat. We could decompose th further by identifying that we need to address food access both in school and out of school. While it's important to address the needs of each student individually, creating general cases allows us to create solutions with the farthest reaching impact.

FRIDAY INSTITUTE Computational Thinking and Design - Spring 2017


Introduction

Dig In!

Discuss: Brainstorming

Core Resources

**Earn a Micro-credential**

Discuss: In My Classroom

Unit Feedback

Course Home

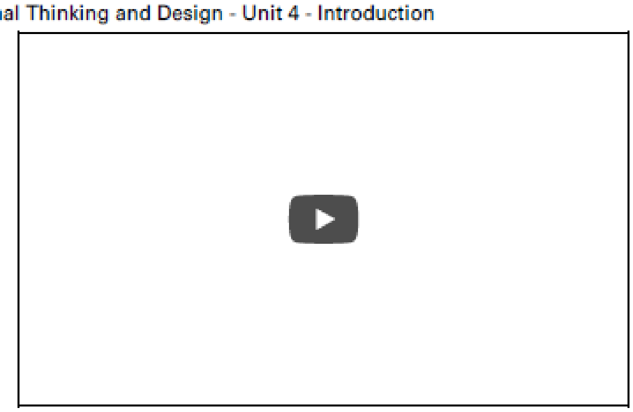

# Introduction

*Bertie County, North Carolina, is predominantly an agricultural community. Despite this community focus, it is also considered t be a food desert. Much of the population lived in poverty, and the entire county had only one grocery store with limited access t fresh, local produce. So local leaders decided to build a community farmers' market run by and for the local population.*

The scenario depicted above isn't a hypothetical one — this was a real design project undertaken by local high school students working on the Studio H program, and was depicted in the award-winning documentary *If You Build It.* After engaging in a process of data collection and problems definition, the students identified this challenging problem to attempt to solve. At each step of the design and build process, the students gathered input from the community, collected more data, and made revisions to their design based on the feedback that they received. It's important to underscore the distinction in this process: in design thinking, you are designing **with** your stakeholders, whereas in traditional design you are designing **for** them.

At this point in the course, we've engaged in a process of data collection and problem definition. Now, it is finally time to start *creating solutions.*

## Computational Thinking and Design - Unit 4 - Introduction



[ Listen to a Podcast | Read a Transcript ]
Save To My Bookmarks

Rate this Video:

Average of ratings: 3.8 (4 Submitted)

The core principle of design thinking is **human-centered design**. In this type of design, you are attempting to solve *real* problems f *real* people. As a result, every solution will be specific to its individual context. For example, a house designed for a large family in rural area would be different than one for an individual person in a dense urban area. The problem that has been identified needs be responsive to the data that has been collected and the people who will be impacted by your solutions. When creating solutions stakeholders should always be closely involved, having the opportunity to provide feedback and suggest revisions. Design thinkin is by its very nature **iterative**, so you should always be revisiting your original findings, collecting more data, asking more questior and refining your design.

In the previous unit, we discussed the computational thinking skills of **decomposition** and **abstraction**. Once problems have been decomposed, you'll then face the challenge of having to address them. In computational thinking, the process of defining which o your procedures will run simultaneously is referred to as **parallelization**.

These steps are are often creative processes in themselves and require thoughtfulness and creativity. In computational thinking, a set of steps to complete a task are referred to as an **algorithm**. In computer science, algorithms are often extremely detailed (e.g., move forward 10 meters, stop, turn right, move forward 5 meters, etc.), whereas with people these steps can be a more general. Either way, all good algorithms reflect a series of steps that any person or machine can implement exactly the same way every tim In the upcoming sections of this unit, you will learn more about designing algorithms to solve real problems with human beings a the center.

FRIDAY INSTITUTE Computational Thinking and Design - Spring 2017

Introduction

Dig In!

Discuss: Brainstorming

Core Resources

Earn a Micro-credential

Discuss: In My Classroom

Unit Feedback

Course Home

# Dig In!

## Designing Solutions

At this stage in the design process, we have invested significant amounts of time and energy in order to understand what our stakeholders need from whatever solution we identify. We have defined these requirements, identified the problems, and decomposed them into approachable tasks. After all of this work, we can *finally* get to create something new. Some design model break out the "designing solutions" component into two separate components: **ideation** and **prototyping**. Since these two components are intricately intertwined, we are including them together.

Beyond everything already shared in the course, there are four key considerations for designing solutions in a design thinking context:

1. **Throw things against the wall:** It's unusual (and potentially problematic) for there to be only one solution developed and agreed-upon solution. The first step in design thinking is to create as many ideas as possible — some of them will be bad, some will be unfeasible, some will be overly complex or will miss some of the requirements, and most will be thrown out. However, when all of the ideas are laid out and reviewed, a small number will probably stand out as promising. But all of the other ideas on the table may serve to improve the standouts, either immediately or later on in the process.
2. **Failure is a learning experience:** Designs fail and not all ideas are workable. However, in design thinking, every failure is an opportunity to refine future designs, learn what doesn't work, and get additional feedback. For that reason, it's important that the team has a shared understanding that no idea is too crazy and that no work is useless or discarded.
3. **Designs are iterative:** In traditional design methods, a finished product is presented for evaluation. In design thinking, ideas a presented in unfinished forms several times during the design process. Feedback from stakeholders is solicited and designs are tested. Changes are made frequently and early in the process.
4. **Use rapid prototyping:** It's easier for people to react to something they can see clearly. Rapid prototyping is a technique whe designers create simple, easy-to-create prototypes to test designs and feedback for future iterations. They may be a set of drawings, a 3D print, a unit of functionality in code, a mock-up of a finished product, etc. The general idea behind rapid prototyping is that you are creating a simplified version of the product for the purposes of getting feedback and making improvements.

This is an intensive process, and it's important for designers to resist the temptation to start building immediately. While this feedback cycle can be frustrating, it is ultimately the most beneficial part of the process (for students and teachers alike). It is important to return to the data and our stakeholders often — to make sure that the things that we build work for our stakeholders, meet the requirements, and solve the problems that we have identified.

## Parallelization

As solutions are being created, it's logical to break them up into small pieces that can be addressed separately or by different wor teams. In computational thinking, the process of identifying tasks that can be run in parallel is called **parallelization**, and tasks are sequenced in one of three ways:

1. *In parallel.* Problems that can be approached in parallel rely on work streams that can be undertaken at the same time becaus the work on one stream won't conflict with another. There are times, however, when the order in which workstreams comple is important, and results can vary if they are not done in a particular order. In parallel tasks, a **race condition** occurs when the result of one or more tasks depends on the order in which they complete.
2. *Sequentially or serially.* Tasks are undertaken one at a time. When one task is complete, the next one can begin. This is generally slower, but is necessary in some cases. Sequential tasks are tasks completed one at a time in a specific order. Seria tasks are done one at a time, but order doesn't matter. Tasks are usually completed serially due to a lack of resources, and ar usually completed sequentially because the order in which tasks are completed is important to the result.
3. *Parallel-sequential or parallel-serial.* This is essentially a combination of the two previous approaches. Tasks are grouped, an each group is completed in parallel. Within the group, tasks may be completed sequentially, serially, or in parallel.

Cooking provides us with great examples for all three scenarios.

1. Cooking meat and vegetables for a meal can be done in parallel (i.e., they usually happen at the same time but in different dishes). If you're using a slow cooker, you may be cooking several things in parallel, but the order of completion does matter (the meat can't finish cooking too much earlier than the potatoes or it may not taste good) which may result in a race condition.
2. Decorating a cake is something that happens sequentially — you have to bake the cake, cool the cake, and decorate the cake in that order. And setting the table is done serially — whether you put the plates out first or the glasses doesn't matter, but y probably don't want to do both at the same time.

3. The entire process of preparing, cooking, and serving the meal is a series of parallel-sequential and parallel-serial events.

## Algorithm Development

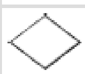
Within each task, it may be necessary to create an **algorithm**. An algorithm is a detailed step-by-step set of rules or procedures to b followed. We follow algorithms every day, whether we are driving to work, baking a cake, grading papers, or playing a board gam Algorithms have two main characteristics:

- **They are specific.** Don't skip steps (also: don't make assumptions). Consider the classic example for teaching about algorithm making a peanut butter and jelly sandwich. While we assume everyone knows that they have to open the package of bread t get the bread out, an algorithm would make no such assumption. Getting a piece of bread from the package involves picking up the package, removing the twist tie, opening the package, reaching in, grabbing a piece of bread, and removing it from th package.
- **They are replicable.** Given identical starting conditions, anyone or anything that implements an algorithm would achieve exactly the same result, every time.

Algorithms generally consist of several elements:

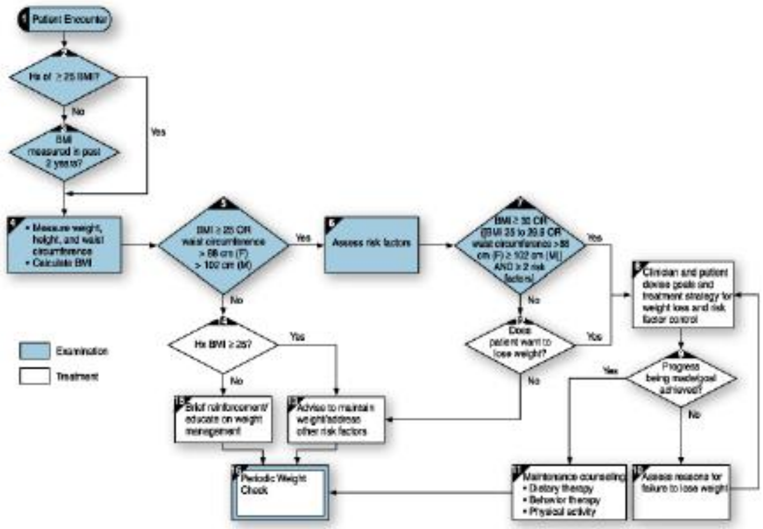
- The individual component steps in the algorithm.
- Conditional statements (if/else). Conditional statements are decision points within the algorithm that allow **branching** — different routes to be taken in the algorithm based on if a condition is true or false. For example, *if the traffic light is red then stop, otherwise, keep going.*
- Loops which define that a series of steps should be repeated either a certain number of times (repeat 10 times), or until a conditional changes (*drive straight ahead until you see a gas station on your right*).

Flowcharts are a great tool for displaying an algorithm. In computer science, the following shapes are used in a flowchart to help t reader understand each element:

| Name | Symbol | Purpose |
|---|---|---|
| Oval or Circle |  | Denotes the beginning of an algorithm; may also be used to denote the end |
| Double Box |  | Denotes the end of an algorithm or a stop point |
| Flow Line/Ray |  | Indicates direction of the flow of the algorithm |
| Parallelogram |  | Represents input or output received in the program |
| Rectangle |  | Denotes a process that the program will carry out |
| Diamond |  | Denotes a conditional decision point; typically will have one flow line entering and two leaving (on for a true condition, one for false) |



Consider the following medical example below. A doctor may use this algorithm to determine if the patient is at risk for weight-related illness. The doctor will begin by assessing the BMI and taking measurements if needed. Based on risk factors, they'll make decisions about how best to advise the patient.




* This algorithm applies only to the assessment for overweight and obesity and subsequent decisions based on that assessment. It does not include any initial overall assessment for cardiovascular risk factors or diseases that are indicated.

Image via Wikipedia

**Algorithmic thinking** is a critical component of computer science (and arguably math and science in general). They general princip of algorithmic thinking is that nothing happens by accident or by magic — that every output is a result of clearly defined inputs following a clearly defined process. In mathematics, for example, it is useful to both identify the type of problem to be solved, and from that data, identify the specific step-by-step method to be used (for example, quadratic equations):

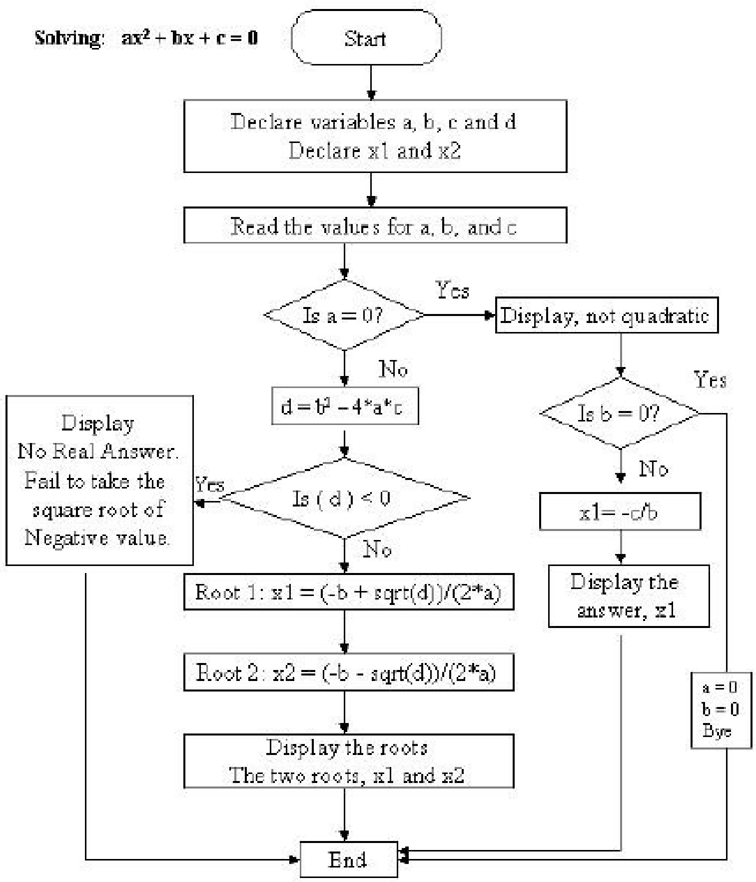



Image via Chad Sprensky

FRIDAY INSTITUTE Computational Thinking and Design - Spring 2017

Introduction
Dig In: Building a Testing Plan
Activity: Learning From Failure
Core Resources
Earn a Micro-credential
Discuss: In My Classroom
Activity: Putting It All Together
End-of-Course Survey
**Download Your 10-Hour Certificate of Completion**
Course Home

# Introduction

*The failure of the Samsung Galaxy Note 7 is one of the most significant product failures of the past few years. The Note 7 was widely anticipated, expected to be extremely popular, and heralded as a step forward in mobile phone technology. However, shortly after customers started receiving their phones, reports started to emerge of phones catching on fire. Despite a series of countermeasures and recalls, phones continued to explode. Making the problem worse was Samsung's inability to recreate the failure in their labs and to identify a resolution. They were forced to recall and scap the phones at a cost of billions of dollars. Th phones were banned from air travel (and airlines had to purchase special fire-containment gear in case someone brought one o board anyway) and cellular providers eventually banned them from their networks.*

While Samsung did a thorough job of testing the components of the phone (and the phone itself), they didn't do so in real world conditions — they failed to meet the requirements of the **user** of the phone. This failure cost Samsung significantly, and could potentially have compromised their standing in the mobile phone market.

## Computational Thinking and Design - Unit 5 - Introduction

[ Listen to a Podcast | Read a Transcript ]
Save To My Bookmarks

Rate this Video:

Average of ratings: 4.5 (2 Submitted)

In education, assessment has a stigma attached to it that we're going to ask you to put aside for this unit. The stigma of testing an assessment in education and in other fields make testing and assessment the most maligned, misunderstood, and overlooked element of the design process. Designers are often tempted to skip testing and move on to the next thing, and can sometimes be afraid of the failures made visible in testing. However, rigorous testing enables designers to create the best possible product and create something that people are excited about using.

In a design thinking context, assessment is **informative and iterative**, versus traditional assessment, which is **summative.** In desigr thinking, failure is an opportunity for growth. With each testing and feedback cycle, there's an opportunity to iterate and tweak the design to make the product better with each round.

While testing and assessment is the last unit in our course, it isn't the "final" component of the design process — it's simply the ne phase. After testing, we may decide our product needs to be different or we may need to collect more data, in which case we woul move back to those particular phases of the design process. The goal is to repeat the process until you have something that is don and ready for use. During testing, you should also be collecting additional data. In seeing how users interact with the product or service, we will have additional data that may encourage us to modify our design or original conclusions from our data collection.

Another key component of design thinking is that solutions should be easy and obvious for end users to navigate, so testing with actual users is vital. For this reason, many software companies increasingly turn to public beta testing for additional feedback.

There are two computational thinking skills that are useful in evaluation:

> **Simulation** allows us to create models of our real-world solutions that are realistic but easier to create and implement. Simulation allows for testing of a solution to be completed, while still allowing the design to be easily tweaked if needed. Examples of simulations include model making, bench testing, and 3D printing.

> **Automation** is the ability of machines to complete repetitive tasks. Automation is both a way of creating a solution and an evaluation measure. As a solution, we offload tasks to a machine or system of machines to be completed. But we can also us machines to test solutions (consider a probability spinner to test for randomness) or a rig to test a new farming system. Automation usually allows reparative tasks to be completed faster and more consistently than can be done by humans.

Introduction
Dig In: Building a Testing Plan
Activity: Learning From Failure
Core Resources
Earn a Micro-credential
Discuss: In My Classroom
Activity: Putting It All Together
End-of-Course Survey
Download Your 10-Hour Certificate of Completion
Course Home

# Dig In: Building a Testing Plan

While testing is introduced in the final unit of the course, it is important for designers to enter the design process thinking about testing. In a design-thinking context, a project isn't considered "complete" until it has gone through several rounds of testing and revision. A comprehensive testing plan helps ensure that the product will work as anticipated and meet the user requirements. In computer science, there are five phases of testing: unit testing, integration testing, system testing, acceptance testing, and regression testing. When a design fails testing, it will usually be due to one of three failure types: failure of requirements, failure of design, or failure of implementation.

A key differentiator in design thinking is that designers use testing to refine their ideas and products and create better outcomes. Testing helps us identify and improve our systems — a "failure" does not indicate a deficiency or inadequacy of the system, but rather an opportunity for making the system better. Additionally, in all digital-age problem-solving contexts, testing and evaluation is formative — it should be done early, often, and consistently. Because evaluation is a key to iteration, we never want to go too fa into a project before conducting real-time evaluations.

## Testing Types

### Unit Testing

Unit testing is the most granular level of testing. Unit testing is where each individual component of the final system is tested in isolation. This is a way to ensure that each individual component is functioning as anticipated. In computer programming, each individual function will undergo unit testing. In auto mechanics, this is where each individual component of the car would be teste individually to ensure that each component is working as expected. In lesson planning, we often do unit testing to test each of the elements of our lesson plan to ensure that they are quality activities that accomplish our learning objectives.

What unit testing looks like will vary based on the type of application. But consider a simple programming function that determine if a number is odd or even. A unit test plan would be similar to the table below — three columns, one for the input, one for the expected result, and one for the testing result. In a different type of system, you could substitute "input" and "output" for other similar words (i.e., "stimulus" and "response", "action" and "result", etc.). The key components of a reliable unit test remain the same:

- **Test error cases.** 0 is technically an even number, but it's an unusual case that needs to be tested. A non-integer value should also produce an error.
- **Test a variety of inputs.** Testing 2, 4, and 6 are valid, but what happens if you input 2,922,423,355,512?
- **Test all possible cases.** If your function has cases to output "odd", "even", "error", and "not an integer", you should develop te cases that will produce each result.
- **Try to break it.** In a function like the one described, could the computer round the number? It's important to test to make sure

| Input | Expected Output | Actual Output |
|---|---|---|
| 0 | Even | |
| 2 | Even | |
| 4 | Even | |
| 2922485884444 | Even | |
| -2 | Even | |
| -5847684416358 | Even | |
| 3.1415 | Not an integer | |
| 5416584.28588 | Not an integer | |
| 5.9999999999999999999999 | Not an integer | |
| 3 | Odd | |
| 3554576327 | Odd | |

| | | |
|---|---|---|
| Hello world! | Error | |

## Integration Testing

Integration testing is the process of testing the larger system by connecting several "functional units" together. In our auto mechanics example, integration testing would test that an entire subsystem of the car works as it's supposed to (i.e., if all of the parts of the transmission pass unit testing, does the entire transmission function as expected in unit testing?). The testing scenario design for integration testing are very similar to unit testing. Just as in unit testing, it's important to test all possible outcomes for the program as well as anti-examples (when users do something wrong or unexpected). Additionally, in integration testing, it is possible for errors to be introduced in the interactions between functions, which may make them harder to diagnose. Consider the space shuttle example from Unit 3, where one function expected metric units and the other expected Imperial. Each would have passed a successful unit test, but would have failed an integration test. It is also possible for each activity in our lessons to be grea but for the entire lesson to not come together as well as we had hoped.

## Requirements Testing (System and Acceptance Testing)

In computer science, requirements testing is split up into two components — system and acceptance testing. They're the same, except one is done by the designer and the other by the end user. In a digital-age problem-solving context, this would be done collaboratively, even if separately.

Requirements testing ensures that the final system meets the initial requirements. In Unit 3 of the course, we discussed "defining the problem" and decomposing that problem into smaller elements to be solved. Requirements testing compares the design and functionality of our solution to our initial requirements and helps us determine if the design we have created actually solves the problems we have identified.

User stories are an essential component of this process. Ideally before the solution is created (so as not to bias test planning), a series of stories are created that define the way a user would interact with the system. For each user action, the story identifies the expected reaction from the design. It's important to design user stories for all possible users, and our empathy maps are great too for creating these stories. User stories usually follow a format such as "I am a _________, and I need to do ____________, so I _______________ and ___________ should happen." There may be many of these cards for each type of users, including failure cas (what happens if a user can't or shouldn't do something they may want to do). Getting feedback on user stories directly from the end user is critically important in the design process to identify both missing requirements and failed user tests. In a lesson plan, a requirements test is the test of student understanding, administrator evaluations, self-assessments, etc.

## Regression Testing

Since design is iterative by nature, it is possible that something that was working is broken by changes elsewhere in the system. Regression testing is a plan, usually involving re-running integration tests and requirements tests, to ensure that no new failures a introduced by a change in the system. Using our car example, when we are repairing our car transmission, we may accidentally disconnect the windshield washer cable.

## Failures

While failures and iteration are important elements of the design-thinking process, understanding the reason behind a failure is essential to resolving it.

The first two failure types are "miscellaneous cases":

- **Failure by design:** Especially in designs where safety can be at risk, some systems will be designed to fail if a certain series of conditions occur. But also, even in simpler systems, designers might choose to build in failure points for user experience or f a variety of other reasons.
- **Failure of assessment validity:** Your system may or may not work as expected, but it fails testing because the expected results of the test are not correct. Sometimes this may be a lack of understanding or a mistranslation of how the system is supposed to work for the purpose of generating test cases, or a typo or similar error can create an assessment which is invalid. Assessments may also not accurately measure the system they are testing. Additionally, many unit and integration tests are automated so that a computer may perform many tests quickly and automatically. An error in the testing system can cause a false error in the unit test results.

The following three failure types are failures at the different layers of the design process, starting with the most granular:

- **Failure of implementation:** This typically occurs at the unit- or integration-testing phase. A failure of implementation means that the design of the system may be correct, but an error in one of the functional units (the implementation of the design) is causing an error.
- **Failure of design:** The requirements may be gathered appropriately, but the design created is flawed or does not meet the requirements or solve the problem.
- **Failure of requirements:** This is the most common failure, and the hardest one to identify. A failure of requirements occurs because our initial data collection and problem definition was missing key data or incomplete, designers didn't correctly identify the problem from the data, or the data we gathered was not clear enough to develop a workable solution to the

problem. New requirements may also come to light based on unforeseen complications or the system may work as expected but be non-obvious to the end users.

Once the failure type has been identified, it's important that we keep our designs and everything we've learned, make a plan to address whatever issue we have identified, and iterate. If our design or requirements are invalid, it may be necessary to return to a earlier phase of the process and begin again. It's perfectly reasonable in a digital-age problem-solving process that our requirements will be incomplete to start with, and we may iterate on a flawed design several times before arriving at a solution tha works well. However, good designers will also be aware of **scope-creep**. With each iteration, stakeholders will be tempted to add additional requirements to any system because it's an "incremental change". However, mainly through experience, good designer: learn when a requirement is a reasonable and necessary addition to a project, and when it simply moves to a new project.