International Journal of Technology and Design Education (2023) 33:1449–1470
https://doi.org/10.1007/s10798-022-09773-y

Programming in school technology education: the shaping of a new subject content

Peter Vinnervik[1]

Accepted: 1 August 2022 / Published online: 19 August 2022

Abstract
In Sweden, as in many other countries, programming has been added to the curriculum in recent years. This paper explores how and under what conditions programming is transformed into teaching in technology education. Teachers in grade 7–9 were interviewed about what and how they teach about programming in the context of technology education, and what influenced the choices they made. Data was analysed using content analysis. The results suggest that teachers mainly see programming as a medium to explore and understand technological systems and construction work. The results further implies that teachers are uncertain of what programming means in terms of practices and concepts, and how programming knowledge can be assessed. The process of transformation presents teachers with several challenges, related to their own knowledge of programming, quality of teaching materials, and knowing pupils' prior knowledge about programming. In addition, the results indicate that technology teachers work in isolation. In consequence, the intended interdisciplinary collaboration between technology and mathematics teachers on programming is largely absent.

Keywords Programming · Curriculum implementation · Compulsory school · Technology education · Teacher challenges

Introduction

Programming is a novel content in Swedish school mathematics and technology education and was introduced through a curriculum reform formally enacted in 2018. This paper explores how, and under what conditions teachers transform programming into teaching content in Swedish technology education. Teachers' perceptions and experiences of what they should teach, why, and how are important to study, and can provide valuable insight into how new policies influence teachers' work, and how new policies are implemented (Cohen & Ball, 1990; Hargreaves, 2005; Sentance & Csizmadia, 2017; Yadav et al., 2016). Ten grade 7–9 (pupils age 13–15) teachers were interviewed about their choices of what

---

✉ Peter Vinnervik
peter.vinnervik@umu.se
https://www.umu.se/en/staff/peter-vinnervik

[1] Department of Science and Mathematics Education, Umeå University, 90187 Umea, Sweden

to teach about programming and technology, and with what resources. The paper further explores intrinsic and extrinsic factors that influence these choices. The aim is to contribute to the understanding of what programming in this particular context means, and under what circumstances the new subject content is being shaped. There are significant reasons for why the process of bringing programming into Swedish technology education is important to study.

First, as programming was recently added to the Swedish school curriculum, there is a lack of history and tradition of how the subject matter can be taught (Vinnervik, 2020). Few teachers have studied programming in their teacher training, and teacher education institutions are still in the early stages of implementing programming in mathematics courses (Swedish Higher Education Authority, 2020). The National Agency for Education (hereafter referred to as NAE) have initiated various professional development (hereafter referred to as PD) initiatives for in-service teachers. However, a recent survey showed that the overall impact of these initiatives thus far have been modest and many teachers perceive a lack adequate knowledge (Lärarnas Riksförbund, 2020). In addition, Parding et al. (2018) have shown that many teachers are dissatisfied with the conditions for formal PD provided by their employers. Despite the efforts from the NAE, it is reasonable to conclude that a strong need for PD on programming remains.

Background

International school policy reform trends emphasize the necessity to increase and improve the use of digital technologies in teaching and learning activities and to increase school children's digital competence (Balanskat & Engelhardt, 2015). Common in these global policy initiatives is the addition of elements of computer science, such as programming, to the curriculum. This addition is often justified as a medium to grow computational thinking abilities. School policy and how programming is contextualised and framed in the curricula differ between countries. In some countries, programming is part of discrete subjects with a more elaborate focus on informatics or computing.[1] The Swedish/Nordic strategy is to integrate programming in mathematics, and in a complementary subject such as crafts, science or technology (Bocconi et al., 2018). Even though there can be significant contextual differences between school systems, it is important that these travelling reform ideas are scrutinized, and experiences shared. From a global perspective, much remains to be learned about how programming can be implemented and utilized in primary education (Crick, 2017; Kjällander & Petersen, 2016; Larke, 2019; Luxton-Reilly et al., 2018; Medeiros et al., 2019).

The role of programming in Swedish compulsory school

The Swedish programming reform is part of a national push to increase the *digital competence* of school children. Digital competence is a set of broader cross-curricular learning goals related to digital and social awareness, critical thinking, creativity and problem solving (Skolverket, 2017). The NAE presents programming as a medium, or a 'learning tool'

---

[1] In England, the subject ICT as a discrete subject was replaced with Computing in 2013, see https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study

for achieving the broader cross-curricular learning goals as well as for achieving subject specific goals in mathematics and technology (Skolverket, 2017). In mathematics, pupils are expected to develop 'basic programming skills' to perform and deepen the understanding of mathematics. In technology, programming mainly revolves around understanding the methodology of control and regulation of programmable electronics in the context of construction work. Research (Vinnervik, 2022) shows that the formal written school curriculum reveals few details about programming knowledge in terms of practices, principles and concepts. A lack of specificity in the curriculum emphasizes the importance of teachers having adequate professional knowledge to make informed interpretations of the intended curriculum.

An investigation of the uptake of programming in grade 7–9 mathematics and technology education, conducted 7–10 months after the enactment of the reform, showed that both mathematics and technology teachers need PD about programming and digital learning (The Swedish Schools Inspectorate, 2019). It was noted that for most of the 27 schools evaluated, programming was not covered to its full intent, and in a few cases not at all. Several schools had begun to introduce programming in mathematics, but not yet in technology education. It should, however, be noted that the evaluation was carried out in the early stages of the change process and that reform work generally is a long-term process (Fullan, 2001; Ornstein & Hunkins, 2018).

Regarding technology education, there are two particular circumstances related to the identity and traditions of the subject that may influence how programming is integrated in teaching. First, technology is the newest subject in the Swedish school curriculum, established in 1994. Up until then, technology education had been closely tied to, or mixed up with, the natural sciences subjects since the 1960s (Hallström et al., 2014). Technology education was initially an optional path of study but became mandatory in 1980. Rooted in the needs of the industry, technology has evolved into a subject which accentuates broader interdisciplinary aspects of technological literacy (Swedish National Agency for Education, 2018, pp. 296–297). Second, technology has had limited space in teacher education. For example, in the teacher education program that was in force between 1988 and 2005, technology was covered over the course of five weeks. This was 1/3 or less of the time devoted to the natural sciences. In addition, statistics from the NAE reveal that less than 50% of Swedish compulsory school technology teachers are formally qualified (Skolverket, 2019). Although no detailed statistics over the level of education among in-service teachers is available, it is reasonable to assume that many teachers have a degree from the 1988–2005 teacher program, and thus have 5 weeks of technology studies as a basis.

Teachers' curriculum work

The following section outlines the theoretical underpinnings of the study reported. First, the theoretical context of the study is established from a curriculum theory perspective. Thereafter follows an introduction of a conceptual framework (Finger & Houguet, 2009) used as an analytical tool to identify and organise factors that influence teachers' work.

Within the field of curriculum theory, several scholars (e.g., Billett, 2006; Goodlad, 1979) have proposed models that represent curriculum policy decision making in different strata. The paper draws on a curriculum model differentiated in three levels, or arenas (Linde, 2012):

1. The formulation arena

2. The transformation arena
3. The realisation arena

The *formulation arena* represents the space where school authorities make decisions about intentions and underpinning philosophy of education, thereafter, communicated through formal curriculum documents. Swedish school curricula follow the northern continental curriculum tradition of Didaktik (Wahlström & Sundberg, 2018). In this model, the curriculum outlines the aims, content and learning objectives of education, but provide few details about teaching methods. Instead, there is leeway for teachers to exert their professional agency (Mølstad & Karseth, 2016) and make decisions about content, means and methods based upon the perceived intentions (Linde, 2012). These decisions are made at the *transformation arena*, where a process of interpretation and recontextualization takes place. Here, teachers unpack the curriculum and "transform the content he or she possesses into forms that are pedagogically powerful and yet adaptive to the variations in ability and background presented by the students" (Shulman, 1987, p. 15). The decisions are thereafter operationalised at the *realisation arena*, which represents the intended curriculum put into practice, that is, what is de facto happening in the classroom. This arena encompasses what students experience, interpret, and learns.[2]

Curriculum unpacking is not a straightforward process for teachers and can be seen as a process that oscillates between the formulation and the realisation arena (Linde, 2012). The decisions teachers make during transformation can be understood through three questions; their understanding of *what* and *how* students should learn and *why* the particular content and methods are selected (Priestley, 2019; Shulman, 1986; Wickman et al., 2018). Teachers' work is influenced by their professional knowledge, personal preferences, experiences, and external conditions (e.g., class size, teaching materials, traditions, school leadership, colleagues) which in turn affect how they design teaching and learning activities, regardless of the intended curriculum (Linde, 2012; Lipsky, 2010). Such pre-conditional differences open up for a wide variety in curriculum interpretations, both conscious and unconscious, between teachers, schools, school districts and municipalities (Jarl & Rönnberg, 2019). This paper utilises the questions of what-how-why as a data analysis tool, as further outlined in the Method section.

In curriculum theory research, it is common to study how predetermined factors influence curriculum enactment (Linde, 2012). In this paper, a conceptual framework developed by Finger and Houguet (2009) that classifies factors which impede or enable reform implementation is used as a data analysis tool to capture and organise factors that were recognized as influential to the transformation process. The factors in this framework are separated in *intrinsic* and *extrinsic* challenges. Intrinsic challenges affect the teacher on a personal level and relate to, for example, knowledge of and attitude towards the reform message. Extrinsic challenges are environmental factors that affect teachers' work and are, for example, related to teaching resources, time and assessment. The framework was developed in Queensland, Australia, during implementation of technology as a discrete school subject. Figure 1 presents an overview of the challenges that constitute the framework. For further details, see Finger and Houguet (2009).

---

[2] Other curriculum enactment models use additional levels to differentiate the student perspective, for example, the perceived curriculum (Billett, 2006), or the experienced curriculum (Goodlad, 1979).

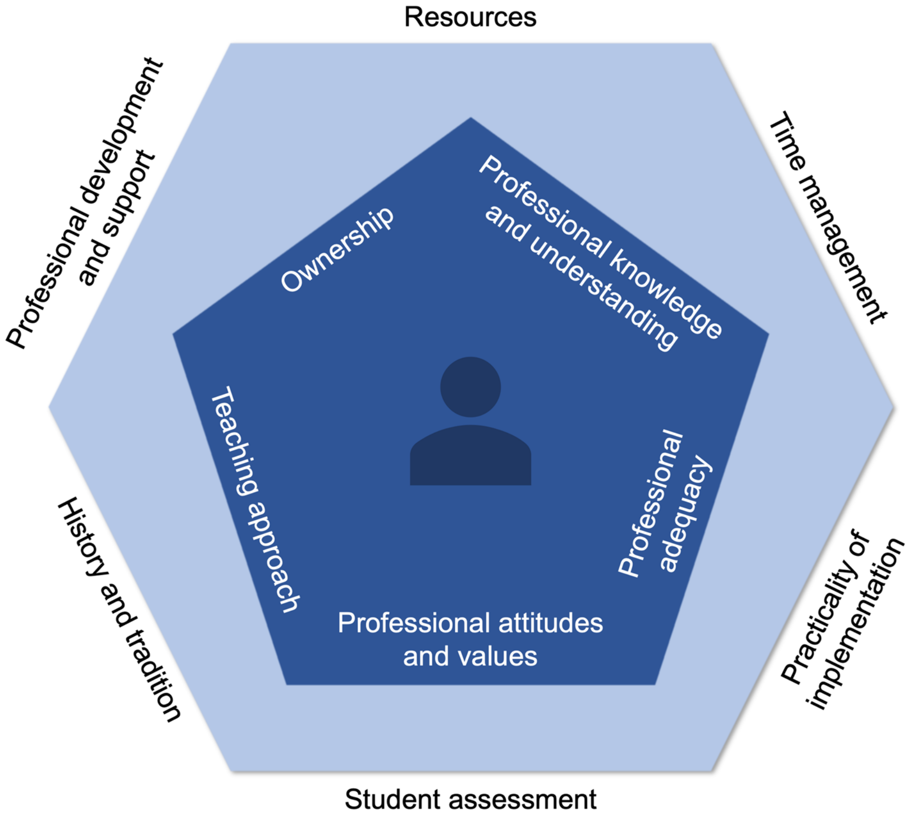

Fig. 1 Author's visual interpretation of the conceptual framework of intrinsic and extrinsic challenges for technology education curriculum implementation (Finger & Houguet, 2009)

Purpose of study

The aim of the study reported is to investigate technology teachers' work in the transformation arena as it affects how programming is contextualised and shaped in the technology classroom. The specific research questions are as follows:

RQ1 How do technology teachers transform programming into technology teaching?
RQ2 What challenges do teachers face in the process of transforming programming into technology teaching?

In a broader perspective, studying the process by which programming in technology education is understood and transformed into teaching can contribute to the understanding of what school children may learn with and about programming.

Table 1 Coding criteria, research question 1: Transformation

Classification category	Coding criteria
What	Knowledge of technology and what programming teachers intend to address in their teaching
How	Organisational aspects of teaching in terms of teaching materials and support
Why	Rationale of choices made about content, learning goals and materials

Method

This section is divided into two parts and outlines the approaches and tools used to answer the two research questions. The first part describes how data were collected while the second part describes the data analysis procedure.

Data collection

Data were collected through semi-structured (Kvale, 1997) online interviews with 10 in-service technology teachers, 27 months into the reform. The teachers were recruited through a combination of convenience and purposeful sampling. Each interview lasted between 42 and 90 min, with an average length of 60 min. In the total sample, 10 different schools are represented. There are small rural schools with a total number of 15 children in grades 7–9, as well as larger schools in regional centres with over 300 children in grades 7–9. The regional centre schools are populated by children from the regional centre itself along with children from less densely populated areas of the same municipality. Represented in the sample are also schools from mid-sized and larger cities. Nine schools are public schools, and one is an independent (private) school.

All participants[3] are certified teachers and have 10 years of teaching experience or more. Three (Alex, Boo, Elliot) have five weeks of studies in technology and technology didactics included in their teacher training. Three (Sam, Kim, and Max) have 20–25 weeks of studies in technology and technology didactics and two (Frankie and Robin) have a rigorous background in technology through degrees in engineering. Two (Taylor and Chris) lack academic credits in technology. Chris has experience from teaching technology at upper secondary school. Three (Alex, Elliot, and Robin) have studied a 5-week introductory university course in Python programming for teachers.

Data analysis

The analysis commenced with reading the interview transcripts. This approach enabled a flexible and holistic overview of the corpus of data. Thereafter the transcripts were analysed using deductive content analysis as described by Elo et al. (2014). In this procedure, the data were reviewed and coded according to predefined categories established from the research questions and the theoretical framework for the study. To answer the first research question of transformation, three sub-questions were used to guide the data analysis

---

[3] Fictive gender neutral names are used to underscore that the participants are real persons.

Table 2 Example of a coded statement, research question 1: Transformation

Statement	Classification category	Colour
You should make a construction which you are able to program	What	🟥
In 9th grade, we are using Arduinos to program electronic components	How	🟨
It fits quite well because in 8th grade, we work with electronics and there is a tradition at my school to work with electronics	Why	🟦

Interviews were conducted in Swedish and illustrative quotes have been translated to English by the article author, with the intended purpose of capturing the essence of what was said, rather than a literal translation

Table 3 Example of a coded statement, research question 2: Challenges

Statement	Classification category	Colour
It is a bit of a hassle. We have Chromebooks which means that I have to use an old laptop cart [to reliably connect to the Arduino controller cards]	Challenge	🟩

process: *what*, *why* and *how* (Priestley, 2019; Shulman, 1986; Wickman et al., 2018). These questions were operationalised as a "theory-based categorisation matrix" (Elo et al., 2014, p. 2) during the transcript analysis. Transcript statements, that is, full sentences or parts of sentences, were coded according to the following criteria (Table 1):

The coding procedure was conducted on printouts of the transcribed interviews. Each category was given a unique colour code and once a statement was found to correspond to a category, the text was marked with the predetermined colour. In some cases, statements were coded with multiple colours. The teachers' narratives are verbose and certain passages in the transcripts were left uncoded. Table 2 provides an example of how statements were coded during the analysis.

The coded statements were thereafter aggregated in two steps. In the first step, the statements for each teacher were collated and organised according to the classification categories (colours). In the next step, each category was reviewed for additional patterns, and statements that conveyed similar meaning were collated in sub themes. For example, statements subordinated to *how,* were first collated into two sub-themes: teaching materials and support. Thereafter, teaching materials were organised into two additional sub-themes: tools and instructional materials. Finally, illustrative findings were selected for presentation in the results section.

To answer the second research question of challenges, a second round of data analysis was conducted which included examining the transcripts for factors that were affecting the teachers' choices. Such factors were identified either by what the teachers specifically said influenced their choices, or by what I, as a researcher, assessed were influential for their decision-making process. The factors were identified and classified by using the conceptual framework for intrinsic and extrinsic challenges (Finger & Houguet, 2009) as a categorisation matrix. The coding procedure was similar to the one used for the first research question. First, each printout was thoroughly reviewed for intrinsic and extrinsic challenges. Statements that revealed information about challenges were jointly coded in a single colour (green) (Table 3).

Next, the green-coded statements were organised into the two main categories of *intrinsic* and *extrinsic* challenges, followed by a breakdown into the specific challenges of each main category. Where applicable, the statements were arranged in sub-themes. For example, the challenge Resources was found to include three themes: monetary issues, teaching materials and IT infrastructure. As a final step, illustrative findings were selected for presentation.

Results

The results are presented in two sections. The first section addresses the transformation of programming (research question 1), and the results are presented in sequential order based on the classification categories of *what*, *how* and *why*. Similarly, the second section addresses challenges that affect the process of transforming programming into technology teaching (research question 2), using the conceptual framework of intrinsic and extrinsic challenges (Finger & Houguet, 2009) to provide structure to the presentation of results. It is implied that the challenges presented need solutions to facilitate the transformation and implementation of programming.

The transformation of programming (RQ1)

What content knowledge is intended to be taught?

The question of *what* intends to illuminate which knowledge of technology and programming the teachers seek to address in their teaching. In terms of specific knowledge about programming principles and concepts, the teachers broadly talked about the importance of structure, accuracy, and logical and sequential thinking. The results suggest that teachers who either have additional formal training or have a certain personal interest were able to speak about aspects of programming knowledge more openly. While doing so, they primarily gave examples of declarative knowledge (McGill & Volet, 1997) and mentioned various programming concepts, for example, step-by-step instructions (sequence of commands, or algorithms), conditional structures (if-else), repetitions and loop structures. Procedural knowledge (McGill & Volet, 1997), that is, the ability to design and successfully implement a computational solution to a problem, was regarded as less important:

> It is about computational thinking, thinking about stepwise instructions, one must first analyse what I want to happen and in what order, to start thinking a little in that direction. Whether they can design the scripts or actual Python code, it's not as relevant (Elliot)

The quote further illustrates the idea that in technology education, understanding the importance of structure and sequence is more important than learning "how to program". Situating programming in the context of technological systems and systems thinking was considered important:

> Most important is really to understand how systems are affected by input and output, systems have sensors that provide feedback to the program which then can respond… They should understand that these systems are totally dependent on me programming them. A computer does exactly what I tell it to do (Alex)

The connection between programming and construction work was mentioned throughout the interviews, for example:

> the core content [section in the technology syllabus] I think points towards the construction of something that you should be able to program … like building a robot… it says that you should make your own constructions (Frankie)

It was emphasized that technology is a multidimensional, well-rounded, and interdisciplinary subject with a structure of knowledge constructed around five core abilities,[4] and that it is important to "put programming and digitalisation in a context", based on these abilities. Especially the societal role of technology emerged as important. It is worth noting that programming as a problem solving activity was rarely mentioned during the interviews. When mentioned, it was stated that "programming is really about solving problems", but that the 'technological problem' is priority:

> It is about own constructions, [and] analysing a problem you must solve […] The problems get gradually more complex. First, you make it [the educational robot] follow a path. Then it should do something. Last time we used it to sort boxes, green and red (Elliot)

How is programming introduced in the technology classroom?

The question of *how* explores organisational aspects of the teachers' work, and specifically relates to the tools and support with which teaching intent is transformed into classroom activity.

The results show that teachers introduce programming in technology education while working with the themes of technological systems and electronics, and BBC Micro:bit micro-controller cards were commonly used. The Micro:bit was appreciated by the teachers for its reliability, cost effectiveness, and extensibility through various accessories. The teachers particularly mentioned sensors, servomotors, and robotics kits.

> I am very grateful for the Micro:bit, you can connect lots of entertaining stuff without adding too much complexity (Elliot)

The teachers also used educational robots, such as LEGO Mindstorms. For example, robots were used to introduce the concept of control and regulation of technological systems, and to address the role of automated systems in society. However, the LEGO robots were perceived to be a better fit for middle grade school (ages 10–12), mainly due to the more structured and controlled environment for construction work that they provide. More complex Arduino micro-controller cards were used, but to a lesser extent. They were believed to provide flexibility beyond what the Micro:bit could offer, albeit at the cost of increased complexity. The Arduino cards were used in constructions that incorporate electronic components attached to breadboards (e.g., diodes, capacitors).

---

[4] The five core abilities of school technology are (Swedish National Agency for Education, 2018, pp. 296–297):

- Identify and analyse technological solutions based on their appropriateness and function.
- Identify problems and needs that can be solved by means of technology, and work out proposals for solutions,
- Use the concepts and expressions of technology.
- Assess the consequences of different technological choices for the individual, society, and the environment, and.
- Analyse the driving forces of technological development and how technology has changed over time.

Computational concepts and principles of control and regulation were also introduced without any hardware involved. In those instances, teachers used programming environments such as Scratch and Blockly Games.[5] Scratch was used to explore principles of simulated control and regulation in more open-ended projects, often games, but also for virtual technological projects, such as designing and programming a virtual robot vacuum cleaner. Blockly Games was believed to provide a novice-friendly and well-structured, linear introduction of some core principles of coding, for example the importance of accuracy and sequence:

> They get to see how to stack things [commands] in a sequence. You have to think things through, you cannot skip any step to get the program where you want (Chris)

The present results suggest that teachers prefer block-based environments and find them sufficient to meet the intentions they perceive are conveyed in the syllabus. Textual programming environments, such as the in-browser integrated development environments (IDE) Replit and the Arduino web editor, were also used. None of these environments were the teachers' first choice, but they were used since pupils were not allowed to install any additional software on their Chromebooks.

In terms of instructional materials, teachers primarily used tutorials and assignments designed by the makers of the hardware and software they use. For example, when working with Micro:bit, teachers to a large extent relied on instructional material published on the BBC website. The few teachers that used (digital) textbooks for programming activities found that the textbooks provided a scaffold to hold on to:

> The core content [of the syllabus] is represented there… it keeps me from going off the rails, it feels right (Boo)

A few teachers had access to external support and collaborated with external school resource centres[6] that advocate for technology and entrepreneurship in school. The centres were highly appreciated both for providing expertise knowledge, and key equipment (Micro:bits):

> the school can definitely not afford the costs [for equipment] (Taylor)

Why is the particular content selected?

The question of *why* explores factors that have influenced the teachers' choices of content, learning goals and teaching materials. First, the section addresses the role of the formal curriculum. Thereafter follows an account for how external input and personal interest influence teachers' decisions to select a particular content or teaching approach.

The syllabus statements that connect programming to control and regulation of programmable digital technology have guided the teachers' transformation process. However,

---

[5] Blockly Games is a web-based application part of the Code with Google initiative: https://blockly.games/about

[6] An external resource centre is typically an organisation outside the school system with a mission to support schools on a regional or a national basis. Some centres provide hands-on support and either visit schools or let schools come visit them. Others are more of a net-presence and provide, e.g., teaching materials. The centres have different profiles, such as science and technology, entrepreneurship, arts, special needs education etc. Some are tied to universities while others are operated by municipalities. Funding may come from tax funds, grants, or the private sector.

their perceptions of what level of detail the syllabus offers differ. Some teachers felt “confident” about the intentions conveyed by the syllabus, while others found the statements vague and that the perceived lack of details could open up for an ‘anything goes’ approach:

> It is rather unclear what words and [programming] concepts you should include… some concepts come naturally, but it can be at any level (Kim)

Swedish subject syllabuses are concise by design. As an optional service for teachers, the NAE provides supplementary texts (commentary materials) that describe the content of a syllabus in more detail. The supplementary texts for mathematics and technology contain information about the role and purpose of programming in compulsory school (Vinnervik, 2022). However, when asked about whether these texts have provided any additional input on what to teach and why, only one teacher claimed to have read the commentary material for technology.

Further, the role of programming as a tool, rather than an end goal, was identified and problematised:

> I think it is a bit unfortunate… from my point of view programming has a value in itself… and as I understand the syllabus, programming is more of a tool… You are not supposed to learn programming for the sake of programming, but you should learn programming to create a technological construction or solve a mathematical problem (Frankie)

This quote reveals a belief that that learning programming as a practice is downplayed in the curriculum and that this complicates the transformation process. The teacher behind this statement further implied that the fundamental functional understanding of programming as a tool is something pupils should (but currently do not) pick up in mathematics class. For now, programming concepts are therefore introduced when the technological context calls for it.

External input seems to influence the teachers’ decisions about what to do in the technology classroom. Some explain that they have used what was already available or what had been done earlier at the school, whereas others have listened to recommendations from colleagues (physical or online), external resource centres, the NAE[7] or vendor representatives at school technology trade shows.[8] Personal interest for programmable technological devices has also influenced teachers’ decisions. For example, one teacher describes that the choice to use Micro:bit was preceded by an evaluation, carried out in spare time, of the device’s characteristics in terms of functionality, extensibility, usability, and cost effectiveness.

---

[7] In 2017, the NAE toured 13 cities to present the new curriculum reform and what it entails about digitalisation and programming. The NAE is generally careful about promoting specific materials, methods, or ideologies but during these events, each participant received a Micro:bit.

[8] For example the British Bett trade fair (formerly British Educational Training and Technology Show) or the smaller Swedish equivalent SETT (Scandinavian Education Technology Transformation) fair.

Challenges that affect the transformation of programming content (RQ2)

Intrinsic challenges

Professional knowledge and understanding All participating teachers wished they had better professional knowledge and understanding of programming, because “you cannot just make things up”. The teachers who have studied Python acknowledge that the course has improved their subject matter knowledge, that is, their understanding of basic principles and concepts of programming: “without it, I would have fallen flat, definitely” (Robin). The course did, however, not address subject didactics of technology education to any significant extent. Gaps in programming knowledge can take on different expressions:

> You can collate a set of instructions and make a ‘snurra’ [spin] or a ‘slinga’ [loop[9]] or whatever you want to call it […] I am not a programming nerd and there are many words that means almost the same thing, which can be quite confusing for me as a teacher (Kim)

The statement illustrates that teachers may struggle with using a conceptually correct language, also identified by Nouri et al. (2020). The quote was made by a well-trained and driven technology teacher, albeit with no formal training in programming, that finds the conceptual terminology confusing.

Professional adequacy This challenge is related to a teacher’s professional knowledge and understanding and captures aspects of trust in one’s abilities to teach according to the perceived curriculum. The following quote illustrates this challenge:

> When it comes to programming, we cannot be the best in the room, just forget about it. We have too little training for that (Alex)

The statement was made by a teacher with over 20 years of teaching experience in both technology and mathematics, and additional training in Python programming. Despite this background, the teacher believes that a child who spends “two weeks and three hours per evening and find it exciting, will know more than I do”.

One of the teachers (Boo) said it is a “feeling of panic before each lesson” and had told the pupils upfront that “here you have an illiterate when it comes to programming”. This teacher had next to no professional knowledge about technology and programming.

A strategy to overcome the feeling of limited professional adequacy was to script teaching activities in more detail, to thereby minimise the risk of having to deal with unexpected code related problems.

Professional attitudes and values The challenge of attitudes and values captures aspects of personal beliefs and preferences regarding the intended change. The teachers in this study appeared to have accepted the basic premise that programming is a natural part of technology education, but two examples stood out. One teacher (Taylor) lacked formal qualifications and experience of teaching technology. Even though the teacher found this lack of professional knowledge troublesome, a personal interest in technology made the teacher embrace the message of reform and look for a way forward. A more

---

[9] In Swedish, ‘snurra’, ‘slinga’ and ‘loop’ are used interchangeably.

restrained attitude was shown by a teacher (Boo) who explicitly said, "I do not want to teach technology", and instead preferred to teach science and mathematics. Unfortunately for this teacher, no other teacher at school had formal qualifications. It should, however, be noted that the teacher seemed to take on the task utmost responsibly. To conclude, the reform message was received with enthusiasm, but for various reasons, also with some degree of hesitancy.

Teaching approach This challenge is related to teachers' flexibility and ability to adapt their teaching approaches to meet the particular "needs of students and the innovation to be implemented" (Finger & Houguet, 2009, p. 316). It is interconnected with teacher professional knowledge and understanding. Programming seems to pose unique challenges that teachers must adapt to:

> If you are coding, it's, as people say, 10% coding and 90% debugging. There are many who do not enjoy this. You want things to happen immediately (Sam)

Although the practice of debugging syntactic and logical errors is a fundamental part of programming, the quote illustrates a struggle to find a reasonable balance between different programming practices and to meet the needs of the pupils, and the perceived intent of the reform. The combination of debugging practices and pupils with literacy difficulties forces teachers to be flexible:

> They can get a little frustrated because if there is a semi-colon in the wrong place, you are smoked… and we have several children in 7th grade who have difficulties to distinguishing between uppercase and lowercase letters (Alex)

The teacher stressed the importance of being a calm voice and help the pupils understand how to use the programming environment to identify errors. The need to be flexible was further illuminated through an example of how pupils had found it difficult to translate pseudo code instructions in Swedish, into valid instructions in a textual programming language in English. To reduce complexity, the pupils were instead given premade code snippets to help them get used to a text-based programming methodology.

Further, teachers apparently have to develop their ability to find bugs in programs they have not designed themselves, and to provide the right amount of feedback to help pupils identify and understand the nature of the problem:

> It can be challenging to discover errors in the program code, and it is not something you are used to, and to also learn which errors may be typical and to be able to provide some sort of feedback so that I do not directly point out where the problem is (Kim)

Ownership The interviews revealed that these teachers do not yet fully "own" the reform message, but some of them are clearly underway. Ownership of a reform grows over time as the intrinsic and extrinsic challenges presented in this results section are resolved.

Extrinsic challenges

Resources Resources include, for example, teaching materials, and the school IT infrastructure. Most teachers mention and return to the issue of resources during the conversations.

In technology education, certain equipment is required, and this is associated with costs. However, when the "municipality's economy is in free fall", as one teacher ironically puts it, the situation can become troublesome. Two schools had their equipment costs covered by external resource centres, and both teachers believed that without the external support, there would be no programmable devices at their respective schools. A few teachers had no programmable devices due to financial restraints. It was suggested that a list of recommended equipment, provided by the NAE, could be beneficial in negotiations over school spending.

In terms of teaching materials, regular textbooks from established publishers were rarely used. Even though most schools had textbooks, these were printed before 2018 and contained little to no information about programming. Two schools used digital textbooks designed for the new curriculum. One of them used a test licence, which eventually expired because the school was unable to afford the costs. Instructional materials provided by the hardware vendors (e.g., LEGO, BBC) were commonly used. However, it is not certain that the available teaching materials meet the expectations of teachers:

> I don't think there has been good enough equipment, nor teacher guides, which means that I have taken it… I am still in a start-up phase (Max)

This quote suggests that the teacher has had time to search for teaching materials but had found it difficult to identify quality materials. The teacher found it important to make informed decisions about which materials to use, but the difficulties finding appropriate resources slowed things down.

Without equipment and up-to-date technology education textbooks, one teacher had decided to rely on a textbook on Python programming for mathematics. The teacher was puzzled about the order in which the reform had been pushed forward:

> It feels a bit forced that we should implement programming without providing the conditions for it (Robin)

However, access to teaching materials is to no avail if the skills and knowledge are missing:

> I know of schools that have bought sets of educational robots […] These schools had early adopters who honed their skills, but then, this person leaves, and the equipment was stowed away in a cupboard. I think this is a greater risk than the equipment becoming outdated… having equipment nobody knows how to use […] The key is to understand how to use it in a pedagogically sound way (Sam)

Regarding general IT infrastructure, iPads, regular laptop computers (Windows/Mac) and Chromebooks were used. The use of iPads was not seamless, either because apps of interest came with a cost, which the school could not afford, or did not support older devices. A similar experience was shared by Chromebook users. Either there was no software available, or they were not allowed to install it. Situations like these forced teachers to revert to solutions that led to other problems, such as with saving code:

> We use repl.it. It is a hassle to write and run the code there. Sometimes it disappears and the children have to start all over again (Robin)

Some teachers described how they occasionally had to use old Windows laptops to be able to establish stable connections with the peripheral devices they used in class, when the iPads or Chromebooks failed to do so.

Time management In Finger and Houguet's (2009) framework, time is considered the most important teacher resource and therefore treated as a separate challenge. For example, finding and learning how to use appropriate teaching materials was described as time-consuming, especially when working in isolation. This forced the teachers to use the classroom as a testbed. Time is required for undergoing PD and the teachers who had studied the Python programming course estimated that they were compensated for approximately half the time that was required (200 h).

In addition, teachers need to estimate how much time should be spent on activities that include programming. It was difficult for the teachers to account for specific figures in terms of how many hours would be spent over the course of three school years. Based on their answers, I made the interpretation that they expected to spend between 10 and 15 h on activities in technology education that include programming over three school years. The general perception among the teachers was that they did not expect to “get very far” in terms of what the children would learn about programming during this time. It was, however, believed that things will improve when more functional interdisciplinary work routines between technology and mathematics are in place.

Practicality of implementation This challenge captures aspects of reform compatibility with already existing subject content, work practices and school culture. As we have previously learned, the way in which the programming message is conveyed in the intended curriculum seems to cause uncertainty and thereby affects the practicality of the implementation:

> It is stated that programming should be used but with little further directions and that is a bit vague. On the other hand, it may be a good thing and opens for different approaches. However, as a less experienced teacher, you may need some help to get started, a bit more guidance. (Chris)

This teacher perceived that the technology syllabus reveals few details about the nature of programming, but at the same time recognizes that this opens a space for teachers to exert agency, available for those who have the capacity. Another teacher noted that “everything is open for interpretation” and believed there will be significant differences in how the intended syllabus is understood and transformed. However, it was also suggested that the lack of details in the technology syllabus would be less of a problem, if the mathematics syllabus had been more detailed about which programming practices and concepts the pupils should learn. If that would be the case, a technology teacher could have a better understanding of what knowledge pupils bring to the technology classroom. In addition, the lack of direction in terms of programming languages was found confusing. Teachers seemed to suggest that Python has become a de facto standard language in mathematics, while they did not see a similar consensus for technology education:

> It is somewhat unclear which programming language will become the main language that we should use. I don’t think you should use different languages. It should have been decided: let’s use this language (Alex)

Learning progression is another challenge of practicality that emerged in the interviews. In grade 7, new classes are often formed, with children coming from different schools. The teachers felt that they were poorly informed about what children from other classes and schools know, and are expected to know, about programming. As a temporary solution, some concluded that it is best to “start from the beginning” but that it would be “much easier if someone had made a learning progression which you could use as a template”.

Student assessment In the Swedish school curriculum, knowledge requirements regulate what knowledge pupils are expected to attain, and they are related to the aims and content of individual subjects. Even though the knowledge requirements emphasize results and achievement (Sundberg & Wahlström, 2012), they provide no specific details about assessment of programming knowledge, and the teachers believed this made the assessment process more complex. The teachers argued that programming knowledge is subordinated to knowledge about technological systems and construction work. Assessment should therefore focus on the broader process of technological work and "not whether they can program. Programming is just a small part". It was further argued that a formative assessment practice makes it possible to follow how the children develop their ideas, constructions, and reasoning, as well as how they drive the work process and tackle problems that arise. Despite the absence of clear assessment guidelines, teachers saw a need for pupils to learn certain fundamentals:

> They should know what a loop is, what a sequence is, an algorithm, but when you look in the knowledge criteria section [in the syllabus], it does not say much about programming (Taylor)

Assessment is a challenge for teachers and the NAE is called out for not providing adequate support:

> It would have been great to have material from the NAE: "We recommend that you do this", that would make it easier for teachers. Programming is a very broad term and encapsulates everything from giving instructions to a friend […] to making advanced codes in a programming environment. It would have been nice to know what [difficulty] level is appropriate for the younger ages, and what is appropriate for the older ages. It would be great to have this kind of support (Taylor)

The NAE provides assessment guidelines online[10] as a service for teachers, but at the time of the present study, there was no support material for assessment of technology education and programming in grades 7–9.

History and tradition Even if some teachers had a couple of years' experience of implementing programming in technology education, the overall message conveyed in the interviews was that they were standing on new grounds. The status and position of the technology subject itself is something one teacher ponders about:

> I believe technology is constantly changing and this is something we must relate to. The subject has not yet settled, has it? I do not think so. I have met colleagues who think it is easy to receive an A in technology… and I go, jeez… you have probably not worked much with technology, because I think it is quite difficult for a pupil to meet the requirements [for A] (Kim)

Professional development and support The programming reform requires large-scale and continuing PD. The few teachers who have had the opportunity to study Python programming still asked for additional subject-didactic PD. In addition, several teachers did not know whether they would teach technology next school year and this uncertainty left the question of PD hanging in the air.

---

[10] The Assessment Portal (Bedömningsportalen) is an online service provided by the NAE that aims to support teachers in the knowledge assessment process, https://bp.skolverket.se

Regarding support, the present study has showed that external resource centres can play a vital role in how the reform is enacted. Further, the National Digitalisation Strategy for the School System (Regeringsbeslut U2017/04119/S, 2017) states that schools should have access to both technical and pedagogical support. Few schools had pedagogical support in terms of ICT educators[11] within the organisation. The perception was, however, that these educators were either very busy, or did not have enough subject knowledge to be able to help improve teaching.

Programming is embedded as an interdisciplinary subject content, and the intended curriculum advocates, albeit somewhat cautiously, collaboration between mathematics and technology (Vinnervik, 2022). Among the teachers, there was a clear perception of what the division of roles between the subjects. At an overall level, their view seemed aligned with the intentions conveyed in the curriculum (Vinnervik, 2022):

> I think it's very clear. I believe and feel that it will be about spending more time in mathematics about learning programming as a language and all concepts and writing code […] while programming in technology is all about artefacts (Max)

Interdisciplinary work between technology and mathematics was largely absent. Where there was ongoing collaboration and mutual support, it was merely due to the teachers teaching both subjects.

Help from pupils was another type of support that teachers used, for example, to help identify and evaluate peripheral devices for construction work. In the classroom, pupils who caught on quickly were asked to help their classmates.

Discussion

This paper explores how technology education teachers contextualise and transform programming into teaching, and what challenges they may face during this process. The study reported was conducted 27 months into the reform which introduced programming into the school curriculum. The knowledge the study provides is of significance to understand how the intended curriculum is interpreted and realised and may inform future curriculum evaluations and reforms.

The results suggest that the participating teachers, at an overall level, shared a mutual understanding of what role programming should play in technology education. A message that emerges from the study is that teachers saw technological work (e.g., construction work or technological systems) as the primary activity, for which programming is a building block that helps shape the understanding of the particular technological context being explored. In other words, the teachers did not see programming knowledge per se as a main goal. This view conforms to the intentions conveyed in the written curriculum (Vinnervik, 2022) There were, however, certain basic concepts of programming that the teachers believed the children should understand, such as sequence of commands and conditional structures. When characterising programming, the teachers rarely described programming as an iterative process. This suggests that programming knowledge, in the context of technology education, currently is shaped into being mostly about gaining some basic and

---

[11] In Sweden, an ICT educator (IKT-pedagog) is generally responsible for support and improvement of the use of digital tools for teaching and learning. There are no formal requirements for such a position.

practical experience of how instructions can be constructed and transferred to a physical computing device, and to understand the broad structures and logic behind some of these instructions.

How curriculum content, for example, programming, is realised is decided both by the intended curriculum and how it is perceived (see Adolfsson & Alvunger, 2018; Lundgren et al., 2004; Wahlström & Sundberg, 2015), but also by the backdrop against which teachers make their decisions about how to approach and transform the new content (Linde, 2012; Lipsky, 2010). A previous study (Vinnervik, 2020) suggests that teachers accept the rhetoric behind the reform message and that programming is seen as a relevant and stimulating contribution to the curriculum. However, the study further shows that teachers may encounter several intrinsic and extrinsic challenges along the way of transformation and implementation. The challenges identified by Vinnervik (2020) are also to a large extent identified in this paper. For example, teachers' perceived lack of professional knowledge and understanding has been identified as a major obstacle in previous research (Sentance & Csizmadia, 2017; Vinnervik, 2020). This paper clearly indicates that technology education teachers, despite having long teaching experience, believe they lack necessary professional knowledge. This perceived knowledge deficit can be remedied through PD. Although no widespread conclusions can be drawn from the study data, it raises concerns about the extent to which offered PD opportunities actually reaches the intended teachers. The PD situation for Swedish teachers is problematic (Parding et al., 2018) and few of the study participants had undergone PD.

Furthermore, several extrinsic challenges emerged through the interviews. For example, the level of clarity with which the reform message is communicated in the curriculum, affects teachers' transformation. This and other research (Vinnervik, 2022) suggests that the intended curriculum mainly concludes that programming should be used, but provides little guidance on what specific content related to programming the children should learn. Programming is designed as an interdisciplinary curriculum content, and perhaps the vagueness of the intended curriculum, experienced by some teachers in this study, would be less of a problem with a more detailed mathematics syllabus. Assessing programming knowledge was identified as a challenge by the participating teachers. Åkerfeldt et al. (2018) note that there are no specific knowledge requirements in the mathematics and technology syllabuses that explicitly address the expected learning outcomes of programming. This was also observed by the teachers in the present study. Åkerfeldt et al. (2018) express concern that the pupils' experiences of programming will be mostly about learning the basic procedures and routines of how a computer program can be constructed in one or a few different programming languages and environments. The present results suggest that even though the teachers apparently intend to teach about procedures, they seem to advocate a more holistic approach to assessment and focus on the broader technological competencies rather than specific "technicalities". On the other hand, one of the teachers, with a background in engineering, expressed a particular concern that programming as practice is downplayed in the curriculum, arguing that programming knowledge in itself is part of the technological knowledge realm. For the school authorities, the dilemma is to understand and determine when and what curriculum change is desirable, feasible and sustainable. This paper shows that most of these teachers work on their own, which is considered challenging (Shulman, 1998; Yadav et al., 2016). Previous research has shown that collegial interplay around programming is appreciated (Vinnervik, 2020), and that collegial conversations can be a powerful tool for PD and improvement initiatives (Langelotz, 2014). Fullan (2001, p. 60) notes that schools that deliberately set up networks, or communities for workplace learning, will get teachers who "constantly search for new ways to make improvements", but at the same time remarks that the working conditions of teachers seldom allows "sustained teacher innovation". Only one of the teachers in this study had nearby colleagues that also taught technology. The number of teachers who work in solitude could be reduced through collaboration with mathematics teachers, in line with the intentions conveyed in the curriculum (Vinnervik, 2022). There are rare examples of interdisciplinary work in this study, mainly occurring when teachers taught both subjects.

It was further found that teachers had limited knowledge of what the children had learned about programming in previous grades, unless they were also teaching those grades themselves, which a few did. The Swedish Schools Inspectorate (2014) has concluded that there is a risk that teachers intentionally teach at a lower level to compensate for a potentially inadequate experience of technology education in earlier school years. This paper presents some evidence of teachers who decided to start from the very beginning, regardless of whether some pupils may have had prior experience to build on. Teachers in this study call for a more coherent and well-designed learning progression than what is currently in place. Research (Vinnervik, 2022) shows that aspects of learning progression are addressed in the curriculum supplementary texts. The teachers in this study did, however, not pay much attention to these supplementary texts. The question is, however, whether they have missed anything of importance, as the usefulness of the learning progression statements provided in these texts is questionable (Vinnervik, 2022).

Conclusions

Enacting long-term, sustainable change is a lengthy and incremental process, particularly when the reform message is complex (Ahtiainen, 2017; Fullan, 2001; Hargreaves & Goodson, 2006). The results of this study support this finding. They reveal a group of teachers who were serious in their efforts to provide the children with insight into what programming means in the context of technological work. For the programming reform to be feasible, effective, and sustainable, this paper points to several intrinsic and extrinsic challenges that must be addressed appropriately. It is on the verge of being unfair to ask teachers to drive educational change without also giving them proper training and resources 'to drive'.

Funding Open access funding provided by Umea University.

Declarations

Conflict of interest The author declares that no conflict of interest exists.

References

Adolfsson, C.-H., & Alvunger, D. (2018). The selection of content and knowledge conceptions in the teaching of curriculum standards in compulsory schooling. In *Transnational Curriculum Standards and Classroom Practices The New Meaning of Teaching* (pp. 98–115).

Ahtiainen, R. (2017). *Helsinki Studies in Education, number 13 Shades of change in Fullan's and Hargreaves's models Theoretical change perspectives regarding Finnish special education reform. Fabianinkatu* (Vol. 33).

Åkerfeldt, A., Kjällander, S., & Selander, S. (2018). *Programmering: introduktion till digital kompetens i grundskolan* (Första upp). Stockholm: Liber.

Balanskat, A., & Engelhardt, K. (2015). *Computing our future Computer programming and coding - Priorities, school curricula and initiatives across Europe*. *European Schoolnet (EUN Partnership AISBL)*. Retrieved from http://www.eun.org/c/document_library/get_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887

Billett, S. (2006). Constituting the workplace curriculum. *Journal of Curriculum Studies*. https://doi.org/10.1080/00220270500153781

Bocconi, S., Chioccariello, A., & Earp, J. (2018). The nordic approach to introducing computational thinking and programming in compulsary Education. *Report Prepared for the Nordic@BETT2018 Steering Group*, 42.

Cohen, D. K., & Ball, D. L. (1990). Policy and practice: An overview. *Educational Evaluation and Policy Analysis, 12*(3), 233–239.

Crick, T. (2017). Computing education: An overview of research in the field. *Cronfa - Swansea University Open Access Repository*, (April), 1–38. Retrieved from http://cronfa.swan.ac.uk/Record/cronfa43589

Elo, S., Kääriäinen, M., Kanste, O., Pölkki, T., Utriainen, K., & Kyngäs, H. (2014). Qualitative content analysis. *SAGE Open, 4*(1), 215824401452263. https://doi.org/10.1177/2158244014522633

Finger, G., & Houguet, B. (2009). Insights into the intrinsic and extrinsic challenges for implementing technology education: Case studies of Queensland teachers. *International Journal of Technology and Design Education, 19*(3), 309–334.

Fullan, M. (2001). *The new meaning of educational change* (3rd Ed.). New York : London: New York : Teachers College Press.

Goodlad, J. I. (1979). *Curriculum Inquiry*. The Study of Curriculum Practice.

Hallström, J., Hultén, M., & Lövheim, D. (2014). The study of technology as a field of knowledge in general education: Historical insights and methodological considerations from a Swedish case study, 1842–2010. *International Journal of Technology and Design Education, 24*(2), 121–139. https://doi.org/10.1007/s10798-013-9252-x

Hargreaves, A. (2005). Educational change takes ages: Life, career and generational factors in teachers' emotional responses to educational change. *Teaching and Teacher Education, 21*(8), 967–983. https://doi.org/10.1016/j.tate.2005.06.007

Hargreaves, A., & Goodson, I. (2006). Educational change over time? The sustainability and nonsustainability of three decades of secondary school change and continuity. *Educational Administration Quarterly, 42*(1), 3–41. https://doi.org/10.1177/0013161X05277975

Jarl, M., & Rönnberg, L. (2019). *Skolpolitik : från riksdagshus till klassrum* (Tredje upp). Stockholm: Liber.

Kjällander, S., & Petersen, P. (2016). *Översikt avseende forskning och erfarenheter kring programmering i förskola och grundskola* (Vol. 1).

Kvale, S. (1997). *The Qualitative Reserach Interview*. Lund: Studentlitteratur.

Langelotz, L. (2014). *Vad gör en skicklig lärare?* Retrieved from https://gupea.ub.gu.se/bitstream/2077/34853/1/gupea_2077_34853_1.pdf

Larke, L. R. (2019). Agentic neglect: Teachers as gatekeepers of England's national computing curriculum. *British Journal of Educational Technology, 50*(3), 1137–1150. https://doi.org/10.1111/bjet.12744

Linde, G. (2012). *Det ska ni veta! En introduktion till läroplansteori* (3., [rev.]). Lund: Studentlitteratur.

Lipsky, M. (2010). *Street-Level Bureaucracy: Dilemmas of the Individual in Public Services*. Russell Sage Foundation. Retrieved from http://www.jstor.org/stable/https://doi.org/10.7758/9781610447713

Lundgren, U. P., Lundahl, C., & Román, H. (2004). *Läroplaner och kursplaner som styrinstrument* (No. 2003:1767).

Luxton-Reilly, A., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Ott, L., Paterson, J., Scott, J. M., Sheard, J., Szabo, C. (2018). *Introductory programming: A systematic literature review. Annual conference on innovation and technology in computer science education, ITiCSE*. https://doi.org/10.1145/3293881.3295779

McGill, T. J., & Volet, S. E. (1997). A conceptual framework for analyzing students' knowledge of programming. *Journal of Research on Computing in Education*, *29*(3), 276–297. https://doi.org/10.1080/08886504.1997.10782199

Medeiros, R. P., Ramalho, G. L., & Falcao, T. P. (2019). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education, 62*(2), 77–90. https://doi.org/10.1109/TE.2018.2864133

Mølstad, C. E., & Karseth, B. (2016). National curricula in Norway and Finland: The role of learning outcomes. *European Educational Research Journal, 15*(3), 329–344. https://doi.org/10.1177/1474904116639311

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry, 11*(1), 1–17. https://doi.org/10.1080/20004508.2019.1627844

Ornstein, A. C., & Hunkins, F. P. (2018). *Curriculum: Foundations, Principles and Issues* (7th Ed.). Pearson.

Parding, K., Sehlstedt, T., Johansson, A., Berg-Jansson, A., & Jakobsson, M. (2018). *Lärares arbetsvillkor i kontexten av marknadisering, privatisering, val och konkurrens*. Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-67117

Priestley, M. (2019). Curriculum: concepts and approaches. Retrieved 10 February 2020, from https://mrpriestley.wordpress.com/2019/01/04/curriculum-concepts-and-approaches/

Regeringsbeslut U2017/04119/S (2017) Nationell digitaliseringsstrategi för skolväsendet. Sweden: Utbildningsdepartementet. Retrieved from http://www.regeringen.se/informationsmaterial/2017/10/regeringen-beslutar-om-nationell-digitaliseringsstrategi-for-skolvasendet/

Lärarnas Riksförbund. (2020). *Programmering — en skolreform utan program*. Retrieved from https://www.lr.se/opinion--debatt/undersokningar/2020/2020-04-30-programmering---en-skolreform-utan-program

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies, 22*(2), 469–495. https://doi.org/10.1007/s10639-016-9482-0

Shulman, L. S. (1986). Those who understand: knowledge growth in teaching. *American Education Research Association, 15*(2), 4–14. https://doi.org/10.1017/CBO9781107415324.004

Shulman, L. S. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review, 57*(1), 1–21. https://doi.org/10.1007/SpringerReference_17273

Shulman, L. S. (1998). Theory, practice, and the education of professionals. *Elementary School Journal, 98*(5), 510–526. https://doi.org/10.1086/461912

The Swedish Schools Inspectorate. (2014). *Teknik – gör det osynliga synligt. Om kvaliteten i grundskolans teknikundervisning* (No. 2014:04). Stockholm. Retrieved from https://www.skolinspektionen.se/globalassets/publikationssok/granskningsrapporter/kvalitetsgranskningar/2014/teknik/kvalgr-teknik-slutrapport.pdf

The Swedish Schools Inspectorate. (2019). *Digitala verktyg i undervisningen*. Retrieved from https://larportalen.skolverket.se/#/modul/0-digitalisering/Grundskola/513-Digitala-verktyg-i-naturvetenskap

Skolverket. (2017). Få syn på digitaliseringen på grundskolenivå – Ett kommentarmaterial till läroplanerna för förskoleklass, fritidshem och grundskoleutbildning. Stockholm: Skolverket.

Skolverket. (2019). Skolverket Statistical database (Siris). Retrieved 24 May 2019, from https://siris.skolverket.se/reports/rwservlet?cmdkey=common&notgeo=&p_verksamhetsar=2018&report=personal_amne2&p_lankod=&p_kommunkod=&p_hman=&p_niva=A&p_amne=18&p_verksform=11&p_hmankod=

Sundberg, D., & Wahlström, N. (2012). Standards-based curricula in a denationalised conception of education: The case of Sweden. *European Educational Research Journal, 11*(3), 342–356. https://doi.org/10.2304/eerj.2012.11.3.342

Swedish Higher Education Authority. (2020). Utvärderingen av ämneslärarutbildningarna klar. Retrieved 26 February 2020, from https://www.uka.se/sitevision/proxy/kvalitet--examenstillstand/resultat-fran-granskningarna-hogskolekollen/hogskolekollen.html/svid12_794d1f901621e65b329c27f/427238778/files/1930738/2075414/Bedömargruppens_yttrande.pdf

Swedish National Agency for Education. (2018). Curriculum for the compulsory school, preschool class and school-age educare.

Vinnervik, P. (2020). Implementing programming in school mathematics and technology: Teachers' intrinsic and extrinsic challenges. *International Journal of Technology and Design Education*. https://doi.org/10.1007/s10798-020-09602-0

Vinnervik, P. (2022). An in-depth analysis of programming in the Swedish school curriculum—rationale, knowledge content and teacher guidance. *Journal of Computers in Education*. https://doi.org/10.1007/s40692-022-00230-2

Wahlström, N., & Sundberg, D. (2015). *En teoribaserad utvärdering av läroplanen Lgr 11*.

Wahlström, N., & Sundberg, D. (2018). Transnational curriculum standards, curriculum reforms and classroom practices - an introduction. In N. Wahlström & D. Sundberg (Eds.), *Transnational Curriculum: Standards and Classroom Practices - The New Meaning of Teaching*. Routledge.

Wickman, P. O., Hamza, K., & Lundegård, I. (2018). Didaktik och didaktiska modeller för undervisning i naturvetenskapliga ämnen. *Nordic Studies in Science Education, 14*(3), 239–249. https://doi.org/10.5617/nordina.6148

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer Science Education, 26*(4), 235–254. https://doi.org/10.1080/08993408.2016.1257418