5-2021

Using Scrum to Teach Standards-Based K-12 Computer Science: A Prosepectus for a Master’s Level Methods Class at Buffalo State

Noah M. Pierce
State University of New York College at Buffalo - Buffalo State College, noahpierce125@gmail.com

Advisor
Dr. Steve Macho
First Reader
Dr. Steve Macho
Second Reader
Dr. Dennis Mike
Third Reader
Dr. John Earshen
Department Chair
Dr. Steve Macho, Chair & Professor of Career & Technical Education

Using Scrum to Teach Standards-based K-12 Computer Science

By

Noah Pierce

An Abstract of a Master's Thesis in Career and Technical Education

Submitted in Partial Fulfillment

Of the Requirements

For the Degree of

Master of Science in Education

December, 2020

SUNY/Buffalo State
Department of Career and Technical Education

Using Scrum to Teach Standards-based K-12 Computer Science

Abstract

Computer Science has been increasingly prevalent in K-12 education in recent decades. Most Americans believe that Computer Science is as important as other skills taught in school; further, parents are putting pressure on districts to offer Computer Science programs (1.1). To meet this demand, many teacher preparation programs are adding Computer Science Education to their offering of degrees. This thesis investigates Agile and Scrum product development as a potential method of Computer Science instruction, explores the standards relevant to a Computer Science teacher, and offers a prospectus for a new Graduate Level Methods class to prepare Computer Science teachers to utilize the Scrum framework in standards-based instruction at the K-12 level (1.3). To create the prospectus, research from peer-reviewed articles, case-studies, and implementation guides relating to the topics of Scrum and Computer Science standards are reviewed. The implementation, validity and importance of Scrum, and its educational variant eduScrum, are compared based on the roles, rituals, and artifacts utilized in each framework. The results justify eduScrum as a valid method for problem-based, constructivist Computer Science instruction (2.10-2.12). The background, validity, and importance of three sets of Computer Science standards (K-12 Computer Science Framework, NYSED, and ISTE) are explored (3.1-3.3). These standards were selected for their relevancy to Computer Science certification in New York State and the support of industry, professionals, and lawmakers. The results justify the inclusion of all three standards as crucial to curriculum in New York State (3.4). The thesis culminates in the creation of a prospectus for the Student Learning Objectives and structure of a Methods of Computer Science Instruction class at the Graduate level (4.1-4.4). The SLO's are created utilizing Bloom's taxonomy (4.1). The prospectus recommends Scrum in the creation of Learning Segments utilizing relevant standards, topics, concepts and research literature. The prospectus models Scrum at all levels and is a valid way to teach constructivist, problem-based learning (4.2). More research is needed on the effectiveness of Scrum with low performing students, the use of eduScrum at the K-12 level and the implementation of the prospectus as a class at SUNY/Buffalo State.

SUNY/Buffalo State
Department of Career and Technical Education

Using Scrum to Teach Standards-based K-12 Computer Science

By

Noah Pierce

A Thesis in
Career and Technical Education

Submitted in Partial Fulfillment
Of the Requirements
For the Degree of
Master of Science in Education

December, 2020

Date of Approval:

____________________ ______________________________
Steve Macho, Ed.D.
Associate Professor and Chair
Department of Career and Technical Education

____________________ ______________________________
Kevin J. Miller, Ed.D.
Dean
The Graduate School

Project Committee Signatory

Steve Macho, EdD
Chair of the Committee
Associate Professor and Chair
Department of Career and Technical Education

John J. Earshen, PhD
Associate Professor and Program Coordinator
Industrial Technology Program
Department of Engineering Technology

Dennis G. Mike Ph.D.
Emeritus Associate Professor
Department of Elementary Education, Literacy, and
Educational Leadership

Acknowledgements

I would first like to thank my advisor, Dr. Macho. His thought-provoking questions always provided me with new avenues of inquiry and his caring demeanor gave me the self-confidence I needed to complete this project.

I would like to show my gratitude to the Academy of Process Educators and Lean Dog for their support of my research efforts. They provided me with a wealth of information and resources that guided the research of this thesis.

I would like to thank my friends, family, and colleagues, who supported me throughout this project. Thank you for being my cheerleaders during this process. Thank you to Jill and Barrett at the Buffalo State Writing Center for helping me revise my writing.

Most importantly, I would like to thank my partner Kristina Rovison. She provided me with the counsel to guide my writing, the knowledge to fix my mistakes, and the courage to continue working when I felt like giving up. Without her, this thesis and my life, would feel incomplete.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1: Introduction to Research

1.1 Introduction

Computer Science classes have grown in popularity across American secondary schools in recent years. Many see developing Computer Science skills as an essential step towards becoming an informed and productive citizen. According to a 2015 Horizon Media study, Americans believe computer science is as important to learn as reading, writing, and math; in fact, most parents want their child's school to offer computer science classes (K-12 Computer Science Framework, 2016, p. 12). This has led to an increased demand for qualified computer science teachers.

Computer Science Education is a relatively new field. As such, not many colleges offer programs that specialize in this field. With the introduction of the K-12 Computer Science framework and the rumors of a Computer Science Education certification in New York State, Buffalo State is in the process of creating a Computer Science Education program within its existing Career and Technical Education department. This Master's class should train teachers in how to utilize the Agile methodology of product development to teach standards-based Computer Science classes.

There is a wealth of research proving Agile's effectiveness at managing software development and its prevalence in different industries. Scrum is the most common Agile framework to be used in industry. A form of Scrum, eduScrum, was developed to apply the Agile methodologies of the Scrum framework to a problem-based, constructivist pedagogy. In this thesis, the researcher will explore the different Computer Science standards and how eduScrum can be used to teach them. This culminates in a proposal to Buffalo State on how to incorporate eduScrum and Computer Science standards into the curriculum for its Master's level Methods of Computer Science Instruction class.

1.1.1 Search Terms

There will be several terms used to search for literature. The basic terms for this thesis are Scrum, Computer Science, and Standards. There are several search terms related to each of these. Combinations of different search terms may be used for different inquiries. The core search terms and related search terms are outlined in the table below:

Educational	Industry
Methods	Agile
Standards	Scrum
Computer Science	Kanban
Classroom Management	User stories
High School	Cards
American Schools	Sprints
Grades 9-12	Iteration
Assessment	Stand-up
Inquiry Based Learning	Backlogs
Project/Problem Based Learning	Retrospective
	Pair Programming

NYSED Computer Science and Digital Fluency Standards K-12 Computer Science Framework ISTE Professional Development eduScrum	Test-driven Development Velocity

1.1.2 Definitions

- Independent Learning - Learning done outside of a structured academic setting such as a high school classroom.
- Effectively structure - The curriculum and implementation most likely to give each student a higher chance of success in the area of instruction.
- Teacher - Mentor, whether in a formal school environment or someone who takes a protege's learning upon themselves.
- Secondary School - American schools containing the grades 9-12.
- Computer Science - All subsections of the study of computers and society's interactions with them. Use interchangeably with Programming, Information Technology, and Computational Thinking.
- Methods - In this context, Methods refers to the methodology and pedagogy of teaching Computer Science. The intention for this thesis is to describe the methods most appropriate for a Master's Level course.

- Agile - Also known as Agile Project Management and Agile Product Development, are a set of beliefs and practices used by the Computer Science industry to increase productivity that was first out lined in the Agile Manifesto of 2001. May be used interchangeably with Scrum in some areas.
- Scrum – A methodology of project planning and implementation used in the Computer Science industry. Is sometimes combined with other methodologies such as Kanban and DevOps, and is considered part of the Agile paradigm. May be used interchangeably with Agile in some areas.
- SLO – Student Learning Objective, or the expectation of what a student should walk out of the class knowing.
- Learning segment – A collection of 3 to 10 lessons that explore a designated topic. These lessons should build off one another and culminate in a project, assessment, or presentation. The exact length of the learning segment required for the students to receive credit in the class outlined in Appendix A is to be determined by the professor teaching the class.
- Rationale – An explanation and justification of the choices made during a particular project.
- Buffalo State – The State University of New York at Buffalo State.
- Waterfall - A project management technique where decisions are made by the project managers at the beginning of the development process and implemented by the developers to completion in the development process.

1.2 Research Questions

In this thesis, the researcher explores the following questions:

- What are Agile and Scrum, and what potential application could they have to a K-12 Computer Science classroom?
- What are the Computer Science standards relevant to New York State K-12 Computer Science education?
- What should a class preparing future educators to teach Agile and Computer Science standards look like at the Master's level?
- What Student Learning Objectives address these needs?

1.3 Method

The method of this thesis will be an integrative literature review of the academic journals and peer-reviewed articles relating to the research questions. All sources will be obtained through the State University of New York at Buffalo State library or the internet.

For the research process, information will be pulled from sources within the scope and criteria below. Chapter 2 of this thesis will discuss Agile, Scrum and eduScrum and their potential applications to the classroom. Chapter 3 will discuss 3 sets of Computer Science and Technology standards relevant to NYS Computer Science teachers. Chapter 4 will model how a Computer Science Education Master’s class could incorporate instruction of Agile methodologies and the standards discussed in Chapter 3. Chapter 5 will discuss the potential merits, pitfalls and options for future research derived from the earlier chapters.

1.3.1 Inclusion Criteria

All articles reviewed for this thesis are relevant to the questions posed and obtained ethically. Both qualitative and quantitative studies may be included. Sources used for this information have been published after January 8, 2002. This date marks the signing of the No Child Left Behind Act, a significant event in the American education political landscape.

1.3.2 Ethical Issues

Any major ethical issues involved in this integrative literature review would deal with information collection or dissemination. All information referenced in this thesis will be obtained legally. This information will be reported accurately and properly cited. If confidentiality is an issue for any subject discussed, appropriate means will be used to protect the privacy of the individuals and organizations involved with that subject.

1.4 Limitations and Assumptions

This thesis will limit information sources to only those accessible through the Buffalo State library circulation system and the internet. The thesis will limit its scope to American public and private schools. This thesis is not focused on the individual concepts taught in Computer Science classrooms; rather, the thesis explores methods of instruction for these concepts and the standards relevant to that instruction.

This thesis assumes that information in literature is cited correctly. Information provided in literature is assumed to be provided without malicious intent. This thesis assumes that the NYSED Computer Science and Digital Fluency Standards, K-12 Computer Science Framework and ISTE standards are the frameworks that most Computer Science classes will utilize when implementing their Computer Science curriculum. It is assumed that all of these frameworks are valid. The only standards used from these frameworks are those that apply to students and teachers. Since the NYSED Standards are currently in their final draft, it is assumed that the final version will be very similar to the draft and the standards cited in this thesis will be present in the final version. Crosswalks between different frameworks are based on the personal judgement of the creator of this thesis. The crosswalks are not meant to explain relationships or interactions, but to highlight areas and topics of interest. The purpose of these crosswalks is for activity design and is considered exploratory in nature.

It is assumed that the common use of frameworks within the Computer Science industries validate them as effective. While these frameworks have been developed for environments other than the classroom, it is assumed that the increased productivity gained by these frameworks would be beneficial to the Computer Science classroom as well. The Student Learning Objectives and curriculum for the Computer Science Methods class are created at the discretion of the author.

The Master's Class modeled in Chapter 4 of this thesis is based on a class containing teams of five students. It is assumed that the workload for the students is appropriate and that previous classes students have taken will prepare them to complete learning segments to the standards expected by Buffalo State. The class is a general recommendation for the structure of a Master's level Methods class, and is not intended to meet all requirements of Buffalo State in their creation and implementation of new classes. The length and standards of these learning segments are up to the discretion of the professor teaching this class. It is assumed that students will benefit from experiencing eduScrum roles, and that learning the eduScrum process while participating in it is the most effective way for them to learn.

Chapter 2: Agile, Scrum, and eduScrum Frameworks

2.1 Background

Historically, the software development industry applied a top down approach to their project management. A project manager or software architect would determine the features and design of a system and the developers would create it. This approach is referred to as Waterfall, for the way information and decisions flow from the top of the organizational chain to the bottom. As described by Lei, Ganjeizadeh, Jayachandran, and Ozcan (2017), the "…Waterfall model assumes that the team has nearly perfect information about the project requirements, the solutions, and ultimately the goal" (p 59). The preplanned approach to system design ultimately increased the cost of the projects and made them inflexible to changing requirements. Stakeholders in the process became aware of the inflexibility of this process. As stated by Lei et al., "…it had become evident that the approach lacked effectiveness in addressing the needs of customers, managing rapidly changing scope, delivery time, and cost of the project" (2017, p. 59).

Lei et al. go on to describe how this led to the development of the Agile movement. The principles of Agile were first outlined in the Agile Manifesto of 2001. Since then, many frameworks have been created that implement the principles outlined in the Manifesto. The first, and most popular of these, is the Scrum framework (Lei et al., 2017). Chapter 2 will explore the elements, principles, and applications of the Agile methodology and Scrum framework in industry and in pedagogy.

2.2 Scrum Framework

As described by Lei et al., the Scrum framework "…is a project management methodology that uses iteration and implementation" (2017, p. 60) and is based on the following three principles:

- "Transparency: The process must be visible to everyone who is involved in the project" (p. 60).
- "Inspection: Scrum users must inspect Scrum artifacts frequently to detect problems in early stages" (p. 60).
- "Adaptation: If an inspector determines that some aspects of the project are unacceptable and outside of the project scope, the process can be adjusted to avoid further problems" (p. 60).

These principles are incorporated into the roles and rituals of the Scrum framework.

As can be seen in the principles above, all stakeholders are involved in the creation and implementation of the project. The members of the team are normally divided into separate roles. Scrum is made up of roles, artifacts, and rituals discussed in Sections 2.10-2.11.

2.3 Application to Education

Table 1: Main differences between traditional behavioral learning and Agile constructivist learning. (López-Alcarria et al., 2019, p. 7)

Term/Concept in PM Environments		Term/Concept in Education Environments
Customers as beneficiaries of a product or a service	⇔	Students as beneficiaries of an education service, and in lower degree, parents and other stakeholders.
Project product	⇔	Knowledge, competencies, attitudes and skills acquired by the student.
Partial Releases	⇔	Completed didactic units.
Final Releases	⇔	Final result delivered to the student when the educational program or course is over in terms of achievement of predefined requirements.
Detailed plan	⇔	Syllabus and didactic units and their execution.
Strict contracts	⇔	Over-specified and rigid syllabus and course planning leaving little autonomy to instructors.
Control Parameters	⇔	Assessment parameters (both external and internal).
Refactoring model	⇔	Plans to continuously improve the educational programs (ideally applied during the courses).
Basic functionality	⇔	List of competencies and basic objectives of the educational program, course, didactic unit, course session, assignment or activity.
Minimum Viable Product	⇔	Minimum expected result of every educational program, course, didactic unit, course session, assignment or activity (in terms of grades and Competencies obtained by students).
Scope of the project	⇔	Scope of every educational program, course, didactical unit, course session, assignment or activity.
Providers	⇔	Providers of educational services (public administration, legislators, instructors, other staff, etc.).
Partner	⇔	Students and their families, other stakeholders collaboration in the educational service such as private companies, foundations, NGOs, etc.
Project delays	⇔	Delays in the execution of didactic units, course sessions, assignments or activities that are finished without meeting the predefined objectives. Any deviation when putting into practice an educational program or a course.
Disagreements between providers and customers	⇔	Conflicts between students and/or their families and faculty about the quality and misalignments between the expected results and the final outcome.
Resisting the change	⇔	Rigid syllabus and educational plans hard to modify. Complex bureaucratic process, slow administration, instructors reticent to adopting new pedagogies, non-participative students.
Agreed Specifications	⇔	Communication between instructors and students about the objectives of the educational program, course, didactic unit, course session, assignment or activity so that the objectives and expected results are clear and known by all stakeholders.
Team structure	⇔	Composition of teams inside an educational project (teams of instructors, teams of students, teams of parents).
Hierarchical teams	⇔	Hierarchical classrooms with instructors as omnipotent and inaccessible figures.
Project manager	⇔	Instructor, program coordinator, Principal.
Micromanagement	⇔	Over specification by the instructors of the syllabus and students' way to work and learn.
Project/Company Quality Standards	⇔	Education quality standards of the instructor, department, school, city, region or country.
Waterfall Project Management	⇔	Educational project management model based on the exhaustive definition of the courses, didactic units, course sessions, assignments and activities prior to the execution, leaving little or no space to modification of the time-boxed sessions and their content. Typically outlined using a Gantt chart.

Agile Software Development is often associated with constructivist learning theory. According to López-Alcarria, Olivares-Vicente, and Poza-Vilches, "…adopting an agile approach in education can be linked to the experiential learning theories of Dewey, Kolb and

Piaget, which all state that knowledge develops as a result of direct experience" (2019, p. 10).

Table 2: Linden's Interpretation of the Agile Manifesto Principles for the Student-Centered Learning Environment. (Linden, 2018, p. 67)

Agile Manifesto Principles (Beck et al., 2001)	**Our Interpretation of Principles in the Student-Centered Learning Environment**
"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."	Students continuously complete tasks and provide deliverables for assessment by teaching staff.
"Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage."	A student may need to adjust his/her learning style to achieve the subject learning outcomes within the expected timeline.
"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."	Due to a semester being a strictly 12 weeks period, students' deliverables should happen from every week to every fortnight.
"Business people and developers must work together daily throughout the project."	A student and teaching staff need to work together for the student to achieve the subject learning outcomes.
"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."	Teaching staff expect students to be motivated to learn and should provide a challenging and exciting learning environment which supports student-centered learning.
"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."	Although a variety of methods are used for modern communications, there should be face-to-face communication opportunities where students support each other as well as get necessary instruction and help from teaching staff during formal and informal sessions.
"Working software is the primary measure of progress."	Students' learning is judged by the quality of the deliverables.
"Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely."	The subject should be designed to maintain a constant learning pace, with regular deliverables. Instead many subjects seem easy in the first 3-4 weeks with an unexpected jump in difficulty levels closer to the middle – end of the semester rapidly increasing workload and amount of stress for students (as well as marking workload for teaching staff).
"Continuous attention to technical excellence and good design enhances agility."	Teaching staff should provide valuable feedback on improvement opportunities and students should be given an opportunity to implement and deliver improved versions of their work as evidence of their learning.
"Simplicity – the art of maximizing the amount of work not done – is essential."	Students should be able to see how much more they could learn beyond the constraints of the subject.
"The best architectures, requirements, and designs emerge from self-organizing teams."	Students being in charge of their learning approaches often find a "study buddy" to maximize their learning, they decide on getting support based on their individual needs.
"At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly."	At the end of each semester students should be encouraged to reflect on their learning approaches to ensure that they learn from the mistakes and adjust their learning style and time management for the next subject.

Lopez-Alcarria et al. go on to describe problem-based learning as a constructivist-learning paradigm (2019). In problem-based learning, "…small groups of students engage in cooperative learning and collaboration to solve complex problems in an authentic project context" (El-Khalili, 2013, p. 1). In the context of education, Agile and Scrum are used as an implementation of problem-based learning. As observed by Linden, "…students' learning needs are affected by many variables and therefore educators should consider Agile teaching approaches" (2018, p. 66).

López-Alcarria et al. (2019) further elaborate on how an Agile based classroom compares with a traditional learning environment: the role of professor as a facilitator of the learning process, the continuous evaluation, and the flexibility to students' interests and performance. The researchers went on to detail how Agile project management terminology relates to an educational setting. The results of this comparison are viewable in Table 1. The researchers reference several other authors' attempts to apply the values of the Agile Manifesto to the field of Education. López-Alcarria et al. (2019) detail these attempts:

- Kamat's Agile Education Manifesto
    - Teachers and Students over Administration and Infrastructure (p. 10)
    - Competency and Collaboration over Compliance and Competition (p. 10)
    - Employability and Marketability over Syllabus and Marks (p. 10)
    - Attitude and Learning skills over Aptitude and Degree (p. 10)
- Peha's Agile Education Manifesto

- Individuals and interactions over processes and tools (p. 10)
- Meaningful learning over the measurement of learning (p. 10)
- Stakeholder collaboration over constant negotiation (p. 10)
- Responding to change over following a plan (p. 10)

Linden interpreted each of the Agile Manifesto Principles through the lens of a Student-Centered Learning environment. The results of their interpretation are detailed in Table 2.

2.4 Implementation of Scrum in Education

2.4.1 Swinburne

During the 2018 school year, Swinburne University of Technology adapted Scrum to "…teaching and learning in the context of the self-regulated learning framework" (Linden, 2018, p. 66). As described by Linden, their interpretation was created to coincide with Young's Social Cognitive Framework for Self-Regulated Learning (2018). As for the role described above, the students would play the role of developers and the professors would serve as the customers. There is no mention of Scrum Master or Product Owner. In their system, students would not be graded on the quality of submitted work, but would get detailed written or verbal feedback from their professor. The student would not get credit for the assignment until the professor marked it as complete (Linden, 2018).

Linden (2018) further elaborates on the grading system implemented. Grading is determined by the difficulty of task attempted by the student. The tasks were rated as pass (P), credit (C), distinction (D), or high distinction (HD) level. Each student would set a goal for themselves for what level they wanted to achieve and were given tasks on their learning platform based on their goal. The students must also complete two closed book tests to ensure their understanding of the material (Linden, 2018).

This lead to some interesting results for the researcher. Students were reaching for higher than they could achieve and would need to backtrack their expectations. As Linden wrote:

> Most students start with aiming at high distinction which reflects on their goal-orientation behavior and their perceived competence. Those who are mastery-oriented usually keep this goal throughout the semester and take action to achieve it. If they scale back, it is usually to distinction level. Students selecting high distinction for ego-social reasons often do not achieve this level when they discover that the learning curve is steep and the tasks are getting more difficult from one week to another and require constant efforts and regular submissions and resubmissions to achieve the required quality. These students try to wear down staff by re-submitting the work with little changes and show a lack of interest in gaining knowledge. (Linden, 2018, p. 69)

Their adaptation of the Scrum framework, while not being true to the rituals and roles of Scrum, added some interesting ideas to how Scrum could be implemented in a more traditional educational environment. The results of this experiment will be analyzed in the Validity (2.6) portion of this chapter.

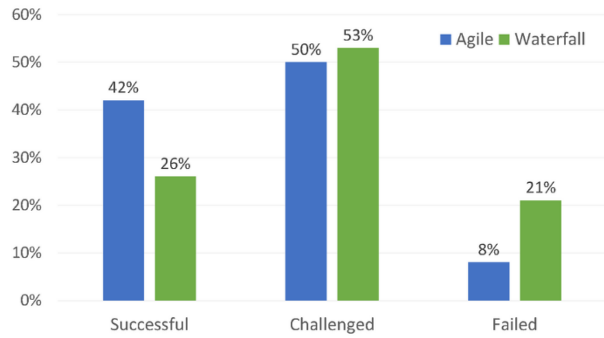
2.5 Implementation in a differentiated K-12 Classroom

Scott et al. (2016) researched into how different learning styles affect the use of Scrum in a learning environment. They organized students into Active and Reflective categories and adjusted their implementation of Scrum accordingly. How they implemented each approach is described in Appendix B. The researchers observed "…that reflective students obtained higher scores… than active students when taught by means of the passive instructional method, whereas active students obtained higher scores on average than reflective students when taught by means of the active instructional method" (Scott et al., 2016, p. 250). This shows us that it is important to consider each student's learning style when implementing an Agile framework in the classroom.

There is very little research on the use of Agile in K-12 Education. Of the papers studied by Salza et al., only "…10%, are papers targeting K-12 students, from a minimum of 4- to 19-year-old (the ranges can change according to different countries). The rest is focused on academy students, where 87.5% is for undergraduates and 18.5% specifically for master students" (2019, p. 28). However, several schools have already implemented Agile programs in their schools. As described by Loewus, several Virginian schools have successfully implemented Agile in their classrooms and central offices. The middle school teachers quoted in the article both responded positively to the use of Agile in their History and Science classrooms. One teacher did, however, have to simplify the Scrum process to make it effective in his classroom (Loewus, 2017). Based on the information above, it is clear that with the right modifications, Agile and Scrum can successfully be used to implement problem-based learning in a Computer Science classroom.

2.6 Validity

2.6.1 Connection to Industry



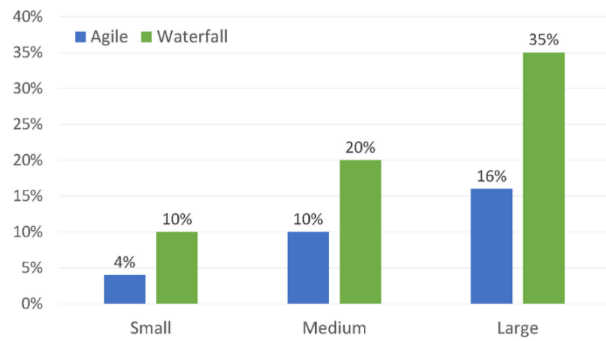


Figure 1: Project success rates depending on project management methodology employed (left). Figure 2: Project failure rate depending on project management methodology employed and project size (right). (López-Alcarria et al., 2019, p. 8)

Agile and Scrum are used by the majority of industry organizations in the Computer Science field. A 2011 survey by Version One ascertained that 80% of the respondents to its survey worked for organizations that had adopted Agile practices. Of that portion, 66% were using Scrum or Scrum variants (El-Khalili, 2013). In a survey 5 years later by the same organization, they reported that the respondents who used Scrum or Scrum variants had grown to 82% (May, York, Lending, 2016). In a personal communication between May et al. and Erica McDowell, a Booz Allen Hamilton executive, in 2015, McDowell discusses the state of Scrum and education:

> In the last three years of my career, I have yet to see one government RFP that did not include some form of a Scrum reference. These days, the Scrum framework and Agile thinking have become the norm. Therefore, we place a strong emphasis on students who have been

exposed to agile thinking in general and the Scrum framework in particular. (May et al., 2016, p. 87)

The Standish Group Chaos Studies found that Agile projects were more likely to be successful than projects conducted in the traditional Waterfall project management technique. Agile is more than twice as effective as Waterfall when it comes to large projects. Please see Figure 1 and 2 above. Figure 1 (on left) describes the project success rates based on which project management technique employed was employed. Figure 2 (on right) describes the failure rate for each project methodology depending on the size of the project (López-Alcarria et al., 2019).

2.5.2 Skills Developed Using Agile

In their paper describing Agile practices for the Environmental Sciences discipline, Lopez-Alcarria et al. (2019) extensively describe the skills developed by students who engage in Agile practices. As stated in the background above, Agile is deeply rooted in the constructivist philosophy of education. Lopez-Alcarria et al. (2019) also provides the following as key competencies of constructivism that are fostered in an Agile approach to education:

- Autonomy in the generation and construction of knowledge. (p. 10)
- Evaluation of alternative solutions. (p. 10)
- Collaboration: merging learning with social and relational context of the individual. (p. 10)

- Critical thinking: meta-cognition and reflection in the process of knowledge construction. (p. 10)
- Systemic thinking: individuals have a general mind map of the knowledge they generate since it springs up from their own experience. (p. 10)
- Use and management of different sources of knowledge. (p. 10)

At the time of writing this thesis, the world is currently in the COVID-19 pandemic. Many school districts (including the author’s employer) are concerned with the implementation of distance learning programs in anticipation of school closure. Lopez-Alcarria et al. (2019) describe how digital platforms used in Agile work (such as Trello, Jira, etc.) fosters connectivism and expands on Vygotsky’s Zone of Proximal Development, which fosters the following competencies in students:

- Ability to understand and visualize connections between different areas, ideas and concepts that generate knowledge (p. 10)
- Decision making (p. 10)
- Ability to innovate and generate revolutionary ideas (p. 10)

This adds to the initial research on how learning styles interact with the implementation of Scrum in a classroom described in the Background of this chapter.

Lopez-Alcarria et al. (2019) further go on to describe how Active learning, a strategy of learning through collaborative experience and self-reflection, develops the following competencies in both students and teachers:

- Comprehension (p. 11)
- Critical thinking (p. 11)
- Reflection (p. 11)
- Reconstruction of knowledge (p. 11)
- Collaboration (p. 11)
- Search, analysis and synthesis of information (p. 11)
- Active problem solving (p. 11)

This collaborative learning environment also leads to the following student competencies, according to López-Alcarria (2019):

- Self-regulation of learning (p. 11)
- Open-mindedness to others' ideas. Identification of strengths of team members (p. 11)
- Learning to learn, building effective knowledge and mental models (p. 11)
- Creative problem solving (p. 11)

Table 3: Key Environmental Sciences Competencies compared to Key Agile Education Competencies. (López-Alcarria et al., 2019)

KEY ESD COMPETENCIES		KEY AGILE EDUCATION COMPETENCIES
Coping with Uncertainty	↔	Agile provides tools to allow adaptation to change processes.
Working creatively and Innovation	↔	Agile favors innovation by fostering creative solutions based on collaborative work. Combines efficiently with other innovation methodologies e.g. design thinking.
Dialog	↔	Agile interaction processes' are based on dialog to create knowledge and increase cohesion of members.
Critical thinking	↔	Agile promotes critical thinking from a continuous reflection process.
Self-motivation and motivating others	↔	Agile empowers individuals to contribute to find solutions and feel valuable within the team.
Planning and implementation	↔	Agile provides long-life and multipurpose project management skills.
Interdisciplinary work	↔	Agile teams are typically composed of multidisciplinary members.
Manage emotions and concerns	↔	Agile favors the pursuit of consensus to reach efficient solutions from the opinions of all stakeholders.
Responsibility	↔	Agile fosters both individual and group responsibility.
Systemic thinking	↔	Agile faces issues and tasks as part of a complex and interconnected system, not as isolated individual elements.

These competencies are crucial to creating proficient Computer Scientists and lifelong learners.

Lopez-Alcarria in the course of their paper mapped out the Key Environmental Science Competencies to the Key Agile Educational Competencies. While these are not explicitly defined as Computer Science competencies, each of the Key ESD Competencies can be seen as an essential skill for a Computer Science student. This comparison can be seen in Table 3 above.

2.7 Pedagogical Implications

In the Swinburne example described in the background, Linden (2018) provides the University's implementation of the Scrum framework. Young's Social Cognitive Framework guided their implementation for Self-Regulated Learning. This framework shows how the

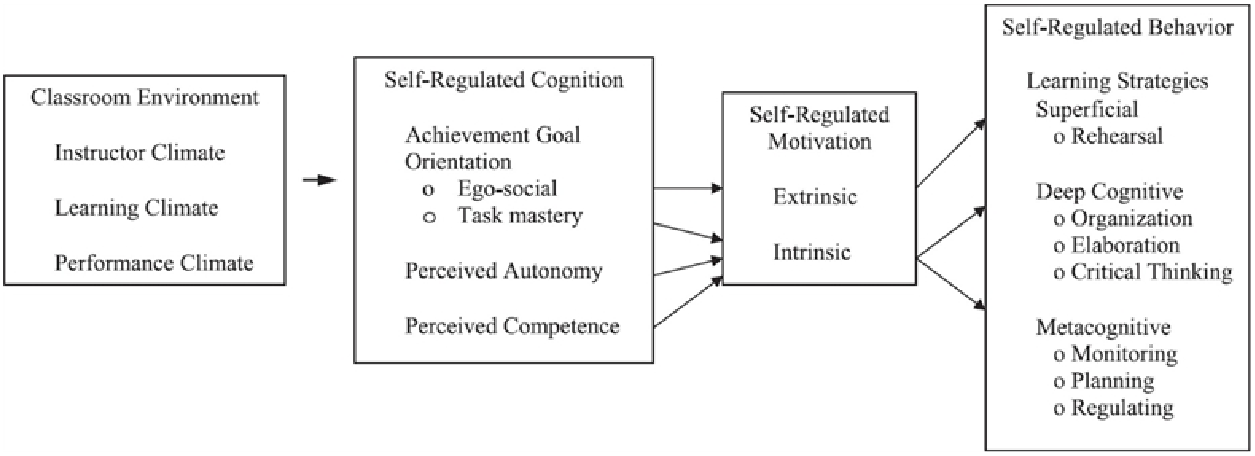


Figure 3: Young's Social Cognitive Framework for Self-Regulated Learning (Linden, 2018, p.68)

classroom environment contributes to self-regulated cognition, motivation and self-regulated behaviors. The model can be seen in Figure 3. Table 4 shows how Linden mapped their process to Young's Framework. Doubtfire, in this case, is the learning management system they used for the Scrum process (Linden, 2018). Linden (2018) observed the following results from their study:

Table 4: Linden's Approach to Self-Regulated Learning Through Scrum Mapped to Young's Framework (Linden, 2018, p. 69)

Classroom environment	**Self-regulated cognition**	**Self-regulated motivation**	**Self-regulated behavior (learning strategies)**
▪ Doubtfire (LMS) ▪ Sprints with pre-set tasks (by difficulty level) ▪ Teaching staff ▪ Formative feedback ▪ Burndown charts	▪ Setting a goal (choice of grade) ▪ Ability to change the goal grade	*Intrinsic*: ▪ setting the goal and sticking to it throughout the semester ▪ accepting feedback positively *Extrinsic*: ▪ superficial approach, resubmissions just to get a "complete" mark ▪ sometimes negative reaction to formative feedback requiring resubmission	▪ Re-submitting tasks ▪ Follow-up on provided feedback ▪ Selecting communication methods with staff ▪ Adjusting order of tasks in the backlog ▪ Dealing with deadlines ▪ Opting to work alone or with classmates

- The results demonstrate that the majority of our students are in favor of the environment that allows them to work using a Scrum approach and supports self-regulated learning (p. 72).
- “Their responses and comments show their satisfaction with the ability to work in short sprints, submitting incremental deliverables, and having a way to keep track of their progress” (p. 72).
- “These responses also illustrate the importance of perceived autonomy and perceived competence” (p. 72).
- “They show appreciation of feedback and the ability to learn from it” (p. 72).

It is important to note that although Linden observed many positive outcomes from their implementation of a Scrum-like framework, they did not meet their goal of reducing student failures in Swinburne’s introductory Programming class. They found that “…38% (of students) completed less than 75% of tasks,” and observed that these students “…demonstrated a superficial approach to their studies and a lack of interest in learning” (Linden, 2018, p. 72). Although the use of Agile did not decrease their number of failing students, Linden did remark positively on the use of the Scrum framework for students who were motivated and interested in learning (Linden, 2018).

Based on surveys of students interacting with a Kanban framework (a different Agile methodology), Saltz and Heckman (2020) found that the majority of responses (73%) showed internalization of at least one Agile concept. Of the 86 students who responded to their survey, 83 demonstrated evidence of internalizing at least one Agile concept. The two that were most commonly identified were reflection and self-organization. (Saltz, Heckman, 2020)

2.8 Importance

It is clear from the surveys of industry that the Scrum framework, and the Agile methodology, are dominating the different Computer Science industries in terms of practice. It can be assumed that any student who decides to enter this field will be exposed to Agile at some point in their career, and therefore that knowledge and skills with an Agile methodology would be valuable to potential employers of a student.

The background with the Scrum framework would give students several essential life skills such as collaboration, time management, reflection, autonomy, and problem solving. These skills make students marketable to the Computer Science industry, but are coveted in most industries and educational institutions. Although Linden's (2018) research didn't prove that this would raise low performing students' academic engagement, Saltz and Heckman's (2020) research show that the majority of students internalize at least some aspect of the Agile methodology.

Not only do students receive essential skills and background in their learning, it also raises students' satisfaction. According to Loewus, several teachers in Virginia who have implemented Scrum in their middle school classrooms have seen an improvement in student engagement (2017). In the Swinburne study, 88.6% of students preferred their version of Scrum over the traditional approach to learning programming concepts (Linden, 2018).

One of the major difficulties in problem-based learning is how to structure student learning in a meaningful way that allows them to explore ideas and topics while still maintaining the structure needed to fit learning objectives and deadlines. Scrum offers a potential solution to this problem. It allows for more ambiguous projects by adapting to the changing needs of stakeholders and structuring daily and weekly rituals involved in the process. These rituals give the students a sense of normalcy while also giving them a say in the creation of project ideas.

The researcher sees implementation of an Agile framework, at some level, to be essential to modern Computer Science education. Following that logic, it is essential for any teacher preparation program to offer a class that prepares teachers to implement it in their classrooms. The Scrum framework is the most widely used, researched, and accessible Agile methodology for this purpose. As the class detailed in this thesis is an upper level Methods of Instruction class, it makes sense for the inclusion and exploration of Scrum rituals and how they apply to the classroom.

2.9 Scrum Methodology

The Scrum methodology can be split into three basic components: rituals, roles, and artifacts. Within the remainder of this chapter, each of the components will be discussed. The roles and rituals will be explained individually, and the artifacts will be discussed in the context of the rituals they support. After a component is discussed, its application to education will be discussed. This will commonly be done through the lens of the eduScrum framework.

The eduScrum framework is a framework developed in the Netherlands in 2011. Alphen aan den Rijn, a chemistry teacher, began implementing a modified version of the Scrum framework with his students between the ages of 12 and 18. This version of Scrum was later codified in the eduScrum guide (Wijnands & Stolze, 2019). As of 2016, several universities across Europe have already modeled entire classes around the eduScrum approach (May et al., 2016).

The eduScrum framework makes several distinct changes from the Scrum framework. These changes will be discussed in each section about Scrum components. It must be stated that the rules of Scrum, as described by Sutherland and Schwaber (2017), are immutable. This means that to change the rules in any way would result in something that isn't Scrum (Sutherland & Schwaber, 2017). Wijnands and Stolze echo a similar sentiment with eduScrum, arguing:

> You cannot do eduScrum halfway. Each part is there for a reason. If one single eduScrum component makes your situation better, it is obviously smart to apply that. Fine. But that does not make your teaching eduScrum yet. You should not seek to adjust eduScrum to your situation because eduScrum, like Scrum, is a system that works like a Swiss clock. Whatever you do and how you apply it, use all the elements. It is a precarious game. If you want to use parts of eduScrum because it seems useful, please feel free to do so. You just do not gain all the benefits that

> can be achieved. eduScrum works as a whole and delivers more than the sum of the parts. (Wijnands & Stolze, 2019, p. 113)

For this reason, the recommendation featured in Chapter 4 of may not be able to technically call itself Scrum or eduScrum (depending on whether the resulting framework meets all the eduScrum criteria).

2.10 Scrum Roles

2.10.1 Product Owner in Scrum

In the Scrum framework, the Product Owner is responsible for managing the **Product Backlog**. The Product Backlog is the list of everything that is needed for a product. It is the source of all requirements needed to implement and change a product. The backlog lists all features, functions, enhancements, and fixes that could change or create the product. The Product Backlog is considered dynamic. This means that it adapt and changes overtime (unlike the product requirements in traditional Waterfall development). It is ordered by priority as determined by the Product Owner (Sutherland & Schwaber, 2017).

The Product Owner's main role is the maintenance and the interpretation of the Product Backlog. According to Sutherland and Schwaber (2017), this can include the following tasks:

- Clearly communicating Product Backlog items
- Putting the items in the Product Backlog in order of importance based on current goals and missions

- Optimizing Development Team work
- Keeping the Product Backlog visible, transparent, and clear to all so that the Scrum Team knows what to work on next
- Ensuring the Development Team understands items in the Product Backlog to an appropriate level

This work may be delegated to the development team, but it is ultimately the Product Owner who is accountable for it. Sutherland and Schwaber are clear that the Product Owner is intended to be one person. While this person may be flexible to the consideration, they clearly stress the importance that the product backlog is maintained by a single entity and not by committee (Sutherland & Schwaber, 2017).

2.10.2 Product Owner in eduScrum

It is not difficult to bring this role to education. The teacher of a K-12 class has traditionally determined what is to be studied and when. The Product Backlog in terms of education is the curriculum to be taught and the standards/objectives to be met. Like the Product Owner, the teacher is accountable for material or standards that are not met. eduScrum describes the teacher as the Product Owner, but also as a servant leader to the teams of students in the class. In eduScrum, the teacher decides what is learned and how much time will be given to it. The teacher determines what projects the students will work on, sets the learning goals, and details how they will measure the student’s work.

2.10.3 Scrum Master in Scrum

The Scrum Master is, according to Sutherland and Schwaber (2017), responsible for the implementation of Scrum on a project. They make sure everything runs smoothly and that the Scrum methods are upheld. They also help everyone understand Scrum theory, rules, practices and values (Sutherland & Schwaber, 2017). According to Sutherland and Schwaber (2017), the Scrum Master serve the Product Owner in the following ways:

- Making sure that goals, scope, and product domain are understood by everyone on the Scrum Team to the greatest extent possible
- Ensuring effective Product Backlog management
- Aiding the Scrum Team to understand the need for Product Backlog items that are clear and concise
- Clarifying how product planning works in an iterative environment
- Mentoring the Product Owner on how to arrange the Product Backlog to create the maximum value
- Demonstrating agility
- Facilitating Scrum events as often as needed

Sutherland and Schwaber (2017) also detail the way in which the Scrum Master serves the development team by:

- Mentoring the Development Team in Scrum Values, Rituals, and Artifacts
- Aiding the Development Team in the creation of value
- Removing obstacles to the Development Team's success
- Facilitating Scrum events as often as needed
- Mentoring the Development Team to implement Scrum in organizational environments in which Scrum is not fully adopted

The role of a Scrum Master is to serve the other members of the Scrum process. The Scrum Master does not command or control the process. Rather, they do everything in their power to champion the process and remove roadblocks for the Development Team.

2.10.4 Scrum Master in eduScrum

Teachers are ultimately responsible for the structure and management of their classroom. Due to this accountability, it is the responsibility in many ways to serve as Scrum Master. The teacher is responsible for the implementation of Scrum/eduScrum and making sure that all stakeholders are aware of relevant practices, theory, and rules of the framework. They also influence student's behavior and workflow through servant leadership.

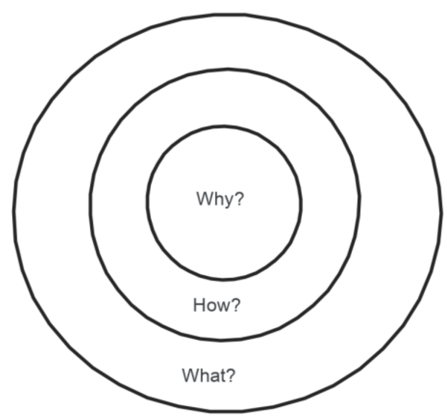


Figure 4: Simon Sinek's Golden Circle (Sinek, 2020)

As a servant leader, the teacher is responsible to assist student teams and answer student questions. In addition to determining ‘what’ assignments the students work on, it is also essential, the teacher determines the ‘why’ of the assignment. Wijnands and Stolze (2019) observe how much more effective a student team is when they understand the ‘why’, or the importance and relevancy, of an assignment. This relevancy and importance should be personal to the students. Wijnands and Stolze (2019) describe it as:

> Start with the students’ ‘why’ to ask questions. Why are they in your class and ‘must’ follow your subject. Explain to them its usefulness, and how they can use and apply it. Then they know and understand why they also need to do things they do not like to do. The ‘why’ is about passion, motivation, your heart-feeling, your inner self. This is not about what people believe in, it’s what they feel. (p. 9)

They recommend Simon Sinek’s Golden Circle as a great place to begin when planning a project. The Golden Circle can be seen in Figure 4. The idea behind the Golden Circle is to start with **why** we do something, then figure out **how** to do it, and create **what** expresses it. In eduScrum, the teacher would decide what the students needed to do and explain why it will be relevant to the students. It is then the students’ responsibility to figure out how to do it. This gives the teacher the executive control of the Product Owner, while still allowing for the flexibility and self-organization of the Scrum framework (Wijnands & Stolze, 2019).

The eduScrum framework also has a Team Captain selected for all student teams. The team captain can be chosen by the teacher or by the class. The Team Captain does not serve as boss of the team, but rather as an echo of the servant leadership of the teacher. They act as an ‘oilman’, who helps the team and coaches them. While the teacher is accountable for the implementation of the Scrum process, the Team Captain helps facilitate and collaborate amongst the team members (Wijnands & Stolze, 2019).

2.10.5 Development Team in Scrum

The Development Team, as described by Sutherland and Schwaber (2017), are a group "…of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint" (Sutherland & Schwaber, 2017). The team defines this "Done" increment and are given autonomy to manage their own work. The teams can range in size from 3-9 members, not including the Scrum Master and Product Owner unless they are executing work on the Sprint Backlog (Sutherland & Schwaber, 2017). As described by Sutherland and Schwaber (2017), Scrum Development Teams share the following characteristics:

- They organize themselves and decide amongst themselves how best to turn the Product Backlog items into releasable Increments
- The Development Team has members with different skills that all contribute to the creation of the product Increment
- There are no titles in Scrum Development Teams

- The Development Team is one team: there are no sub-teams
- Regardless of the individual skills of team members, all team members are accountable for a project’s success

2.10.6 Development Teams in eduScrum

Teams in eduScrum consist of teams of four to five students. This is similar to traditional group projects in K-12 education. eduScrum teams, however, are self-organizing. This means that they figure out amongst themselves how to work together and accomplish the task. Team formation takes place before each sprint. Each team captain randomly selects members for their team based on gender. There is expected to be as even of a gender distribution as possible on Scrum teams. It is done anonymously so that students do not flock to friends or select teams that are single sex. The students stay in these for the duration of the sprint. The students choose a name for their team and begin making arrangements for how the team will work. Trust forms the foundation of these eduScrum teams. According to Wijnands and Stolze (2019), “students will see that if they trust each other and work together with pleasure, a good result is almost self-evident” (p. 99).

2.11 Scrum Rituals and Artifacts

2.1.1 Sprints in Scrum

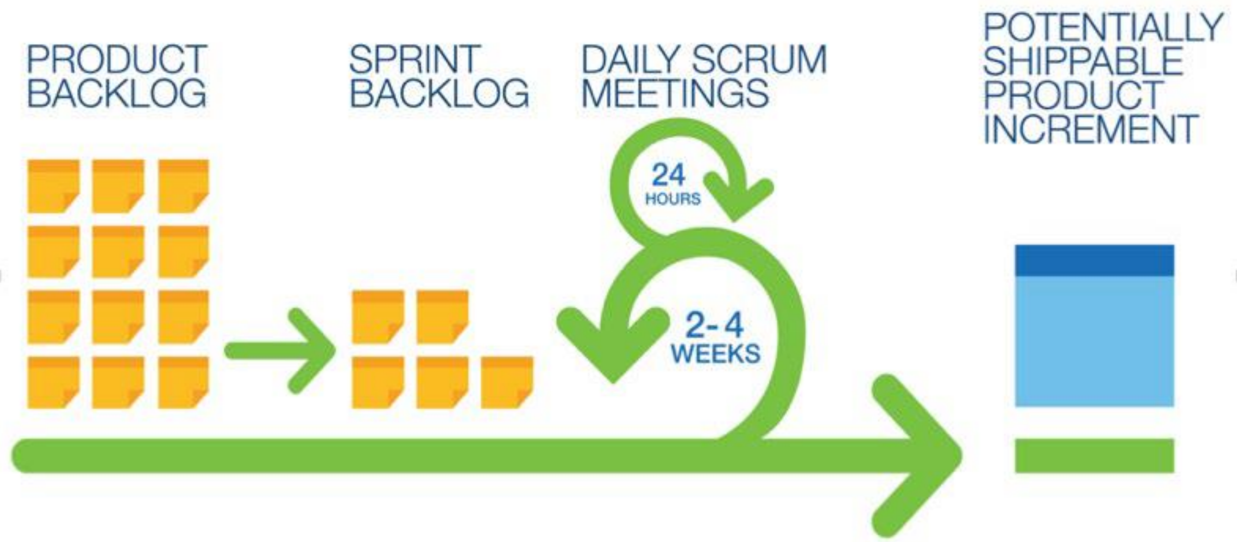


Figure 5: The Sprint Process in Scrum (Lean Dog, 2019, p. 46)

The Sprint is essential to the Scrum process. Every other piece of Scrum is built to support the successful implementations of Sprints. A sprint is a "time-box" in which a useable product Increment is created, tested, and released. In this case, an Increment refers to a portion of the features, fixes, etc. listed in the Product Backlog. It is expected to be usable and implementable, as defined as "Done" by the Development Team before the Sprint starts. Sprints are a "time-box", meaning that they occur within a set period of between one week and one month. This length is consistent throughout the entire development effort. A new Sprint begins as soon as a previous sprint closes (Sutherland & Schwaber, 2017).

The concept of an Increment that is "Done", usable and releasable is essential to the sprint. Sprints are intended to accomplish a goal. Within the Sprint, there are several rituals that occur on a consistent basis. The goal of the Sprint is usually determined in the Sprint Planning meeting. Other rituals include Daily Standup Meetings (or Daily Scrums), the Sprint Review, and the Sprint Retrospective. Each of these rituals supports the Sprint and its goal in a different way. Sprints should not exceed the one-month timeframe (Sutherland & Schwaber, 2017). Other important considerations of the Sprint described by Sutherland and Schwaber (2017) include:

- No additions or changes are made that would endanger the accomplishment of the Sprint Goal
- The quality of the goals should not decrease
- As new things are discovered, the scope can be re-negotiated between the Product Owner and the Development Team

Figure 5 shows the general workflow in Scrum. Work is pulled from the Product Backlog to the Sprint Backlog; the Sprint Backlog is worked in the sprint and released as a working increment of software. The 24 h portion shows the Daily Scrum that occurs daily during the Sprint.

2.11.2 Sprints in eduScrum

In the eduScrum Framework, the Sprint is a time frame defined by the teacher, in which a certain amount of work needs to be completed. The maximum amount of time of a sprint extends out to 2 months for eduScrum Sprints. The fundamental rituals of the Sprint are still relevant. Each sprint begins with planning and ends with a review and retrospective. During the sprint, the team works together to achieve their goals. A sprint review will happen after every 3-4 hours of work. Although students have freedom to determine how they complete the work, the rituals and rules give structure to the framework (Wijnands & Stolze, 2019).

Having used several Scrum rituals in their classroom, the researcher can say that organizing classroom activities into short sprints rather than daily schedules helped my students immensely. In my classroom, the students were given a Flap (discussed below) electronically that contained all assignments the students needed to complete for the week. They were then given the freedom to choose which order they worked on them. The researcher observed a high completion rate amongst my students, but the researcher also had no control group to measure this against.

2.11.3 Sprint Planning and Tracking in Scrum

In a Scrum Sprint, the **Sprint Backlog** is the guiding document for the Sprint. The Sprint Backlog is a set of items selected from the Product Backlog to be worked on during the Sprint. It also includes a plan for delivery of the product Increment and completing the Sprint Goal. The Sprint Backlog is the culmination of all work needed to meet the Sprint Goal. It usually also contains at least one high priority process improvement identified by the development team in the Sprint Retrospective. The work needed to be completed is constantly updated, showing the teams progress and adding new requirements are identified. The amount of work completed each day is tallied and tracked during the Daily Scrum (Stand-Up Meeting) (Sutherland & Schwaber, 2017).

The Spring Backlog and Sprint Goal are determined during a **Sprint Planning** meeting at the beginning of the Sprint. The plan made during this meeting is created in collaboration with all members of the Scrum Team. The Planning meeting is time-boxed based on how long the Sprint will be. Eight hours is the maximum for a one-month Sprint. The Scrum Master is in charge of making sure the event takes place and everyone understands its purpose (Sutherland & Schwaber, 2017). The two major guiding questions of Sprint Planning outlined by Sutherland and Schwaber (2017) are:

- What can be delivered in the Increment resulting from the upcoming Sprint?
- How will the work needed to deliver the Increment be achieved?

These questions help guide the team to the creation of a Sprint Backlog. These questions determine the Sprint Goal. The Sprint Goal is an objective that is to be met by the Development Team during the Sprint and helps guide the team in why they are creating the Increment. At the end of the meeting, the entire team should understand what they are doing, why they are doing it, and how they will organize themselves to accomplish it (Sutherland & Schwaber, 2017).

Scrum teams keep everyone on the same page by using **Information Radiators**. Information Radiators are tool or documents that are shown in public places so everyone can check in on how the team is doing. This helps keep the team to the value of transparency. One example of this is the **Story Card Wall.** A **Story Card** represents a user story and is the smallest piece of a product Increment. Story Cards are 1-2 sentences describing a needed function of the product. An example of Story Cards can be seen in Figure 6. These do not specify detailed

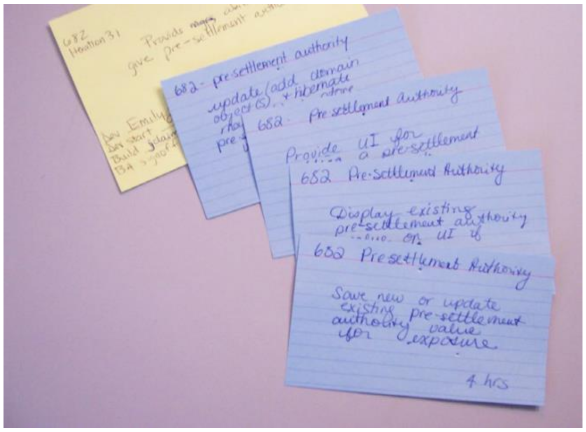


Figure 6: Story Card (Lean Dog, 2019, p. 43)

requirements. Instead, they serve as a "placeholder for conversation". However, they should be testable and include criteria for acceptance into the release. The Story Card Wall is usually broken up into columns representing the function of the cards in it. A very simple version of this is the Kanban board, where there are four columns: to-do, doing, testing, and done. One important factor of Story Card Walls is that any member of the team can move cards from one column to the other at any time. This builds collaboration through inspection and transparency. Another aspect of successful Story Card Walls are work-in-progress (WIP) limits. In an example above, a WIP limit would restrict the amount of cards in doing to one card per person on the team. This forces the team to reach "Done" on each story card before moving to the next one (Lean Dog, 2019).

An important aspect of the Sprint Planning meeting, especially when using Story Cards, is to estimate/size the cards. These sizes are not based solely on time, but also on complexity and uncertainty as well. These planning processes are based on collaboration between the entire Scrum team, and the size is not set in stone until all team members agree. This size may also be changed at any time. There are several methods for this, including planning poker (discussed below), t-shirt sizes, etc. The numerical size determined by the team are normally referred to as **Story Points**. If a card is too big to complete in one Increment, it is usually broken down into less complex Story Cards (Lean Dog, 2019).

To track how close the team is to their goal, the team can use Story Points. The amount of story points complete in an iteration is the team's **Velocity**. By calculating a team's daily velocity and the total amount of Story Points needed in a sprint, a team can determine if they are on track to meet their Sprint Goal (Agile Alliance, 2020). One information radiator that uses Velocity to track a team's progress is called a **Burn Down Chart**. A Burn Down Chart plots the team's daily velocity against the work still needed to be done. This can show whether a team is on track or needs to complete more (Agile Alliance, 2020). Please see Figure 7 for an example of a Burn Down Chart. One criticism of the Burn Down Chart is that it does not specify whether the team is working on the correct things (Agile Alliance, 2020). For example, a team may be completing Story Cards that ignore the core functionality of the Increment, but the Burn Down Chart shows that they are being productive towards the Sprint Goal.

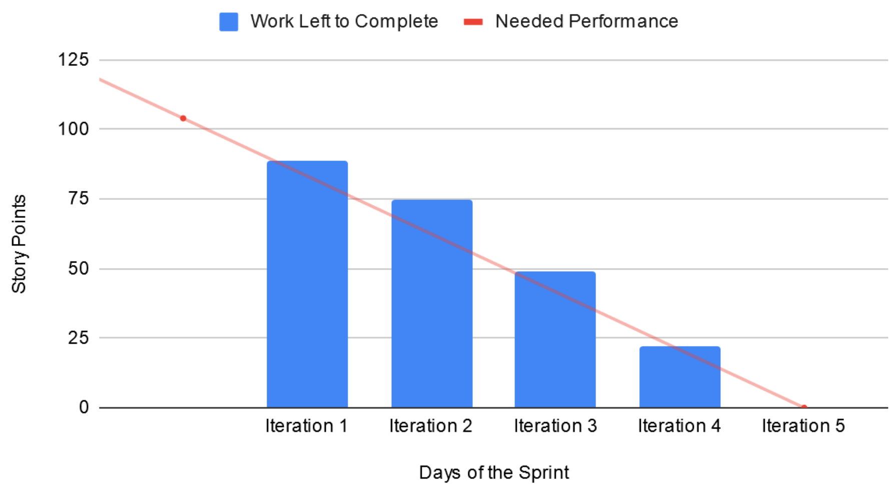


Figure 7: Burn Down Chart

2.11.4 Sprint Planning and Tracking in eduScrum

In eduScrum, the Planning Meeting takes place at the beginning of an assigned project. As stated above, the teacher is responsible for the "what" and the "why" of the assignment. Therefore, it is the teacher's responsibility to ensure that everything is "ready" for the students. This means that all steps of the assignment that must be completed by the teacher before the team starts working on it. The students should walk right into planning and distributing the work. The student teams create a Flap (the eduScrum version of a Story Card Wall discussed below) and plan for "how" to complete the assignment (Wijnands & Stolze, 2019). This plan, as outlined by Wijnands and Stolze (2019), is guided by the following questions the students are expected to ask:

- How much needs to be done?
- How long will this take us?
- How will we distribute the workload?
- What tools do we need access to?

Stories are used to describe the expectation of what the students should deliver. Each story contains the "what" and the "why" about the item to complete. Examples of stories include making assignments, writing a report, preparing a presentation, among other deliverables. The teams divide the assignment project into smaller actionable to-do items. Each to-do post it note is then placed on the teams Flap. Each story should also include **Celebration Criteria.** Celebration

Criteria is the teacher's way of ensuring that learning objectives are met. These can include assessments the students will need to complete, a rubric for the project, learning objectives, among other things. These are included on the Flap (Wijnands & Stolze, 2019).

Figure 8: An Flap of a Student Team (Gomes, 2020)

The **Flap** is the equivalent of the Story Card Wall in Scrum. It is an information radiator that makes sure the teacher and all the students remain on the same page. The Flap contains the Stories, the Celebration Criteria, and the Tasks relevant to each team's project. An example of the Flap can be seen in Figure 8. The tasks are split into the columns: to-do, doing, and done. The to-do column covers all tasks that need to be worked on in the Sprint. The doing column is any task that an individual has chosen to work on after consulting with the team. Some tasks may need to be worked on by ALL teammates (Wijnands & Stolze, 2019). In order for a task to reach the Done column, Wijnands and Stolze (2019) describe, three conditions must be met:

- All team members must be in agreement that the task is complete.
- The result must meet all requirements outlined in the celebration criteria.

- All students should be able to answer any questions the teacher has correctly

The Flap also contains a working agreement for the student team split between the team's **Definition of Doing** and **Definition of Fun**. The Definition of Doing is the actual working agreement of the team. This is composed of statements defining how the team will work towards completing the project (eduScrum, 2020). Examples include "the report meets the requirements stated in the celebration criteria" and "created work is discussed with the team". The Definition of Fun is composed of statements defining how the team will maintain a positive working environment. Examples include "puns will be made as much as possible" and "always giving positive feedback before negative feedback".

Another important aspect of the Flap is **Impediments**. Impediments are obstacles the team is facing that could possibly keep them from successfully completing the project. These can include interpersonal issues, lack of materials, etc. They are also ranked in order if most detrimental. It is the responsibility of the team to remove obstacles, but the teacher may intervene where needed (Wijnands & Stolze, 2019).

The final aspect of the Flap is the Run-Up Chart. The Run-Up chart functions in the same way as the Burn-Down Chart in Scrum. For an example of a Run-Up Chart, please see Figure 9. Each task is assigned a number of points. One way that is utilized to determine the amount of points for a particular task is Planning Poker. In Planning Poker, Fibonacci numbers are used to rate how complex a task will be. Each member of the team submits a number on a card. The team then communicates until they all agree on a point value for that task. Once all tasks are valued, the total amount of points for the Sprint are calculated. A line is drawn between 0 and the total. The team then calculates its daily velocity and compares it to that line. If a task has been marked done, it can be added to the Run-Up Chart. This is done each day during the Stand-Up. This tool helps students determine how they are stacking up to the class as a whole (Wijnands & Stolze, 2019).

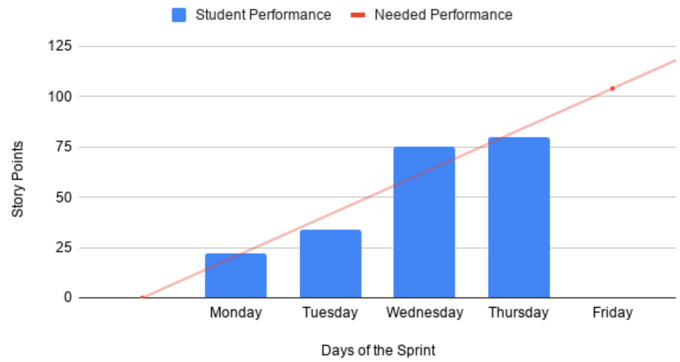


Figure 9: Example of a Burn Up Chart.

The researcher has implemented several of the tools discussed above in my classroom to great effect. In my classroom, each student was working on individual assignments but collaboration was encouraged. Each student had a virtual Flap on Trello containing links to all assignments. At the beginning of each week, the students would be given cards containing links to all of their assignments. In the example of Figure 10, all cards are color coded to their purpose. The researcher observed that the students had a much easier time completing work with it visually laid out for them in this way. The students also appreciated the freedom to choose how they worked on things. The researcher did notice that certain students would move things to the

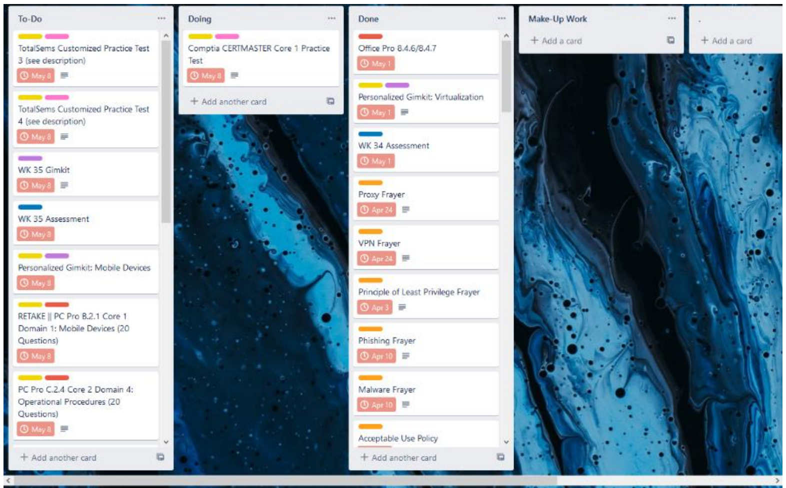


Figure 10: Digital Flap from My Classroom

Done column without actually completing them. Perhaps through the implementations of Definitions and Celebration Criteria, one could keep this from happening in the future.

2.11.5 Daily Scrum in Scrum

In Scrum, the **Daily Scrum** (or Daily Stand-Up) is a time-boxed event that occurs each day during a Sprint. After the Daily Scrum, the development team works for the next 24 hours. The Daily Scrum is always held at the same time and place every day. The structure of the Daily Stand-Up is determined by the Development Team (Sutherland & Schwaber, 2017). Most Stand-Ups, as described by Sutherland and Schwaber (2017), include answering the following questions:

- What have I completed since yesterday that has contributed to the Sprint Goal?
- What can I do today to contribute to the Sprint Goal?
- What impediments could prevent me or the Development team from meeting the Sprint Goal?

The Daily Scrum is usually followed by more detailed discussions with relevant team members. The Scrum Master ensures that the meeting takes place, that every team member understands the importance of the meeting, and that visitors to the meeting do not interrupt it (Sutherland & Schwaber, 2017).

The importance of this meeting cannot be overstated. It makes sure that all team members understand the progress of the Sprint and agree on what they should be working on during that day. It also gives them a chance to update the information radiators. This is in alignment with the values of transparency, adaption, and inspection.

2.11.6 Daily Scrum in eduScrum

In eduScrum, the Daily Stand-Up takes place at the beginning of each learning unit. Students come into the classroom and immediately put their Flap on the wall and begin updating it. These meetings are restricted to the first 5 minutes of class. It is necessary that all team members attend the meeting (Wijnands & Stolze, 2019). The meetings, as described by Wijnands and Stolze (2019), have a similar three questions to the meeting in Scrum:

- What have I completed since the last Stand Up?
- What can I complete before the next Stand Up?
- What obstacles or impediments are in my or the teams way?

Like in Scrum, there is no deeper discussion on the answers to these questions until after the Stand-Up meeting. The team captain ensures that this happens on a daily basis. After the standup, the team updates the Flap and Run-Up Chart (Wijnands & Stolze, 2019).

The researcher started using a Daily Standup to transition my students into the class. He would run it with his class, and each student would share their progress on their individual assignments. The researcher also added the question “What did we learn yesterday”. While the researcher does not think it helped the students on the level the Flap did, he does believe it helped the students’ transition into the classroom mindset.

2.11.7 Sprint Review in Scrum

A **Sprint Review**, sometimes called a Show and Tell, is held at the finish of a Sprint. The purpose of this meeting is to inspect the Increment, share updates with all stakeholders, and adapt the Product Backlog. During the Sprint Review, all stakeholders meet to talk about what was completed and give feedback on the Increment. The meeting is a great time for discussions about the priority of Product Backlog items to be included in future increments. The meeting is a time-boxed event, normally timed to about 4 hours for a one-month Sprint. The meeting is considered informal; its intention is to elicit feedback and determine items for future sprints (Sutherland & Schwaber, 2017).The meeting, as described by Sutherland and Schwaber (2019), normally include the following items:

- The whole Scrum Team and any key stakeholders should be in attendance
- The Product Owner details which Product Backlog items have been “Done” and which haven’t
- The Development Team discusses their performance, what problems they ran into, and how they solved these problems

- The Development Team demonstrates the Increment and answers any questions stakeholders have about it
- The Product Owner discusses the current state of the Product Backlog and updates target and delivery dates
- Through collaboration, the group determines priorities for future Sprints
- The group discusses potential changes to the marketability of the product and how this affects the priority of items in the Product Backlog
- The timeline, budget, potential capabilities, and marketplace for new changes are reviewed in the context of future releases

2.11.8 Sprint Review in eduScrum

In an eduScrum classroom, Sprint Reviews happen much more frequently. Reviews normally occur every 3-4 hours of work. The Sprint Review is an opportunity for the student team to demonstrate what they have learned during the most recent cycle and receive feedback from the teacher. The type and structure of the meeting is determined by the teacher. These reviews allow the students to adapt their self-developed content and allow the teacher to check if the assignments are actually being completed. This should be communicated on a personal level as well as what was accomplished as a team (Wijnands & Stolze, 2019). A good Sprint Review, as stated by Wijnands and Stolze (2019), should answer the following questions:

- Are all team members satisfied with the results?

- Has the results met all the celebration criteria?
- If not, how can the team address the missed celebration criteria?
- What additional support or assistance can the teacher provide?

Possible ways that students can communicate what they have learned include posters, presentations, and videos. This can be determined by the teacher or can be left up to the students. An eduScrum project usually culminates with a larger review meeting at the close of the project (Wijnands & Stolze, 2019).

2.11.9 Sprint Retrospectives in Scrum

While the Sprint Review focuses on improvements made to the product over the course of the Sprint, the **Sprint Retrospective** focuses on improvements that can be made to the team and the Scrum structures. This meeting normally occurs between the Sprint Review and planning the next Sprint. It is limited to 3 hours for a one month Sprint (Sutherland & Schwaber, 2017). The purpose of the Sprint Retrospective, as described by Sutherland and Schwaber (2017), includes:

- Discuss how the impact of people, relationships, process, and tools on the last Sprint
- Determine what went well and what improvements could be implemented in future Sprints
- Create a plan for implementing improvements determined by the Scrum Team

The Scrum Master ensures that this meeting takes place, that it stays positive and productive, and participates as a member of this meeting with equal accountability over the Scrum process. The Sprint Retrospective is an essential implementation of the values of Inspection and Adaptation. By the end of the Sprint Retrospective, the team should have a plan of how they will improve the Development process during the next Sprint (Sutherland & Schwaber, 2017).

2.11.10 Sprint Retrospectives in eduScrum

Just like in Scrum, the student teams complete a Retrospective at the end of each project. Each team reflects on their on their achievement and discusses how they will do things better for the next project. The students also reflect on their individual progress and role within the team. Each student rates the other team member based on their qualities and skills (Wijnands & Stolze, 2019). Guiding questions for an eduScrum Retrospective, as outline by Wijnands and Stolze (2019) include:

- What did we do well?
- How can I improve my performance? How can I contribute to improvements in others' performance?
- How can we improve as a team?
- What should we no longer do in the future?
- What specifically could we do to improve in the next Sprint?
- What do we know about the quality of our performance?

- What did I contribute to the team? What have I learned from my team members?
- For things that went well or didn't go well, why did it turn out that way?
- What felt like a waste of time? What did we do that really contributed to the quality of our project?
- What processes should we keep in the next Sprint?

These can be outlined by the teacher or determined by the team.

The Retrospective is one of the most important rituals in the eduScrum process. The Retrospective trains the students to self-reflect and give constructive feedback. Any postponement of this meeting is a major missed opportunity. A good retrospective includes a coach to guide the team through the process. The team should have a plan to improve their efforts in the next Sprint. This is also a good time to improve the Definition of Doing and the Definition of Fun (Wijnands & Stolze, 2019).

2.12 Conclusion

Scrum offers an interesting methodology for classroom procedures. It gives the students the ability to plan their own work, while increasing collaboration, time-management, and self-reflection skills in the process. It allows them to pursue more ambiguous and difficult projects, while still providing the structure necessary to bring success for students. There is not enough research to claim that Scrum is more beneficial for low-performing students, but the research justifies that even low performing students internalize at least some aspect of the Scrum process.

Perhaps most importantly, for institutions preparing students for a career or collegiate experience in Computer Science, it aligns itself to what the students will experience in their future. It is an important methodology to be studied, and justifies its inclusion in this thesis.

Chapter 3: New York State K-12 Computer Science Standards

3.1 K-12 Computer Science Framework

3.1.1 Background

The "K-12 Computer Science Framework" (2016) was created in response to questions such as "What should students be able to know and do in a K-12 computer science pathway?" and "What does computer science look like in the elementary, middle, and high school?" (p. 43). The framework was developed for states, districts, schools, and organizations to answer these questions and provide guidance for the development of standards and curriculum ("K-12 Computer Science Framework", p. 1). The framework does not outline expectations for specific courses; it outlines guiding principles for course development. In the words of the "K-12 Computer Science Framework" (2016):

> It does not provide grade level-specific outcomes, nor does it define course structure (the scope and sequence of topics in a particular course) or course pathways (the scope of topics and sequence across multiple courses). The five core concepts of the framework were not designed to serve as independent units in a course or separate topics defining entire

> courses; instead, the framework's concepts and practices are meant to be integrated throughout instruction. (p. 15)

The purpose of the framework is twofold: to outline concepts that should be touched on in computer science courses and detail practices that computer literate students should actively engage in ("K-12 Computer Science Framework", p. 3).

The framework outlines its vision to create students who are informed citizens. According to the "K-12 Computer Science Framework" (2016), students who are informed citizens can:

- Critically engage in public discussion on computer science topics (p. 10)
- Develop as learners, users, and creators of computer science knowledge and artifacts (p. 10)
- Better understand the role of computing in the world around them (p. 10)
- Learn, perform, and express themselves in other subjects and interests (p. 10)

The "K-12 Computer Science Framework" acknowledges how many stakeholders are involved in creating students with capabilities above. These stakeholders are considered the primary audience for the document. According to the "K-12 Computer Science Framework" (2016), these stakeholders include:

- State/district policymakers and administrators (p. 15)

- Standards and curriculum developers (with sufficient computer science experience) (p. 15)
- Current and new computer science teachers, including teachers from other subject areas and educators in informal settings (p. 15)
- Supporting organizations (nonprofits, industry partners, and informal education) (p. 15)

In order to ensure the needs of the developing students and other stakeholders are met, there are several themes woven through the different concepts and practices in the framework. The four themes prevalent in the work can be viewed in Table 5. These themes, and the concepts and practices developed from them, are reflect the current research in computer science education. The development of the framework was also highly dependent on feedback from its stakeholders. According to the "K-12 Computer Science Framework" (2016):

> Where specific computer science education research is lacking, the framework relies on the existing knowledge base of the practitioner community and research from other related content areas to guide decisions such as the developmental appropriateness of particular concepts. (p. 17)

In this context, the "K-12 Computer Science Framework" was created.

Table 5: Adapted from the "K-12 Computer Science Framework" (2016, p. 3)

Theme	Description
Equity	Issues of equity, inclusion, and diversity are addressed in the framework's concepts and practices, in recommendations for standards and curriculum, and in examples of efforts to broaden participation in computer science education.
Powerful ideas	The framework's concepts and practices evoke authentic, powerful ideas that can be used to solve real-world problems and connect understanding across multiple disciplines.
Computational thinking	Computational thinking practices such as abstraction, modeling, and decomposition intersect with computer science concepts such as algorithms, automation, and data visualization.
Breadth of application	Computer science is more than coding. It involves physical systems and networks; the collection, storage, and analysis of data; and the impact of computing on society. This broad view of computer science emphasizes the range of applications that computer science has in other fields.

3.1.2 Validity

The validity of the "K-12 Computer Science Framework" is connected to professional organizations in the field. Many of the ideas of the "K-12 Computer Science Framework" can be traced to the "US: A Model Curriculum for K-12 Computer Science, 2nd Edition" and the "CSTA K-12 Computer Science Standards" ("K-12 Computer Science Framework", 2016, p. 43). The organizations that published these documents also became part of the steering committee for this framework and several of the writers of the "CSTA K-12 Computer Science Standards" worked as writers of the "K-12 Computer Science Framework" as well (p. 43). The frameworks from several other countries were used to "benchmark the concepts and practices of the framework" ("K-12 Computer Science Framework", p. 43). The framework also utilized the AP Computer Science curriculum and the Association for Computing Machinery's to pinpoint what level of knowledge and skill students would need to reach to continue with their computer science education after their K-12 program ("K-12 Computer Science Framework, p.16).

3.1.3 Importance

The "K-12 Computer Science Framework" has garnered a lot of support from major educational and technology organizations. Some educational organizations that have announced their support in the “Statement of Support” (n.d.) for the framework include:

- CSTA
- ISTE
- NYC Department of Education
- Project Lead the Way
- Code.org

Several large technology companies have also vocalized their support of the framework. These companies are some of the largest employers of people in the computer science field, according to the “Statement of Support” (n.d.) including:

- Google
- Microsoft

- Amazon
- Apple
- Expedia
- SAP

The vocal support from these educational organizations and major companies in the technology industry has led to the use of the "K-12 Computer Science Framework" in the creation of state standards and curricula. Both California and Virginia have both utilized the framework to create their respective, state K-12 computer science standards (Lambert, 2018; Deruy, 2016). North Dakota did not explicitly state that it used the framework in the creation of its computer science standards, but is partnered with both Code.org and Microsoft in the creation of its statewide computer science education initiative (Foresman, 2018). It is safe to assume because it has partnered with two of the supporting organizations, the framework will be utilized to guide the development of this initiative. The framework is prevalent through many of the regulatory and supporting organizations of computer science education.

3.1.4 "K-12 Computer Science Framework" Practices

The two major components of the "K-12 Computer Science Framework" are concepts and practices. It was determined since this thesis is limited to pedagogy and not conceptual knowledge related to computer science, the practices are of more interest than the concepts. There are seven practices listed in "K-12 Computer Science Framework" and each practice has several indicating behaviors listed accompanying it. ("K-12 Computer Science Framework", p. 3) The practices and their indicating behavior are shown in Appendix E.

3.2 New York State Education Department Computer Science and Digital Fluency Learning Standards

3.2.1 Background

The New York State Department of Education has established a preliminary set of Computer Science and Digital Fluency standards. According to a press release from the New York State Department of Education, the department created the standards with a variety of stakeholders in order to keep them relevant to a diverse population of students (New York State Board of Regents, 2020). The NYSED webpage devoted to these standards (Computer Science and Digital Fluency Learning Standards) further elaborates on the process for creating and approving the standards. In October 2018, an Authoring Workgroup and Review Panel were formed to ensure representation from a diversity of stakeholders. Between then and March 2019, the Authoring Workgroup produced the first draft. In April of that year, the Authoring Group compared the standards written by themselves and their colleagues and provided this feedback to the Education Department (New York State Education Department, 2020).

According to the NYSED website, the second draft of the standards was then reviewed by the Review Panel. The Review Panel went through several in-depth reviews of the standards. According to the New York State Education Department (2020), each review occurred through one of the following lenses:

- Clarity and Focus (p. 6)
- Coherence and Progression (p. 6)
- Equity (p. 7)
- Interdisciplinary Connections (p. 7)
- Rigor (p. 7)
- Relevance and Engagement (p. 7)
- Specificity (p. 7)

In July and August of that year, the NYSED staff and a Computer Science education consultant revised the standards to address the Review Panel's feedback. The draft was presented to the Executive Standards Committee and Department Senior Leadership in September. The input from this meeting was included in another revision distributed for stakeholder feedback.

After receiving stakeholder feedback through a distributed survey, NYSED created a Workgroup to ensure that the standards reflected stakeholder feedback. This version of the standards is the version we see in circulation at the time of this thesis (New York State Education Department, 2020).

3.2.1a The Standards

According to a report by the New York State Teacher's Union, the NYSED referenced the K-12 Computer Science Framework in their creation of the standards (NYSUT Research and Educational Services, 2018). The core concepts are further outlined in the Press Release by the New York State Board of Regents (2020):

> The New York State K-12 Computer Science and Digital Fluency Standards are organized into five Concepts: Impacts of Computing, Computational Thinking, Networks and Systems Design, Cybersecurity, and Digital Literacy. Each Concept contains two or more Sub-Concepts. Within the Sub-Concepts are a number of standards. The standards are grouped into grade-bands: K-2, 3-5, 6-8, and 9-12. Students are expected to master the standards by the end of the last year of the grade band. (para. 4-6)

To see the standards in their entirety, please refer to Appendix F.

3.2.2 Validity

The third draft of the standards were published for feedback from teachers, administrators and other professionals between October 15 and November 15, 2019. According to the report on the standards from the New York State Education Department (2020) website, the "…majority of responses were from K-12 educators and administrators; feedback was also received from higher education, advocacy groups, business/industry, nonprofit organizations, parents, students, and school board members" (p. 7). Included in the report is a response from the survey: "Approximately 60 percent of respondents indicated that they either moderately or strongly supported the standards overall; however, several themes clearly emerged as priorities for immediate revision" (New York State Education Department, 2020, p. 7). From the report, we can ascertain that the majority of stakeholders approve of the standards. The standards that are currently in circulation at the time of writing this thesis incorporate that feedback.

The Board of Regents believes that these standards are representative of the skills 21st century citizens require. In a statement in the official press release of the standards, Board of Regents Chancellor Betty A. Rosa stated:

> We know that computer science and STEM fields are the jobs of the future, so it's important that we invest in our children and provide them with access to training in these areas to ensure that all of New York's students are prepared to compete for 21st century jobs…" (New York State Board of Regents, 2020, para. 2).

As stated in the March 2018 Regents item:

> Through these concepts, students [will] engage in a variety of activities including: creating prototypes that use algorithms to solve computational programs; comparing interactions between application software, system software, and hardware layers; refining computational models based on data; evaluating the ways that computing impacts social and economic practices; and comparing various security measures of a computing system. These types of activities immerse students in creative problem solving where they learn how to identify and present problems that

> computers can solve and how computers can solve them. (NYSUT Research and Educational Services, 2018, "What is Computer Science?")

The support from the Board of Regents supports the validity of the Computer Science and Digital Fluency Standards.

It is important to point out that the Computer Science and Digital Fluency Standards pay specific attention to younger populations of students. According to Board of Regents Chancellor Betty A. Rosa, the education of Elementary and Pre-K students in Computer Science is a priority. She stated in the official press release: "Further work to ensure the standards are developmentally appropriate for our youngest learners will ensure New York's children are exposed to these vital skills early on" (New York State Board of Regents, 2020, para. 2). This sentiment is echoed in the next steps planned for the standards. According to the NYSED website, one of the next steps is "Engage further with early learning experts to ensure the K-2 grade band standards are developmentally appropriate, and that both the clarifying statements and provided examples are helpful and relevant to K-2 teachers" (New York State Education Department, "Computer Science and Digital Fluency Learning Standards", 2020). The urgency of early childhood education in Computer Science adds a validity not seen in other standards studied.

3.2.3 Importance

The New York State Computer Science and Digital Fluency standards should be included for study in this thesis and the graduate class it outlines for two major reasons. The first is that the Methods class outlined in this thesis is for a teacher's preparation program in New York State. The standards fulfill expectation outlined in the 2010 USNY Statewide Technology plan that "students, teachers, and leaders will have clear standards for what students should know and be able to do with technology" (New York State Board of Regents, 2020, para. 4). Board of Regents Chancellor Betty A. Rosa shares a similar sentiment in the Board of Regents press release. She states "…as the Board of Regents and the Department work to ensure that all students have access to a high-quality education, it's critical that a comprehensive computer science curriculum is available to our students" (New York State Board of Regents, 2020, para. 2). The Board of Regents, as outlined on the NYSED webpage, "…the Board of Regents conditionally approved New York State's Learning Standards for Computer Science and Digital Fluency" (New York State Education Department, 2020, para. 1) in January 2020. According to the "Standards Development Process" portion of the webpage, one of the next steps is to "begin to develop resources and guidance to aid the field in implementing the standards in accordance with the proposed implementation timeline" (New York State Education Department, "Computer Science and Digital Fluency Learning Standards", 2020). This adds additional relevancy to this thesis, as it can serve as a proposal as guidance for incorporating the standards into teacher preparation.

The other major reason for inclusion is the link between the standards and the process for Computer Science accreditation in New York State. According to the "Computer Science Certificate Coursework Guidance" listed on the New York State Department of Education (2020) website, candidates need to complete a total of 12 hours of coursework in that addresses content in the following five concepts:

- Algorithms and programming (para. 1)
- Computing systems (para. 1)
- Data and analysis (para. 1)
- Impacts of computing (para. 1)
- Networks and the internet (para. 1)

These are the same core concepts addressed in the K-12 Computer Science framework and the NYSED Computer Science and Digital Fluency standards. This means that teachers training in Computer Science Education will be exposed to the concepts listed above and a Methods class that links them to their eventual classroom will be beneficial. The certification requirements outlined in “Computer Science Certificate Coursework Guidance” (2020) “mandate that students learn “the American Disabilities Act (ADA) website accessibility compliance requirements and how to code for accessibility” and “how computers can be used in educational settings to meet the needs of all learners, including those with learning differences” (New York State Department of Education, para. 3). Both of these topics are explored later in this thesis.

The changes brought on by the introduction of the standards not only affect new teachers, but also teachers currently teaching Computer Science at the K-12 level. According to NYSUT, “certified teachers who are or will be teaching computer science courses within the 5 years prior to September 1, 2022 can apply using TEACH for the Statement of Continuing Eligibility

(SOCE) in Computer Science” (NYSUT Research and Educational Services, “Transition to Computer Science Certificate”, 2018). This gives current Computer Science teachers the ability to continue teaching their courses, but it does have an end point. According to NYSUT, the “SOCE is valid for a period of 10 years from the date it is issued” (NYSUT Research and Educational Services, “Transition to Computer Science Certificate”, 2018). It can be assumed that any Computer Science teacher planning to practice after 2032 will need to achieve the Computer Science Certification, extending the relevant reach of this thesis and the Methods class it outlines.

3.3 International Society for Technology in Education Standards

3.3.1 Background

What is now known as the ISTE Standards for Students was originally developed as the National Educational Technology Standards. According to Niederhauser et al. (2007), the "NETS*S were developed to provide standards and guidelines to help teachers effectively and meaningfully use technology with their students" (p. 484). These standards addressed the basic principles of student technology use and as Niederhauser et al. stated were "...aligned with the broader constructivist-based content-area curricular reform efforts that occurred in the 1980s and 1990s" (2007, p. 484). The standards are constantly updated.

In the words of Dondlinger et al. (2016), the standards "describe both 'what' our students need to learn and the 'ways' they need to learn and think" (Dondlinger et al., 2016, p. 260). When describing the standards at greater length, Dondlinger et al. (2016) clarifies:

> It's important to note that although ISTE labels these “Standards,” they don't describe narrow, content-specific, performance objectives, such as those assessed by standardized tests. Instead, they describe broader intellectual competencies vital to productivity in a digital age—an age requiring more than mere proficiency with technology tools. (p. 259)

The ISTE standards aren't meant to only address the content and proficiency of students. They are meant to address the abilities required to be a productive citizen in the modern digital age (Dondlinger et al., 2016, p. 260).

ISTE has adopted a holistic approach to standards development. In addition to the ISTE Standards for Students, ISTE has created standards for Administrators, Teachers, Coaches, Computer Science educators, and one for Computer Science educators with specific regards to Computational Thinking. Crompton (2014) outlines the uses for each of these standards (with the exception of the ISTE Standards for Computational Thinking) in her article. An adapted version of her table featuring the standards and their accompanying uses can be seen in Table 6. According to "NETS are now ISTE Standards" (2013), the specific benefits of using the standards include:

- Improving higher-order thinking skills, such as problem solving, critical thinking, and creativity (p. 8)
- Preparing students for their future in a competitive global job market (p. 8)
- Designing student-centered, project-based, and online learning environments (p. 8)

- Guiding systemic change in schools to create digital places of learning (p. 8)
- Inspiring digital age professional models for working, collaborating, and decision making (p. 8)

The ISTE standards are one holistic approach to the development of curricula and classroom instruction in order to create more technologically literate students.

Table 6: ISTE Standards and their Uses (Crompton, 2014, p. 39)

ISTE Standards	The ISTE Standards were used for:
ISTE Standards for Students	• Evaluating students' skills as they complete high school and go on to college • Measuring student and teacher technology use at different grade levels • Examining what needs to be better addressed in teacher education programs to help provide recommendations for addressing neglected areas
ISTE Standards for Teachers	• Measuring the TPACK confidence of in-service science teachers

	• Assessing whether teachers are following the standards when faced with new technologies • Measuring teacher candidate proficiencies in the final year of study • Examining college/university faculty use of the standards • Evaluating and finding exemplary models of teacher education • Exploring adaptations of the standards for use in higher education
ISTE Standards for Administrators	• Determining what technology skills administrators have and what they are lacking • Examining the competencies found in unique societies (e.g., Native American schools) to determine what skills were lacking and how this society could be supported
ISTE Standards for Coaches	• Examining how tech coaches support teachers and how these skills are connected with the TPACK framework

ISTE Standards for Computer Science Educators	• Considering the roles and responsibilities of the computer science community • Examining ideas for new curricula

3.3.2 Validity

The ISTE standards are one of the most reputable technological standard sets out there presently. The standards are constantly used as a benchmark when judging the effectiveness of technological initiatives and programs in studies. The standards for teachers are directly aligned with the National Council for the Accreditation of Teacher Education (NCATE) standards (Friedman, Bolick, Berson, & Porfeli, 2009). The TPACK framework is a well-researched methodology that combines technical, content and pedagogical knowledge. The standards are well aligned with the TPACK framework of technological integration with pedagogy. According to Ronan (2018), this makes them the "leading set of standards for technology integration in education" (p. 7).

According to ISTE, this focus on pedagogy is the core of the ISTE standards (International Society for Technology in Education, 2016, p. 2). The development of the ISTE standards combines research with consultations with experts and opportunities for public feedback. This is a similar process to the one used by "the Council for the Accreditation of Educator Preparation (CAEP), the National Board for Professional Teaching Standards (NBPTS), the American Library Association (ALA) and others." (International Society for

Technology in Education, 2016, p. 4). ISTE conducted a literature review to ensure that the standards were up to date with modern technological and educational research. As stated by ISTE (2016):

> In addition to feedback from experts and other stakeholders from the field, ISTE did a literature review to scan up-to-date thinking about the field of education technology. Even more importantly, however, was seeking research that showed the efficacy and overall value of various education practices and focus areas and to reflect in the 2016 ISTE Standards for Students rigorous approaches to learning and teaching with technology backed up by research, thought leadership and other data. These sources are primarily research papers and reports derived from academic, nonprofit or governmental studies but they also include a handful of illustrative or argumentative examples from the press or other mainstream sources. (p. 5)

The ISTE standards are one of the most established and utilized set of educational technology standards.

3.3.3 Importance

The ISTE standards are used widely throughout the country. The ISTE standards were utilized during the revision of the 2017 Ohio Learning Standards in Technology ("Ohio's Learning Standards for Technology", n.d.). The Connecticut Commission for Educational

Technology adopted the ISTE standards as their Digital Learning Standards in 2016 (Connecticut Commission for Educational Technology, n.d.). The reach of the ISTE standards extends beyond the United States. Dr. Crompton details how these standards are extensively used in other countries. As Crompton (2014) states:

> Researchers in Turkey and China, in particular, appear to be regular users of the standards, as numerous studies originated in these two countries. (p. 38)

The standards are being adopted by many important educational institutions all over the world.

3.3.4 ISTE Standards

The "ISTE Standards for Coaches" and the "ISTE Standards for Administrators" were considered outside the scope of this thesis, as they are outside the scope of what a teacher would use daily. The "ISTE Standards for Students" are detailed in Appendix D.

3.4 Conclusion

The K-12 Computer Science Framework was created to provide states guidance in the creation of their standards and curriculum. It was utilized by the New York State Department of Education in their development of the Computer Science and Digital Fluency standards. This central link between the two sets of standards validates their inclusion in this thesis. The K-12 Framework has multiple stamps of approval by industry powerhouses and Computer Science Education organizations. NYSED’s support of their standards necessitates that all teachers graduating from teacher preparation programs in the state should be well versed in them. The relevancy and connection of these two frameworks justifies their inclusion and use in this thesis.

The ISTE standards are the most established standards discussed in this thesis. Although they are the oldest educational framework discussed in this thesis, the continual updates they receive keep them relevant and modern. ISTE also was one of organizations that signed a letter of their support for the K-12 Computer Science Framework, which increases the connection between the three frameworks. The crosswalks between the K-12 Computer Science Framework and the ISTE Standards for Students (Appendix C-E) identifies interesting opportunities for instructional activities.

The Methods class outlined in Chapter 4 of this thesis will feature all three of these frameworks, incorporated into the instructional activities outline in the curriculum. Coding for accessibility and meeting a diversity of learning styles, mentioned in the certification requirements of New York State for Computer Science, will also be incorporated into the curriculum of the Method's class.

Chapter 4: Prospectus for a Master's Level Methods Class at Buffalo State

Chapter 4 will proceed to outline my proposal of a Master's level Methods Class for Computer Science Education Masters candidates at Buffalo State. The proposal will outline several Student Learning Objectives that correlate with standards discussed in Chapter 3 and the

Scrum methodologies discussed in Chapter 2. The outline of the class will then be modeled and explained. Then a few recommendations regarding grading and classroom procedures will be discussed. It is assumed in this Chapter that classes will be 15 weeks long, with a 3-hour class once a week. Schedules that deviate from this format will discussed during the recommendations section.

4.1 Student Learning Objectives

The class is divided into 3 Student Learning Objectives. Each Student Learning Objective details a particular aspect of the research done in this thesis. The first of the three SLO's defines the students understanding of the Scrum and Agile processes. It reads as follows:

- The students will be able to demonstrate use of the Agile framework through the implementation of eduScrum in the process of creating collaborative projects.

This SLO defines that the students will show their understanding of Agile and Scrum processes using eduScrum. The class will be ran as an eduScrum classroom with the Professor functioning in the Teacher role and the students functioning in Student Teams. By participating in the class Sprints, the students will demonstrate their knowledge of the Agile framework. Demonstration falls under the Application umbrella in Bloom's Taxonomy of Cognitive Development (Krathwohl, 2002).

The next SLO defines the students understanding of the different computer science standards discussed in Chapter 3. It reads as follows:

- The students will be able to justify the inclusion of standards from ISTE, the K-12 Framework, and NYSED in the creation of Computer Science curricula.

During the creation of their projects, in this case Learning Segments for a K-12 Computer Science class, the students will be asked to include and justify several standards within their Learning Segments using relevant research. Some research will be provided to them, while other portions of research will be found by the students as needed. Justification falls under the Evaluation portion of Bloom's Taxonomy (Krathwohl, 2002).

The final Student Learning Objective is for the students to design a Learning Segment, using Agile principles, in collaboration with other students. It reads as follows:

- The students will be able to collaboratively create a Learning Segment that utilizes relevant research and standards to implement Agile and teach specific concept areas to designated student populations.

Each sprint will give the students an opportunity to create a Learning Segment for a designated student population. Through this process, participants will learn how to plan collaboratively, utilize relevant research/standards, and meet the needs of specific populations. Creating is part of the Creation portion of Bloom's Taxonomy (Krathwohl, 2002).

4.2 Class Structure

The class will be divided into 7 sprints. The first sprint will begin on the first day of class. Each sprint shall last two weeks. Each sprint shall begin during the first class of that two week period with a Sprint Planning meeting. In this meeting, the Professor will act as the Scrum Master, reminding everyone of the importance of the event and that everyone understands its purpose. In the Sprint Planning meeting, the Professor will provide the students will a list of Stories and the Celebration criteria. Please see Appendix A for the full list of Stories and Celebration Criteria. The students will determine their list of To-Do's during the Sprint Planning, their Definition of Doing, and their Definition of Fun. The students will also elect a Team Captain for that iteration, who is responsible for daily check-ins with the team and updating the information radiators. It is expected that each student will serve as a Team Captain at least once.

On the next class after the Sprint Planning meeting, the second class of the Sprint, the students will have the opportunity to conduct a Sprint Review with the Professor. The Professor will provide direct feedback on what the students have completed up to that point and address any impediments that the students are not able to work out.

Since the students are only meeting once a week. It is expected that they will conduct Daily Scrums through digital means, but it is up to the students to figure out how they will do this and complete a record of their interactions to be shown to the professor during the Sprint Reviews.

Another Sprint Review will take place on the last day of the Sprint. The students will present the iteration they made to the professor and any other student teams of the class. The professor will give them direct feedback, but will grade the project after that class. The class will then begin a Sprint Retrospective. The structure of this Retrospective will be determined by the students. Finally, the professor will give the student teams the next set of Stories and Celebration Criteria, and a new Sprint will begin.

The last day of class will be a longer, more involved Retrospective. This will be referred to as the Class Retrospective. In the Class Retrospective, the students will reflect on their performance in the class, the merits and improvement areas for future iterations of the class, and describe ways they will use what they learned in the class in their teaching practice. Improvements recommended will be taken into consideration for the next iteration of the class.

In this way, Agile methodologies are instituted at every level of the class. The students are detailing how they will utilize Agile in their classrooms to teach standards to different student populations. In the process of doing this, the students are utilizing Agile processes to manage collaborative work for the Method's class. The Method's class itself is also considered an iteration, improving and adapting with each new class of students.

Each sprint will have a "theme", coinciding with a Concept from either the NYS K-12 Computer Science and Digital Fluency Standards or the K-12 Computer Science framework. The Concepts include:

- Impacts of Computing (New York State Department of Education, "Computer Science and Digital Fluency Learning Standards", 2020)
- Computational Thinking (New York State Department of Education, "Computer Science and Digital Fluency Learning Standards", 2020)

- Networks and System Design (New York State Department of Education, "Computer Science and Digital Fluency Learning Standards", 2020)
- Cybersecurity (New York State Department of Education, "Computer Science and Digital Fluency Learning Standards", 2020)
- Digital Literacy (New York State Department of Education, "Computer Science and Digital Fluency Learning Standards", 2020)
- Data and Analysis (New York State Department of Education, "Computer Science Certificate Coursework Guidance", 2020)

Out of all these Concepts, Computational Thinking is the largest, has the most Sub-Concepts and Standards. For this reason, Computational Thinking is used twice on the outline. The student teams are encouraged to focus on any Sub-Concept or subject that fits within this concept.

Each Sprint includes a designated student population to accommodate the Learning Segment too. These populations include:

- 12th Grade Suburban AP Computer Science Students
- 8th Grade Rural Intro to Computing Students
- 5th Grade Urban Students
- 9th Grade Alternative Education Students
- 10th Grade Urban Robotics Students
- 11th Grade Computer Career and Technical Education Students
- 1st Grade Rural Students

These populations cover a wide range of different age and location demographics teachers leaving the Master's program may run into. The order of these populations is based on the order outlined in Appendix A. This also calls on teachers from different teaching populations to utilize their particular expertise.

The demographics listed above are suggestions, and the professor of the class has ultimate say in. The outline in Appendix A is a suggestion for how the class should operate. It is up to the professor and the class to determine how to implement eduScrum. In the outline, a general outline of each sprint is given. It states the student population, concepts, and key standards the students of the Method's class need to tailor their Learning Segment to. A suggested research article is provided for the students Method's class to reference in their justification of the Learning Segment. Several example 'Student Stories' are given, or what the students who would be completing the learning segment can expect to have accomplished by the end the Learning Segment. The Celebration Criteria for the Sprint are then provided. The Student Stories are suggestions, but the Celebration Criteria are expected to be met.

4.3 Recommendations

Here are a few recommendations for the professor that would teach this class. The first is about sizing the classes. The amount of work for this class was planned for at least five students per team. The amount of work for this class should be adjusted to how many students are in the class. If a class only has three students enrolled, it would create a much larger workload. If there are 14 students in the class, it would be best to split them into two teams.

The professor should communicate his expectations to the class. These expectations should include equal distribution of role of responsibility. The intention is to have each students work on each part of the Learning Segment at least once. Each student should have done relevant research and presented on it to the team and the professor at least once. Each student should fill the role of team captain at least once. The intention of Scrum is to distribute the work, but also that each team member has experience with every step of the process.

Lastly, the professor should consider how the role of digital resources and information radiators should be used. The 'Suggested Research Literature' featured in the outline is all from a digital publication named 'Agile and Lean Concepts for Teaching and Learning' published by Springer. This book is available online through the Buffalo State library circulation. Considering the effect of COVID-19 on Higher Education and the fact that the class only meets once a week, it is important that all information radiators and other important documents are accessible by all students digitally. The students should be prepared to work online digitally as much as in person. It should be the team captain's responsibility to keep these information radiators updated and current.

4.4 Conclusion

Above the researcher has modeled an example of how Agile concepts can be used in a classroom that prepares Master's level teachers to utilize Agile to meet the needs of a diverse range of students and to meet the different Computer Science standards. Everything stated above is a suggestion to Buffalo State in their creation of the Computer Science Educations Masters curriculum.

Chapter 5: Discussion

5.1 Merits

5.1.1 The Case for Agile in a Constructivist Classroom

The rigid roles, rituals and artifacts of Scrum and eduScrum (2.10, 2.11) offer a structure to the process of problem-based learning (2.3). This is crucial to the field of Computer Science. The problems and projects of Computer Science are typically more ambiguous than projects and problems of other fields of study and often have no clear end point or completion criteria. In programming specifically, there may be no endpoint as there can always be new requirements added and new functionality needed. Requirements of a program are limited to the needs and desires of the users/clients. What are the limits on needs and wants?

The Scrum framework was developed by industry professionals to approach this theoretically limitless needs and wants in an iterative and incremental process (2.1). The eduScrum framework is deeply rooted in the constructivist theory of learning, which aligns the learning to processes of discovery that students would experience in the 'real world' (2.5.2). It is clear from the data on the use of Scrum in industry that if the students want an experience similar to that utilized in the 'real world', that Scrum or a Scrum variant is the best option for them (2.6.1). The initial assumption I made was that Scrum should be used in the classroom because it is used in the industry. If we want to prepare the students for a career in Computer Science, then exposing them to industry practices will make them more marketable. While this is important, it ignores the fundamental purpose of Agile. Agile methodologies were created organically in order to facilitate collaboration, and approach large, ambiguous projects in the most effective way possible (2.1). In constructivist educational theory and problem-based learning, effective, incremental project management is valuable regardless of its connection to industry.

The use of Agile methodologies is inherently process-oriented. It creates an environment where students are responsible for the construction of knowledge and improvement of the learning process (2.5.2). Most importantly, it supports goal-oriented behavior, where students set their own goals and develop structures to achieve them (2.5.2). Saltz and Heckman's research validates that a large majority of students internalize at least one Agile concept. The most common of these are self-organization and reflection (2.7). It also inherently embraces the use of digital literacies (2.5.2). This is extremely important, especially for students moving into the workforce or onto higher education.

In the COVID-19 Pandemic, schools were shut down and students were required to do work online and independently. While there is no formal research on the affect this had on students at the time of writing this thesis, in my personal practice I have noticed a severe drop-off in the student completion and understanding of material when removed from an environment of structure that a school provides. To prepare students to be lifelong learners, it is our responsibility to provide students with an education that prepares them for the independent learning and effort of the 'real world'. Agile frameworks (such as eduScrum discussed in this thesis) provide students training in the skills to organize and attend to their learning outside of the structure of a school, regardless of whether this occurs from a pandemic or graduation from the school.

5.1.2 The Case for eduScrum as A Method to Prepare Teachers to Teach Standards-Based Instruction

The prospectus for a Method’s class to teach eduScrum as a method of standards-based instruction is outlined in Chapter 4 of this thesis. It is important that a Method’s class prepare teachers for accreditation in their educational field. The standards discussed in Chapter 3, especially those from the draft of the NYSED Computer Science and Digital Fluency Standards, are directly aligned with the requirements for teacher certification in Computer Science (3.2.3). The need is not just for the teachers to have knowledge in each area of Computer Science, but to be able to use this knowledge in order to achieve the educational objectives outlined by these organizations in their classrooms.

The outline and curriculum discussed in Chapter 4 gives students in the Teacher Preparation program the practice of incorporating these standards in the education of different student populations. Each student population utilized in Chapter 4 exists in the Western New York area surrounding Buffalo State, so it is reasonable to expect that any teacher graduating from the Buffalo State Computer Science Education Master’s program has a chance to encounter any of those student populations over the course of their career.

Through the use of eduScrum, the students are exposed to a cooperative and constructivist approach to education. This approach is modeled at all levels of the class (2.3). By experiencing eduScrum in the procedures of the class, through the creation of Learning Segments and from educational research on Agile in the classroom, the teachers will be given the opportunity to interpret Agile from all roles in the process. Through collaboration with students of other experience, each teacher candidate will be able to draw on that experience to improve their instruction of a diverse range of student populations. This is important, because effective collaboration with colleagues is a valuable skill for teacher candidates to have (2.5.2). Lastly, the teachers will walk out of the class with several Learning Segments they can use in their classrooms, that utilized constructivist and cooperative learning strategies to create a Learning Segment that teach standards-based instruction.

5.2 Potential Pitfalls

5.2.1 Criticism of the use of Scrum (or Scrum Variants) in a K-12 Classroom

While much of the research sees utilization of Agile as favorable for student learning outcomes, there is very little research on how it applies to the K-12 classroom (2.5). This creates a fair amount of uncertainty to the effectiveness at reaching educational objectives, especially with younger populations of students. There is a lot of research on the effectiveness of Scrum and eduScrum at the collegiate level; however, it is questionable that the same results would be reflected at the K-12 level. It is unclear how the maturity of college students affects the success of Agile methods of instruction at the university level.

In Loewus' article (2017) describing Scrum being implemented at the middle school level, one teacher stated that he had to simplify aspects of Scrum in order to implement it in his classroom. The Scrum framework states that any deviation from the framework creates a framework that cannot be called Scrum (2.9). In Wijnands and Stolze's paper describing eduScrum, they detail how problems in the utilization of eduScrum are usually related to deviations from eduScrum (2.9). I used some eduScrum rituals and artifacts in my classroom, but did not find success with all of them (2.4.1). Some questions to consider regarding the implementation of eduScrum in a K-12 Classroom include:

- If a teacher needs to modify eduScrum in order to implement it in their classroom, will it still achieve the results that the research imply still take place?
- Could this modified version of eduScrum still be considered eduScrum?
- The creator of eduScrum originally implemented this with Middle and High School age students, but what about students younger than that?

When discussing the educational outcomes of students, it is important to note that Swinburne found no improvement in the productivity or educational performance of low performing students (2.4.1). Constructivist education results in high educational outcomes for students who are already motivated, but so do many other theories of education. As educators, we need to consider the needs of all students when planning classes. The main question to consider regarding implementation of Scrum with low performing students: Is a framework that does not improve the educational performance of low-performing students worth implementing? While I believe that it is, considering all of the benefits discussed above, some may not agree.

5.2.2 Potential Problems with the Prospectus for Buffalo State

One of the potential issues of the class recommendation in Chapter 4 is that is limited as a general recommendation for a Master's class (1.4). The SLO's were not approved by Buffalo State and it is not tailored to the output required from Buffalo State classes. Significant changes may need to be made to the class in order to align itself with Buffalo State's requirements for classes, especially since a Method's class is a required class. If it were an elective offered to Buffalo State students, perhaps it would be under less scrutiny.

Several other assumptions made in the writing of this thesis could potentially create problems for the professor and students. I assumed that the workload outlined in Chapter 4 is appropriate for a team of 5 students (1.4). Depending on the actual size of the class, this will require large changes to the structure of the class. If student team members are absent from class meetings, this will prove a significant detriment to the teams. It is also assumed that sharing the workload will result in the best overall learning for each student (1.4). Besides research on Agile methodologies, there is no research in this thesis stating that to be the case.

Lastly, the class as outlined in Chapter 4 is heavy on practice and light on direct instruction (4.2). It is assumed that students will learn the Scrum process while they are engaging in it. Scott et al. discovered that student's predilection towards particular learning styles (Active and Passive) lead to increases in success to students who were given instruction in their preferred learning style (2.5). In Chapter 4, the proposal for a Method's class leans heavily to the Active learning style. This could cause students predisposed to the Passive learning style to not achieve the same level of success as their Active peers.

5.3 Options for Future Research

More research in needed on the implementation and effectiveness of Agile, Scrum and eduScrum at the K-12 educational level. It was nearly impossible to find any research on how this could be utilized at the elementary level. There needs to be more research around the use of Agile methodologies with low-performing students. There is not enough research to definitively say that Agile is not effective for low-performing students, or which aspects of Agile might be helpful for low performing students. For Agile as a whole, we don't know what students get from individual rituals, roles, and artifacts. We only know how Agile as a whole improves students' outcomes, so this could create an interesting opportunity for research.

eduScrum is modified to make Scrum fit within the structure of education (2.9). It is unclear whether students who are taught using the eduScrum framework directly understand how this relates to industry. An interesting opportunity for future research is whether students understand this connection and if not, what teachers can do to make this connection clearer. There are many opportunities for research in how to make Agile more effective in the classroom.

Tailoring the outline to Buffalo State's expectations is another potential area for research. There were many assumptions made about what Buffalo States expects from a Master's level class. More research is needed on what Buffalo State requires out of curriculum and SLO's in its teacher preparation classes. As these requirements become more apparent, it will most likely require changes to the outline in order to meet them. It also may be necessary to adjust the curriculum to changes in the final version of the New York State Computer Science and Digital Fluency Standards.

5.4 Concluding Statement

> Children should be able to do their own experimenting and their own research. Teachers, of course, can guide them by providing appropriate materials, but the essential thing is that in order for a child to understand something, he must construct it himself, he must re-invent it. Every time we teach a child something, we keep him from inventing it himself. On the other hand that which we allow him to discover by himself will remain with him visibly for the rest of his life. (Piers, Piaget, 1972, p. 27)

For those of us who subscribe to the constructivist view of education, we acknowledge the need for problem-based learning. Above, I have outlined how eduScrum can be used to successfully in both a teacher preparation program and in the K-12 classroom. Teachers in New York are required to implement standards into their teaching practice. I believe that problem-based learning is the most effective way to teach these standards, and that eduScrum offers an appropriate methodology for implementing problem-based learning in a K-12 classroom.

References

Adnan, N., & Ritzhaupt, A. (2017). Software Engineering Design Principles Applied to Instructional Design: What can we Learn from our Sister Discipline? *TechTrends*, 62(1), 77–94.

Agile Alliance. (2020). *Burndown Chart*. Retrieved from Agile Alliance: https://www.agilealliance.org/glossary/burndown-chart/#q=~(infinite~false~filters~(postType~(~'page~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'burndown*20chart))~searchTerm~'~sort~false~sortDi

Agile Alliance. (2020). *Velocity*. Retrieved from Agile Alliance: https://www.agilealliance.org/glossary/velocity/#q=~(infinite~false~filters~(postType~(~'page~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'velocity))~searchTerm~'~sort~false~sortDirection~'asc~p

Anderson, L. W., Krathwohl, D. R., & Bloom, B. S. (2001). *A Taxonomy for Learning, Teaching, and Assessing : a Revision of Bloom's Taxonomy of Educational Objectives / Editors, Lorin W. Anderson, David Krathwohl ; Contributors, Peter W. Airasian ... [et Al.]*. New York: Longman.

Connecticut Commission for Educational Technology. (n.d.). *Digital Learning Standards.* Retrieved from https://portal.ct.gov/DAS/CTEdTech/Commission-for-Educational-Technology/Initiatives/Digital-Learning-Standards

Crompton, H. (2014). ISTE standards in the research. *Learning & Leading with Technology*, 41(6), 38+.

Deruy, E. (2016, October 19). *A Plan to Teach Every Child Computer Science*. Retrieved from The Atlantic: https://www.theatlantic.com/education/archive/2016/10/a-plan-to-teach-every-child-computer-science/504587/

Dondlinger, M. J., McLeod, J., & Vasinda. (2016). Essential Conditions for Technology-Supported, Student-Centered Learning: An Analysis of Student Experiences With Math Out Loud Using the ISTE Standards for Students. *Journal of Research on Technology in Education*, 48:4, 258-273.

El-Khalili, N. (2013). Teaching Agile Software Engineering Using Problem-Based Learning. *International Journal of Information and Communication Technology Education*, 9(3), 1–12.

Foresman, B. (2018, December 7). *North Dakota coordinates statewide Hour of Code. Scoop News Group. Retrieved from the State Scoop*. Retrieved from State Scoop: https://statescoop.com/north-dakota-coordinates-statewide-hour-of-code/

Friedman, A., Bolick, C., Berson, M., & Porfeli, E. (2009). National educational technology standards and technology beliefs and practices of social studies faculty: Results from a seven-year longitudinal study. *Contemporary Issues in Technology and Teacher Education*, 9(4).

In Piers, M. W., Piaget, J., Institute., E., Chicago., L. U., & Chicago., L. U. (1972). *Play and development: A symposium.* New York: Norton.

International Society for Technology in Education. (2016). *Redefining learning in a technology-driven world: A report to support adoption of the ISTE Standards for Students.* Retrieved from ISTE: https://id.iste.org/docs/Standards-Resources/iste-standards_students-2016_research-validity-report_final.pdf?sfvrsn=0.0680021527232122

K-12 Computer Science Framework. (2016). Retrieved from https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science- Framework.pdf

K-12 Computer Science Framework: View by Progression. (2016). Retrieved from https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf

Lambert, D. (2018, September 5). *California adopts first computer science standards for K-12 students*. Retrieved from EdSource: https://edsource.org/2018/californias-first-computer-science-standards-set-for-approval/601985

LeanDog. (2019). *LeanDog Agile Discussion Guide 4th Edition.* Cleveland: LeanDog.

Lei, H., Ganjeizadeh, F., Jayachandran, P., & Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, 43, 59–67.

Linden, T. (2018). Scrum-Based Learning Environment: Fostering Self-Regulated Learning. *Journal of Information Systems Education*, 29(2), 65–74.

Loewus, L. (2017). Schools Take a Page From Silicon Valley With “Scrum” Approach. *Education Week*, pp. 37(11), 1–.

López-Alcarria, A., Olivares-Vicente, A., & Poza-Vilches, F. (2019). A Systematic Review of the Use of Agile Methodologies in Education to Foster Sustainability Competencies. *Sustainability (Basel, Switzerland)*, 11(10), 2915–.

May, J., York, J., & Lending, D. (2016). Play ball: bringing scrum into the classroom. *Journal of Information Systems Education*, 27(2), 87–.

Krathwohl, D. R. (2002). A Revision of Bloom’s Taxonomy: An Overview. *Theory Into Practice*, *41*(4), 212. https://doi-org.proxy.buffalostate.edu/10.1207/s15430421tip4104_2

NETS are now ISTE Standards. (2013). *Learning & Leading with Technology*, 41(3), 8.

New York State Board of Regents. (2020, January 13). *Board of Regents Acts to Conditionally Approve First Ever Learning Standards for Computer Science and Digital Fluency*. Retrieved from New York State Department of Education:

http://www.nysed.gov/news/2020/board-regents-acts-conditionally-approve-first-ever-learning-standards-computer-science

New York State Department of Education. (2020). *Computer Science Certificate Coursework Guidance*. Retrieved from New York State Department of Education: http://www.nysed.gov/college-university-evaluation/computer-science-certificate-coursework-guidance#a

New York State Education Department. (2020). *Computer Science and Digital Fluency Learning Standards*. Retrieved from New York State Education Department Curriculum and Instruction: http://www.nysed.gov/curriculum-instruction/computer-science-and-digital-fluency-learning-standards

Niederhauser, D., Lindstrom, D., & Strobel, J. (2007). Addressing the NETS*S in K-12 Classrooms: Implications for Teacher Education. *Journal of Technology and Teacher Education*, 15(4), 483-512.

NYSUT Research and Educational Services. (2018, April 13). *Fact Sheet 18-6 Certification in Computer Science.* Retrieved from NYSUT: https://www.nysut.org/resources/all-listing/research/fact-sheets/fact-sheet-certification-in-computer-science

Ohio's Learning Standards for Technology. (n.d.). Retrieved from http://education.ohio.gov/Topics/Learning-in-Ohio/Technology/Ohios-Learning-Standards-for-Technology

Parsons, D., & MacCallum, K. (2018). *Agile and Lean Concepts for Teaching and Learning: Bringing Methodologies from Industry to the Classroom.* Springer.

Piers, M. W., Piaget, J., Erikson Institute., Loyola University Chicago., & Loyola University Chicago. (1972). *Play and development: A symposium*. New York: Norton.

Rafael Gomes, N. (2017, April 17). *eduScrum goes international!* Retrieved from Medium: https://medium.com/seriouslearning/eduscrum-goes-international-b768c6755d6a

Rasmusson, J. (2020). *Burndown Charts.* Retrieved from Agile in a Nutshell: http://www.agilenutshell.com/burndown

Ronan, B. (2018). Standards-Based Technology Integration for Emergent Bilinguals. *Multicultural Education*, 25(2), 7.

Saltz, J., & Heckman, R. (2020). Exploring Which Agile Principles Students Internalize When Using a Kanban Process Methodology. *Journal of Information Systems Education*, 31(1), 51–60.

Salza, P., Musmarra, P., & Ferrucci, F. (2019). Agile Methodologies in Education: A Review. In D. Parsons, & K. MacCallum, *Agile and Lean Concepts for Teaching and Learning.* Singapore: Springer.

Scott, E., Rodríguez, G., Soria, Á., & Campo, M. (2016). Towards better Scrum learning using learning styles. *The Journal of Systems & Software*, 111, 242–253.

Sinek, S. (2020). *The Golden Circle: Presenter Slides and Notes.* Retrieved from Simon Sinek: https://simonsinek.com/product/share-the-golden-circle-presenter-slides-and-notes/

Sinek, S. (2020). *The Golden Circle: Presenter Slides and Notes.* Retrieved from Simon Sinek: https://simonsinek.com/product/share-the-golden-circle-presenter-slides-and-notes/

Statements of Support. (n.d.). Retrieved from K-12 Computer Science Framework: https://k12cs.org/statements-of-support/

Sutherland, J., & Schwaber, K. (2017). *Scrum Guide.* Scrum.org.

Sutherland, J., & Schwaber, K. (2017). *Scrum Guide.* Retrieved from Scrum.org: https://www.scrumguides.org/scrum-guide.html

Wijnands, W. (2020, January). *The eduScrum Guide.* Retrieved from https://drive.google.com/file/d/1lDFc9cgEJ-v9PQYWox-YjG8Z15gzE8R3/view

Wijnands, W., & Stolze, A. (2019). Transforming Education with eduScrum. In D. Parsons, & K. MacCallum, *Agile and Lean Concepts for Teaching and Learning.* Singapore: Springer.

Appendices

Appendix A: Outline of Sprints for the Recommendation to Buffalo State

Sprint #	1
Student Population	12th Grade Suburban AP Computer Science Students
Concept	Computational Thinking
Key Standards	9-12.CT.8 Identify a relevant module, library, or API and use it in a computer program to add a feature or functionality. 9-12.CT.10 Develop a program that effectively uses control structures in order to create a computer program for practical intent, personal expression, or to address a societal issue.
Suggested Research Literature	Transforming Education with eduScrum by Willy Wijnands, Alisa Stolze eduScrum Guide by the eduScrum team
Student Stories	"I will be able to utilize Agile methodologies in the collaborative creation of a computer program." "I will create a program for a 'client', making sure that the needs of the client are met." "I will understand the importance of Agile rituals, artifacts, and roles."
Celebration Criteria	"I have created a unit plan that teaches the use of Agile methodologies." "I have communicated my understanding of relevant research to the professor, and how I have applied that research to the creation of the unit plan." "I have justified my use of at least 4 standards, and clearly communicated this justification to the professor." "I have created a unit plan that features all relevant instruction to prepare the students for a final project, presentation, or assessment."

	"I have clearly communicated to the professor how all the lessons and projects in the unit plan contribute to the celebration criteria, objectives, and standards in the unit plan and build on each other." "I have clearly communicated to the professor how my unit plan meets the needs of my designated student population." "I have effectively utilized collaboration and Agile practices in the completion of this product."

Sprint #	2
Student Population	8th Grade Rural Intro to Computing Students
Concept	Digital Literacy
Key Standards	6-8.DL.3 Compare types of search tools, choose a search tool for effectiveness and efficiency, and evaluate the quality of search tools based on returned results. ISTE for Students 3b Students evaluate the accuracy, perspective, credibility and relevance of information, media, data or other resources.
Suggested Research Literature	Getting Agile at School by Paul Magnuson, William Tihen, Nicola Cosgrove, Daniel Patton
Student Stories	"I clearly communicate the difference between different search tools and media sources." "I understand the impacts of media on the interpretation of events in the news and popular culture." "I can clearly communicate the way that search tools affect the kind of information I receive." "I understand how media I consume affects the way I view the world."
Celebration Criteria	"I have created a unit plan that teaches the use of Agile methodologies."

	"I have communicated my understanding of relevant research to the professor, and how I have applied that research to the creation of the unit plan." "I have justified my use of at least 4 standards, and clearly communicated this justification to the professor." "I have created a unit plan that features all relevant instruction to prepare the students for a final project, presentation, or assessment." "I have clearly communicated to the professor how all the lessons and projects in the unit plan contribute to the celebration criteria, objectives, and standards in the unit plan and build on each other." "I have clearly communicated to the professor how my unit plan meets the needs of my designated student population." "I have effectively utilized collaboration and Agile practices in the completion of this product."

Sprint #	3
Student Population	5th Grade Urban Students
Concept	Cybersecurity
Key Standards	3-5.CY.1 Explain why different types of information might need to be protected. 3-5.CY.2 Describe common safeguards for protecting personal information.
Suggested Research Literature	Teaching and Fostering Reflection in Software Engineering Project Courses Håkan Burden, Jan-Philipp Steghöfer
Student Stories	"I know what data is given away when I use the internet and why it's important to protect it." "I know how to guard my identity when online." "I can describe different ways people try to steal my identity and how to prevent them."

Celebration Criteria	“I have created a unit plan that teaches the use of Agile methodologies.” “I have communicated my understanding of relevant research to the professor, and how I have applied that research to the creation of the unit plan.” “I have justified my use of at least 4 standards, and clearly communicated this justification to the professor.” “I have created a unit plan that features all relevant instruction to prepare the students for a final project, presentation, or assessment.” “I have clearly communicated to the professor how all the lessons and projects in the unit plan contribute to the celebration criteria, objectives, and standards in the unit plan and build on each other.” “I have clearly communicated to the professor how my unit plan meets the needs of my designated student population.” “I have effectively utilized collaboration and Agile practices in the completion of this product.”

Sprint #	4
Student Population	9th Grade Alternative Education Students
Concept	Impacts of Computing
Key Standards	9-12.IC.1 Evaluate the impact of computing technologies on equity, access, and influence in a global society. ISTE for Students 3d Students build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.
Suggested Research Literature	Lean Learning of Risks in Students’ Agile Teams by Wentao Wang, Chaitra Thota, Xiaoyu Jin, Nan Niu, Carla C. Purdy
Student Stories	“I have related what I am learning in my computer class to events in the news or history.” “I have chosen an opinion on computing and defended it.”

	"I have found an issue related to computing and made a recommendation to fix it."
Celebration Criteria	"I have created a unit plan that teaches the use of Agile methodologies." "I have communicated my understanding of relevant research to the professor, and how I have applied that research to the creation of the unit plan." "I have justified my use of at least 4 standards, and clearly communicated this justification to the professor." "I have created a unit plan that features all relevant instruction to prepare the students for a final project, presentation, or assessment." "I have clearly communicated to the professor how all the lessons and projects in the unit plan contribute to the celebration criteria, objectives, and standards in the unit plan and build on each other." "I have clearly communicated to the professor how my unit plan meets the needs of my designated student population." "I have effectively utilized collaboration and Agile practices in the completion of this product."

Sprint #	5
Student Population	10^{th} Grade Urban Robotics Students
Concept	Data and Analysis
Key Standards	9-12.CT.2 Collect data from multiple sources for use in a computational artifact. ISTE for Students 6c Students communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.
Suggested Research Literature	Criterion-Based Grading, Agile Goal Setting, and Course (Un)Completion Strategies by Petri Ihantola, Essi Isohanni, Pietari Heino, Tommi Mikkonen

Student Stories	"I have collected data in the creation of my robot." "I have developed conclusions from the data in the creation of my robot." "I have clearly communicated to my teacher how I will use this data to improve my robot design."
Celebration Criteria	"I have created a unit plan that teaches the use of Agile methodologies." "I have communicated my understanding of relevant research to the professor, and how I have applied that research to the creation of the unit plan." "I have justified my use of at least 4 standards, and clearly communicated this justification to the professor." "I have created a unit plan that features all relevant instruction to prepare the students for a final project, presentation, or assessment." "I have clearly communicated to the professor how all the lessons and projects in the unit plan contribute to the celebration criteria, objectives, and standards in the unit plan and build on each other." "I have clearly communicated to the professor how my unit plan meets the needs of my designated student population." "I have effectively utilized collaboration and Agile practices in the completion of this product."

Sprint #	6
Student Population	11^{th} Grade Career and Technical Education Students in a Computer Trades Class
Concept	Networks and System Design
Key Standards	9-12.NSD.1 Design a solution to a problem that utilizes embedded systems.

	9-12.NSD.3Develop and communicate multi-step troubleshooting strategies others can use to identify and fix problems with computing devices and their components.
Suggested Research Literature	Red-Green-Go! A Self-Organising Game for Teaching Test-Driven Development by Suzanne M. Embury, Martin Borizanov, Caroline Jay
Student Stories	"I have a created a device that utilizes a network to solve a problem." "I have tested the device and have demonstrated that it works as intended." "I have clearly communicated how to operate the device and how to fix it if it malfunctions."
Celebration Criteria	"I have created a unit plan that teaches the use of Agile methodologies." "I have communicated my understanding of relevant research to the professor, and how I have applied that research to the creation of the unit plan." "I have justified my use of at least 4 standards, and clearly communicated this justification to the professor." "I have created a unit plan that features all relevant instruction to prepare the students for a final project, presentation, or assessment." "I have clearly communicated to the professor how all the lessons and projects in the unit plan contribute to the celebration criteria, objectives, and standards in the unit plan and build on each other." "I have clearly communicated to the professor how my unit plan meets the needs of my designated student population." "I have effectively utilized collaboration and Agile practices in the completion of this product."

Sprint #	7

Student Population	1st Grade Rural Students
Concept	Computational Thinking
Key Standards	K-2.CT.12Use a planning process to outline the steps taken to solve a problem or complete a task. K-2.CT.10 Develop an algorithm that uses repetition structures for creative expression or to solve a problem.
Suggested Research Literature	Using Agile Games to Invigorate Agile and Lean Software Development Learning in Classrooms by Rashina Hoda
Student Stories	"I have planned how I will solve a problem." "I have communicated this plan to others." "I have repeated this plan multiple times to solve the problem multiple times."
Celebration Criteria	"I have created a unit plan that teaches the use of Agile methodologies." "I have communicated my understanding of relevant research to the professor, and how I have applied that research to the creation of the unit plan." "I have justified my use of at least 4 standards, and clearly communicated this justification to the professor." "I have created a unit plan that features all relevant instruction to prepare the students for a final project, presentation, or assessment." "I have clearly communicated to the professor how all the lessons and projects in the unit plan contribute to the celebration criteria, objectives, and standards in the unit plan and build on each other." "I have clearly communicated to the professor how my unit plan meets the needs of my designated student population." "I have effectively utilized collaboration and Agile practices in the completion of this product."

Appendix B: Differentiation of Scrum Topics by Learning Style

Table 2
Organization of the Scrum topics by instructional method.

	Main topic		Sub-topic	Practices for Reflective Students (Passive teaching style) After explain the concept, the professor takes a pause and allow students to think about...	Practices for Active Students (Active teaching style) After a brief explanation of the topic, the professor helps to ...
1.	Where we stand in the Scrum framework?		-		
2.	User stories	2.1.	User stories parts	... different roles that could need something from the system	... identify user stories' parts (e.g. role, acceptance criteria)
		2.2	When should you stop to disaggregate?	... a single feature to be developed	... elaborate user stories from a given requirement. Then, disaggregate them correctly
		2.2.	Functional and non-functional requirements on user stories	... how to specify quality attributes in US	... add to the US quality attributes
		2.3.	Epics	... if it is possible to develop the US	... identify a potential Epic from the USs
3.	Agile estimating	3.1.	Size, velocity and duration of sprints	... how much work can the team complete along the Sprint?	... suppose different values of size, velocity of each team and calculate the number of Sprints estimated
		3.2.	Story points and working hours	... why is important to understand that Story Points cannot be mappable to working hours	... discuss within the team about the relationship between story points and working hours
4.	Agile planning	4.1.	Planning poker introduction	... a consensus-based game	
		4.2.	Complexity units in the estimation	... a matter of comparing different opinions on the effort required by a task	... discuss within the team about the use of a fixed scale or a Fibonacci's scale
		4.3.	How to proceed to perform planning poker?	... the synergy generated in the game and the estimation results	... perform planning poker in the group for assigning story points to each US previously identified
		4.4.	Release plan	... how the team should organize all the features to be developed along the iterations	... organize the estimated US along the number of iterations previously calculated

(Scott et al., 2016, p. 246)

Appendix C: Crosswalk Between K-12 Computer Science Framework and ISTE Standards

Below you can find the results of a crosswalk performed during the research for this thesis. This Crosswalk is exploratory in nature, showing connections between the two frameworks for use in assignment creation. The crosswalk was created based on my personal judgement. For a more detailed description of each ISTE standard and the K-12 Computer Science Framework Practices that relate to it, please see Appendix D. For a more detailed description of each K-12 Computer Science Framework Practice and the ISTE Standards that relate to it, please see Appendix E.

An X on the chart signifies a strong connection. An O signifies a weak connection. If there is an explanation point next to the notation, it shows a strong connection when viewing from one framework over the other.

K-12 Computer Science Framework Practices

ISTE Standards

	1.1	1.2	1.3	2.1	2.2	2.3	2.4	3.1	3.2	3.3	4.1	4.2	4.3	4.4	5.1	5.2	5.3	6.1	6.2	6.3	7.1	7.2
1.A		X!					O				O	O				O				!O		
1.B					!O		!O				O		O									
1.C	O				!O											O		O				
1.	O										O			O				O	O	O		
2.A		X										O		O!			O					

O						O	X							X!	O!			
				O				!O	!X	!X					O			O
!X			!O		!O		O	X!	O!					!O				
		!O				X	!O	O!			!O	X						
	O	X		O	O!		!O				O!		O	O	O			
O		O	!O	X!	!O		O				O	O	O	O!	O!			O
	O!	O!	O!			O		O	O	O					O			
	O	X	O					O	O							O		
					O	!O		O!	!O	O	X	X!	O	O				
X			O		X!			O	O									O
	O	O	O		O			X	O!			O	O		O			
	X	O					O		!O									
				X!			O				O	O		X	O			
														O				
					O!									!X				
										!O								
									O									
										!O								
O					!O					!O								
O		O		O!	X!	O	O!		X!	O	O!	O!	O!					
									O!	O!	O!	O!	O!		O			
O	O					O!			O							O		
2.B	2.C	2.	3.A	3.B	3.C	3.	4.A	4.B	4.C	4.	5.A	5.B	5.C	5.	6.A	6.B	6.C	6.

7.A			O			O																
7.B			O					O	O							O				O		O
7.C	O		O	O		O	O								O							
7.						O	O		O											O		

Appendix D: K-12 Computer Science Framework Practices Relevant to Each ISTE Standard

Standards that were determined to be highly related are bold.

1. Empowered Learner STUDENTS LEVERAGE TECHNOLOGY TO TAKE AN ACTIVE ROLE IN CHOOSING, ACHIEVING AND DEMONSTRATING COMPETENCY IN THEIR LEARNING GOALS, INFORMED BY THE LEARNING SCIENCES.	
ISTE Standard	*Related K-12 Computer Science Framework Practice*
1.A articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes.	• 2.2 Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness. • 2.4 Evaluate and select technological tools that can be used to collaborate on a project. • 5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.
1.B build networks and customize their learning environments in ways that support the learning process.	• 2.1 Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities. • 2.2 Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness.
1.C use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.	• 2.3 Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders. • 5.3 Modify an existing artifact to improve or customize it. • 6.2 Identify and fix errors using a systematic process. • 6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.
1.D understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their	• 2.4 Evaluate and select technological tools that can be used to collaborate on a project. • 4.3 Create modules and develop points of interaction that can apply to multiple situations and reduce complexity. • 5.3 Modify an existing artifact to improve or customize it.

knowledge to explore emerging technologies.	
2. Digital Citizen STUDENTS RECOGNIZE THE RIGHTS, RESPONSIBILITIES AND OPPORTUNITIES OF LIVING, LEARNING AND WORKING IN AN INTERCONNECTED DIGITAL WORLD, AND THEY ACT AND MODEL IN WAYS THAT ARE SAFE, LEGAL AND ETHICAL.	
ISTE Standard	*Related K-12 Computer Science Framework Practice*
2.A cultivate and manage their digital identity and reputation and are aware of the permanence of their actions in the digital world.	• 4.2 Evaluate existing technological functionalities and incorporate them into new designs. • 5.3 Modify an existing artifact to improve or customize it.
2.B engage in positive, safe, legal and ethical behavior when using technology, including social interactions online or when using networked devices.	• 1.1 Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products. • 1.3 Employ self- and peer-advocacy to address bias in interactions, product design, and development methods. • 2.1 Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities. • 5.3 Modify an existing artifact to improve or customize it.
2.C demonstrate an understanding of and respect for the rights and obligations of using and sharing intellectual property.	• 1.1 Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.
2.D manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online.	• 1.3 Employ self- and peer-advocacy to address bias in interactions, product design, and development methods. • 5.3 Modify an existing artifact to improve or customize it
3. Knowledge Constructor	

STUDENTS CRITICALLY CURATE A VARIETY OF RESOURCES USING DIGITAL TOOLS TO CONSTRUCT KNOWLEDGE, PRODUCE CREATIVE ARTIFACTS AND MAKE MEANINGFUL LEARNING EXPERIENCES FOR THEMSELVES AND OTHERS.	
ISTE Standard	*Related K-12 Computer Science Framework Practice*
3.A plan and employ effective research strategies to locate information and other resources for their intellectual or creative pursuits.	• **7.1 Select, organize, and interpret large data sets from multiple sources to support a claim.**
3.B evaluate the accuracy, perspective, credibility and relevance of information, media, data or other resources.	• 7.1 Select, organize, and interpret large data sets from multiple sources to support a claim.
3.C curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions.	• 4.1 Extract common features from a set of interrelated processes or complex phenomena. • 4.4 Model phenomena and processes and simulate systems to understand and evaluate potential outcomes. • **7.1 Select, organize, and interpret large data sets from multiple sources to support a claim.** • 7.2 Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.
3.D build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.	• 1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability. • 1.3 Employ self- and peer-advocacy to address bias in interactions, product design, and development methods. • **3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally.** • 3.2 Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures. • 4.4 Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.

	• 5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue.
4. Innovative Designer STUDENTS USE A VARIETY OF TECHNOLOGIES WITHIN A DESIGN PROCESS TO IDENTIFY AND SOLVE PROBLEMS BY CREATING NEW, USEFUL OR IMAGINATIVE SOLUTIONS.	
ISTE Standard	*Related K-12 Computer Science Framework Practice*
4.A know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.	• 1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability. • 3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally. • 3.2 Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures. • 4.2 Evaluate existing technological functionalities and incorporate them into new designs. • **5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations**. • **5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue.** • 6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.
4.B select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.	• **2.4 Evaluate and select technological tools that can be used to collaborate on a project.** • 3.3 Evaluate whether it is appropriate and feasible to solve a problem computationally. • 4.2 Evaluate existing technological functionalities and incorporate them into new designs. • 5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.

4.C develop, test and refine prototypes as part of a cyclical design process.	• 4.2 Evaluate existing technological functionalities and incorporate them into new designs. • 4.3 Create modules and develop points of interaction that can apply to multiple situations and reduce complexity. • 5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue. • 5.3 Modify an existing artifact to improve or customize it. • **6.1 Systematically test computational artifacts by considering all scenarios and using test cases**. • **6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.**
4.D exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.	• 3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally. • **3.2 Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.** • 3.3 Evaluate whether it is appropriate and feasible to solve a problem computationally. • 5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.
5. Computational Thinker STUDENTS DEVELOP AND EMPLOY STRATEGIES FOR UNDERSTANDING AND SOLVING PROBLEMS IN WAYS THAT LEVERAGE THE POWER OF TECHNOLOGICAL METHODS TO DEVELOP AND TEST SOLUTIONS.	
ISTE Standard	*Related K-12 Computer Science Framework Practice*
5.A formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.	• **3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally.** • **3.2 Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.** • 5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.

	• 5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue.
5.B collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.	• 2.4 Evaluate and select technological tools that can be used to collaborate on a project. • 3.3 Evaluate whether it is appropriate and feasible to solve a problem computationally. • 4.1 Extract common features from a set of interrelated processes or complex phenomena. • 4.2 Evaluate existing technological functionalities and incorporate them into new designs. • 4.4 Model phenomena and processes and simulate systems to understand and evaluate potential outcomes. • 6.1 Systematically test computational artifacts by considering all scenarios and using test cases. • 6.2 Identify and fix errors using a systematic process. • 6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility. • **7.1 Select, organize, and interpret large data sets from multiple sources to support a claim.** • **7.2 Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.**
5.C break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.	• 3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally. • **3.2 Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.** • 4.1 Extract common features from a set of interrelated processes or complex phenomena. • 4.4 Model phenomena and processes and simulate systems to understand and evaluate potential outcomes. • 7.1 Select, organize, and interpret large data sets from multiple sources to support a claim. • 7.2 Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.
5.D understand how automation works and use algorithmic thinking to	• 3.2 Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.

<table>
<tr><td>develop a sequence of steps to create and test automated solutions.</td><td>4.3 Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.
4.4 Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.
5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.
6.1 Systematically test computational artifacts by considering all scenarios and using test cases.
6.2 Identify and fix errors using a systematic process.</td></tr>
<tr><td colspan="2">6. Creative Communicator

STUDENTS COMMUNICATE CLEARLY AND EXPRESS THEMSELVES CREATIVELY FOR A VARIETY OF PURPOSES USING THE PLATFORMS, TOOLS, STYLES, FORMATS AND DIGITAL MEDIA APPROPRIATE TO THEIR GOALS.</td></tr>
<tr><td>ISTE Standard</td><td>Related K-12 Computer Science Framework Practice</td></tr>
<tr><td>6.A choose the appropriate platforms and tools for meeting the desired objectives of their creation or communication.</td><td>1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.
2.4 Evaluate and select technological tools that can be used to collaborate on a project.
3.3 Evaluate whether it is appropriate and feasible to solve a problem computationally.
4.2 Evaluate existing technological functionalities and incorporate them into new designs.
5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.
7.1 Select, organize, and interpret large data sets from multiple sources to support a claim.</td></tr>
<tr><td>6.B create original works or responsibly repurpose or remix digital resources into new creations.</td><td>1.1 Include the unique perspectives of others and reflect on one’s own perspectives when designing and developing computational products.</td></tr>
</table>

	• 5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations. • **5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue.** • **5.3 Modify an existing artifact to improve or customize it.**
6.C communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.	• 2.4 Evaluate and select technological tools that can be used to collaborate on a project. • 4.3 Create modules and develop points of interaction that can apply to multiple situations and reduce complexity. • **4.4 Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.** • 7.1 Select, organize, and interpret large data sets from multiple sources to support a claim. • **7.2 Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.**
6.D publish or present content that customizes the message and medium for their intended audiences.	• **1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.** • 2.3 Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders. • 4.3 Create modules and develop points of interaction that can apply to multiple situations and reduce complexity. • **5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue.** • 5.3 Modify an existing artifact to improve or customize it.
7. Global Collaborator STUDENTS USE DIGITAL TOOLS TO BROADEN THEIR PERSPECTIVES AND ENRICH THEIR LEARNING BY COLLABORATING WITH OTHERS AND WORKING EFFECTIVELY IN TEAMS LOCALLY AND GLOBALLY.	

ISTE Standard	*Related K-12 Computer Science Framework Practice*
7.A use digital tools to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning.	• **1.1 Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.** • 1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability. • 1.3 Employ self- and peer-advocacy to address bias in interactions, product design, and development methods. • **2.1 Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.** • 2.3 Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders.
7.B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.	• **1.1 Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.** • 1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability. • **2.1 Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.** • 2.2 Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness. • **2.3 Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders.** • 2.4 Evaluate and select technological tools that can be used to collaborate on a project. • 6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.
7.C contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.	• 1.1 Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products. • 1.3 Employ self- and peer-advocacy to address bias in interactions, product design, and development methods.

	• 2.1 Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities. • **2.2 Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness.** • 2.3 Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders. • 2.4 Evaluate and select technological tools that can be used to collaborate on a project. • 5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations. • 7.2 Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.
7.D explore local and global issues and use collaborative technologies to work with others to investigate solutions.	• **1.1 Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.** • **1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.** • 1.3 Employ self- and peer-advocacy to address bias in interactions, product design, and development methods. • 2.1 Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities. • 2.2 Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness. • 2.3 Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders. • 2.4 Evaluate and select technological tools that can be used to collaborate on a project. • **3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally.** • 3.2 Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures. • **5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue.**

(International Society for Technology in Education, 2016, p.14-16)

(“K-12 Computer Science Framework”, 2016, p. 74-83)

Appendix E: ISTE Standards Relevant to Each K-12 Computer Science Practice

Strong connections are marked in bold.

K-12 Computer Science Framework *Practice 1. Fostering an Inclusive Computing Culture*	
Practice from K-12 Framework	**Aligning ISTE Standards**
1.1 Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.	• 1.C use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways. • 6.C communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations. • **7.B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.**
1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.	• **1.C use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways** • 3.D build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions. • **6.D publish or present content that customizes the message and medium for their intended audiences**. • 7.A use digital tools to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning. • 7. B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints. • **7.D explore local and global issues and use collaborative technologies to work with others to investigate solutions.**
1.3 Employ self- and peer-advocacy to address bias in interactions, product design, and development methods.	• 3.D build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions. • 7. B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.

	• **7.D explore local and global issues and use collaborative technologies to work with others to investigate solutions.**
Practice 2. Collaborating Around Computing	
Practice from K-12 Framework	**Aligning ISTE Standards**
2.1 Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.	• **7.A use digital tools to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning.** • 7.B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints. • 7.D explore local and global issues and use collaborative technologies to work with others to investigate solutions.
2.2 Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness.	• 1.A articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes. • 1.B build networks and customize their learning environments in ways that support the learning process. • 7.B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints. • **7.C contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.** • 7.D explore local and global issues and use collaborative technologies to work with others to investigate solutions.
2.3 Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders.	• **1.C use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.** • 4.D develop, test and refine prototypes as part of a cyclical design process. • 6.D publish or present content that customizes the message and medium for their intended audiences. • 7.B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.
2.4 Evaluate and select technological tools that can be used to collaborate on a project.	• **4.B select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.**

	• 5.B collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making. • 6.C communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations. • 7.A use digital tools to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning. • **7.B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.**
Practice 3. Recognizing and Defining Computational Problems	
Practice from K-12 Framework	**Aligning ISTE Standards**
3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally.	• **3.D build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.** • 4.A know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems. • 4.D exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems. • 5.A formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions. • 5.C break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving. • 7.B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.
3.2 Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.	• 3.D build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.5 • 4.A know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems. • **4.D exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.**

	• 5.A formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions. • **5.C break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.** • 7.B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.
3.3 Evaluate whether it is appropriate and feasible to solve a problem computationally.	• 4.D exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems. • 5.A formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.
Practice 4. Developing and Using Abstractions	
Practice from K-12 Framework	**Aligning ISTE Standards**
4.1 Extract common features from a set of interrelated processes or complex phenomena.	• 1.D understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. • **3.C curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions.** • **5.B collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.** • 5.C break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
4.2 Evaluate existing technological functionalities and incorporate them into new designs.	• **1.D understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.** • 3.A plan and employ effective research strategies to locate information and other resources for their intellectual or creative pursuits.

	• 4.A know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems. • **4.B select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.** • 4.C develop, test and refine prototypes as part of a cyclical design process. • 5.C break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
4.3 Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.	• 1.B build networks and customize their learning environments in ways that support the learning process. • 1.D understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. • **5.D understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.** • 6.C communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.
4.4 Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.	• 1.D understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. • 3.C curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions. • 3.D build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions. • 5.B collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making. • **6.C communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.**

Practice 5. Creating Computational Artifacts	
5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.	• **1.A articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes.** • **4.A know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.** • 4.B select and use digital tools to plan and manage a design process that considers design constraints and calculated risks. • 4.C develop, test and refine prototypes as part of a cyclical design process. • 4.D exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems. • 5.A formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions. • 5.D understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions. • **6.A choose the appropriate platforms and tools for meeting the desired objectives of their creation or communication.**
5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue.	• 1.A articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes. • 1.C use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways. • **3.D build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.** • 4.A know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems. • 4.D exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems. • 5.A formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.

	• 6.A choose the appropriate platforms and tools for meeting the desired objectives of their creation or communication. • 6.D publish or present content that customizes the message and medium for their intended audiences. • 7.B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints. • **7.D explore local and global issues and use collaborative technologies to work with others to investigate solutions.**
5.3 Modify an existing artifact to improve or customize it.	• 1.C use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways. • 1.D understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. • 4.C develop, test and refine prototypes as part of a cyclical design process.
Practice 6. Testing and Refining Computational Artifacts	
6.1 Systematically test computational artifacts by considering all scenarios and using test cases.	• 1.C use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways. • 1.D understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. • **4.C develop, test and refine prototypes as part of a cyclical design process.** • 5.B collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making. • 5.D understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.
6.2 Identify and fix errors using a systematic process.	• 1.C use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways. • 1.D understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and

	troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. • **4.C develop, test and refine prototypes as part of a cyclical design process.** • 5.B collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making. • 5.D understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.
6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.	• 1.C use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways. • 1.D understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. • **4.C develop, test and refine prototypes as part of a cyclical design process.** • 5.B collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making. • **6.D publish or present content that customizes the message and medium for their intended audiences.** • 7.D explore local and global issues and use collaborative technologies to work with others to investigate solutions
Practice 7. Communicating About Computing	
7.1 Select, organize, and interpret large data sets from multiple sources to support a claim.	• **3.C curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions.** • 5.B collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making. • 5.C break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving. • **6.C communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.**

	• 6.D publish or present content that customizes the message and medium for their intended audiences.
7.2 Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.	• 3.C curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions. • 5.B collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making. • 5.C break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving. • 6.C communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations. • **6.D publish or present content that customizes the message and medium for their intended audiences.** • 7.B use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints. • 7.C contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.

(International Society for Technology in Education, 2016, p.14-16)

(“K-12 Computer Science Framework”, 2016, p. 74-83)

NYS K-12 Computer Science and Digital Fluency Standards

Impacts of Computing

Computing affects many aspects of the world at local, national, and global levels. Individuals and communities influence computing through their behaviors and cultural and social interactions. In turn, computing influences new cultural practices. Informed citizens understand the ethical and social implications of the digital world, including equity and access to computing and computing technologies.

The Impacts of Computing standards promote an understanding of the evolving impact of computing technologies on society through many lenses, including personal, social, cultural, accessibility, legal, economic, and ethical.

Society	Computing can change or reinforce cultural practices and equity within society. Human social structures that support education, work, and communities have been affected by the ease of communication facilitated by computing. Governments enact laws to influence the impact of computing technologies on society.
Ethics	Computing is not done in a vacuum. The question of ethics in computing is for both creators and users of technology. If *computer scientists* and end users do not take into account biases and ethics of what has been built, algorithms and programs may have unintended impacts on societies.
Accessibility	The development and design of computing systems needs to take into account the needs and wants of diverse end users and purposefully consider potential perspectives of users with different backgrounds and ability levels. Identifying potential personal bias during the design and implementation process maximizes accessibility in product design, and awareness of professionally accepted accessibility standards helps to evaluate computational artifacts for accessibility.
Career Paths	The increased connectivity between people in different cultures and in different career fields has impacted the variety and types of careers that are possible. There are also many possible career paths within computer science itself, as well as different specialties within each field, that make computer science a broad and encompassing opportunity.

Impacts of Computing

Society

K-2	3-5	6-8	9-12
K-2.IC.1 **Identify how individuals lived and worked before and after the adoption or implementation of new computing technology.**	3-5.IC.1 **Describe computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.**	6-8.IC.1 **Compare and contrast tradeoffs associated with computing technologies that affect individuals and society.**	9-12.IC.1 **Evaluate the impact of computing technologies on equity, access, and influence in a global society.**
Clarifying Statement: The focus should be on how computing technology has changed the way people live and work, including the similarities and differences of a task with or without computing technology.	Clarifying Statement: The focus should be on how computing technologies both influence and are influenced by society and culture.	Clarifying Statement: Topics that could be addressed include, but are not limited to, free speech, communication, and automation.	Clarifying Statement: The focus should be on how computing technologies can both perpetuate inequalities and help to bring about equity in society.
For Example: Students could discuss how words and phrases can easily be translated into other languages using computing technology and how this is accessible to anyone with an internet connection. Students could then discuss how people would have translated words and phrases from one language to another in the past, without computing technology.	For Example: Students could discuss how technology has influenced communication, relationships, and the practices of cultural traditions and customs. Alternatively, students might discuss how technologies, such as GPS systems, have changed how people travel, as well as the way they explore new places.	For Example: Students could identify trade-offs with a new and emerging technology, discussing how the technology could improve convenience, but also impact personal privacy.	For Example: Students could research how better access to information and/or resources affects a population and develop a strategy and/or recommendation to address the issue.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Impacts of Computing

SOCIETY

K-2	3-5	6-8	9-12
K-2.IC.2 **Identify rules related to computing technologies and digital information.**	**3-5.IC.2** **Explain how laws impact the use of computing technologies and digital information.**	**6-8.IC.2** **Evaluate the impact of laws or regulations on the development and use of computing technologies and digital information.**	**9-12.IC.2** **Debate laws and regulations that impact the development and use of computing technologies and digital information.**
Clarifying Statement: At this level, the focus is on how rules influence computing technologies.	Clarifying Statement: At this level, the focus is on how laws regulate the use of computing technologies and what might happen if those laws did not exist.	Clarifying Statement: At this level, the focus is on the potential consequences of laws related to computing technologies.	Clarifying Statement: At this level, the focus is on developing and defending a claim about how a specific law related to computing technologies impacts different stakeholders.
For Example: Explain how rules impact the use of computing technologies (ex. don't share your password).	For Example: Students could explain how government regulation of the internet affects people's access to information.	For Example: Students could research how laws protect intellectual property rights of digital materials and how those laws changed the music industry.	For Example: Students could write a persuasive essay about a legal dilemma related to an individual's right to privacy being at odds with the safety, security, or well-being of a community.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Impacts of Computing

Ethics

K-2	3-5	6-8	9-12
K-2.IC.3 **Identify computing technologies in the world around us.**	3-5.IC.3 **Explain current events that involve computing technologies.**	6-8.IC.3 **Identify and discuss issues of ethics surrounding computing technologies and current events.**	9-12.IC.3 **Debate issues of ethics related to real world computing technologies.**
Clarifying Statement: Computing technologies can be identified through personal experience or through text/media presented in class.	Clarifying Statement: Explanations should be grade level appropriate to ensure understanding of current events and the related computing technologies.	Clarifying Statement: At this level, students may require teacher support to discuss the possible ethical implications of computing technologies.	Clarifying Statement: The focus is on developing and defending a claim about a specific ethical dilemma related to computing technologies.
For Example: A teacher might keep a class log of all the different computing technologies that they use, see, or read about throughout one school day and compare it to other classrooms.	For Example: Students might read an informational text about an interdisciplinary topic and be able to explain the connection with computing technologies that were presented in the text.	For Example: A teacher might have students find current articles about computing technologies and discuss them in terms of ethical decisions and actions.	For Example: Students might develop and present an argument related to the ethical responsibilities of technology companies.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Impacts of Computing

Ethics

K-2	3-5	6-8	9-12
K-2.IC.4 **Compare the ideas of public and private places and public and private information.**	**3-5.IC.4** **Explain who has access to data in different digital spaces.**	**6-8.IC.4** **Identify and discuss issues of ethics related to the collection and use of data with different computing technologies.**	**9-12.IC.4** **Assess personal and societal trade-offs related to computing technologies and privacy.**
Clarifying Statement: Physical and digital spaces may be public, semi-public, or private.	Clarifying Statement: The focus is on identifying different groups who might have access to data stored or posted in different places, including companies.	Clarifying Statement: The focus is on exploring the ethics of data collection, biases in the data collection, and use by different stakeholders for a range of purposes.	Clarifying Statement: The focus is on discussing the personal and societal benefits and drawbacks of different types of data collection and use, in terms of ethics, policy, and culture.
For Example: Students could describe where a parent could help them post a picture that they want anyone to see, only family to see, or that they don't want anyone to see.	For Example: Students could explain that things posted to online accounts can be accessed by "friends" and "strangers" that they share data with.	For Example: Students could describe how facial recognition surveillance video is used in a store to track customers for security or information about purchase habits. Students might discuss who owns that data and what it is acceptable to do with the data.	For Example: Students could discuss the monitoring of road traffic. They might discuss the trade-offs: changing signals in real time to improve road efficiency and safety versus concerns around consent for personal data collection and potential sharing of personal data with other agencies like the police department or insurance companies.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Impacts of Computing

Ethics

K-2	3-5	6-8	9-12
K-2.IC.5 **Explain how computer decision making is used in daily life.**	**3-5.IC.5** **Explain how computer systems play a role in human decision-making**	**6-8.IC.5** **Analyze potential sources of bias that could be introduced to complex computer systems and the potential impact of these biases on individuals.**	**9-12.IC.5** **Describe ways that complex computer systems can be designed for inclusivity and to mitigate unintended consequences.**
Clarifying Statement: The focus is on identifying AI and understanding that it is a branch of computing technology.	Clarifying Statement: The focus is on explaining a range of ways that humans interact with AI to make decisions.	Clarifying Statement: The focus is on understanding different factors that introduce bias into an AI system and how those biases affect people.	Clarifying Statement: The focus is on applying an understanding of bias and ethical design in order to make recommendations for designing with inclusivity and social good in mind.
For Example: Students could name three characteristics for their perfect pet, and based on the combination of those variables, a "perfect pet" would be suggested by the teacher. The teacher might lead a discussion about how the computers make decisions by comparing information given to determine the best result or answer.	For Example: Students could discuss how recommendation algorithms influence what people select on video and music websites and applications. Alternatively, students could discuss AI that is designed to help professionals make decisions like algorithms that help doctors diagnose patients.	For Example: Students could argue that facial recognition software that works better for certain skin tones was likely developed with a homogeneous testing group and could be improved by sampling a more diverse population. Alternatively, Students could use a search engine to search images and search on the word "grandma" and discuss whether the results of the images are expected, representative, in what way biased.	For Example: Students might consider the ethical and social implications of police departments using artificial intelligence to identify and respond to potential criminal activity. Then make recommendations for how to make such a tool increase equity in policing and mitigate unintended bias caused by the system.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Impacts of Computing

Accessibility

K-2	3-5	6-8	9-12
K-2.IC.6 **Identify factors that make a computing device or software application easier or more difficult to use.**	3-5.IC.6 **Identify ways to improve the accessibility and usability of a computing device or software application for the diverse needs and wants of users.**	6-8.IC.6 **Assess the accessibility of a computing device or software application in terms of user needs.**	9-12.IC.6 **Create accessible computational artifacts that meet standard compliance requirements or otherwise meet the needs of users with disabilities.**
Clarifying Statement: The focus is on identifying in general terms that developers of computing devices and software make design decisions that affect product users.	Clarifying Statement: The focus is on identifying the needs and wants of diverse end users and purposefully considering potential perspectives of users with different backgrounds, ability levels, points of view, and disabilities.	Clarifying Statement: The focus is on testing and discussing the usability and accessibility of various technology tools (e.g., apps, games, and devices) with teacher guidance.	Clarifying Statement: At this level, considering accessibility becomes part of the design process and awareness of professionally accepted accessibility standards.
For Example: Students might compare a travel keyboard with a standard keyboard and note that one is easier to carry around but difficult to type with, while the other might be easier to type with but difficult pack into a bag.	For Example: Students could use both text and speech when they create and convey information in a game that they program. Students might make recommendations for making an app easier to navigate.	For Example: Students might notice that allowing a user to change font sizes and colors will not only make an interface usable for people with low vision but also benefits users in various situations, such as in bright daylight or a dark room.	For Example: Students could make sure that a website they are designing is ADA compliant. Students might consider the needs of users with learning disabilities when designing an educational app.

Impacts of Computing

Career Paths

K-2	3-5	6-8	9-12
K-2.IC.7 **Identify a diverse range of roles in computer science.**	3-5.IC.7 **Identify a diverse range of role models in computer science.**	6-8.IC.7 **Explore a range of computer science-related career paths**	9-12.IC.7 **Investigate the use of computer science in multiple fields**
Clarifying Statement: The focus is not just on the roles in computer science, but also the skills and practices that are necessary for careers in that field.	Clarifying Statement: The emphasis of this standard is the opportunity to personally identify with a range people in fields related to computing technologies.	Clarifying Statement: At this level, the focus is on building awareness of the many different computer science-related careers.	Clarifying Statement: At this level, the focus is on making connections between computer science and the fields of interest of individual students.
For Example: Students could take on the role of "programmer" during computer science lessons. A teacher might emphasize that programmers collaborate to solve problems with code.	For Example: A teacher might provide leveled articles for students to read about people in computer science that reflect diversity in race/ethnicity, gender, disability, sexual orientation, and other characteristics.	For Example: A teacher might spotlight different careers and then have students develop a mind map for the classroom wall that connects all the different career pathways.	For Example: A student interested in fashion design could conduct interviews and do research to find out how computer science intersects with that field.

Computational Thinking

Computational thinking involves thinking about and solving problems in ways that can be carried out by a computer. Computational thinking not only underpins all theory and application of computer science, but also influences many other subject areas. Computational thinking includes both core *concepts*, such as algorithms and variables, and core *practices*, such as abstraction, decomposition, data analysis, modeling, and simulation, that are vital not only to the design and development of computer programs but also to the strategic use of computational power to solve problems across disciplines.

The process of creating meaningful and efficient solutions, often done in collaboration with others, typically involves these steps: defining the problem, breaking apart large problems into smaller ones, recombining existing solutions, analyzing different solutions, using data to inform new potential solutions, and looking at information in new ways to develop innovative solutions.

Computational thinking plays an important role in supporting the creation of solutions to problems, both large and small. Algorithms, programs, simulations, and data are essential to all computing systems, empowering people to communicate and collaborate with others around the world. The standards promote development of foundational skills, knowledge, and experience to solve problems by creating solutions that utilize computational thinking concepts and practices.

Modeling and Simulation	Modeling is the process of representing a system to allow one to observe, understand, or simulate it. Models can be used to simulate real world phenomena that are not easy to observe or reproduce, and often generate simulated data that can further understanding of the system or make predictions.
Data Analysis and Visualization	Data analysis is the process of cleaning, transforming, organizing, clustering, and categorizing data to discover useful information, draw conclusions, and aid in making decisions. Data can be visualized in a variety of ways (including graphs and charts) to aid in and communicate the results of the analysis.
Abstraction and Decomposition	Abstraction is the process of reducing complexity by focusing on key elements. The study of a complicated system often starts by simplifying it and addressing just the most important parts. Complex computer programs also rely on abstraction to isolate particular routines or tasks, especially if those tasks are common. A programmer can then call on that routine, often written by others, without needing to understand its details. Decomposition is the process of strategically breaking complicated problems or tasks into smaller parts that are simpler to understand, program, and debug.
Algorithms	An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms can be translated into programs, or code, to provide instructions for computing devices. Algorithms are central to programming.
Programming	Programming is the process of designing and developing code to perform a specific task. It includes the transformation of an algorithm into a specific language that a computer can read and execute, testing code under controlled conditions to ensure its accuracy, debugging the code to resolve errors, and producing documentation both for end users to understand how to use the program and for other developers to assist in following the logic within the program.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

MODELING AND SIMULATION

K-2	3-5	6-8	9-12
K-2.CT.1 **Recognize and extend a pattern that exists in a natural or designed model.**	3-5.CT.1 **Develop a model of a system that shows changes in output when there are changes in inputs.**	6-8.CT.1 **Compare the results of alternative models or simulations to determine and evaluate how the input data and assumptions change the results.**	9-12.CT.1 **Create a simple digital model that makes predictions of outcomes.**
Clarifying Statement: The emphasis is on learning about observed patterns in an existing model and then using that information to extend parts of the model.	Clarifying Statement: The emphasis is on understanding, at a conceptual level, that models or simulations can be created to respond to deliberate changes in inputs.	Clarifying Statement: The focus is on understanding that models or simulations are limited by the data that they use, rather than understanding specifically how they use that data.	Clarifying Statement: Students will use data to build alternative numerical models that can best represent a data set.
For Example: Students could start with a shape made of four tangram manipulatives and extend the pattern to create a continuous mosaic using the combined shape.	For Example: Students could use the movement of a rope to simulate a sound wave and then explain what happens (in terms of pitch) if they slow down (lower pitch) or speed up (higher pitch) the oscillations modeled by the simulation of sound waves using the rope.	For Example: Students could compare the accuracy of weather models based on research of the inputs.	For Example: Students collect data and use graphing software to create a linear graph, logarithmic graph, and polynomial graph to determine which best addresses the required output.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

Data Analysis and Visualization

K-2	3-5	6-8	9-12
	3-5.CT.2 **Collect digital data related to a real-life question or need.**	6-8.CT.2 **Collect and use digital data in a computational artifact.**	9-12.CT.2 **Collect data from multiple sources for use in a computational artifact.**
	Clarifying Statement: Students should consider using digital tools to collect and organize multiple data points.	Clarifying Statement: Emphasis is on designing and following collection protocols. Data sources include, but are not limited to sensors, surveys, and polls.	Clarifying Statement: Emphasis is on designing and following collection protocols. Data sources include, but are not limited to sensors, web or database scrapers, and human input.
	For Example: Students could create a classroom poll or survey using digital tools and report the results to the class.	For Example: Students could collect temperature data with a sensor and distribute a digital form to community members for a community planning project in which they make recommendations about recreational needs in different types of weather.	For Example: Students could gather and analyze data on the mood and tone of different music genres using a variety of different tools. Students could use a web scraper or API to count the frequency of specific words in the song lyrics, a sound sensor to measure pitch, or a digital survey to capture people's moods after listening to each song.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

Data Analysis and Visualization

K-2	3-5	6-8	9-12
K-2.CT.3 **Present the same data in multiple visual formats in order to tell a story about the data.**	3-5.CT.3 **Visualize a data set in order to highlight relationships and persuade an audience.**	6-8.CT.3 **Refine and visualize a data set in order to persuade an audience.**	9-12.CT.3 **Refine and visualize a large data set using an appropriate tool in order to persuade an audience.**
Clarifying Statement: The emphasis is on using the visual representation to make the data meaningful. Options for presenting data visually include tables, graphs, and charts.	Clarifying Statement: The emphasis is on identifying and organizing relevant data to emphasize particular parts of the data in support of a claim.	Clarifying Statement: Refining includes, but is not limited to, identifying relevant subsets of a data set, deleting unneeded data, and sorting and organizing data to highlight trends.	Clarifying Statement: Large data sets require use of a software tool or app to cross-reference, analyze, refine, and visualize subsets of the data.
For Example: Students could collect temperature data over a week then use it to create a data table and line graph. They could then use the graph to communicate what the weather was like that week. Alternatively, students could count and chart the number of pieces of each color of candy in a bag of candy, such as Skittles or M&Ms.	For Example: Students could use a spreadsheet program to create a data table and graph of student interests and hobbies in their class and sort them by category. Alternatively, students could sort a data set of sports teams by wins, points scored, or points allowed.	For Example: Students could access government data sets for science (tide, hurricane data, sunrise/sunset) and sort and analyze the data to get specific information to support a claim. Students can use a database program to create a pivot table to summarize multidimensional player stats from their favorite sport in order to tell a story or support a claim about a player's career.	For Example: Students could build a custom filter for a mapping application that overlays weather data, then cross-reference weather with traffic patterns to find connections. Students could combine a data set on average household income and a data set on health by zip code in order to cross-reference instances of asthma and conditions related to the environment in order to persuade an audience to take action on environmental social justice issues.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

ABSTRACTION AND DECOMPOSITION

K-2	3-5	6-8	9-12
K-2.CT.4 **Break down a larger problem into multiple smaller problems (decomposition).**	**3-5.CT.4** **Decompose a problem or task into smaller and smaller parts in order to identify where there is repetition.**	**6-8.CT.4** **Decompose a program into distinct parts in order to write functions to reduce repetition and increase readability.**	**9-12.CT.4** **Decompose a program into parts in order to understand how the program should be organized and written.**
Clarifying Statement: Students should start with a completed task and break it down into steps.	Clarifying Statement: Students should identify smaller steps that are used to solve a larger problem and will break down those steps until each is manageable. Students should notice where steps are repeated.	Clarifying Statement: Students should identify where there is repetition in code or potential to reuse a segment of code in order to develop that segment into a function or procedure.	Clarifying Statement: Students should think about the program they want to develop and break it down into sub-systems based on functionality. They should then break the subsystems into programmable parts, using functions or procedures when necessary.
For Example: Students could think about the steps needed to get ready to go home from school.	For Example: Students could plan a classroom party by separating the task (party) into subtasks such as food, activities, and prizes. The subtasks could then be broken down into further into steps like determining which activities could be present and planning what order to do each activity.	For Example: Give students a program that draws a number of boxes by having instructions to draw all lines. The students can decompose the program by writing a single function to draw a box (abstracting it out) and using it or calling it the same number of times.	For Example: Students who want to create an app that solves a community problem might first break down the project as: front-end, back-end, and data/API. They could then take one subsystem at a time and break it down further by programmable features (i.e. The front-end might need a form, a button, a menu, and a list of links.)

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

ABSTRACTION AND DECOMPOSITION

K-2	3-5	6-8	9-12
K-2.CT.5 **Identify multiple layers that make a common object work**	**3-5.CT.5** **Identify multiple layers of abstraction and perform actions on/with each layer**	**6-8.CT.5** **Identify multiple layers of abstraction and create generalizations that can be placed into one or more abstraction(s)**	**9-12.CT.5** **Create or remix one or more abstraction(s) utilizing multiple existing abstractions**
Clarifying Statement: Students should start with a physical object that contains multiple dependent working parts and then discuss how each part works, as well as how it contributes to the overall functioning of the object itself.	Clarifying Statement: Students should identify each part of a physical object, explain how each part works, and describe how each part contributes to the overall functioning of the object. Students should use parts individually and then together as a whole to accomplish a goal.	Clarifying Statement: Students should identify patterns within the parts of an object so that they can describe common behaviors. Subsequently, students should describe those common behaviors such that they can repeatedly be applied to the functioning of multiple parts within the object.	Clarifying Statement: Students should demonstrate an understanding of the concept of abstraction such that they are able to create or modify one working element of an object, resulting in a predictable outcome. Students should notice that patterns can be made general for reuse.
For Example: An analog wall clock is an abstraction that people use to tell time. Looking behind the clock face, we can see how time is represented by the gears, motors, and battery.	For Example: A calendar or assignment book uses abstraction to represent a year, month, or week in terms of days and hours, helping us to organize our time.	For Example: Students can combine variables that hold similar, related data to make an array. Alternatively, Students can generalize many similar functions to one function such as generalizing individual functions that draw a square, draw a triangle, and draw an octagon to a single function that draws a polygon based on an input for the number of sides.	For Example: Show how popular songs are abstractions of multiple parts of a song (intro, verse, chorus), each part of the song is itself an abstraction of the instruments in that section (drums, piano, bass, vocals, etc.), and that each instrument is an abstraction of the music in that section (notes, rhythm, meter, tempo). Students use this understanding to create remixes or original compositions based on different levels of abstraction using a medium of their choice.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

ALGORITHMS

K-2	3-5	6-8	9-12
K-2.CT.6 **Follow an algorithm to complete a task and create simple algorithms.**	**3-5.CT.6** **Create and test an algorithm to complete a task.**	**6-8.CT.6** **Design an algorithm and use it within a program.**	**9-12.CT.6** **Demonstrate how at least two classic algorithms work.**
Clarifying Statement: The task can be a familiar, daily activity or more abstract. Algorithms at this stage may be short, containing at least three steps, and focus on sequencing.	Clarifying Statement: Student algorithms can be longer than in K-2 and reflect a task with a specific result that can be checked. Students will revise their algorithms based on the test results.	Clarifying Statement: Student algorithms may be approximations. Algorithms can be represented in a range of formats, including flowcharts, pseudocode, or written steps. Planning the output of a program, such as with a storyboard or wireframe, is not sufficient on its own.	Clarifying Statement: Students can meet this standard through unplugged modeling or through programming. Students should understand that classic algorithms are solved problems that can be reused. Students are not expected to memorize algorithms.
For Example: A teacher might lead students in following an algorithm that tells the class how to line up for recess. Students could create an algorithm on how to build a simple structure with manipulatives (with blocks, cups, etc.) or how to complete a simple classroom task. .	For Example: Students could develop an algorithm for creating a pixel art image and follow the algorithm to test if it produces the intended image. Alternatively, students could create an algorithm for moving a robot or other character through a maze and test the algorithm by programming the robot or other character and watching to see if it follows the intended route.	For Example: Students using block code could arrange blank cut-outs of the blocks in order, then write a pseudocode description on each paper block. Alternatively, students could draw a flowchart of how their program will run from start to finish.	For Example: Students could use merge sort and bubble sort to order a set of playing cards. Alternatively, students could remix a simple program that uses insertion sort to order a class generated data set.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

Algorithms

K-2	3-5	6-8	9-12
K-2.CT.7 **Create and compare two or more algorithms for the same task.**	**3-5.CT.7** **Compare two or more algorithms and select the most appropriate one for a task.**	**6-8.CT.7** **Compare and refine algorithms for a specific task.**	**9-12.CT.7** **Analyze trade-offs related to two or more algorithms for completing the same task.**
Clarifying Statement: The task can be a familiar, daily activity or more abstract. The focus is on finding more than one way to reach the same goal.	Clarifying Statement: Tasks can be unplugged or related to a computer program.	Clarifying Statement: The steps of an algorithm can be implicit. Tasks should be culturally relevant and familiar to students.	Clarifying Statement: The focus of this standard is a high-level understanding that algorithms involve trade-offs, especially related to memory use and speed. Students are not expected to get into specifics about the trade-offs.
For Example: Students could plan two routes for a robot or other character to reach the same location. Alternatively, students could write “How To” guides for the same task. Students might showcase their work, explaining the steps.	For Example: Students could compare algorithms for making a culturally relevant food item (i.e. pb&j, doner, bahn mi, etc.). Students could then choose an algorithm and explain the reason for their choice. Possible reasons students may provide for their choices might include the detail, the fewest steps, or an algorithm that describes a process most similar to how they make the food item at home.	For Example: Students could compare routes suggested by a mapping app and refine the route based on knowledge of the area near their school or home. Alternatively, students could write an algorithm to draw a geometric shape and refine the algorithm by creating new versions of it until it has no unnecessarily repeating code.	For Example: Students could be asked to look for a specific value in a sorted data set using a sequential search and then a binary search. They can count the number of comparisons it takes to find the value. Alternatively, students could model sorting algorithms with books on a bookshelf and contrast different methods in terms of shelf space and the time spent.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

Algorithms

K-2	3-5	6-8	9-12
K-2.CT.8 **Modify an algorithm for personal expression or to solve a problem.**	**3-5.CT.8** **Remix an artifact for personal expression or to solve a problem.**	**6-8.CT.8** **Modify, remix, or incorporate portions of an existing program into one's own work.**	**9-12.CT.8** **Identify a relevant module, library, or API and use it in a computer program to add a feature or functionality.**
Clarifying Statement: The emphasis is on making changes to an existing solution. Modifications could involve starting with a base algorithm and moving, taking away, or adding steps to make it their own.	Clarifying Statement: The emphasis is on remixing as a form of abstraction. Remixing could involve remixing an algorithm or program in a new context. Students should give attribution when remixing.	Clarifying Statement: The emphasis is on abstraction. This could include using portions of code from an existing program and modifying it to meet the needs of a new program. Students should give attribution when remixing.	Clarifying Statement: To meet this standard, students can use a module, library, or API, but do not need to use all three. The emphasis is on understanding that these are ways of accessing pre-built features or application data.
For Example: A teacher could provide students with an algorithm representing a process of getting ready for school in the morning. Student then might revise the algorithm to reflect their own morning routines.	For Example: Students could combine pictures and text into their own meme. Alternatively, a teacher might give all the students a chorus and have students incorporate or remix the chorus to create their own song.	For Example: A teacher might provide code that counts the score in one game and have students modify that code for another program in order to keep track of the number of times an image is clicked.	For Example: A student could import a library into their program that has built-in functionality for analyzing large data sets. Alternatively, a student could use a social media API to create a program that determines the most common geographical areas for particular hashtags.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

Programming

K-2	3-5	6-8	9-12
K-2.CT.9 **Categorize and label data in multiple ways.**	**3-5.CT.9** **Work with variables in an algorithm or program.**	**6-8.CT.9** **Utilize variables to store and modify data when designing or remixing a program.**	**9-12.CT.9** **Design or remix a program that utilizes data structures to store and modify a set of related data.**
Clarifying Statement: At this level, the emphasis is on exploring categories and recognizing patterns that different types of data fit into and using a label as a stand-in for data.	Clarifying Statement: Unplugged options would involve modelling the use of variables. Programming options would involve declaring and initializing variables with a constant value.	Clarifying Statement: This would involve programming or modelling the use of variables with both constant and mutable values.	Clarifying Statement: Data structures include, but are not limited to, arrays and trees. Data structures are how data types in the program are related.
For Example: Given a set of animals, the teacher might have students come up with different possible labels for groups of like animals (i.e. birds, baby animals, plant-eating animals, animals that live in the Arctic).	For Example: Students could write out an algorithm for a repeating rhyme or song. They can use a variable to indicate where a word is changed each time the rhyme or song is repeated, as in “animal” in “Old MacDonald had a Farm” or “name” in “Happy Birthday.” Alternatively, students could use variables to set the drawing pen color in a program.	For Example: Students could create a fill-in-the-blank story that stores user input in different variables and displays the completed story back to the user. Alternatively, students could program a game that uses a score variable to store the users points while playing the game.	For Example: Students could create a website that includes a form field that stores user input and adds it to a dictionary with key/value pairs. Alternatively, students could create a unique class to group all the different enemy objects together in a video game.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

PROGRAMMING

K-2	3-5	6-8	9-12
K-2.CT.10 **Develop an algorithm that uses repetition structures for creative expression or to solve a problem.**	**3-5.CT.10** **Develop an algorithm or program that uses both repetition structures (e.g. loops) and conditionals for creative expression or to solve a problem.**	**6-8.CT.10** **Develop or remix a program that effectively combines one or more control structures for creative expression or to solve a problem.**	**9-12.CT.10** **Develop a program that effectively uses control structures in order to create a computer program for practical intent, personal expression, or to address a societal issue.**
Clarifying Statement: Repeating a set of instructions is common in algorithms and programs.	Clarifying Statement: In a programming context, a single program or unplugged model would include a "for loop" or "while loop" and single level "if statement." They should be in the same algorithm or program, but do not need to work together.	Clarifying Statement: A single program should combine control structures in such a way that they work together to achieve an outcome that they could not achieve using only one.	Clarifying Statement: Different types of loops and conditionals should be used appropriately in order to achieve the desired output of a program created with a specific purpose in mind.
For Example: Students could choreograph a dance using an algorithm. They would indicate that dance steps are repeated.	For Example: Students could guide a paper mouse through a maze to find cheese by developing a set of rules for the "mouse" to follow. Rules could include the following: move forward one space and repeat until the mouse hits a wall, and if there is a wall, turn left then move forward. Alternatively, students could program a math quiz that uses conditionals to check the user's answers and display a response. The students could use a loop to make a sprite dance when the user completes the quiz.	For Example: Students could use a nested loop to draw a grid. Alternatively, students could use a loop and compound conditional to print *all multiples* of three between one and one hundred.	For Example: Students could program a choose-your-own-adventure game that uses multiple choice options and probability to determine outcomes. Alternatively, students could program a game that utilizes multiple control structures within a game loop.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Computational Thinking

PROGRAMMING

K-2	3-5	6-8	9-12
K-2.CT.11 **Explain the steps of an algorithm for the purposes of debugging.**	3-5.CT.11 **Explain each step of an algorithm that includes loops and conditionals for the purposes of to find and correct errors (debugging).**	6-8.CT.11 **Read and interpret code to predict the outcome of various control structures for the purposes of debugging drawing on a range of debugging strategies.**	9-12.CT.11 **Systematically test and refine programs using a range of test cases, based on anticipating common errors and user behavior.**
Clarifying Statement: This involves evaluating the steps in an algorithm to determine if they have the desired outcome.	Clarifying Statement: Debugging frequently involves stepping or tracing through a program as if you were the computer to reveal errors	Clarifying Statement: Programs can be debugged in numerous ways, including tracing and trying varying inputs. Perseverance is important in finding errors.	Clarifying Statement: The emphasis is on perseverance and the ability to use different test cases on their programs and identify what issues are being tested in each case.
For Example: Students could write, draw, or speak each step of an algorithm for brushing their teeth.	For Example: Students could predict how a sprite will behave in a certain condition. Alternatively, students could consider code snippets with bugs and collaborate with peers to find the errors by reading and discussing the code.	For Example: Students could trace through a program using a variety of inputs to determine the result.	For Example: Students could test the boundaries of input values and the outcome of each branch in a conditional statement.

Computational Thinking

Programming

K-2	3-5	6-8	9-12
K-2.CT.12 **Use a planning process to outline the steps taken to solve a problem or complete a task.**	**3-5.CT.12** **Describe the steps taken and choices made to design and develop a solution using an iterative process.**	**6-8.CT.12** **Document the iterative design process of developing a computational artifact that incorporates user feedback and preferences.**	**9-12.CT.12** **Collaboratively design and develop a program or computational artifact for a specific audience and create documentation outlining implementation features to inform collaborators and users.**
Clarifying Statement: The planning process can be performed using writing, drawing, or speaking.	Clarifying Statement: An iterative design process involves defining the problem or goal, developing a solution or prototype, testing the solution or prototype, and repeating the process until the problem is solved or desired result is achieved. Describing can include talking or writing about the design and development choices for a computational artifact.	Clarifying Statement: At this level, the emphasis is on using the iterative design process to create a solution or prototype with the end user in mind and to document the steps taken by the student to gather and incorporate information about the user into the computational artifact.	Clarifying Statement: The focus is on the collaborative aspect of software development, as well as the importance of documenting the development process such that the reasons behind various development decisions can be understood by other software developers.
For Example: A common writing graphic organizer (e.g. story map or storyboard) can be used as a planning document with drawings that illustrate a story about walking from a classroom to the cafeteria that includes a beginning, middle, and end.	For Example: Starting with a specific issue or topic (e.g. animal extinction, bullying, hunger), students use the iterative design process to explore the issue or topic and then create and deliver a presentation to the class describing the different steps that were taken to arrive at a solution.	For Example: Conducting 'empathy interviews' (as part of the design thinking process), students can discover a particular problem or issue a person wants solved. Then, using this information, students can design a program/'app' that is meant to solve the identified problem in a meaningful way.	For Example: Using a web-based version control platform to share and comment on a program/app, students can engage in collaborative practices common among software developers. Additionally, writing in-line comments within one or more source code file(s) allows students to communicate how a particular part of a program is intended to function.

Networks and Systems Design

Computing devices typically do not operate in isolation. Networks connect computing devices to share data and resources and are an increasingly integral part of computing. Networks and communication systems provide greater connectivity in the computing world by providing fast, secure communication, and facilitating innovation.

Individuals interact with data using a variety of input and output devices that are part of a more complex computing system. The hardware and software that make up a computing system process data in digital form. A basic understanding of hardware and software is useful when troubleshooting a computing system that does not work as intended.

The Networks and Systems Design standards aim to prepare students to understand the basic functioning of the computing systems and networks that are used as fundamental tools in our personal and professional lives.

Hardware & Software	A computing system is composed of hardware, software, and the individuals who use them. Hardware refers to the physical components that make up a computing device. Software refers to the program instructions that operate on such hardware.
Networks & The Internet	Networks are formed by connecting individual devices in a variety of ways. Data is stored on one or more devices in a network and transferred between devices using a set of protocols or rules. The internet is an example of a global network that transmits data between many devices around the world.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Networks and Systems Design

HARDWARE & SOFTWARE

K-2	3-5	6-8	9-12
K-2.NSD.1 **Describe different ways that humans can interact with computers.**	3-5.NSD.1 **Propose improvements to the design of a computing technology based on an analysis of user interactions with that technology.**	6-8.NSD.1 **Design a user interface for a computing technology that takes into account usability, accessibility, and desirability.**	9-12.NSD.1 **Design a solution to a problem that utilizes embedded systems.**
Clarifying Statement: The emphasis is on understanding that humans and computers interact through inputs and outputs. This includes identifying the components of a computer system that help people input information and the parts that produce output.	Clarifying Statement: The emphasis is on thinking about the purpose of the computing technology and how the user interface could be optimized for that purpose.	Clarifying Statement: The emphasis is on designing (but not necessarily creating) a user interface. Designs could include things like written descriptions, drawings, and/or 3D prototypes.	Clarifying Statement: The emphasis is on designing (but not necessarily creating) solutions with embedded systems. Systems can be biological, mechanical, social, or some other type of system. Designs could include written descriptions, drawings, and/or 3D prototypes.
For Example: Students could label a diagram of a computing system with the words input and output. Alternatively, students could sort images of computer components into input and output columns on a t-chart.	For Example: Students could make recommendations on how to improve a tool, device, or app based on their experiences or those of their classmates.	For Example: Students could design a game controller that is accessible for a person with limited hand and arm movement. Students could design apps that encourage healthy living and take into account factors like *motivation to use* the app and ease of use.	For Example: Students might design medical devices that can be embedded inside a person to cure a specific illness, regulate a specific function of the body, or give enhanced ability. Students might propose embedded systems that address public health and safety such as coming up with solutions that use embedded systems in a car to address car accidents, texting while driving, pets overheating when left alone in a car, etc.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Networks and Systems Design

Hardware & Software

K-2	3-5	6-8	9-12
K-2.NSD.2 **Explain the function of common hardware and software components in computing systems, using descriptive/precise language.**	**3-5.NSD.2** **Model how computer hardware and software work together as a system to accomplish tasks.**	**6-8.NSD.2** **Design a project that combines hardware and software components to collect and use data to perform a function.**	**9-12.NSD.2** **Explain the levels of interaction existing between the application software, system software, and hardware of a computing system.**
Clarifying Statement: Hardware includes, but is not limited to, desktop computers, laptop computers, tablet devices, monitors, keyboards, mice, and printers. Software includes, but is not limited to, apps, web browsers, and operating systems.	Clarifying Statement: At this stage, a model should only include the basic elements of a computer system, including input, output, processor, and storage.	Clarifying Statement: The focus is on designing (but not necessarily creating) a system that involves collecting and exchanging data including input, output, storage, and processing.	Clarifying Statement: At this stage, knowledge of specific advanced terms of computer architecture and how specific levels work is not required. Rather the progression, in general terms, from voltage to binary signal to logic gates and so on to the level of human interaction, should be explored.
For Example: Students could label images of components and match components with their descriptions.	For Example: Students can draw the computing system, program an animation of how the computer system works, or act it out in some way.	For Example: Students could design an app for finding free filtered water stations in the area that would use GPS, magnetometer, and touch screen sensors as well as the phone's wifi and a map API.	For Example: Students could create a diagram representing the levels of interaction involved in text editing. They would show that software interacts with the operating system to receive input from the keyboard, convert the input to bits for storage, and interpret the bits as readable text to display on the monitor.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Networks and Systems Design

Hardware & Software

K-2	3-5	6-8	9-12
K-2.NSD.3 **Describe basic hardware and software problems using descriptive/precise language.**	**3-5.NSD.3** **Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.**	**6-8.NSD.3** **Identify and fix problems with computing devices and their components using a systematic troubleshooting method or guide.**	**9-12.NSD.3** **Develop and communicate multi-step troubleshooting strategies others can use to identify and fix problems with computing devices and their components.**
Clarifying Statement: At this stage, the focus is on communicating a problem with appropriate terminology, although students do not need to understand the causes.	Clarifying Statement: Examples of common troubleshooting strategies include: rebooting device, checking for power, checking network availability, closing and reopening an application, making sure speakers are turned on or headphones are plugged in, making sure that the caps lock key is not on, try using a different browser, and checking your settings within an application in an attempt to solve problems.	Clarifying Statement: The focus is on using a structured process, such as a checklist or flowchart, to troubleshoot problems with computing systems and to ensure that potential solutions are not overlooked.	Clarifying Statement: Some examples of multi-step troubleshooting problems include resolving connectivity problems, adjusting system configurations and settings, ensuring hardware and software compatibility, and transferring data from one device to another.
For Example: Students might notify a teacher when an application or device is not working as expected. Rather than saying, "It doesn't work," A student might describe things like, "The device will not turn on," or "The sound doesn't work."	For Example: A teacher might lead students in creating a classroom checklist for basic problems, such as the device not responding, no power, no network connection, application crashing, no sound, or password entry not working.	For Example: Students could follow a troubleshooting flowchart that guides them through a process of checking connections and settings, changing software to see if hardware will work, and swapping in working components.	For Example: Students could create step by step instructions for a help desk employee. Alternatively, students could create a troubleshooting flowchart for anyone using a school device.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Networks and Systems Design

Networks and the Internet

K-2	3-5	6-8	9-12
K-2.NSD.4 **Describe how protocols are used to facilitate communication.**	3-5.NSD.4 **Model how data is structured to transmit through a network.**	6-8.NSD.4 **Design a protocol for transmitting data through a multi-point network.**	9-12.NSD.4 **Describe the design characteristics that allow data and information to be moved, stored and referenced over the Internet.**
Clarifying Statement: The focus is on identifying rules and conventions that allow for communication and data sharing.	Clarifying Statement: The focus is on understanding that data is broken down into smaller pieces and labeled to travel through a network and reassembled.	Clarifying Statement: The focus is on understanding how protocols enable communication and what additional data is necessary for transmission. Knowledge of the details of how specific protocols work is not expected.	Clarifying Statement: The focus is on understanding the design decisions that direct the coordination among systems composing the Internet that allow for scalability and reliability. Discussions should consider historical, cultural, and economic decisions related to the development of the Internet.
For Example: Students could explain how they would communicate an idea using the appropriate gestures or words to someone who doesn't speak their language.	For Example: The teacher could run a series of live simulations in which students act out the flow of information through servers, routers, and other devices to transmit a message. Alternatively, a teacher might have students cut up a map of the United States, then place the states in envelopes and transmit the "packets" through a physical network of students. At the destination, the packets could then be reassembled back into a map of the United States.	For Example: Students could devise a plan to represent a long text-based message as chunks of data and how it would be reassembled at the destination. An unplugged example would include the use of zip codes and barcodes for a letter to travel through the post office system.	For Example: Students could explain how hierarchy in the DNS supports scalability and reliability. Alternatively, students could create a computational artifact that explains the path of data transmission from their device to a website hosted on another continent and back using the network (including but not limited to servers, routers, etc.).

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Networks and Systems Design

NETWORKS AND THE INTERNET

K-2	3-5	6-8	9-12
K-2.NSD.5 **Explain how data travels through a network.**	3-5.NSD.5 **Describe that data can be stored locally or remotely in a network.**	6-8.NSD.5 **Summarize how remote data is stored and accessed in a network.**	9-12.NSD.5 **Describe how emerging technologies are impacting networks and how they are used.**
Clarifying Statement: The focus is on explaining that data can only be shared between computing devices or individuals when they are connected to the same network. Additionally, there should be an emphasis on understanding that there are multiple ways to connect to a network.	Clarifying Statement: The focus is on describing that data must be stored on a physical device. Access to remotely stored data is restricted by the networks, and to access non-local data a connection to the network is required.	Clarifying Statement: The focus is on explaining where the data associated with different apps, devices, and embedded systems is stored, how the data is synchronized, and how to connect to it.	Clarifying Statement: The focus is on discussing how specific emerging technologies impact networks in terms of scale, access, reliability, and security, and user behavior.
For Example: Students might explain various ways they can send information to someone else, such as car, letter, email, phone call, or video chat if they both have access to that shared network.	For Example: Students could explain the difference between owning a book (local copy) and borrowing it from a library (remote copy), or what information they might no longer be able to access if the internet went down.	For Example: Students could create a diagram that illustrates the use of remote storage in cloud computing, a school's data server, or distributed media. Students could discuss how local copies of data are synced with data from the remote server.	For Example: Students could discuss how the Internet of Things has impacted the privacy and security of networks. Alternatively, students might discuss how cloud computing affects the scale of networks and access to shared resources.

Cybersecurity

In a digital world, all individuals have a responsibility to protect data and the computing resources they access. Cybersecurity encompasses the physical, digital, and behavioral actions that can be taken to increase this security. These measures are meant to ensure the confidentiality and integrity of data and computing resources, as well as ensure that they are accessible to the users who are supposed to have access to them. Digital security includes understanding and identifying risks, implementing appropriate safeguards, and being prepared to respond to potential attacks.

The Cybersecurity standards prepare students to understand why data and computing resources need to be protected, who might access them, and why they might do so whether intentionally malicious or not. It is important that students know how to employ basic safeguards to protect data and computing resources and how to appropriately respond if a breach occurs.

Risks	Risk is a combination of a vulnerability, the likelihood that the vulnerability will be exploited, and the severity of consequences if the vulnerability is exploited. It is important to understand why data and resources need to be protected and how they might be compromised so the correct safeguards can be put into place.
Safeguards	Programmers and individuals must know how to protect their data and computing resources with common safety measures. When combined, various physical, digital, and behavioral precautions can create a level of digital security.
Response	When a security breach occurs, individuals must decide what actions to take. This takes into account what type of breach occurred and how to improve security moving forward.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Cybersecurity

RISKS

K-2	3-5	6-8	9-12
K-2.CY.1 **Compare and contrast information that should be kept private with information that might be made public.**	3-5.CY.1 **Explain why different types of information might need to be protected.**	6-8.CY.1 **Determine the types of personal information and digital resources that an individual may have access to that needs to be protected**	9-12.CY.1 **Determine the types of personal and organizational information and digital resources that an individual may have access to that needs to be protected.**
Clarifying Statement: The emphasis is on discussing the reasons that people would want to keep some types of information private and may want to share some information publicly.	Clarifying Statement: The emphasis is on discussing different reasons that adversaries may want to obtain, compromise, or leverage different types of information. At this stage, students should be focused on general concepts.	Clarifying Statement: The emphasis is on identifying personal information and devices that an individual may have access to and that adversaries may want to obtain or compromise. At this stage, students should focus on specific data and devices that they have access to.	Clarifying Statement: At this stage, the emphasis is on identifying both personal information and organizational information that an individual may have access to and that adversaries may want to compromise or obtain. Additionally, there is a focus on identifying devices and embedded systems that an individual may have access to and that adversaries may want to leverage in some way.
For Example: Students could take strips of paper with information like phone numbers, birthdays, pets names, passwords, etc. Then place the paper strips into the categories “ok to share with everyone,” “ok to share with people you know,” and “keep private” on a shared chart.	For Example: Students could discuss the type of data needed for different adversarial behaviors such as information that can be used for identity theft, cyberbullying, political influence, or ransomware attacks.	For Example: Students could think about their personal information and devices that need to be protected and discuss how adversaries might use the data or computing resources if accessed.	For Example: Students could research events in business, industry, and government involving organizational security breaches and pinpoint the type of data and resources compromised and how it was used.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Cybersecurity

SAFEGUARDS

K-2	3-5	6-8	9-12
K-2.CY.2 **Explain ways that information can be kept secure.**	3-5.CY.2 **Describe common safeguards for protecting personal information.**	6-8.CY.2 **Describe physical, digital, and behavioral safeguards that can be employed in different situations.**	9-12.CY.2 **Describe physical, digital, and behavioral safeguards that can be employed to protect the confidentiality, integrity, and accessibility of information**
Clarifying Statement: The emphasis is on recognizing and avoiding potentially harmful behaviors, such as sharing private information online or walking away from public devices after using them without logging off.	Clarifying Statement: The emphasis is on describing common safeguards such as protecting devices and accounts with strong passwords, keeping software updated, and not sending sensitive information over SMS.	Clarifying Statement: The emphasis is on recommending different types of security measures including physical, digital, and behavioral, for a given situation.	Clarifying Statement: The emphasis is on considering the CIA Triad when recommending safeguards for a specific application or device.
For Example: Students could demonstrate that they know how to log in and out of any devices and accounts used for classroom work or other applications.	For Example: Students could create a guide to everyday digital security safeguards for students in another grade. The guide could teach them how to implement different safeguards in the classroom and at home.	For Example: The teacher might provide different scenarios and students can pick safeguards appropriate to the situation from a list that the class generated together.	For Example: Formulate recommendations for setting up a secure home or small business network.

Cybersecurity

Safeguards

K-2	3-5	6-8	9-12
K-2.CY.3 **Identify the benefits and drawbacks of sharing accounts, app access, or devices.**	3-5.CY.3 **Describe trade-offs between allowing information to be public and keeping information private and secure.**	6-8.CY.3 **Describe trade-offs of implementing specific security safeguards.**	9-12.CY.3 **Explain specific trade-offs when selecting and implementing security recommendations.**
Clarifying Statement: The focus is on explaining how user habits and behaviors should be adjusted based on who shares a device.	Clarifying Statement: The focus is on considering the trade-offs of data sharing in different contexts.	Clarifying Statement: The focus is on thinking about how a specific safeguard impacts the confidentiality, integrity, and access of information. Additionally, there should be a focus on discussing if specific safeguards strengthen one part of the triad, but adversely affect another part.	Clarifying Statement: The focus is on making security recommendations and discussing trade-offs between the degree of confidentiality, the need for data integrity, the availability of information for legitimate use, and assurance that the information provided is genuine.
For Example: Students could discuss who has access to shared accounts and why it might be both helpful and risky. They might consider an account that is shared with family members to stream movies or an educational app that is shared by the entire class.	For Example: Students could list the pros and cons of sharing pictures and information about their activities on social media.	For Example: Students could examine the pros and cons of using different methods of authentication, for example passwords, biometrics, or key-fobs and the trade-offs of using single-factor vs multi-factor authentication.	For Example: Students could analyze high profile cybersecurity breaches from the perspectives of competing audiences, including individuals, corporations, privacy advocates, security experts, and government.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Cybersecurity

SAFEGUARDS

K-2	3-5	6-8	9-12
K-2.CY.4 **Model different ways to transmit information confidentially.**	**3-5.CY.4** **Model simple cryptographic methods.**	**6-8.CY.4** **Describe the limitations of cryptographic methods.**	**9-12.CY.4** **Evaluate applications of cryptographic methods.**
Clarifying Statement: The focus is on thinking of different ways that they might convey confidential information and test them out.	Clarifying Statement: The focus is on using cyphers to encrypt and decrypt messages as a means of safeguarding data.	Clarifying Statement: The focus is on recognizing that cryptography provides a level of security for data, and some types of encryption are weaker than others.	Clarifying Statement: The focus is on analyzing the role that cryptography and data security play in events that have shaped history and impact the future.
For Example: Students could brainstorm different ways to send a secret message.	For Example: Students could use a Caesar Shift or Vigenere Square to encrypt a message for a classmate. the classmate can use the same cypher to decrypt the message.	For Example: Students could do a basic frequency analysis of a message encrypted with a Caesar Shift to determine how easy it would be to break it.	For Example: Students could research the role of Navajo Code Talkers and the Enigma machine during World War II and how it relates to the use of private and public keys. Alternatively, students could do a report on the cryptography used to secure Bitcoin and what general ways it could be improved.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Cybersecurity

Response

K-2	3-5	6-8	9-12
K-2.CY.5 **Identify unusual behaviors of applications and devices that should be reported to a responsible adult.**	**3-5.CY.5** **Explain unusual behaviors of applications and devices.**	**6-8.CY.5** **Describe actions to be taken when an application or device reports a security problem or behaves unexpectedly.**	**9-12.CY.5** **Recommend multiple potential actions to take in response to various types of digital security breaches.**
Clarifying Statement: The emphasis is on recognizing situations in which students should notify a trusted adult when a device or application does not behave as expected. At this stage, they do not need to understand why the behavior is unusual or pinpoint what is unusual about the behavior.	Clarifying Statement: The emphasis is on describing simple forms of unusual behavior in common applications and devices, including unusual data or links.	Clarifying Statement: The emphasis is on explaining appropriate actions for common situations.	Clarifying Statement: The emphasis is on analyzing different types of breaches and planning appropriate actions that might be taken in response.
For Example: Students could explain that they should not click on pop-ups in an app or online unless instructed by an adult.	For Example: Students could review sample email messages and describe features that suggest suspicious behavior.	For Example: Students could explain the value of running malware scans and removal tools on devices as soon as unusual behavior is observed. Alternatively, students could recommend changing passwords immediately after an account is compromised.	For Example: Students could discuss how organizations could respond to data theft involving customer information.

Digital Literacy

Digital literacy is a multifaceted concept that extends beyond skills-based activities and incorporates both cognitive and technical skills. It refers to the ability to leverage computer technology to appropriately access digital information; to create, share, and modify artifacts, and to interact and collaborate with others. Digital literacy includes understanding the benefits and implications of using digital technologies to be successful in our contemporary world.

Digital Use	Computers are a part of everyday life. A variety of digital tools exist to create, revise, and publish digital artifacts, as well as communicate and collaborate with others.
Digital Citizenship	Digital citizenship focuses on empowering learners to use online resources, applications, and spaces to improve communities, make their voice heard, and curate a positive and effective digital footprint. It encourages students to engage respectfully online with people with different beliefs and better determining the validity of online sources of information.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Digital Literacy

Digital Use

K-2	3-5	6-8	9-12
K-2.DL.1 **Identify, locate, and use the main keys on a keyboard.**	**3-5.DL.1** **Type on a keyboard while demonstrating proper keyboarding technique.**	**6-8.DL.1** **Type on a keyboard while demonstrating proper keyboarding technique, with increased speed and accuracy.**	**9-12.DL.1** **Type proficiently on a keyboard.**
Clarifying Statement: In K and 1, students should explore keyboards. In 2nd grade, students should be introduced to keyboarding. The main keys include the alphanumeric, punctuation, enter, shift, backspace, and delete keys, as well as the space bar.	Clarifying Statement: Students should receive direct instruction in keyboarding beginning in 3rd grade. Instruction should focus on form over speed and accuracy.	Clarifying Statement: Students should continue to improve keyboarding skills, with a focus on increasing speed as well as accuracy.	Clarifying Statement: Students should demonstrate proficient keyboarding skills by the end of 12th grade.
For Example: Students use a keyboard to type a narrative written during writing workshop to create a class book of stories.	For Example: Students use a school-selected online keyboarding program to learn the fundamentals of keyboarding.	For Example: Students regularly type on keyboards as they use technology throughout the school day.	For Example: Students are able to type on a keyboard with enough automaticity that they can fluently and fluidly transfer thoughts to computer.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Digital Literacy

Digital Use

K-2	3-5	6-8	9-12
K-2.DL.2 **Communicate and work with others using digital tools to build knowledge and convey ideas.**	**3-5.DL.2** **Communicate and collaborate using digital tools to learn with others.**	**6-8.DL.2** **Communicate and collaborate with others using a variety of digital tools to create and revise a collaborative product.**	**9-12.DL.2** **Communicate and work collaboratively with others using digital tools to support individual learning and contribute to the learning of others.**
Clarifying Statement: The focus should be on teaching students that people use digital tools to share ideas and work together. Communication and collaboration should be under teacher supervision with school-approved audience(s).	Clarifying Statement: Students progress from understanding *that* people use digital tools to communicate and collaborate to *how* they use the tools. Communication and collaboration should be purposeful and, when possible and appropriate, with an authentic audience.	Clarifying Statement: Students connect with others (students, teachers, families, the community, and/or experts) to further their learning for a specific purpose, give and receive feedback, and created a shared product.	Clarifying Statement: Digital tools and methods should include both social and professional (those predominantly used in college and careers). Collaboration should occur in real time and asynchronously, and there should be opportunities for students to both seek and provide feedback on their thoughts and products.
For Example: Students collaboratively build a list of their favorite books, and the teacher posts the list on their class website.	For Example: Students use a school-approved digital tool to type a request to an expert (author, zoologist, museum curator), asking him or her to speak to their classroom; collaboratively generate a list of questions to ask; and connect with the expert over a digital conferencing tool.	For Example: Students communicate through digital conferencing tools with students from other countries about voting rights. Students collaboratively create an original product (report, presentation, podcast) based on the conversations, post to a shared site, and provide feedback to peers on their products.	For Example: Students identify a local issue of interest/concern, collaborate on a solution, collaboratively create a digital product, and give presentations to authentic audiences.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Digital Literacy

Digital Use

K-2	3-5	6-8	9-12
K-2.DL.3 **Conduct basic searches based on keywords provided.**	**3-5.DL.3** **Conduct and refine advanced multi-criteria digital searches to locate content relevant to varied learning goals.**	**6-8.DL.3** **Compare types of search tools, choose a search tool for effectiveness and efficiency, and evaluate the quality of search tools based on returned results.**	*(mastery reached by grade 8)*
Clarifying Statement: The teacher will provide key words to help students conduct searches and sorting of objects, etc..	Clarifying Statement: Focus should be on the quality of results a search generates, and how to improve search results based on the task or purpose by defining multiple search criteria and using filters.	Clarifying Statement: Mastery of this standard implies an understanding of how different search tools work, why different search tools provide different results, and how and why some websites rise to the top of a search.	
For Example: Students use a key word to find appropriate picture(s) related to a search.	For Example: Students search for articles published after 2018 and pictures licensed under the Creative Commons Non-Commercial license to create a presentation on endangered ecosystems.	For Example: Students compare results when they search on multiple engines; conduct a search, clear their cache/cookies and then conduct a *search* again; and conduct a search on a mobile device versus a desktop.	

Digital Literacy

Digital Use

K-2	3-5	6-8	9-12
K-2.DL.4 **Use a variety of digital tools and resources to create simple digital artifacts.**	**3-5.DL.4** **Use a variety of digital tools and resources to create and revise multimedia digital artifacts.**	**6-8.DL.4a** **Select and use digital tools to create, revise, and publish digital artifacts.**	**9-12.DL.4a** **Independently select advanced digital tools and resources to create, revise, and publish complex digital artifacts or collection of artifacts.**
Clarifying Statement: Different digital tools are used for different purposes, such as communicating, collaborating, researching, and creating original content. Students may work collaboratively or independently, under supervision.	Clarifying Statement: Multimedia artifacts include images, audio, video, animation, etc., though not necessarily all formats in one single artifact.	Clarifying Statement: Teachers should designate a school-approved location for students to publish artifacts for an audience to view. Advanced digital tools may refer to the tool itself (i.e. the tool is more advanced) or to utilization of more advanced features on a tool.	Clarifying Statement: Mastery of this standard implies an ability to choose and use the technology tool or resource best suited for a task or purpose.
For Example: By the end of second grade, students will have communicated with a school-approved audience, worked together, researched a topic, and created something, all using different tools. When introducing a new tool, the teacher thought aloud/modeled how to choose the right tool for the task.	For Example: Students create a digital story to demonstrate understanding of a concept, such as the branches of government.	For Example: In collaborative groups, students create anti-cyberbullying commercials and an accompanying infographic for parents on ways they can help kids spot and report cyberbullying.	For Example: For a project that allows students to represent learning in different/multiple ways, students choose the tools to use and write a justification why they were the best choices.

DRAFT NYS K-12 Computer Science and Digital Fluency Standards

Digital Literacy

Digital Use

K-2	3-5	6-8	9-12
		6-8.DL.4b **Transfer knowledge of technology operations in order to explore new technologies.**	**9-12.DL.4b** **Transfer knowledge of technology operations in order to use new and emerging technologies on multiple platforms.**
		Clarifying Statement: New technologies could include different tools for collaboration, creation, etc. that the student has not used before.	Clarifying Statement: New technologies could include different tools for collaboration, creation, etc. that the student has not used before. Platforms could include devices running different operating systems or could be emerging STEAM technologies. Digitally fluent individuals can move between platforms and can use that knowledge when encountering new technology.
		For Example: Students familiar with a desktop presentation software (PowerPoint/Keynote) use an online presentation tool to create a presentation.	For Example: Students choose an emerging technology and use it create a simulation of a principle of physics.

Digital Literacy

Digital Citizenship

K-2	3-5	6-8	9-12
K-2.DL.5 **Provide examples of online information about real people, and identify ways that people put their own information into online spaces.**	3-5.DL.5 **Describe persistence of digital information and explain how actions in online spaces can have consequences in the "real world."**	6-8.DL.5 **Explain the connection between the persistence of data on the Internet, personal online identity, and personal privacy.**	9-12.DL.5 **Actively manage digital presence and footprint to reflect an understanding of the permanence and potential consequences of actions in online spaces.**
Clarifying Statement: Concepts related to personal information sharing/privacy should be introduced.	Clarifying Statement: In order for students to be able to effectively manage their digital identities, it should be understood that online information doesn't "go away," and that information posted online can affect their "real lives," even years in the future.	Clarifying Statement: A focus should be on learning about privacy settings on social media accounts, exploring the concept of a positive online presence/identity, and identifying behaviors and information that could potentially affect them now and when they are ready to transition to college and careers.	Clarifying Statement: Active management implies an understanding of how intentional and unintentional actions can affect a digital presence.
For Example: The teacher leads a discussion about photos found online: how people post them, how they sometimes let other people see them, and sometimes they choose to keep them private, and how if the students are playing a game and the game wants to take their picture, they need to ask a responsible grown-up if it's ok.	For Example: Students use a tool that displays archived versions of websites (such as "Wayback Machine") to research how information is available even if it seems to be deleted.	For Example: Students create guides for an adult (family member, celebrity, fictional character) on how to manage online identity and actions that affect someone's digital footprint.	For Example: a. Students create diagrams / infographics that illustrate the myriad sites that might collect data on an individual, the accumulation of which is a digital footprint. b. Students create an online portfolio showcasing sample work and resume that could be shared with potential employers or college admissions boards.

Digital Literacy

Digital Citizenship

K-2	3-5	6-8	9-12
K-2.DL.6 **Identify actions that promote good digital citizenship, and those that do not.**	3-5.DL.6 **Identify actions in online spaces that could potentially be unsafe; describe cyberbullying and actions to take if cyberbullying is witnessed or experienced.**	6-8.DL.6 **Describe safe, appropriate, positive, and responsible online behavior; identify types of cyberbullying, and identify strategies to combat cyberbullying/harassment.**	9-12.DL.6 **Design and implement strategies that support safety and security of digital information, personal identity, property, and physical and mental health when operating in the digital world.**
Clarifying Statement: Good digital citizens practice safe, responsible, ethical, and positive behavior when they use technology and/or are in online spaces. Just like in "real-life," good citizens know to follow the rules, be safe, and be respectful to one another.	Clarifying Statement: Potentially unsafe actions include – but are not limited to – sharing personal information, clicking on pop-ups/advertisements/phish-bait, allowing access to camera.	Clarifying Statement: Types of cyberbullying include – but are not limited to – harassment, trolling/flaming, excluding, outing, dissing, masquerading, and impersonation.	Clarifying Statement: Strategies that support positive mental health in the digital world include both ways to avoid or handle cyberbullying and ways to interact positively and constructively with others in connected spaces.
For Example: Students hold up red light/green light signs at teacher prompts about actions with technology/in online environments, such as "Share your password," "Go to sites linked from our class webpage," "Write something mean about someone," etc. Actions could be added to a running list on a chart displayed in the classroom.	For Example: Students create PSAs on online safety and cyberbullying to include in district/school newsletters/newspaper or make posters to put up in the middle school.	For Example: Students work in collaborative groups to create action plans to decrease instances of cyberbullying among teens.	For Example: Students create an individual action plan on how they would prevent multiple types of cyberbullying and/or a compromise of their digital identity.

(New York State Education Department, “Computer Science and Digital Fluency Learning Standards”, 2020, p. 14-54)