



Implementing programming in school mathematics and technology: teachers' intrinsic and extrinsic challenges

Peter Vinnervik¹

Accepted: 6 June 2020 / Published online: 17 June 2020
© The Author(s) 2020

Abstract

The 2017 reform of the Swedish national curriculum requires that all compulsory school mathematics and technology teachers integrate programming into their teaching. The new programming policy poses a particular challenge since a majority of the affected teachers have little or no previous programming experience. This paper reports on a study of teachers preparing to implement the new policy. Insight into the preparation process was made possible through recorded group conversations and data were collected in March 2018, less than 4 months before the formal enactment of the new curriculum. The results, conceptualised by using a framework for intrinsic and extrinsic challenges, reveal several challenges that can potentially affect the uptake of the programming policy and the quality of implementation such as uncertainty about the subject content, unequal professional development opportunities, lack of teaching materials and recurring problems with school IT infrastructure. This study seeks to provide knowledge about teachers' concerns and expressed needs while negotiating programming as new curriculum content and thus aims to contribute to the understanding of teachers' strategies to approach the 2017 Swedish educational reform that introduces programming. Such knowledge is valuable for the possibilities to better understand under what circumstances programming is included in school mathematics and technology. The results illustrate the complexity of curriculum reform implementation and may be of value for decision makers at all levels of school policy and also for providers of both in-service and preservice teacher training.

Keywords Challenges · Curriculum implementation · Programming · Technology education · Compulsory school · Educational change

Introduction

This paper explores consequences of the inclusion of programming in the 2017 Swedish curriculum for compulsory school from a teacher perspective. Programming was added to the syllabi of mathematics and technology and consequently requires that all

✉ 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

mathematics and technology teachers adopt the new policy. The study reveals teachers' perceptions about what the curriculum change may entail for their teaching profession with respect to challenges and needs.

Background

The Swedish education system conforms with recent international curriculum development efforts which have pushed for improved adoption and integration of digital technologies (school digitalisation) in general and computer science education in particular (Balanskat and Engelhardt 2015; Bocconi et al. 2016; Mannila et al. 2014). In 2015, the Ministry of Education and Research (Utbildningsdepartementet) appointed the National Agency for Education (NAE) to propose a new national IT strategy for the entire Swedish school system and also to propose changes to curricula and syllabi in order to reinforce and clarify programming as new core content (Utbildningsdepartementet 2015b). During 2016, substantial parts of the then prevailing curriculum, Lgr11, was "digitally remodelled", a process that introduced programming as core content to the mathematics and technology subjects. In March 2017, the government announced a new national curriculum for compulsory school, Lgr11/2017. The new curriculum was made public in June 2017, to be enacted no later than July 1, 2018.

Even though the new curriculum should be seen in light of recent years' international curricular development trends, programming has been included in the Swedish national curriculum before. In the late 1970s and early 1980s, when technological advancements brought computers that were manageable size wise and also affordable to a broader audience, schools included, *Datalära* (old term for Information science, or Informatics) was added to the 1980 curriculum, Lgr80. *Datalära* was integrated into the lower secondary mathematics syllabus and linked to both natural science subjects and civics. Programming was part of this new interdisciplinary subject which, in terms of content and underlying motives, demonstrates distinct similarities with the additions to the 2017 curriculum about digital competence (Skolverket 2017a). The purpose of programming was never to learn how to program per se, but to use programming as a medium for problem solving, primarily intended for high achievers (Skolöverstyrelsen 1984). However, for various reasons, *Datalära* did not have the impact Swedish school authorities had hoped for and it was omitted from the next national curriculum, Lpo94 (1994). It was found that schools on average assigned *Datalära* less than one-third of the recommended time of 80 h and that professional development initiatives were scarce (Riis 1987). During this period, there was also increased access to more purposeful software which lessened the need to spend time on making programs (Rolandsson 2015). It turned out that access to software alone was not enough when the teachers did not receive adequate training on how to best utilize the software in their teaching. The school digitalisation efforts of the 1980s would later be described as top-down initiatives where teachers' experiences and professional development needs were not sufficiently recognized (Farkell-Bååthe 2000; Hylén et al. 2002) and pedagogy, teaching methods and subject integration remained more or less unchanged (Riis 1991 in Farkell-Bååthe 2000).

Rationales for re-adding programming to the curriculum

A common argument in favour of programming in compulsory school curricula, which seems to have influenced recent international school policy decision making, refers to the idea that programming fosters computational thinking (CT) skills (Clements 1999; Grgurina et al. 2017). The generally accepted view is that the notion of CT first emerged in Seymour Papert's seminal book on the role of computers and computing for knowledge construction, *Mindstorms* (1980). Papert was never particularly elaborate about the constituents of CT and in 2006, Wing revived the notion by describing CT as "thinking like a computer scientist". Wing argued that computational thinking describes a particular approach to problem formulation and solving, characteristic of computer science but beneficial for all disciplines (Wing 2006, 2010). Besides Wing's description of CT, there are several others from different stakeholders, e.g. Barefoot Computing,¹ BBC² and Google³ and also from scholars (Barr and Stephenson 2011; Grover and Pea 2013) but there is currently a lack of a common definition of the constituents of CT and how it relates to other ways of thinking, e.g. procedural thinking and algorithmic thinking (Bocconi et al. 2016; Román-González et al. 2017; Weintrop et al. 2016). Some scholars also associate CT not only with skills but also with attitudes (Barr et al. 2011). CT skills are often attributed a cross-curricular value, transferable and beneficial for the understanding of many other knowledge domains (Wing 2006, 2010); skills that can become constructs in a person's generic problem solving toolbox. Literature, however, reveal that there is still much to learn about under what conditions and to what extent generic and transferable problem solving skills exist (Clements 1999; Goldenson 1996; Pea 1983; Scherer 2016; Straw et al. 2017). Billett (2003, p. 8) argues that "Proposing problem solving as a generic process, not requiring detailed knowledge of the situation where the activities are to be enacted and the kinds of solutions that are acceptable, is fanciful and flawed". A recent study by Doleck et al. (2017) explored the effects on learning and academic outcomes of CT skills (algorithmic thinking, cooperativity, creativity, critical thinking and problem solving) and the results suggest that there is a "lack of association between computational thinking and academic performance" (Doleck et al. 2017, p. 366). The authors question the current curricular push for CT and 21st century skills and suggest that better curricular alignment between instructional objectives and evaluation approaches may improve the situation. Tedre and Denning (2016) present a commendable overview of how the debate about "general-purpose thinking tools" has evolved from the 1960's and onwards and conclude that results in support of the claim of transferability are ambiguous or meagre at best.

In the Swedish curricular context, programming is attributed the ability to function as a medium that provides pupils with digital competence (Skolverket 2017a), for example the essential knowledge to handle their digital everyday lives and digital tools, but also stimulate problem solving skills, creativity and logical thinking (Utbildningsdepartementet 2015a, b). Furthermore, programming is believed to help raise girls' interest in technology, which in turn may lead to more female applicants for technology oriented higher education programs (2015b). These overarching aims are to be pursued through combined efforts in all subjects in which programming has been included.

¹ <https://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/>.

² <https://www.bbc.com/bitesize/guides/zp92mp3/revision/1>.

³ <https://edu.google.com/resources/programs/exploring-computational-thinking/index.html#!ct-overview>.

Change bring challenges

Schools are continuously asked to act upon educational reforms (Börjesson et al. 2013), spreading around the world like a “policy epidemic” (Levin 1998). Such initiatives can be of a more organizational nature, seek a change in ideology (e.g. addressing inequities) or focus on improving teaching and learning outcomes (curriculum reforms). Educational policy changes may affect local, regional or national levels and involve many different stakeholders (Ornstein and Hunkins 2018). Changes, for example, the adding of programming to the curriculum, are often urgent in the sense that they address “current needs of students and society in an increasingly changing world” (ibid.). Furthermore, most reforms significantly affect the work of teachers (Ryder 2015) and present them with challenges that in many cases are universal, subject independent.

Teachers—key policy enactors at the core of change

School academic activity is governed by the national curriculum and teachers are recognized as key policy enactors (Braun et al. 2012; Cohen and Ball 1990), entrusted to make professional and well-informed interpretations of the curriculum guidelines and design educational activities accordingly. The interpretation process poses several challenges: What intentions with respect to programming does the curriculum entail, how well are they described and clarified, and to what extent do the teachers possess adequate knowledge to make well-informed interpretations, design and evaluate classroom activities that align with the intended curriculum? Cheng and Wang refer to Spillane et al. (2002) and paint a complicated picture of curriculum reform implementation:

... teachers often lack the capacity—the knowledge, skills, personnel, and other resources—necessary to work in ways that are consistent with the policy. Spillane et al. warned that even if teachers understand the policymakers’ intentions, they still may not have the necessary skills and the human and material resources to accomplish what they perceive the policy to be asking of them. (Cheng and Wang 2009, p. 139)

The underlying reasons for this lack of capacity are important to study. Cohen and Ball (1990, p. 234) argue that “little is known about how teachers perceive instructional policies, how they interpret them, and how different kinds of policies influence teaching and learning”. Hargreaves (2005, p. 981) suggests that “understanding how teachers experience and respond to educational change is essential if reform and improvement efforts are to be more successful and sustainable”. Sentance and Csizmadia (2017) refer to Diethelm et al. (2012) and argue that the teacher perspective is of importance in order to understand the outcomes of teaching. Yadav et al. (2017) state that little is known about the experiences and challenges computer science teachers face in the computer science classroom.

Conceptual framework

Some reforms are well received and implemented while others pass more or less unnoticed, and a majority of efforts seem to be unsuccessful (Fullan 2001; Fullan and Pomfret 1977; Ornstein and Hunkins 2018). Today, there is a large base of curriculum implementation research and different scholars (Altrichter 2005; Barnes 2005; Finger and Houquet 2009; Fullan 2001; Fullan and Pomfret 1977; Ornstein and Hunkins 2018;

Ryder (2015) have identified a number of characteristics, or key factors, associated with change processes that are of significance for the outcome. Ryder (2015) conducted a review of research on curriculum reforms in science education that have examined “teachers’ beliefs, practices and reflections, the response of teacher communities to reform (e.g. within school departments), and teachers’ (and other stakeholders’) experiences across school systems” (Ryder 2015, p. 87). Several key factors that affect how teachers choose to approach reform processes were identified and organised under three different categories: personal, internal and external. The personal factors relate to e.g. teachers’ subject knowledge, pedagogical skills, beliefs and professional identity. Internal factors relate to for example, school culture, working practices and leadership, and support structures such as teaching materials and other resources. The external factors affect at a systemic level and address the flexibility or rigidity of the new ideas in the curriculum and how it relates to or is affected by other reforms.

This study draws on a conceptual framework developed by Finger and Houguet (2009) who studied intrinsic and extrinsic challenges for implementing technology education in Australia. Intrinsic challenges are described as challenges that “teachers may face on a personal level, such as teacher knowledge and understandings of the intended curriculum” while extrinsic challenges are “external or environmental factors that may impede the implementation process, such as insufficient resources and a lack of professional development” (Finger and Houguet 2009, p. 311). The framework positions the curriculum implementation process in a sequence where the intended curriculum precedes and the implemented and received curriculum follows the implementation phase. The challenges in this framework are in many ways similar to the key factors identified by Ryder but the context of technology education makes the framework particularly appropriate for the context of this study. The framework will not be used in its entirety but contributes with its classification of intrinsic and extrinsic challenges. This classification will be used both as lens in the analysis process of the data transcripts and as a structural tool for presenting the findings. Table 1 presents the challenges identified by Finger and Houguet.

The intrinsic challenges are in many ways both interlinked and interdependent. For instance, the challenge of professional adequacy which concerns “the level of teacher confidence in their ability to teach technology” (Finger and Houguet 2009, p. 327) is correlated to the challenge of professional knowledge and understanding. The more solid subject knowledge base teachers have, the greater the trust in one’s ability to teach the subject. In addition, professional attitudes and values and teaching approach are also considered to be interlinked with professional knowledge and understanding. A lack of professional knowledge and understanding combined with a hesitancy to alter or adapt teaching approaches

Table 1 Intrinsic and extrinsic challenges for technology education curriculum implementation

Intrinsic challenges	Extrinsic challenges
I1. Professional knowledge and understanding	E1. Resources
I2. Professional adequacy	E2. Time management
I3. Professional attitudes and values	E3. Practicality of implementation
I4. Teaching approach	E4. Student assessment
I5. Ownership	E5. History and tradition
	E6. Professional development and support

may affect the enthusiasm a teacher carries for a subject or subject specific content. And vice versa, enthusiasm may be a valuable brick in a knowledge construction process.

Some extrinsic challenges are interlinked in a similar way, for example, the challenge of time management can be considered as a resource but stands as a separate challenge in this framework. The element of practicality is dependent on the level of intrusion the innovation entails for existing classroom practice. The less departure from established behaviour the better the chances are for a more successful reform (Careless 1997). For further, and more in-depth details about the framework, see Finger and Houquet (2009).

Purpose of study

The overall purpose of this study is to describe challenges related to the programming integration that teachers identify as urgent and necessary to the new curriculum policy. Curriculum change is a process with different stages (Fullan 2001; Ornstein and Hunkins 2018). The study seeks to provide insight into the early stages of such a process and help construct an understanding of whether the programming policy is being effectively enacted or not. The results can be valuable for subsequent studies that wish to further the understanding for teachers' conditions for implementing this particular reform but may also be relevant to studies of reform implementation related to digital competence and reform studies in general. Furthermore, the results may be of value for decision makers at all levels of school policy and also for providers of both in-service and preservice teacher training.

The study is based upon two assumptions: (1) teachers have concerns and challenges they must relate to and manage before they are able to integrate programming in their teaching, and (2) programming as core content may pose a particular kind of challenge to the teachers. The specific research question is:

What challenges for integrating programming are expressed by in-service teachers?

Method

Context of study

Finger and Houquet (2009) identify professional development as one of several, interconnected, extrinsic key factors that either inhibit or enable change. Again, a large number of teachers are required to undergo professional development, and this time, the focus is on programming. The current situation is that most Swedish teachers lack both formal and informal programming education and have little or no experience of teaching with/about programming. The NAE recommends all teachers study their 16 h online introductory course *About programming*, whether they are directly affected by the new policy or not. Mathematics and technology teachers are implicitly expected to take part in a multi-step professional development process which includes taking at least one 7,5 credit university course in introductory programming in addition to the introductory course. The same applies for most of the student teachers who graduate in a near future since, at the time of study, most Swedish universities are yet to integrate programming in relevant teacher training programs. To complicate matters further, there is an already existing need for professional development amongst technology teachers. Statistics from the NAE for school

year 2018/19 reveal that 55%⁴ of all grade 1-9 teachers in technology lack formal eligibility to teach the subject (Skolverket 2019b). A 2014 evaluation of technology education in compulsory school points out an undesirable abundance of unreflected activity in the technology classroom, where the learning goals and contexts of the activities are not made clear enough for the students (Skolinspektionen 2014). The evaluation also revealed that teachers, both qualified and unqualified, expressed uncertainty about the subject's distinctive characteristics, overall purpose and how to teach the subject matter. Further, the report revealed that "teachers with insufficient skills often, and unreflected, rely on pre-made teaching materials" (ibid, p. 26), something that may undermine the purpose, core content and learning goals of the subject.

Description of participants

The study reports from three group conversations between in-service compulsory school teachers ($n = 19$) in mathematics and/or technology who are preparing to meet the requirements of the new curriculum. The teachers, representing grades 1–9, were invited as part of a short professional development program funded by the NAE. The purpose of the program was to extend previous government funded professional development programs and thereby maintain and strengthen university-school collaboration and provide additional opportunities for collegial learning amongst teachers. For the particular occasion when data collection took place, the teachers had specifically requested time for exchange of experience and to talk about how to approach the new curriculum content that was added as a consequence of the new national IT-strategy, programming in particular.

All participants have a key function at their respective schools in the sense that they are either holding a career position as *first teachers*⁵ or have undergone a previous subject related professional development program provided by the NAE (e.g. the Mathematical Boost⁶). A teacher in such a key position is experienced and carries a certain responsibility for, for instance, collegial support and guidance and/or subject related development activity (Skolverket 2015).

Data collection

The sample is based on a combination of opportunity sampling and purposive sampling (David and Sutton 2010). The group studied was not assembled specifically for this study, these teachers would have had these conversations with or without data collection. The particular group constitution with teachers who hold key functions at their respective schools as described above, made it a relevant and purposeful sampling group to study. These teachers are presumably more conversant with subject and curricular development than their colleagues. Thus, the combination of both opportunity and purposive sampling. Fourteen different schools are represented in the data, 12 of which belong to the same mid-sized Swedish municipality and the remaining two to an adjacent municipality.

⁴ Grade 1–3: 42% of the teachers have formal eligibility, grade 4–6: 45% have formal eligibility, grade 7–9: 51% have formal eligibility.

⁵ The first teacher reform was instigated in 2013 and allowed teachers with particular skills to apply for a salary increase (Erlanson et al. 2018).

⁶ <https://larportalen.skolverket.se/#/moduler/1-matematik/alla/alla>.

The data collection phase took place at the local University in March 2018, less than 4 months prior to the formal enactment of the new curriculum. The teachers were divided into three groups based on what stage they teach. The first group comprised four primary school (grades 1–3, ages 7–9) teachers, the middle school group (grades 4–6) was comprised of eight teachers and the third group of seven lower secondary teachers (grades 7–9). The group conversations lasted approximately 70 min each. The participants were informed of the purpose and focus of the study in the beginning of the session and asked whether they would like to participate or not. All agreed to participate but they were also explicitly verbally informed that they could opt out by switching off the recording device (surf tablet) anytime, regardless of reason. Neither group chose to finish the recordings in advance and while picking up the recording devices, all participants were still fully aware that they had been recorded. The participants received the initial instruction to individually depict the current preparation status at their respective schools with regard to the 2017 curricular clarifications and reinforcements concerning school digitalisation and programming. While describing the situation at their respective schools, they were asked to share and discuss experiences, if any, and what challenges and possibilities they could identify.

The data collection method differs from the more commonly used focus group interview (David and Sutton 2010) setup in the sense that no interviewing researcher was present in any of the groups. Instead, the conversations were moderated by the participants and subsequent conversation topics were decided by the group members themselves. The data collection method aligns with the overall purpose of the study to discover issues, related to the integration of programming, teachers identify as urgent or important to address prior to the enactment of the new curriculum. Furthermore, letting the participants control the conversation topics is a way to ensure that focus is on real teacher concerns and potentially makes the teachers more at ease and willing to both share personal experience and ask questions.

Data analysis

The aim of the study was to explore and illuminate the breadth of concerns expressed from a group of teachers and not to quantify the results. This means that aspects voiced by the single teacher are considered equally important as aspects shared by many. The data analysis began with the initial transcription phase. To get thoroughly familiarized with the raw data set, all transcription was carried out by the article author alone (step 1). Most audible content of the data set was transcribed, with parts of the conversation that did not inform the study (e.g. private conversations) left out. The transcribed material was imported into the OpenCode v4.03 software for coding. The initial steps in the analysis were conducted by using an inductive thematic analysis approach following Braun and Clarke (2006). During the first cycle coding, the transcripts were coded in an inductive and mainly linear process, but it was occasionally necessary to go back-and-forth for recoding or refining initial codes (step 2). The process was executed with the aim to stay as close to the data as possible, code extensively in order to open up as many potential patterns as possible and to remain open to whatever theoretical directions that may be discerned in the data. Once the initial coding phase was completed, a large number of emergent codes (> 350) had been established. Each code is a word or short phrase that aims to capture the essence of a chunk of data. The analysis process progressed by collating and, when necessary, rephrasing or merging codes in order to start identifying comprehensive patterns (step 3). Once the codes were considered to be consistent, the process proceeded to the next step (4). In this phase, the conceptual framework by Finger and Houguet (2009) was applied and the

Table 2 Transcript and coding examples

Analytical unit	Code	Type of challenge
... we have tested a little here and a little there, testing different programs as well, but there hasn't been any structure in the teaching around programming [...]	Unstructured testing	Teaching approach (I4)
I do not feel that I have control,	Lack of control	Ownership (I5)
I think it is almost unpleasant so...	Feeling unpleasant	Professional adequacy (I2)
I would not refrain teaching it, but	Sense of responsibility	Beliefs and attitudes (I3)
I have so many issues about this,	Having issues	Professional adequacy (I2)
how should I assess,	Assessment	Student assessment (E4)
how should I do...	Lack of knowledge and guidelines	Teaching approach (I4) Professional knowledge (I1)

I1-I5 and E4 refer to the categorization of intrinsic and extrinsic challenges by Finger and Houguet (2009) See Table 1 for complete list of challenges

deductive part of the analysis commenced. Each inductively created code was reviewed and assigned to the relevant type of challenge from Finger and Houguet's conceptual framework (Table 1). The following step (5) was to return to the coded data extracts to search for coherency or anomaly between the clustered codes. The analysis identified codes related to all intrinsic and extrinsic challenges, with a particular overweight for codes related to Resources ($n=95$) and Professional development and support ($n=70$). Table 2 illustrates the relation between transcribed text (analytical units), codes and type of challenge (intrinsic or extrinsic).

During the penultimate step in the analysis process (6) the codes were further grouped to form sub-themes for the extrinsic challenges Resources and Professional development and support. The results are presented with illustrative quotes from the data corpus (step 7). The transcription examples (Table 2) and the quotes presented in this paper have been translated to English by the article author.

Results

The results are presented in two sections: (1) Intrinsic challenges for implementation of programming, (2) Extrinsic challenges for implementation of programming.

Intrinsic challenges for implementation of programming

Professional knowledge and understanding (I1)

Teachers' professional knowledge is a fundamental base for teaching, and the results of this study show that the teachers' knowledge of programming varies. A few of the teachers have some prior knowledge of programming, either due to private interest or because they have

made the decision to precede the policy change. However, a key issue that is illuminated in these conversations describes the lack of professional knowledge and understanding of the challenge at hand, the teachers question their own knowledge base. The teachers express concerns about the lack of both content knowledge and pedagogical content knowledge:

what I lack is... basic knowledge of programming, what it is, and some kind of a plan, what do we want to achieve, what steps do we need to take, there is no structured teaching and perhaps that is what I would like to see to be satisfied, so... but I do not yet have enough knowledge...

Another teacher's voice illuminates the knowledge void but also how the lack of any previous experience (e.g. from own schooling or teacher education) creates a feeling of insufficiency. This teacher also puts his/her experienced knowledge insufficiency in a greater context, referring to teachers as a group where presumably "most" colleagues share a similar need for 'help' to meet the policymakers' intentions:

Oh, help, I know nothing about programming... I think many feel the same... there are probably few who... who feel... you have nothing to fall back to, you have no personal experience of it at all...

Professional adequacy (I2)

Closely related to the challenge of professional knowledge and understanding is the challenge of professional adequacy. Teaching experience is another key factor that contributes to a teacher's feeling of adequacy. For this specific study, this challenge is displayed through the expressions of uncertainty some teachers share and which are related to how they perceive their ability to initiate and lead classroom work that aligns with the requirements of the new programming policy. As witnessed by one teacher who described a feeling of discomfort, induced by the insight that standard teacher procedures are challenged by this curricular change:

... I do not feel that I have control, I think it is almost unpleasant so... I would not refrain teaching it, but I have so many issues about this, how should I assess, how should I do, how do I get I control...

Professional attitudes and values (I3)

During the conversations, there is no questioning of the re-entrance of programming in the curriculum. There seems to be a general consensus that the content is both timely and valid as they talk about the role of technology in today's society as something that is relevant, independent of whether you work as an 'electrician' or a 'truck driver'. Even though great uncertainty for the task ahead is expressed during these conversations, they also reveal a change prone attitude and a group of teachers who are looking forward to change and "want to get started right now":

... I think it adds some spice to the life as a teacher... [it is] fun when something new comes along...it's like: Yes, now something is happening!

The following quote from another teacher depicts a similar attitude of anticipation, but also seems to convey a story of how to approach change with composure, an ability that is

linked to the teacher's professional adequacy, built by knowledge and understanding of the subject content and teaching experience:

... we have not come that far [in our preparations] either, but think it is going to work out fine [*easy laughter*] ... but as you say, it feels exciting and interesting.

Despite being late with preparations, the pressure to enact a new curriculum seems not to turn into stress. There are, however, concerns raised that the implementation process will increase the workload of the teachers, that a crowded curriculum gets even fuller and that there is a question of what they are "supposed to omit".

Given the particular roles these teachers have at their respective schools, educational change seems to be a generally accepted phenomenon, but the continuous stream of innovations and initiatives is something that these teachers perceive as disruptive:

...there is this school digitalisation initiative, the mathematical boost, the reading boost... this is what I think, something has to be removed...

Another topic raised in these conversations revolves around programming as an instrument that can help create more varied teaching and learning situations. They also discuss if such activities can be beneficial for students who may be struggling with motivation or concentration problems by providing a more multimodal and hands-on way to explore a particular subject content. Another view is that programming is not necessarily a subject content or tool exclusively for mathematics and technology, but something that can enrich other school subjects too.

The attitude towards programming as a knowledge domain is also briefly made visible in the conversation of the grade 1-3 teachers. At this stage in the preparation process, they convey a tool-centric, rather than a competence-based, view towards programming in the curriculum as they touch upon the progression from unplugged programming (i.e. learning programming away from computers) to block-based and eventually text-based programming (both in later stages of compulsory school) and conclude that text-based programming feels complicated and also that the occupation as programmer requires expert knowledge.

Issues related to ethical considerations surfaced in one of the groups' conversations when they discussed possible consequences of future technological development for, for example, old-age care. This is an aspect that is briefly mentioned in the syllabus for technology in general terms but was not made visible in the preparatory work that preceded the reform.

Teaching approach (I4)

As described in the section about attitudes and values, some teachers believe that children can be motivated by using interactive digital tools and are occasionally using the online platform Code.org⁷ to create variation and explore computational concepts:

So, for them it is just to grab a computer, I have the computer cart outside [the classroom] ... then they can work for 20 min at Code.org, all by themselves. I don't even have to prepare anything...

⁷ Code.org is a free-to-use, industry supported, online learning platform targeting school children and teachers globally with the aim to expand access to computer science in schools.

The quote above also reveals how the use of digital tools can free up teacher time, but self-facilitated work can also come with a cost, as addressed further under Practicality of implementation (E3).

Most of these teachers seem to have a pragmatic approach to teaching which, in combination with solid teaching experience, allows them to further their knowledge and understanding of this reform by both starting out early as some of them have done, or by not worrying over having to take risks and construct their understanding during actual classroom implementation, alongside the pupils.

Several teachers have chosen to precede the enactment of the new programming policy and initiated small scale class projects in order to find out more about what the new policy may mean for them. Most of these initiatives seem to be experimental as illustrated by the following quote:

it [programming] has been addressed to a small extent, but rather unstructured, we have tested a little here and tested a little there, testing different programs as well, but there is no structure in teaching with programming...

These discussions also reflect doubt in how these exploratory initiatives are designed and conducted and that they are making “tentative attempts with funny things” and introduce the children to “gimmicky events”, that also risk becoming one-time happenings. It is recognized that what is fun for the teacher is not necessarily fun for the student. One teacher self-reflectingly describes experimental classroom work in a negative way that reveals a sense of discomfort in launching unreflected activities where ‘fun’ seems to be the guiding principle:

... we have been working with programming, but it is like you take something that you think is good, and then you try it a little, we run these robots, then we try something else, but really, it has been more about picking what I personally think is fun and try to start somewhere...

This view is confirmed by another teacher who adds that more experimental approaches may result in learning activities where the children may fail to understand the aim of the activity:

Well, I think it is great fun to use digital gadgets, but sometimes when you use these things, the children do not really understand what they have been doing

According to Finger and Houquet (2009), teachers have to be flexible and willing to adopt new and modify current teaching strategies. The curriculum specifically states that “exploration, curiosity and desire to learn should form the foundations for school activities” (Skolverket 2017b) and there is no contradiction between entertaining activities and well-designed learning situations. However, the activities must pass beyond the state of being exploratory (for the teacher) and ‘fun’ one-time events, into also being aligned with curriculum intentions. This challenge is interrelated to and dependent on other challenges, for example, professional development and understanding and resources. This finding indicates that as long as such related challenges remain unsolved, there will be an increased risk that the programming related activities will be few and fragmented. Something that in turn may slow down the development of teachers’ professional adequacy and ownership of the reform, and as a consequence, the entire programming policy.

Ownership (I5)

Ownership is another intrinsic challenge that can considerably affect the uptake of reforms. Studies have shown that a feeling of ownership of the innovation is built up over time and more easily grows when teachers are included in decision making processes. Thus, simultaneous bottom-up strategies in top-down reforms facilitates teachers' ownership of a reform (Careless 1997; Fullan 2001). There are several findings in this data set that can be of importance for how these teachers construct their ownership of the innovation. Many aspects presented under the sections of teaching approach, professional development and support and time management influence the growth of ownership. These teachers recognize and accept the rhetoric of the reform and seem also willing to commit to a period where exploratory classroom work and professional development may go hand in hand.

Extrinsic challenges for the implementation of programming

Resources (E1)

Much time during the conversations was spent describing and discussing practical classroom work related to the questions of what to do, how to do it and what kind of resources to use. The teachers also discussed programming in the wider perspective of school digitalisation and how quality aspects of the school IT infrastructure greatly affects the uptake of information technology.

Teaching materials The teachers who have initiated programming activities in their classrooms have done so by using online learning platforms such as Code.org and Scratch or tangible computing devices such as educational robots (Bluebots and Lego Mindstorms) and MakeyMakey. The Swedish children's television show *Programmera mera*⁸ has also been used as a starting point for classroom activities. Code.org seems to be used by a few teachers and is appreciated for its streamlined step-by-step introduction into computational concepts:

In Code.org, you are presented to different concepts as you progress, you do not see all the features and that is really good... I would definitely recommend it because of the way it is designed, you learn one concept at a time, and then Scratch as a next step...

However, one of the teachers that used Code.org also discovered that the children tend to skip through content and exercises in a way they will not do while working with traditional textbook-based assignments:

There are many [children] who skipped directly to Flappy [Birds], without taking the steps in between...[...] they follow some of the steps in the first [task], some steps in the second, and then... awesome! Flappy Birds!... [...]

This teacher concluded that this jumping problem solving strategy leads to a situation where the children may fail to grasp the main principles of the learning sequence. It is not only the introduction of a totally new knowledge domain that complicates things for the teachers. The introduction of new tools also seems to challenge teachers' ability to stay

⁸ Programmera mera (Program more) is a children's tv show targeted for children between ages 6-9, produced by the Swedish Educational Broadcasting Company (UR), <https://urskola.se/Produkter/196673-Programmera-mera/Visa-alla?cmpid=del:cl:20190601:urskola>.

ahead and adapt to the teaching situation, calling for flexibility in their teaching approach. Being aware is a first step in finding solutions to these situations.

However, there is an implied request for pre-made teaching materials to use and relate to while constructing an understanding of what the new programming policy may mean for themselves and for the pupils. They also recognize a current lack of teaching materials from established textbook providers but expect that this is a situation that will change once the next semester begins. At the same time, they also conclude that new textbooks do not come without a cost and that there is also an ongoing discussion in school that concerns strengths and shortcomings of digital textbooks.

A teacher in one of the groups pitched the idea to initiate a shared cloud-based workspace where they could share teaching materials and other useful resources that may help construct an understanding of how to approach the new policy.

IT infrastructure The last two decades, various initiatives to extend and improve the digital infrastructure in Swedish schools and/or the digital competence of teachers have been launched. During this period, one laptop per student initiatives (1:1) have become increasingly more popular amongst schools who are trying to keep up with national school digitalisation policies, not only in Sweden but worldwide (Tallvid 2015). While the use of computers can enable and augment learning situations, teachers unfortunately also have to get used to malfunctioning and low capacity IT infrastructure (Håkansson Lindqvist 2015; Skolverket 2019a; Tallvid 2015). The results of this study are no exception as the teachers describe facing problems with hardware, software and network connections. One teacher illustrates how working with the popular educational robotics kit from Lego (Mindstorms) can become an annoyance:

... there were batteries that discharged, and there were synchronization problems between robots and computers, and some were not local administrators on the computers which meant that they were unable to install the necessary software and it is things like this... and then we had no network connection... and yeah you know... This is what I think has been most problematic, the rudimentary things you expect will work, otherwise... it drains your energy.

Some schools have classroom sets of laptops in carts that teachers can book in advance. One teacher has nearby access to the cart and has found a routine to book the cart once a week for 20 min of self-directed learning at Code.org. Another teacher is very satisfied with the school computers and describes a more or less problem free situation, while several other teachers share a fundamentally different experience from working with their carts:

We are six classes that share one laptop cart with 24 computers, but right now, several are broken, and the largest class has 24 children [...] they are so bad, buttons are missing, and they are not being replaced [...] we were promised new computers a year ago ... it makes you wonder what to do in the mean time?

Engaging with the computers is described as tiring and cumbersome, a “hurdle to overcome”, “resource demanding”, and as something you use only when there is a teaching assistant present, to avoid having to “run around and fix things the entire lesson”.

In 1:1 schools (lower secondary schools) the problems seem to be fewer and also a little less obtrusive for the individual teacher. What may happen is that the children either forget or choose to leave their computers at home. This is a phenomenon that is believed to

happen on days when there is physical education and the school bag “gets heavy” because of clothes and other equipment needed for the lesson. It becomes very clear that there are severe problems with the IT infrastructure at several of these schools and that this is hampering the ambitions and motivation of the teachers. As a consequence, teachers choose to avoid using the IT equipment, since they believe it cannot be trusted to work as intended. Another challenging situation that may arise is when new investments are imminent. A teacher describes a situation from one of the schools where a team of teachers have been involved in the process of identifying and evaluating appropriate hardware. The teacher depicts the process as a “struggle” between the teacher team on the one hand and the municipality IT Office on the other.

Monetary resources All example activities seen in this data sample are dependent on access to hardware. The school computers are used to interact with free-to-use online services such as Scratch and Code.org and some schools also have access to educational robots for teaching activities. Most schools assign a certain amount of money to each subject which is meant to cover costs for subject specific needs and teaching materials over the course of a school year. While costs for a school’s IT infrastructure, such as school computers, normally are carried by the municipality and not by individual schools, costs for any additional teaching resources, such as educational robots, will be charged to a subject’s yearly, and tight, budget. One of the schools represented in the data set financed their purchase of educational robots using a scholarship. The following quote illustrates the traditionally weak position of the technology subject in the Swedish school system. Furthermore, it also reveals a situation where teachers have to make specific requests for investments:

You know, we got... we had 0 SEK... when I started working at my school there was no technology institution, it was an integrated part of the science institution... so when the proposal came [from the government] to disconnect technology from the science subjects [and earmark 200 study hours for the technology subject], we just said [to ourselves]: we need money! Then we had enough to buy the Lego robots and after that we went to the school district manager to request an additional amount of money to make further investments...

The episode above seems to have turned out well but indicates that the outcome was dependent on the teacher’s personal interest for the technology subject and/or tangible computing devices in combination with the ability to argue for the subject’s specific needs. It also seems as if there was a certain amount of economic latitude in this particular school district when this episode took place.

Time management (E2)

Time is a challenge that greatly affects the extent to which a curriculum implementation is successful and emerges as a key theme in this study. There is a perceived shortage of time that inhibits the “time consuming” process of finding and evaluating teaching materials. One teacher describes a situation where there is an activity example available, but lack of time restricts to what extent the teacher can evaluate the quality, purpose and adequacy of the activity. The teacher feels hurried into a situation of action, something that seems to be unsatisfying:

... I just grab this sandwich lesson... but you would like some time to assess, what is this [good for], is it something [useful]...

The perceived lack of time for collegial interaction and “no time to reflect” is a feeling that seems to be shared by several of the teachers. There are concurrent professional development initiatives at the schools, but “too few in-service training days”. Such days are an important resource for the teachers to catch up and plan ahead, both individually and in teams:

... it is going to be difficult... you really have to find time to work together... it is a pity ... but sometimes it is that time aspect... or that you are split on too many things so you don't really know what you are doing...

One teacher explains that, despite having monetary resources to make new investments for the technology classroom, almost 4 months have passed, and the new equipment (3d printer) remains unpacked. It is not only about unpacking; time is also needed for setting up the equipment and to learn how it works and can be used.

Another conversation that concerns temporal aspects of the new reform is about whether the inclusion of programming means that they should spend less time with other core content or if programming can be beneficial for the understanding of other core content. One of the teachers who already has begun to explore how programming can be implemented in teaching shares her perspective on the issue:

Of course, you would like more time, but there are lots of things you can bring in, I can tick off things from technology... mathematics for instance, and also in geography where we had a really nice exercise, I think they learned more and faster than what they would have done otherwise...

The teacher behind the quote above is willing to explore different means and methods to approach core content and, as a middle school teacher, appears to have access to an arena that allows her to explore possibilities for subject integration. The quote also pinpoints an important aspect of programming in the compulsory school curriculum: What is the intended purpose of the reform and how is it interpreted by the key policy enactors, that is, the teachers?

Practicality of implementation (E3)

For these teachers, who have change embedded in their work descriptions, the practicality of the implementation may be closer at hand than for their colleagues, but at the same time, as shown in this study, these teachers also face several challenges related to, for example, IT infrastructure and professional attitudes and values, that affect the practicality of implementation. One teacher concluded that the situation calls for a period of acclimatisation and that putting too much time and effort into an early stage may be counterproductive. Instead, it would be better to let the magnitude of the reform grow over time, as a consequence of professional development and increased access to teaching materials:

It feels like it has to be like this for a while, you do some testing and things will grow from there, it may be unwise to start with something too structured, and go too deep... maybe you put down quite a lot of work without having the knowledge to be able to do the job well.

Even though it may sound reasonable, there is also a risk that such an approach may contribute to the delay of a reform or that it will never be considered practical enough. An incremental growth does not happen by itself. School-wide approaches supported by active

and initiated leadership are more likely to become successful than initiatives which rely on individual initiatives (Treagust and Rennie 1993).

Another aspect related to the practicality of implementation revolves around issues of learning progression through the education levels:

Independent of what level you are at in the education system, all start at the same spot... no matter if you are in lower secondary school, or primary school, most pupils have no experience of programming... this means that all [pupils] start the same way...

How do you take this further; we are currently looking at the learning progression... what to do in different grades, what teaching materials should we have, what is missing?

This is a situation they expect will change over time. One teacher elaborates on what the potential consequences may be of having to welcome new children who have some programming experience from primary school:

... I cannot do the same things with them when they come to me as of 4th grade, they would know these things already... that will be untested waters for me, I have never had pupils who knows something I have not taught them... about programming ... but it will be very exciting and it will also affect lower secondary teachers and pupils [in a similar manner]

The tool-centric view on programming once again becomes visible during conversations about learning progression. It seems more urgent to talk about the age appropriateness of certain tools rather than how a learning progression of programming practices and concepts can be constructed.

Student assessment (E4)

The teachers very briefly touch upon potential assessment challenges. One teacher's feeling of discomfort for not having control over the teaching situation (see Professional adequacy) seems to be partially caused by uncertainty about the assessment procedure. Another teacher proposed that the quality of the program code can form basis for assessment:

But programming, whether the code is smart or not can be assessed... If you use a loop instead of putting 38,000 commands in succession... Then it is a sign that you have understood, then you can assess...

Another topic for discussion during these conversations illuminates a perspective on what skills and abilities teachers may associate with programming activities. While describing a situation of classroom work, one of the teachers argues that programming activities with educational robots in the technology classroom fosters a more open and natural systematic trial-and-error approach towards problem solving. This teacher also believes that the immediate feedback the robot provides by showing how it moves facilitates the assessment procedure:

I can immediately see if the Lego robots... [...] it does not follow the planned route...

Another teacher in the same conversation group too recognizes the value of immediate feedback and adds the notion that programming can be a catalyst for children's

communicative skills during mathematical problem solving processes. This is a skill that this teacher believes may be beneficial for and transferable to several subjects and “not only for the math class”. The same teacher illustrates further possible synergy effects with a parable about how the process of writing a text is similar to writing effective code. There are often ways to improve a text and, for example, make it more coherent, clearer and shorter, and that this can be compared with using repetition statements in program code. This line of reasoning around optimisation fits well with one of the knowledge requirements (assessment criteria) of mathematics, which emphasizes the ability to choose and apply effective mathematical methods.

History and tradition (E5)

The kind of tradition a certain school or a municipality leans against to meet educational change can affect the uptake of this reform. In Sweden, no compulsory school has a history or tradition to fall back on when it comes to programming. The reform from early 1980's came and went without making any considerable impact (Riis 1987) and there are few teachers from that period who are still working today. Furthermore, the history and tradition of the technology subject itself is problematic (Skolinspektionen 2014). More than 50% of the active technology teachers lack formal eligibility and the need for professional development is extensive.

In this study, the lack of history and tradition becomes evident through the findings presented for several intrinsic and extrinsic challenges. There is a telling quote under the section for the challenge of professional knowledge and understanding in which one teacher notes that “there is nothing to fall back to, you have no personal experience of it at all”.

Also when it comes to the history and tradition of Swedish school digitalisation initiatives, the situation is fragmented (Skolverket 2019a). There are unequal conditions regarding access to and quality of the IT infrastructure and this also becomes evident in this study, see the Resources section for details.

Professional development and support (E6)

Professional development opportunities and school leadership In Sweden, municipalities are the main organizers of public schools, but school districts and individual schools have a certain amount of autonomy to design local school policies, for example, for professional development. There are already ongoing and concurrent professional development initiatives (e.g. about sustainable development, inclusive pedagogy and co-operative learning) at the teachers' schools and the conditions seem to vary between different schools, and also within schools. Some of these initiatives are initiated at a local level while others are municipality-driven efforts (e.g. learning about sustainable development). For some teachers, the adding of another focus area, digitalisation and programming, makes it more difficult to keep track and understand what goals they are currently committed to. A theme that emerged in the conversations was the lack of clear and firm directives from the school leadership. Some of the teachers did not know when and under what conditions they will get the opportunity to learn more about programming and they are unable to see a long-term strategy for professional development, as illustrated by the following quotes:

they need to be clearer, the principals... let's remove this... they could say, and then we add this [instead]...

You mention [the course] *About programming*, I have heard others who have taken this course already and you get this feeling that you should have done it long ago... Now we're being told that, well, maybe we'll do it in the fall, but I believe that you should have been there already ... to get a good start.

The second quote above is revealing for a situation where the school leadership is unable to provide professional development opportunities in a timely manner. This study presents no evidence to what extent and why such situations occur, but a contributing factor could be that the leadership too "may have lots of expectations from above to satisfy...", as suggested by one teacher, and therefore are short on both time and resources. Some teachers argue that a top-down municipality-driven approach would have been better for this particular situation where all teachers would have studied the introductory course simultaneously, for better impact. There are also teachers representing schools where the decision has been made to focus on a few selected development goals:

... if you would have asked someone at our school a year ago, no one could have mentioned a single [improvement] goal, we have had so many goals and if you wanted to dedicate a staff meeting [to one specific goal], then it would have been every 10th meeting you would work with... [the same goal] [...] so we decided to work with sustainable development and inclusive pedagogy alone...

Making explicit decisions to limit the number of concurrent development goals may, however, bring other consequences in times of change:

...we have got started with [...] co-operative learning throughout F-6,⁹ it is because one of the principals is pushing strongly for co-operative learning [...] which means it is difficult to find time for something else...

This quote illustrates a professional development organization that is focusing a single goal and where personal commitment amongst the school leadership can result in certain commitments getting more attention. However, as an effect, it may mean less flexibility for schools to make rapid adjustments if and when necessary. A similar situation that shows how personal interest can affect investments is described in the following quote:

It may depend on what principal you have [...] our principal is really interested in digital learning, and he has spent lots of money on this [...] primary school has iPads...

The strategies for professional development also seem to differ within the same school depending on what grade you teach. The following quote reveals two different approaches to study the introductory course *About programming*:

4-6 [the middle school teacher team] is really good, they read, do some course modules and then they use the next teacher team meeting [to discuss the course content] [...] 7-9 [the lower secondary teacher team]... I am part of this team... we read freely... we do not have [any scheduled meetings for this particular course]... now I feel I have to try to make sure that we can have at least one check-off meeting

⁹ F-6 are all grades from F (pre-school class for 6-year-olds) to grade 6 (12-year-olds).

This teacher suddenly seems to realize that there are opportunities for collegial interaction and support that are being wasted and feels obliged to make an effort to improve the situation for the grade 7–9 teacher team.

At the same time, at another school:

...we study at the same pace [...] and [after completing the introductory course] we will try to divide us into different tracks [depending on subject] ... we are doing this in a 4-9 constellation and, what should we do together...

This teacher describes a more coordinated effort to approach the consequences of the new programming policy. The teachers at this school do their initial development work as a team regardless of whether they are mathematics or technology teachers and what age group they teach. This way of organizing in-service training is probably more efficient in achieving coherence both in terms of teachers' understanding of programming and how the new content is integrated in teaching activities.

To summarize, the findings around in-service professional development opportunities describe a fragmented situation. Some have studied the introductory course while others do not yet know when they will be given the opportunity to do so. It seems as if being part of the same municipality or, even, school is no guarantee for equal access to professional development.

Support structures There are several ways in how support can be provided to teachers. The teachers in this study have used different support structures in order to help them realize their intentions. Some of the teachers have initiated programming activities at their schools thanks to pedagogical and technical support from a local ICT teacher:

there are many [teachers] who have organized it in such a way that he has visited their classes to work with programming and these teachers have received some professional development at the same time...

While the situation at another school is such that the teachers in grades 1-6 no longer have access to an ICT teacher who can provide support. The school still has one ICT teacher, but he is tied to grades 7–9 and therefore not available for the lower grades, despite their efforts. There is also a school where the local IT technician is considered a teaching resource who can step in and help teach programming.

A few teachers have had the opportunity to bring their classes to a nearby Science Centre which offers school classes ½-day workshops about introductory programming with educational robots. They appreciate the convenience of arriving to a set table where “everything is setup and ready” and meet Science Centre personnel who are perceived to be both competent and committed. One of these teachers is thinking about the possibility of outsourcing curricular core content of the technology subject (regarding control and regulation) to the Science Centre and thinks that such a visit also may be enough to cover the element of block-based programming:

I think it feels quite... the part about controlling objects, that you should control something, a physical device, it is something that is included in the technology subject, why not go to the Science Centre and get it covered, and afterwards you tick it off on the list.... Then you can spend more time thinking about text-based programming activities in your own unit... That is how I think anyway... I don't know...

It is an explicit ambition of the school authorities that all schools should have access to well-functioning, high capacity IT infrastructure and adequate, competent and timely support. The findings of this study indicate a situation where access to pedagogical IT support (e.g. from an ICT teacher) is unequal amongst the schools. The situation seems to be similar at a national level, as shown in a recent report from the NAE (Skolverket 2019a).

For those schools who can provide their teachers with ICT support, the introductory phase may turn out to be a little easier. However, there is also a risk that such schools may rely too heavily on the ICT teacher or an external resource to do the teaching. This in turn could lead to fewer programming activities and also to activities that are only loosely linked to the aim and core content of the subject. It could further undermine the intentions of the entire professional development initiative, and, as a direct consequence, the implementation of the programming policy. In addition, there is no particular evidence available showing that ICT teachers are better prepared to learn and teach programming than a regular mathematics or technology teacher.

Discussion

This study explored the following research question: *What challenges for integrating programming are expressed by in-service teachers?* This was done by analysing a group of teachers' conversations about an upcoming policy change which introduces programming as core content in mathematics and technology. By doing so, the purpose of the study was to learn about what challenges teachers perceive that this particular educational change will bring about, in line with recommendations from Hargreaves (2005) and other scholars. The data collection took place less than 4 months before the enactment of the new curriculum. This study has identified challenges that represent real concerns for these teachers, and how and to what extent these challenges are handled will affect the implementation process. Some of the teachers in this study are likely to be the ones who, to some extent, are expected to 'pave the way' for their colleagues, either through their own teaching, by providing collegial support and/or by driving subject development which concerns the implementation of programming.

Intrinsic challenges

The necessity of having a solid and adequate knowledge base as a teacher is unquestionable. Shulman (1986) concludes that such a base of professional knowledge and understanding (II) should cover both content knowledge, pedagogical knowledge and pedagogical content knowledge. The pedagogical content knowledge is the particular kind of knowledge a teacher needs to teach a specific subject content and implies that the teacher is well-acquainted with both various teaching methods as well as what learning challenges a specific subject content may convey. The teacher is thereby able to interpret and transform content into subject matter for teaching and adapt to the needs and abilities of a particular student or group of students (Thornton 2001). This study shows that the 2017 programming reform presents teachers with a challenge that makes them question their own level of professional knowledge and understanding, something that evokes a feeling of uncertainty. This uncertainty, which may even lead to borderline discomfort, emerges as a consistent theme throughout this study.

The challenge of professional knowledge and understanding is closely related to the challenge of professional adequacy (I2), i.e., the teacher's belief in her ability to meet the requirements for being a teacher in a satisfactory manner. An insufficient or fragmented knowledge base undermines the chances of obtaining a good level of professional adequacy. Furthermore, professional adequacy is interlinked with a teacher's professional attitudes and values (I3). The study shows that there is a general consensus amongst the teachers in this particular group that the programming reform is relevant and timely from a societal perspective. The overall attitude towards the change in this group is positive, which may seem reasonable given the fact that these teachers carry a certain responsibility for development and/or support at their respective schools. This change prone attitude may prove to be valuable for the uptake of programming. Nevertheless, as this study shows, there are several concerns raised amongst this group of teachers about how to initiate the reform, with only a few months to go before the actual enactment of the new curriculum.

The study also shows that there is a willingness to try amongst these teachers and to explore both programming as subject content and appropriate teaching approaches (I4) during actual teaching practice. However, the results also reveal hesitancy towards a situation where teaching activity designs are based on more or less arbitrary decisions or on what they find entertaining, instead of being guided by the aim and goals of the subject.

Furthermore, it may also be relevant to reflect about what the constituents of a solid and adequate programming knowledge base are, given the particular context. What is a reasonable level of professional knowledge and understanding about programming for, for example, a lower secondary school teacher, in order to teach the subject content with agency? How can it be ensured that in-service training and national teacher training programs cover the broader aspects of teacher knowledge necessary to teach with and about programming (Reding and Dorn 2017) and not only focus on the content knowledge domain?

Extrinsic challenges

The results from this study reveal a request for pre-existing teaching materials (E1) which can help teachers deliver what they perceive is expected from them. At the time of study, these teachers tend to be tool-centric in their approach to programming and primarily prefer to talk about different resources and tools and how these may be used. Teachers continuously rely on pre-made teaching materials to some extent in many subjects and the process of evaluating and adopting existing teaching materials for classroom use is an essential part of a teacher's curricular knowledge (Shulman 1986; Thornton 2001) and also part of the teacher's own learning process (Falkner and Vivian 2015). The new programming policy demands the same kind of teacher agency, but without proper professional development in place, this may not be a reasonable expectation. Previous research has shown that teachers "typically plan by concentrating on the content they intend to cover" (Brophy 1999, p. 13) and pay less attention to goals and learning outcomes. This aligns with the critique directed to the technology subject that illuminated the risk for unreflected use of pre-made teaching materials (Skolinspektionen 2014).

Pre-made material is traditionally curated and provided by educational publishers but the perceived lack of curated materials (e.g. textbooks) have made teachers and schools to use generic 'coding' resources. There are a plethora of services and instructional materials available online that claim to help children develop programming, coding or computational thinking skills. These services are sometimes free to use (e.g. Scratch or Code.org) or available at a cost. Falkner and Vivian (2015) have conducted a semi-systematic review of

the quality of online computer science resources from an Australian context and conclude that there are concerns over the completeness of resources, many focus on the narrower notion of “coding” whereas the Australian, and also the Swedish curriculum embrace a broader view of computing.

Some of the schools represented in this study have had the opportunity to invest in tangible computing devices, such as educational robots, and there seem to be an active interest amongst teachers to approach the programming content with support from this type of digital artefacts. Such devices can offer the user an immediate and kinaesthetic feedback which is considered to be advantageous (Ben-Bassat Levy and Ben-Ari 2018; Elkin et al. 2016; Goadrich 2014; Shute et al. 2017). However, there may also be reason to retain composure and make sure that any investments made are well-considered. The quality of resources, both tangible and online, can vary widely, and some of the online services serve the interest of commercial stakeholders, even though the service itself may be free to use. Schools have become a market for corporate information technology providers and many claim they can provide the ‘right’ solutions and hereby become contributing parts in shaping both practice and school policy (Player-Koro et al. 2017).

For a teacher, who has limited training and experience, and perhaps only a moderate interest in programming, it may be a daunting task to identify and evaluate teaching materials that can be used in conjunction with the perceived intentions of the Swedish curriculum. The perceived shortage of time (E2) for this particular teacher task adds to the complexity. As a result, there is a risk that teaching activities are modelled after what the tool can provide rather than a teacher’s careful curricular considerations. Such approaches may consolidate well-known problem areas of the technology subject, such as the occurrence of perceived misalignment between activities and aims and teaching that fails to capture the breath and unique characteristics of the subject (Skolinspektionen 2014). A study by Player-Koro (2013) suggests that despite new digital tools, there is a risk that they will be used in such a way that they consolidate more traditional ways of teaching. Some indication of such a risk was seen in this study.

Another key factor that can either enable or impede the programming implementation that becomes visible in this study is related to the quality and inequality of the IT infrastructure in schools (E1). The lack of access to adequate and functional IT is something that can severely affect a teacher’s ambitions and motivation to engage in technology dependent activities such as programming. As seen in this study and described elsewhere (Skolverket 2019a), there is an ongoing need to maintain and improve the IT infrastructure of schools and new needs seem to emerge as a consequence of the programming reform. Part of the IT infrastructure consists of the supportive human resources schools and teachers have access to. This study points to a situation of inequality between schools (Skolverket 2019a). Some schools have access to dedicated ICT teachers, others have relied on a science centre or a local IT technician while others lack local support. Programming is supposed to be an integral part of both mathematics and technology. However, it is reasonable to believe that there is an increased need for external teacher support as a direct consequence of the programming reform. This will be one of the key factors that affect both quality and uptake of the reform.

Even though programming has been added as new core content to the syllabi for mathematics and technology, the knowledge requirements from the previous curriculum remain unchanged. Neither the curriculum nor any related policy documents, such as the subject specific commentary material, reveal much about possible consequences for the assessment process (E4). This study shows that there are some concerns about assessment, but this is not addressed to any great extent in the conversations. This may possibly be explained by

the fact that assessment often takes place at a later stage in the process of teaching and learning, but on the other hand, a teacher should preferably have a solid plan for the teaching activities, and such plans should, among other things, also address assessment means and methods.

The fragile history and tradition (E5) of programming in compulsory school education and of the technology subject as a whole adds further uncertainty to the outcome of the reform. Elements of computing and computer science have previously been integrated in the Swedish curriculum. Experiences from the eviscerated reform of the 1980s (Datalära) should be of relevance for today's stakeholders. Research has shown that the reform was hampered by complicated technology, immature software and a lack of responsiveness to the professional development needs of teachers (Farkell-Bååthe 2000; Hylén et al. 2002; Riis 1987, 1991).

Together with Resources (E1), issues related to Professional development and support (E6) stands out as particularly urgent to talk about. The study illuminates that these teachers recognize the need for, and welcome professional development initiatives, but that conditions for professional development vary both between and within schools. The formal enactment of the new curriculum is only a few months away and a sense of inadequate preparation is present amongst the teachers. Several teachers expect their school leaders to take control of the situation and prioritize and organize professional development. School leaders are important actors in reform implementation (Winter et al. 2017), but they too need professional development in order to lead digital transformation (Håkansson Lindqvist 2019).

The school authorities have made an effort to ensure that this reform is supported by offering a selection of teacher training courses. Introductory programming courses procured from higher education institutions are offered since fall 2017. The starting point for most teachers, the 16-h online course *About programming* mentioned in this study, opened early 2018, 6 months prior to the formal enactment of the new curriculum. This effort to support professional development signals an awareness that marks a distinct difference from the reform of the 1980s (Datalära) and may prove beneficial for the uptake and longevity of the 2017 reform. However, a thick course catalogue is not helpful if the conditions for professional development are insufficient. Research reveals a bothersome situation where many teachers have to invest their spare time in order to pursue professional development needs (Parding et al. 2018).

Limitations of the study

The data collection method applied in this study with unmoderated focus groups meant that neither conversation topics nor the depth of conversations could be controlled. A more regular focus group study design in which the researcher participates and moderates the discussions could have been helpful in directing the conversations to issues that were not addressed during the conversations. For instance, neither group came to talk about how the programming content is represented in the formal written curriculum and what guidance and support the curriculum may provide. A potential strength with the unmoderated setup could be that it allows the teachers to express themselves more freely and that you, as a researcher, get a deeper insight into their 'inner feelings'.

Another limitation of the study is the sample of teachers. All participants display an active interest in professional development by taking part in the NAE funded professional development program. All also hold a key position at their respective schools with a certain

responsibility for collegial support and guidance, or subject related development activity. This is a signal that these teachers carry a particular interest for the subjects they teach and/or for educational development work. On the other hand, these teachers are spending most of their time performing the same duties as their colleagues. The time allocated for the first teacher position is usually limited (Skolverket 2015) and the questions and challenges this group of teachers identify may well coincide with those of their colleagues.

Implications and future directions

The challenges that have been identified through this study represent aspects that have occurred during the conversations. There has been no intention to represent the challenges in a quantitative manner, instead the motive for the study was to shed light over a challenging situation and how teachers react while constructing an understanding of how to address new curriculum content, programming in this case. Given this background, there may be a reason to make a follow up study with the same group of teachers to see if there has been any progress with respect to the challenges highlighted in these conversations. Has the NAE course *About programming*, and any other professional development initiatives, provided more input about how to make sound didactical designs with respect to mathematics, technology and programming, or are the teachers still unsure about what they are expected to do and, if so, why? There may also be a reason to investigate whether there is a gap between governmental expectations as expressed, explicitly and implicitly, in the curriculum and what is reasonable to expect from teachers, given the circumstances evident in this study.

During the conversations, the teachers did not discuss programming related content of the syllabi for either mathematics or technology in any detail. A reasonable explanation for this is that most of these teachers did not have enough subject matter knowledge or experience to embark into more detailed discussions. One possible direction for future research could be to study teachers' interpretations of the curriculum core content related to programming and explore how these interpretations shape lesson-planning, teaching and learning.

The overarching push for school digitalisation implicates an increased use of digital learning resources such as digital textbooks, digital tools and online resources. These conversations reveal that digital material is making more and more impact in the school system. Remillard and Heck (2014, p. 717) concludes that "Understanding the impact that the new instructional materials market is having on the curriculum policy, design, and enactment system is a critical focus for research".

We also need to know more about under what premises teachers design their teaching activities and what kind of instructional practice and learning goals that appear in these activities. This is a key component to construct a better understanding of what teachers believe the children should learn about programming, why and how. A parallel approach would be to explore what kind of knowledge children construct from participating in programming related teaching activities and how this knowledge aligns with the intentions of the teacher and the interpreted curriculum. The clarity of the programming message conveyed in the intended curriculum may need to be examined more closely. This could be done by examining how teachers interpret the curriculum programming policy and through a document analysis of education policy documents.

The Swedish programming reform is one of several examples of international education policy initiatives aimed at strengthening school children's digital competence.

Similar reforms are already enacted (European Commission/EACEA/Eurydice 2019) or underway (e.g. South Africa, Norway). Studies of England's computing curriculum reform from 2014 identify implementation challenges similar to those revealed in this study, for example related to resources (time, support, technical infrastructure) and teachers' level of professional knowledge (Larke 2019; Sentance and Csizmadia 2017). As a consequence, teachers may choose to neglect parts of the new curriculum content (Larke 2019). While it may be difficult to draw in-depth conclusions when curricula and country contexts differ, these reforms share common features and challenges, and possibly also solutions.

Concluding remarks

The study presents an insight into a series of challenges that the new programming policy may entail for a teacher in mathematics or technology. Furthermore, it discusses some potential consequences of the new programming policy for the technology subject. This information can be valuable for school leaders who aim to improve an implementation process (Hargreaves 2005), for studies of reform implementation related to digital competence and reform studies in general. The study has captured challenges identified by teachers prior to the implementation of the reform.

What is notable in this study is that, with less than 4 months to go before the enactment of the new programming reform, there is a distinct feeling of uncertainty present amongst these teachers. Despite being appointed the responsibility for collegial support and/or subject related development, the conversations reveal that also teachers in such positions may carry the feeling of not being sufficiently prepared, even though there are also early adopters in the group who have decided to start in advance on their own. This clearly indicates that the process from decision to implementation has been forced and that schools have struggled to keep up.

Furthermore, the study shows that there is an inequality amongst schools with respect to the quality of IT infrastructure, access to ICT support and professional development efforts and that the teachers expect the school leadership to be more active leaders in time of digital transformation. In order to reach a sustainable level of programming implementation, the challenges and its potential consequences described in this study must be taken seriously. Recognising and acting constructively upon experience from similar past and ongoing reforms is also of significant importance.

Acknowledgements Open access funding provided by Umea University.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Altrichter, H. (2005). Curriculum implementation—Limiting and facilitating factors. In P. Nentwig & D. Waddington (Eds.), *Making it relevant—Context based learning of science* (pp. 35–62). Münster: Waxmann.
- 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.
- Barnes, R. (2005). Moving towards technology education: Factors that facilitated teachers' implementation of a technology curriculum. *Journal of Technology Education*, 17(1), 6–18. <https://doi.org/10.21061/jte.v17i1.a.1>.
- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. Learning and leading with technology. *ISTE Learning and Leading with Technology*, 5191(March/April), 21–22.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking To K-12. *ACM Inroads*, 2(1), 48–54. <https://doi.org/10.1145/1929887.1929905>.
- Ben-Bassat Levy, R., & Ben-Ari, M. (2018). The evaluation of robotics activities for facilitating STEM learning. In W. Lepuschitz, M. Merdan, G. Koppensteiner, R. Balogh, & D. Obdržálek (Eds.), *Robotics in Education* (pp. 132–137). Cham: Springer.
- Billett, S. (2003). Vocational curriculum and pedagogy: An activity theory perspective. *European Educational Research Journal*. <https://doi.org/10.2304/eeerj.2003.2.1.11>.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education—Implications for policy and practice*. <https://doi.org/10.2791/792158>.
- Börjesson, A., Karlsson, L., & Lindgren, L. (2013). Policy making “by the way” in a Crowded Education Policy Space : The case of the Swedish teacher certificate reform, 21(4).
- Braun, A., Ball, S. J., & Maguire, M. (2012). *How schools do policy. Policy enactments in secondary schools*. London: Routledge.
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101. <https://doi.org/10.1191/1478088706qp063oa>.
- Brophy, J. (1999). *Teaching*. Educational Practices Series: International academy of education. [https://doi.org/10.1016/S0167-8922\(00\)80004-8](https://doi.org/10.1016/S0167-8922(00)80004-8).
- Careless, D. R. (1997). Managing systemic curriculum change: A critical analysis of Hong Kong's target—Oriented curriculum initiative. *International Review of Education*, 43(4), 349–366.
- Cheng, L., & Wang, H. (2009). Factors affecting teachers' curriculum implementation. *The Linguistics Journal*, 4(2), 135–166.
- Clements, D. H. (1999). The future of educational computing research : The case of computer programming. *Information Technology in Childhood Education*, 147–179.
- Cohen, D. K., & Ball, D. L. (1990). Policy and practice: An Overview. *Educational Evaluation and Policy Analysis*, 12(3), 233–239.
- David, M., & Sutton, C. D. (2010). *Social research : an introduction*. (C. D. Sutton, Ed.) (2nd ed.). London: SAGE.
- Diethelm, I., Hubwieser, P., & Klaus, R. (2012). Students, teachers and phenomena: Educational reconstruction for computer science education. In *Proceedings—12th Koli Calling International Conference on Computing Education Research, Koli Calling 2012* (pp. 164–173). <https://doi.org/10.1145/2401796.2401823>.
- Doleck, T., Bazelais, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, 4(4), 355–369. <https://doi.org/10.1007/s40692-017-0090-9>.
- Elkin, M., Sullivan, A., & Bers, M. U. (2016). Programming with the KIBO Robotics Kit in Preschool Classrooms. *Computers in the Schools*, 33(3), 169–186. <https://doi.org/10.1080/07380569.2016.1216251>.
- Erlanson, P., Karlsson, M. R., & Karlsson, M. R. (2018). theory and practice From trust to control—The Swedish first teacher reform From trust to control—The Swedish first teacher reform. *Teachers and Teaching*. <https://doi.org/10.1080/13540602.2017.1379390>.
- European Commission/EACEA/Eurydice. (2019). *Digital Education at School in Europe*. Eurydice Report. Luxembourg: Publications Office of the European Union.

- Falkner, K., & Vivian, R. (2015). A review of Computer Science resources for learning and teaching with K-12 computing curricula: an Australian case study. *Computer Science Education*, 25(4), 390–429. <https://doi.org/10.1080/08993408.2016.1140410>.
- Farkell-Bååthe, S. (2000). *Datorn som pedagogiskt hjälpmedel: effekter och erfarenheter av datorstöd i matematik*. Stockholm: Institutionen för individ, omvärld och lärande, Lärarhögsk.
- 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: Teachers College Press.
- Fullan, M., & Pomfret, A. (1977). Research on Curriculum and Instruction Implementation. *Review of Educational Research*, 47(1), 335–397. <https://doi.org/10.3102/00346543047002335>.
- Goadrich, M. (2014). Incorporating tangible computing devices into cs1 *. *Journal of Computing Sciences in Colleges*, 29(5), 23–31.
- Goldenson, D. R. (1996). Why Teach Computer Programming? Some Evidence About Generalization and Transfer*. In *Call of the North, NECC'96. Proceedings of the Annual National Educational Computing Conference* (pp. 144–158).
- Grgurina, N., Barendsen, E., Suhre, C., Veen, K. van, & Zwaneveld, B. (2017). Investigating Informatics Teachers' Initial Pedagogical Content Knowledge on Modeling and Simulation (Vol. 10696, pp. 65–76). <https://doi.org/10.1007/978-3-319-71483-7>.
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the State of the Field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>.
- Håkansson Lindqvist, M. (2015). *Conditions for technology enhanced learning and educational change: A case study of a 1:1 initiative*. Diss. Umeå: Department of Education, Umeå University.
- Håkansson Lindqvist, M. (2019). School leaders' practices for innovative use of digital technologies in schools. *British Journal of Educational Technology*, 50(3), 1226–1240. <https://doi.org/10.1111/bjet.12782>.
- 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>.
- Hylén, J., Andersson, A., Caesar, M., Ekberg, K., Isberg, G., Nydahl, L., & Persson, H.-I. (2002). E-lärande som utmaning. Retrieved from <https://www.regeringen.se/rattsliga-dokument/departementsserien-och-promemorior/2002/10/ds-200255-/>.
- 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>.
- Levin, B. (1998). An epidemic of education policy: (What) Can We Learn from Each Other? *Comparative Education*, 34(2), 131–141. <https://doi.org/10.1080/03050069828234>.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational Thinking in K-9 Education. In *ITiCSE-WGR'14*. <https://doi.org/10.1145/2713609.2713610>.
- Ornstein, A. C., & Hunkins, F. P. (2018). *Curriculum: Foundations, Principles and Issues* (7th Ed.). Pearson.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Parding, K., Sehlstedt, T., Johansson, A., Berg-Jansson, A., & Jakobsson, M. (2018). *Lärares arbetsvillkor i kontexten av marknadsiering, privatisering, val och konkurrens*. Retrieved from <http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-67117>.
- Pea, R. (1983). Logo Programming and Problem Solving. *Paper Presented at Symposium of the Annual Meeting of the American Educational Research Association(AERA), "Chameleon in the Classroom: Developing Roles for Computers."* Montreal, Canada, April 1983.
- Player-Koro, C. (2013). Hype, hope and ICT in teacher education: A Bernsteinian perspective. *Learning, Media and Technology*, 38(1), 26–40. <https://doi.org/10.1080/17439884.2011.637503>.
- Player-Koro, C., Bergviken Rensfeldt, A., & Selwyn, N. (2017). Selling tech to teachers: education trade shows as policy events. *Journal of Education Policy*, 0939, 1–22. <https://doi.org/10.1080/02680939.2017.1380232>.
- Reding, T. E., & Dorn, B. (2017). Understanding the “Teacher Experience” in Primary and Secondary CS Professional Development. *Proceedings of the 2017 ACM Conference on International Computing Education Research—ICER'17*, 155–163. <https://doi.org/10.1145/3105726.3106185>.
- Remillard, J. T., & Heck, D. J. (2014). Conceptualizing the curriculum enactment process in mathematics education. *ZDM—International Journal on Mathematics Education*, 46(5), 705–718. <https://doi.org/10.1007/s11858-014-0600-4>.

- Riis, U. (1987). *Datalära på grundskolans högstadium—Utvärdering av en treårsattsning*. Tema-T Arbetsnotat 36. Linköpings universitet.
- Riis, U. (1991). *Skolan och datorn. Satsningen Datorn som pedagogiskt hjälpmedel 1988-1991*. Linköpings universitet.
- Rolandsson, L. (2015). *Programmed or Not—A study about programming teachers' beliefs and intentions in relation to curriculum*. KTH School of Education and Communication in Engineering Science.
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678–691. <https://doi.org/10.1016/j.chb.2016.08.047>.
- Ryder, J. (2015). Being professional: Accountability and authority in teachers' responses to science curriculum reform. *Studies in Science Education*, 51(1), 87–120. <https://doi.org/10.1080/03057267.2014.1001629>.
- Scherer, R. (2016). Learning from the past—the need for empirical evidence on the transfer effects of computer programming skills. *Frontiers in Psychology*, 7(SEP), 1, 7–10. <https://doi.org/10.3389/fpsyg.2016.01390>.
- Sentance, S., & Cszmadia, 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>.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. Retrieved from <https://doi.org/10.1016/j.edurev.2017.09.003>.
- Skolinspektionen. (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>.
- Skolöverstyrelsen. (1984). *Datalära i grundskolan*, Pub. L. No. Lgr 80, Studieplan 16, 31.
- Skolverket. (2015). *Vad gör försteläraren?* Stockholm: Skolverket. Retrieved from <https://www.skolverket.se/download/18.6bfaca41169863e6a65b742/1553966098154/pdf3423.pdf>.
- Skolverket. (2017a). *Få syn på digitaliseringen på grundskolenivå—Ett kommentarmaterial till läroplanerna för förskoleklass, fritidshem och grundskoleutbildning*. Stockholm: Skolverket. Retrieved from <https://www.skolverket.se/publikationer?id=3783>.
- Skolverket. (2017b). *Läroplan för Gr, Fö och Fri 2011 rev 2017*. Stockholm: Skolverket. ISBN 978-913832691-6.
- Skolverket. (2019a). *Digital kompetens i förskola, skola och vuxenutbildning*. Stockholm: Skolverket. Retrieved from https://www.skolverket.se/sitevision/proxy/publikationer/svid12_5dfce44715d35a5cdfa2899/55935574/wtpub/ws/skolbok/wpubext/trycksak/Blob/pdf4041.pdf?k=4041.
- Skolverket. (2019b). Skolverket Statistical database (Siris). Retrieved May 24, 2019, from https://siris.skolverket.se/reports/rwservlet?cmdkey=common¬geo=&p_verksamhetsar=2018&report=personal_amne2&p_lankod=&p_kommunkod=&p_hman=&p_niva=A&p_amne=18&p_verksform=11&p_hmankod=
- Spillane, J. P., Diamond, J. B., Burch, P., Hallett, T., Jita, L., & Zoltners, J. (2002). Managing in the middle: School leaders and the enactment of accountability policy. *Educational Policy*, 16(5), 731–762. <https://doi.org/10.1177/089590402237311>.
- Straw, S., Bamford, S., & Styles, B. (2017). *Randomised controlled trial and process evaluation of code clubs*. Slough: NFER.
- Tallvid, M. (2015). *1:1 I KLASSRUMMET—Analyser av en pedagogisk praktik i förändring*. University of Gothenburg.
- Tedre, M., & Denning, P. J. (2016). The long quest for computational thinking. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research—Koli Calling'16*, 120–129. <https://doi.org/10.1145/2999541.2999542>.
- Thornton, S. J. (2001). From content to subject matter. *The Social Studies*, 92(6), 237–242. <https://doi.org/10.1080/00377990109604009>.
- Treagust, D. F., & Rennie, L. J. (1993). Implementing technology in the school curriculum: A case study involving six secondary schools. *Journal of Technology Education*, 5(1), 38–53.
- Utbildningsdepartementet. (2015a). *Programmering in på schemat i ny skolstrategi*. Retrieved from <https://www.regeringen.se/debattartiklar/2015/09/programmering-in-pa-schemat-i-ny-skolstrategi/>.
- Utbildningsdepartementet. (2015b). *Uppdrag att föreslå nationella it-strategier för skolväsendet*, Pub. L. No. U2015/04666/S. Retrieved from <http://www.regeringen.se/4a80e6/contentassets/a22b7decc51047a790f68d63c64920cb/uppdag-till-skolverket-att-foresla-nationella-it-strategier-for-skolvasendet.pdf>.

- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., et al. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127–147.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Wing, J. (2010). Computational Thinking: What and Why? *TheLink—The Magazine of the Carnegie Mellon University School of Computer Science*, (March 2011), 1–6. Retrieved from <http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>.
- Winter, S., Kjer, M. G., & Skov, P. R. (2017). *Gør skoleledelse en forskel?*
- Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2017). 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>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.