Teaching Programming in Secondary Schools: Stepping and Stumbling Stones

Majid Rouhani, Veronica Farshchian, Monica Divitini

Norwegian University of Science and Technology, NO-7491 Trondheim, Norway
{majid.rouhani, veronfa, divitini}@ntnu.no

Abstract. Programming is introduced in secondary education in a growing number of subjects. This results in an increasing number of teachers teaching programming in their classes, often without proper training. Learning programming might be complicated, even more so is teaching it. In this context, there is a need to understand teachers' perspectives on teaching programming. This paper aims to identify challenges that teachers in secondary schools face and might negatively impact their teaching, i.e., stumbling stones, as well as elements that promote teaching and give motivation, i.e., stepping stones. The paper is based on the analysis of reflection notes delivered by in-service teachers attending a university-level course on teaching programming. The teachers compile the reflection notes after they complete their final project. Projects are centred around the definition of teaching plans to be tried out in class. The reflection notes of 173 students are analysed to identify issues related to: programming; teaching programming; recurrent didactic issues; and external challenges. The analysis is then summarised in a set of stumbling and stepping stones. For example, time is identified as one of the main stumbling stones by teachers. On the other side, motivation is one of the central stepping stones that we can identify in the data, often connected to the excitement of teaching something that was not previously taught in schools or that teachers perceive as highly relevant for society and the future job market. Implications for teacher training are also identified.

Keywords: Computing Education, Secondary education, In-service teachers, Challenges of teaching programming

1 Introduction

Digital competencies are recognised as an important part of education. In the context of smart environments, students need to gain not only the basic skills required to use digital tools, but also the advanced programming skills that are required to innovate through computer-based tools, as exemplified in e.g., [1, 2]. Different studies show that this training also adds to students' systematic reasoning and critical thinking abilities [3-6].

During the last years there has been a growing interest in offering activities for learning programming, both in informal, e.g. [7], and formal settings, e.g. [8]. As a result of the growing awareness of its importance, this literacy is being included in more and more national curricula.

The research question addressed by this research is: Which are, in the experience of secondary school teachers, the elements that promote teaching of programming, i.e., stepping stones, and challenges that might negatively impact their teaching, i.e., stumbling stones? To answer, we have analysed the final reflection notes of the teachers who participated in a training program at the university level. As a compulsory assignment, over a period of six weeks, teachers created teaching plans to teach programming to their students. Teachers were then asked to deliver reflection notes about their experience with creating the plan, and possibly implementing it in their class. The reflection notes have been coded to identify issues connected to (i) programming, (ii) teaching programming, (iii) didactic issues, and (iv) external challenges. In this way, the analysis is expected to capture a broad understanding of the teachers' experiences in their transition into teaching the new topic. The qualitative analysis of the delivered reflection notes is presented in this paper.

The main contribution of the paper is in providing a teacher-centered perspective of teaching programming in secondary schools, with focus on in-service teachers that are beginning to teach programming and integrating it in different subjects. The focus is on understanding the overall experience of teachers, acknowledging that these experiences might differ considerably depending on a number of local and individual factors.

2 Literature review

A growing body of literature reports on experiences with the introduction of programming in different educational systems, as for example computer science education in Israel and the United States [9]; in Brazil [10]; in the Nordic Countries [11]; in Finland, Sweden, and Lithuania [12]. A recent report on the status of Informatics education in Europe advocates the introduction of Informatics at all school levels and recommends that the teaching should be undertaken by teachers who have formal education in the subject. However, this is far from the current status in many countries, with teachers who might be teaching without having received the proper training [13]. A significant part of the duty of implementing programming education plans falls upon the educators. This is particularly problematic, considering that many of them have insufficient preparation to teach programming and computational thinking [14]. In this paper, we address the challenge of training in-service teachers to empower them to teach their students the programming skills that are critical for smart citizens.

Several challenges related to the teaching of computer programming have been reported, and teachers often do not know how to obtain adequate training and how to integrate the acquired knowledge into their educational environments [15]. Programming is a troublesome subject to learn and master [16]. It is a complicated subject that requires constant exertion, distinctive methodology, and multi-layer expertise. The way toward getting these aptitudes is a trial-and-error experimentation procedure and persistence [17]. Teachers are often left to teach something that they themselves do not feel comfortable with, considering that it is generally agreed that it takes about ten years to transform an amateur into a specialist developer [18].

Teaching principles will unmistakably influence the results of courses that educate programming [19]. Jenkins [20] explains that instructors of writing computer programs are not communicators of information like in numerous different subjects. Their essential job is rather motivating the students. The students must be roused to take part in undertakings that will cause them to learn, and it is the instructor's job to guarantee this. The world is filled with problems that require complex solutions and issues that are not reported in books and manuals. It is therefore naive to try to "drill" and "order" students, so they know the correct answers [21]. It is important to connect with students and engage them, so they assume liability for their learning and have the drive to make their future. In this perspective, it must be an instructor's primary task to guarantee that every one of their students is appropriately inspired. However, measuring or finding what drives people are not straightforward [20].

This short overview of related work clearly identifies teaching programming as an important but rather challenging endeavour, from a technical and pedagogical perspective. With our study we aim at contributing to the growing body of knowledge about teaching programming in K-12. Our focus is on teachers and how they experience teaching programming on the overall, rather than on specific technical or pedagogical issues.

3 Case and research method

Our study is connected to the in-service teacher training program offered by our university to in-service teachers who have to teach programming courses or integrating programming in other subjects. After completing the program, teachers are expected to be able to teach programming and use it in their discipline. In the following sections, we will briefly explain the context and the case being investigated in this study.

3.1 Context

The program consists of two courses of 7,5 ECTS (European Credit Transfer and Accumulation System) each, over two semesters. The first course focuses on learning basic programming and the second on teaching programming (for more information on the courses, refer to [22]). Though there are no requirements for teachers to follow both courses, most do. The study program is aimed at in-service teachers in grades 8-13 (secondary school). The program is an online study, with web-based lectures and weekly activities such as online lectures and regular compulsory work exercises. Students participate in the courses with the support of their school, which is committing to provide some free time to teachers to complete the course (one day a week), though they continue their primary duties during the two semesters. The additional costs for the schools are partly covered by a national program of the Ministry of Education. This support leads to a very high completion rate.

3.2 Course Description

The focus of this study is on the second course, aiming at helping teachers to develop the competencies needed to teach programming in their subjects, after they have acquired basic programming skills. Since most of the teachers attended the first course, the participants to our study are all expected to have at least a basic understanding of programming concepts and of Python, that is the main programming language studied in the course. The 2020 cohort consisted of 173 teacher students. Fig. 1 shows at which school level participants are teaching. More than 90% of the teachers are teaching in secondary schools, with the large majority in upper secondary schools (In the national educational system this approximately corresponds to the last three years of the K-12 system.) More than 80% of the teachers are teaching STEM-related subjects. Others teach in different subjects such as language, history, music, arts and crafts, etc. 24% of participants have already taught programming in their schools. 61% of participants had some level of programming knowledge before starting the first course. Most have been doing block-based programmings such as Scratch, LEGO Mindstorm, Arduino, micro:bit, and Blockly.

The participants are from the same country but coming from different schools and districts. All schools follow the same national curriculum. Participants vary in terms of teaching experience, including teachers who just started their practice and others who have been teaching for many years. The motivation to take these subjects courses also varies. Most say they are motivated, while others are more skeptical about the inclusion of programming in their subjects. A survey distributed at the end of the program indicates high levels of satisfaction with the course.

On the overall, the participants are diverse in terms of their background knowledge in programming, their interests in learning and teaching it, and the school level. We have not addressed demographic information in this study as we aim to include a large group with different backgrounds, needs, desires, and motivations.

To support each teacher in identifying their own learning trajectory, the course is organized around a 6-week project aiming to define and evaluate activities to be used in teacher's classrooms when teaching programming. As an outcome, teachers create a teaching plan in programming for their specific subject that they can use in their practice [22]. The project is mandatory and part of the final assessment. Teachers receive feedback from the teaching staff during the semester and must pass a first pilot phase to continue with the actual implementation.

Inputs to this project include (1) Teacher's disciplinary knowledge: Knowledge of the subjects they are teaching might help teachers in designing a teaching plan where programming is applied. Teachers can identify relevant problems to be resolved by pupils when learning to program, which may lead to increased learning outcomes in the subject; (2) Learning outcomes during the program: Teachers learn the basics of programming concepts, algorithmic thinking, and problem-solving; (3) The existing teaching plans: The teaching plans that have been created by other teachers who have completed this program in the past are also available to be used as part of creating a new and customised teaching plan; (4) External resources: Teachers can also use other resources such as teaching plans created outside of this program/course; and (5) Support from instructors: The project consists of several phases (analysis, design, and implementation). Instructors and teaching assistants support teachers during each stage by evaluating and giving constructive feedback [23]. This form of feedback may provide teachers with the information they need to improve their learning (formative assessment) [24].

Based on a predefined template, teachers start investigating the purpose and objectives of their teaching plan. Besides, instructors can support participants and give constructive feedback before implementing the teaching plan. Ovando [25] concludes in his research that constructive and meaningful feedback may lead to successful teaching and learning and personal achievement. Feedback should be deliberately given to educators and students occupied with instructing and learning [25].

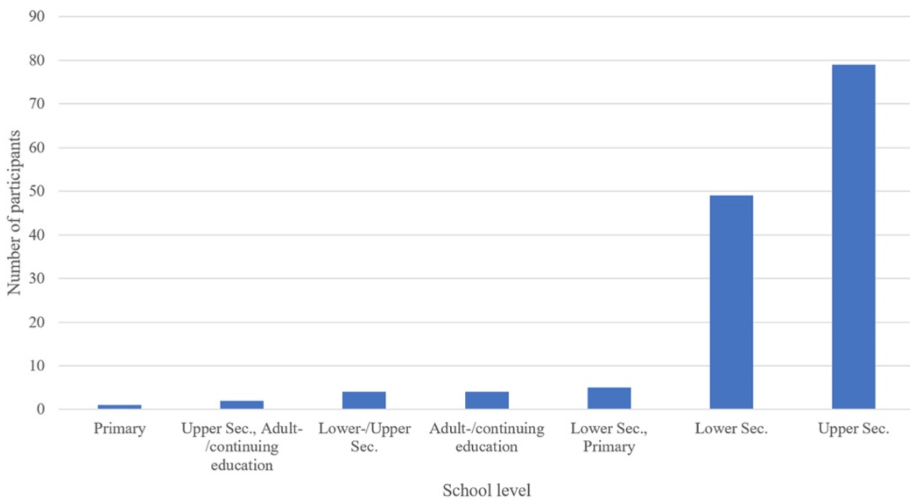

Fig. 1. School level at which participants teach

3.3 Data set

The final delivery of the project includes reflection notes. There is no predefined template for writing them. We simply ask participants to reflect on their experience during the project, including the definition and the implementation of their teaching plan. The 173 participating students' reflection notes are the basis for the study presented in this paper. The number of participants and the type of the documents resulted in a large and rich data set.

We informed students that we would use anonymised data in research. Before we started the analysis, an assistant outside the authors' team anonymised all the reflection notes.

The reflection notes' length and quality vary considerably, with some being very shallow notes and others offering very in-depth and lengthy analyses. This must be considered carefully in the study because the quality of the documents varies, and the opinion of some teachers, the ones who have written most, might be predominant. However, the advantage is that not having a template, teachers are free to focus on what they experience as relevant and essential. In this way, it is easier to capture teachers' perspectives, with less framing from the teachers and the researchers.

3.4 Method of Analysis

To address our research question, we conducted a thematic analysis of the data [26]. The analysis started with themes already identified during the analysis of a similar data set [27]. The previous study, involving a different cohort of teachers attending the same course, explored the impact of the training program on teachers' self-efficacy. Though the research questions addressed in the two studies are different, they share the overall aim of understanding challenges connected to teaching of programming and implications for professional development. In this perspective, general themes related to programming; teaching programming; recurrent didactic issues; and external challenges have been assessed as relevant also for the study presented here. In addition, new codes were added as emerging [28]. Our research is therefore based on both deductive and inductive thematic analysis.

Fig. 2. Word-cloud illustrating code frequency

Data was coded by one of the authors using Nvivo. To limit the thread to validity connected with a single coder, we added an initial phase for quality assurance. Two reflection notes were first coded together by the coder and another author, to help the coder to familiarise with the material and the existing codebook. The coder then went ahead coding 15 additional notes. At this point, the coder and one of the authors thoroughly discussed the coding and the revision of the codebook. No new themes were identified at this point, but some codes were added to avoid bias in the analysis and help capturing both negative and positive issues. After this initial phase, the coder continued with the independent coding of the material but discussing with the other authors whenever doubts arose. At the end, the coding resulted in the following distribution: external challenges with 5 codes occurring 578 times in 153 reflection notes; teaching programming with 8 codes occurring 539 times in 147; didactic issues with 5 codes occurring 429 times in 130 and programming skills with 5 codes occurring 220 times in 99. (Be aware that one participant's reflection notes could receive multiple codes and each code could appear multiple times in a document.) Fig. 2 shows a word cloud illustrating code frequency.

It is important to note that this study is mainly intended to identify trends and main issues and it does not aim at performing a quantitative analysis of the occurrence of specific codes. From this perspective, when we report the number of times a specific code is occurring, it should be mainly interpreted as an indication of the magnitude rather than an exact number.

After the coding, all the authors were involved in the analysis and discussion. In particular, one of the authors had read all the reflection notes before the discussion.

4 Analysis

In this section, we present some of the results of the study. Given length constraints, we only provide an overview of each theme. Text in italics indicates direct quotes from the reflection notes, translated from Norwegian by the authors.

4.1 Issues with programming

Adequate programming level. Many of the teachers were satisfied with the programming skills that they acquired, with 67 of them explicitly mentioning it in their reflection notes. These teachers seemed confident in their programming skills and felt they had learned enough to teach programming to their pupils successfully: *After spending a lot of time on programming in the last year, one is sitting with necessary programming skills, an understanding of what programming is, and how it happens, and one knows a little about what possibilities and challenges the course entails.*

It is also interesting to note that several teachers feel that, even if they have not reached full proficiency, their level might be "good enough." These teachers seemed to believe that they learned enough basic principles to teach pupils confidently about programming: *I feel that my code's level is within what can be achieved by students at 1P* [note by the authors: a math subject in upper secondary school] *without previous programming experience*. Some teachers expressed their concern that the pupils' increasing digital competencies in the class will soon exceed the teacher's digital capability: *One of the challenges has been to get to the right level concerning the students' competence in programming. As I mention in the actual teaching plan, most students, as of today, will have little expertise in programming. Still, it probably does not take many years after introducing subject renewal before the students can do this at a higher level than me.*

Challenges with programming. Some teachers reported challenges with programming, with the topic being explicitly addressed by 39 teachers. This often refers to making small mistakes that drastically change the results of a project: *It has taken many hours of troubleshooting and frustration over things that at first seem fine, to suddenly no longer work. During this project, I have also learned how little it takes for things not to work and how it becomes appropriate to connect everything.*

However, it is essential to underline that this is not necessarily connected to a perceived lack of skills but seen as part of the programming process: *When I have programmed the solutions, I have very often received error messages along the way because I have forgotten something, or used the wrong variable name or such simple things. What I notice about myself when this happens is that I think of this as a natural part of writing code.*

Challenges with pupils' homework. The experience that teachers gained helps them to understand better the challenges that pupils might meet. In particular, one teacher who tested the project with pupils reported that some pupils needed help and support from their parents. Still, the parents were unable to help due to a lack of programming knowledge: *Two students, who need a lot of support and regular confirmations, found it very difficult with programming and home-schooling. In the end, we gave them a different task, and we will look at the programming together when the schools reopen. These students were entirely dependent on parents sitting with them all the time, and the parents also did not know programming and Scratch.*

This poses a challenge that may be an issue for many pupils once programming becomes mainstream in school. Those who need more help while doing schoolwork might lack the necessary support to complete the program, considering that they might miss access to someone with programming skills.

4.2 Teaching programming

Teaching for the future. Many teachers (63) report that they are looking forwards to teaching programming. A vast number of teachers recognise the value of programming in schools and are looking forward to implementing them in their classes. A factor that might influence this attitude is the perceived novelty and relevance of the subject, with teachers feeling responsible for introducing it engagingly: *And ultimately, it is essential that the teacher can teach programming in a relevant, engaging, and lifelong way that engages students. Programming has come to stay in this world, on a par with other subjects we have had for decades. We create education for the future!*

Teachers' confidence. Some teachers (23) report that the knowledge and experience they have gained make them confident about their role. One teacher writes: *I can now meet the new school year with a kind of calm as I have a plan on how to start the work with programming.* Another teacher claims to be ready for the new challenges that might come with the new national curriculum: *In the autumn, we will implement new curricula in science, and there will undoubtedly be opportunities to use other methods along the way. Then I know how it can be done!*

Teachers seem to bring their pedagogical experience and approach into the new teaching of the subject, finding ways to preserve the way they see themselves as teachers, building on their previous experience: *I am mainly an active supervisor who talks to the students during the project and gives detailed and learning-promoting feedback*. Teachers are aware that they might need to acquire additional competencies, but they might still have developed the capability to do it: *Finally, I can say that these two programming courses have made me well enough equipped to complete teaching with programming in the classroom, although there is still much more to learn. In any case, I have seen new opportunities, and I look forward to using the skills I have gained in the classroom.*

External resources. External resources available online are often reported as an important source of inspiration for teachers (69): *I have, through the course, become more proven on various resources available on the Internet. I have learned about various websites I did not know about before. In this way, I feel better equipped to find good plans online when I need to create teaching plans with programming as an element in the future*. It is however, important to underline that finding, selecting, and adapting existing resources might be a challenging and time-demanding process. As explained by one teacher: *I used a lot of time looking for good examples that I could use to teach MicroPython. ...Something was easy to understand, but I had to test out things, take them apart, and put them together again into simpler programs that are easier for pupils to understand. I used a lot of time for this.*

Developing a teaching plan. Even with the available resources, some teachers (37) reported that it was challenging, but at the same time instructive and rewarding to create a teaching plan that was interesting, engaging, and suitable for the pupil's knowledge level and their own. Some teachers also encountered technical difficulties that made it challenging to complete the project. Others experienced as challenging to adapt programming to their teaching, especially for subjects where programming is not explicitly included: *One challenge was that I do not teach subjects that today say something explicit about programming, and I, therefore, had to work a little extra to try to sneak programming into the topics I have.*

In general, teachers value the capability that they have developed to introduce programming in schools, as evident in the two following examples: *I feel I have had a good repetition of some programming concepts for my part, but most importantly for my learning was to try to understand how I think the students will remember and try to imagine what challenges they will have*. Another teacher says: *The most significant learning I got when I discovered how I learned to see the possibility of connecting programming to already existing teaching programs. This made programming a positive contribution to the subject instead of an "additional burden." Incorporating the programming into teaching programs that are already in use is perceived as meaningful and useful.*

Some teachers (19) reported being worried about the teaching of programming. In most cases, they do not refer to themselves, but the perceived lack of knowledge among teachers, as evident in the two following excerpts: 1. *Many of today's teachers have little experience and knowledge in programming from their education, including me. Even though I am a relatively newly qualified secondary education teacher with a master's degree, graduating in spring 2017, my background knowledge would not be enough to teach students programming*. 2. *Many of my colleagues are unsure whether they will master providing training in a whole new skill as programming is. At my* *school I am it the only mathematics teacher with continuing education in programming. I hope that more people can use the teaching program I have created.*

Worth to mention are also some concerns about the time needed to define good teaching plans, with the need to redefine the level of ambition. As one teacher says: *I want to continue to build on the course material, with more themes and tasks. The plan was to have more content, but this is what I managed in these weeks.* The incremental approach that is suggested by teacher seems to be a good way to proceed for many teachers struggling with time limitations.

4.3 Didactic issues

Pupils' motivation. Very few teachers (3) have reported challenges in motivating and creating engagement among pupils. One teacher reports that although many of the pupils use various forms of technology, it is difficult to convince them of the importance of learning programming. They are satisfied in their roles as consumers and have low motivation for learning basic programming. Another teacher reports that students are often used to completing tasks with a single solution and are therefore struggling with programming's ambiguous nature. Another teacher said they struggled to find a balance between the pupil's independent learning and teacher guidance: *Regarding the students' learning outcomes in this project, I have asked myself the following questions: how much of the teaching should be controlled by the teacher, and how much should the students themselves be allowed to try and fail? This is a pedagogical challenge that all teachers face when planning to teach.*

At the same time, other teachers also see programming as motivating. E.g., one teacher mentioned that some boys are not interested in STEM and that they were excited to test their project because they believed it would increase their motivation: *I even imagine some boys with a low interest in science subjects, which could come out immensely strengthened from an encounter with programming. It suddenly becomes relevant in a completely different way, and if the teachers dare to let the students explore this universe themselves, it can go a long way. I'm excited.*

Adapted education. A majority of the teachers (101) reflected on adapting their teaching plan to the pupils' digital competencies in the class. This issue was mentioned in 101 different reflection notes. Most teachers were aware that pupils in the school have varying digital skills. Therefore, they created a teaching plan that would be adaptable and suitable for different competencies: *Suppose a teaching plan is to be designed around the learner's prerequisites. In that case, there is one utterly certain thing, and that is that you must include opportunities for differentiation. Students will encounter programming in school with very different prior knowledge. Some students may already master several programming languages; there will be others who have never worked with it. Although this will change in the time after introducing subject renewal, students will always need challenges at their level.*

This clearly poses a challenge to the creation of inclusive teaching plans, especially when teachers do not have a clear overview of pre-existing knowledge: *I have no knowledge of what they can of programming before.* Many teachers perceive the pupil's digital competence to be somewhat low, except for some pupils who have a particular interest in computers and programming who attempted to learn more in their free time.

Focus of teaching. The teacher's teaching plans had different objectives. What appears to be a common theme amongst many of the teacher's teaching plan (68) was to create motivation for learning programming amongst the pupils in their class. Another common focus was to implement programming interdisciplinary in the courses they were already teaching. For many, the focus was not on teaching programming as its own subject, but rather to understand how it could be used as a resource in other classes: *Our goal was to create a teaching plan for the students that create engagement, arouses interest in programming, and links it to other subjects the students have.*

Assessment of pupils. Very few teachers (15) reflected explicitly around pupils' assessment. This might be because few teachers had the opportunity to test the teaching plan with their pupils. Another reason might be connected to the curriculum's unclarity regarding the learning objectives of programming when integrated into other subjects and the uncertainties related to the upcoming introduction of a new national curriculum: *I have not set up detailed criteria concerning what the students should be able to do in programming. I have deliberately done this as it does not say anything about this in the curriculum. But I interpret the curriculum so that one can use programming as a tool to explore in mathematics. This is also why I have chosen that the assessment situation should be a presentation of what the students have done.*

Formative assessment seems to be an essential element, with some of the teachers concerned about pupils getting continuous feedback about their work and progress. Some teachers also underline the importance of pupils taking an active role in their learning. This relates to their learning process and their participation in shaping the learning activities, as in this example: *As a teacher, I (and we) are concerned with assessing learning and not with learning. Therefore, the task was not just to make something, but also to show something about the process along the way. I think it is important that students think about their role in learning. The student's voice is important, and I think the surveys that were conducted were good tools for mapping how we should set up the teaching in joint sessions, group sessions, and individual sessions. I could also catch those who had problems along the way and who did not say this directly. I think it worked well, and I was motivated to do the best for the students to come as far as they could in the time we had available. The last week was set up for the students to present the result, one by one, and have a conversation with a teacher, where they received feedback on the work.*

Cooperation and Community. Many teachers reported collaborating with other teachers (76) and collaborating with and among pupils in their class (68). Here we include (a) teachers collaborating with other teachers in the course; (b) collaborating with other teachers at their place of employment; and (c) collaboration with and among the pupils in their class.

Due to the ongoing pandemic, many teachers had not been able to test their projects and were, therefore, unable to reflect on the teaching plan's success. This made even more critical cooperation in the creation of lecture plans and their evaluation: *With the students' feedback, it could have been easier to adjust the plan along the way. Therefore, I have spent a lot of time discussing the teaching plan with* *my partner, so it has helped me work in pairs*. Some teachers created their teaching plan with colleagues from their school in mind: *It has been a goal that other teachers can use the teaching program, and it was therefore important that the description shows both what we should do and why! The rationale is central for the teacher to use the teaching program as a bridge between different representations of growth speed, thus helping the students increase their mathematical understanding*.

Many teachers also recognised that collaboration between students in the class was crucial: *Being able to use fellow students in discussions also provides more learning for the student. They must precisely explain each other, use correct subject terminologies, and not least have problems described by more than one person (teacher). All this can give students motivation to create something on their own and/or together with others*. Another teacher says, *I have also thought that it would be wise to organise the students in learning pairs in this programming task. There is a lot of learning in that students can discuss possible algorithms with a learning partner. They then get new ideas and reflect along the way in the work. They also dare to try more when working together in pairs*.

While many teachers recognise the importance of cooperation among pupils, it should be considered that this requires time and must be properly integrated in the planning. As explained by one teacher *My experience with this project is that programming takes time. The pupils need to get enough time to familiarise with the problem definition, to discuss the solutions, to discuss how a program must look like and answer the problem. I must set aside time enough so that pupils can discuss and reflect, not only programming*.

An interesting aspect is inter-generational collaboration. Many teachers reported that they got help from people from younger generations, such as their children or pupils. They said that they had learned a lot from more youthful individuals: *The actual maneuvering and how to build and destroy blocks and some commands I learned from my two sons*. This shows that young people may be an untapped resource for the teachers who plan on teaching programming.

Gender gap. An underwhelming number of teachers (7) mentioned the gender gap in programming in their reflection notes. Very few of the teachers who mentioned it reflected on ways to reduce this gap or make their teaching plan more appealing to girls, but mostly they acknowledged the issue. One female teacher explicitly mentioned that she aimed at being a role model for her pupils, while another teacher

reported designing a project, an automatic greenhouse, that could be more attractive for girls

One teacher said that she was a female science teacher and wanted to use her position to motivate her female students to learn to program: *First, I have become more proven in my role in teaching. As a female science teacher, I want to inspire and motivate more girls to choose technology education. Simultaneously, I want to use the position I have to show that programming is not just for the guys, possibly some nerdy stuff that not everyone can. If the threshold for starting is just low enough, most people will experience mastery in the beginning*.

One teacher mentioned that they created a teaching plan based on topics and themes of interest to girls: *One of my main goals is to get more girls interested in programming. In a preliminary study, several girls showed greater interest if the programming could be linked to biotic factors such as exercise, diet, and food. This* *gave me the idea to create a fully automatic greenhouse in the school's basement where Arduino devices monitored light, fertilisation, and irrigation.*

One teacher who had the opportunity to test their teaching plan in their high school 'higher math class experienced that the girls in their class, many of whom were academically strong in the subject, found programming to be challenging: *Part 2 of the teaching plan was tested on an R1 class due to the Coronavirus and the schools closed. I knew it would present some challenges, but this was instructive for me. We had two sessions (2 and 3 hours), where we worked on text programming with Python. In these sessions, I saw how big a difference is in the students' starting point for learning programming. What perhaps surprised me the most was that some students, who are usually the best (especially girls), did not do so well. They are conscientious and have the right work strategies, but I observed that they became more restrained and found this difficult in the second session.*

The same teacher also reported that the girls showed low interest and motivation in programming and struggled to partake in class discussions about the new teaching plan. The teacher believed that this might be because they focus on their academics and fail to show interest in things that they perceive not to be related to or important to the subject: *One of the nice things was that we had some good discussions about my scripts, and the students showed how they thought the program could be set up. But in this discussion, some of the students and especially some of the girls became a little absent. It may be that some of the schoolchildren's interest was not so great to learn this since it is not part of the curriculum in R1 this year, and they instead prioritise learning the curriculum in which they will be assessed.*

4.4 External challenges

In the analysis of the reflection notes, we identified several external challenges, i.e., challenges that the teachers did not perceive as related to their programming or pedagogical skills. These are all issues that are, to a large extent, outside the sphere of inference of individual teachers.

Use of programming in other subjects. Many of the teachers (57) reflected on teaching programming's challenges integrated into other topics. Some teachers noted that the math used in programming sometimes was challenging for the pupils to understand, requiring the teacher to consider yet another variable in adapting their project: *The students' varying prior knowledge in mathematics will also be essential for the program's educational approach, most of the assignments contain both mathematical algorithms and concepts.*

Some teachers also reflected positively towards using programming in math, possibly increasing the pupil's interest in math, as in the following example: *Through the work with this teaching plan, more people will hopefully get some mathematical "aha-experiences." Numerical methods open up a new field in mathematics that students have not seen before. This will probably be engaging for many students. They know that programming can be the fastest (or only) way to solve some mathematical challenges.*

COVID-19. COVID-19 was the most significant external challenge for this year's teachers (95). All schools were shut down, and all lectures were transferred online.

This confusion prevented most of the teachers from completing their teaching plans in class with their pupils. Due to this, it was difficult for teachers to reflect on their project and how it would work in practice. Some teachers lost access to necessary school resources due to the schools shutting down. Some teachers also reported that they struggled personally with adapting to the situation, and the same was for their pupils. Changing to online schooling as a result of COVID-19 was somewhat tricky and time-consuming. This caused a lot of teachers to experience a lack of time to work on new things.

Time constraints. Seventy-eight teachers mentioned issues connected to time in their reflection notes. Some issues have already been reported discussing, e.g., challenges with programming, relating to the time required to get things working, both for themselves and when planning activities in class. Time-related issues have also been reported in connection to pedagogical issues. Here we identify additional issues connected to elements that are largely outside the control of individual teachers. For example, one teacher expressed concerns about the time that has been allocated for programming in school and the connected challenges: *It is difficult for me to see how I can manage to teach to my students a traditional programming language in the little time that I have available in my subject, when they come to my class without previous knowledge*. A similar concern is also expressed by another teacher that says: *The first thing that struck me is that I cannot teach everything I have learned and experienced, without spending a lot of time....Time is a scarce resource in the classroom*.

Technical issues. Some teachers (30) did stumble upon some minor technical problems, though they were mostly resolved quickly. This also inspired the teachers to revise their activities to make them more robust. For example, one teacher states: *After many runs of the system, it did not always work, which created great frustration, I have been made aware that this built-in temperature sensor in the micro bit can register a little high temperature. This is because it is located inside the processor, and it is hot. In a proper implementation with students, I will have an extra thermometer on the teacup so that one can take control measurements of the temperature*.

School resources. Some teachers (20) mentioned access to school resources in their reflection notes. Most of the schools seem to supply the necessary resources to implement the teaching plan. One teacher noted that their municipality-provided teachers with Chromebooks but that they were not allowed to download apps on it, which was challenging: *The municipalities I work in have decided that we should only have Chromebooks and that no apps should be installed. Therefore, we need to find useful web resources that can be used on Chromebooks*.

5 Stumbling and Stepping stones

After the analysis, we categorised the identified issues into stumbling and stepping stones, based on whether they are perceived as impacting negatively or positively on teachers' experience.

5.1 Stumbling stones

Covid-19 lockdown is clearly emerging as a stumbling stone encountered by this cohort of teachers. This has impacted teachers' capability to implement their teaching activities as planned and created a general stressful situation. However, this is not the focus of the paper, where we instead aim to identify recurrent issues independently by the peculiarity of 2020.

Time is one of the main stumbling stones emerging from the analysis. This relates first and foremost to the problem for teachers to find time for completing their project. Even for this group of teachers, who have got official time for their school training, time was scarce. We can expect to find time for training to be even more difficult for other teachers. The paucity of time is a significant concern because it impacts teachers' capability to up-skill, keep their competencies updated with technological and pedagogical innovations, and ultimately create meaningful and engaging learning activities for their pupils.

Concerns about time are also connected with the implementation of activities in schools, especially in light of formative assessment and continuous support to pupils.

A number of teachers also reported challenges with programming, often requiring more time than expected to fix bugs and getting things working. This frustration might then turn into a lack of motivation [20].

This last point also relates to the perception of programming being difficult, in line with existing literature [14, 16, 17, 29, 30].

Developing programming skills and teaching programming are, not surprisingly, often intertwined in the perception of teachers. *Competence about the subject matter* is essential to give the teacher the confidence to teach [14, 30]. However, teachers might feel comfortable and do an excellent job also with limited programming skills as long as they can embrace the uncertainty and perceive that they know how to look for the missing information and seek for help.

Another possible stumbling stone is *multidisciplinarity*. Though teachers are not necessarily negative, many express concerns connected to the definition of relevant projects, the required knowledge for pupils in different subjects, and issue connected to meeting different learning objectives. Another concern is emerging when we get an eye-bird perspective of the teaching plans developed by the teachers. A majority of the teachers created interdisciplinary programming projects for STEM classes such as higher math, physics, biology, and chemistry. This might be problematic in terms of "Programming for All" [31] since it cuts out Arts and Humanities.

Another challenge worth to mention is the varying level of knowledge and motivation among students. Many teachers acknowledge the need to adapt their teaching to individual pupils. This is clearly a stepping stone for pupils, who can get tasks and scaffolding adequate to their needs. However, *adapted education* might turn into a stumbling stone for teachers since this requires additional time and competencies both in the planning and in the implementation of any teaching activities.

5.2 Stepping stones

Motivation is one of the main stepping stones that we can identify in the data, often connected to the excitement of teaching something that was not previously taught in schools or that teachers perceive as highly relevant for society and for the future job market. Even teachers who might be struggling themselves with programming, seem mostly be motivated to teach it and learn more, overcoming the challenges that they might encounter. This motivation is also reflected in the way many teachers look at their projects, striving to creating activities that are engaging, innovative, and inclusive. Hopefully, this can create a virtuous cycle that can benefit the motivation of both teachers and pupils, sustaining it in time.

Another stepping stone connected to motivation is the application of programming in different disciplines. Teachers are motivated to learn and teach programming when they see the benefit of applying programming in their subject areas to increase pupils' subject understanding. Also, the pre-existing confidence in their main subject might help them to deal better with the uncertainties of programming.

Collaboration and community participation are the other major stepping stone for many teachers. As presented in our analysis, this includes: (i) the cooperation with peers, (ii) the cooperation with students, and (iii) the use of resources from different communities as a starting point for their projects.

6 Turning stumbling stones into stepping stones

Based on the results and the previous discussion, in this section we draw some implications for teacher training. Before starting, we want to underline that the results show that a stepping stone for one might be a stumbling stone for another. Even Covid-19 was seen by some teachers as an opportunity to discover new tools or to find new forms of cooperation with their peers. This does not intend to underestimate the challenges that teachers meet that are concrete and legitimate given their context. The main point here is that teacher training should focus on giving teachers the conceptual tools and skills to address known challenges and limit their impact on their practice.

The course that we studied supports flexible learning trajectories, i.e. teachers can focus on different topics and choose how they intend to teach programming to their students. When defining their projects, they can select relevant subjects, the appropriate level of complexity, and align this with the learning objectives. This seems to be a promising approach that gives motivation. When struggling with time, working with something that one perceives as relevant and directly usable in class is important. However, data shows that teachers choose themes related to the subjects they teach. Since most of the teachers in this study program teach STEM subjects, the teaching plans are also associated with them. As previously discussed, this might influence negatively the development of inclusive programs that would benefit from a higher level of multi-disciplinarity. This should be addressed both at the policy level, when deciding which teachers should get access to training or admission criteria and the existing programs, for example promoting awareness of projects bringing together different subjects outside STEM.

For teachers, the process of learning to program themselves and teaching it to others often goes hand in hand. This is extra demanding because programming in itself is complicated and will take years before one can become steady in programming. Therefore, it is essential during training to increase teachers' awareness about learning as a continuous process, and that their skills must be continuously maintained [21], also for experienced teachers [32]. This is challenging since time is one of the main constraints. Still, if teachers take small steps and are concerned with creating engagement and interest in students [33], it can lead to students who become more autonomous and more active in the learning process [18].

The availability of online resources is clearly an advantage that many teachers are building on. However, for other teachers it is overwhelming. Training of teachers might consider this issue by explicitly providing information about reliable resources and about how to search and adapt them. This issue might be seen under the broader umbrella of cooperation and community building. Teacher training should explicitly address collaboration and community participation, identifying successful forms of cooperation in different settings. Some teachers explicitly mentioned the challenges connected to the lack of peers they can relate to. We can expect this to be particularly serious for small and rural schools. Learning how to create connections among teachers outside owns school [34] is therefore paramount. In this context, we want to underline that large international communities of teachers have been only marginally mentioned. Language might be a limiting factor, together with differences in national curricula. Training should therefore include awareness of local and international communities, different forms of participation, and promote a culture of sharing, where teachers are not only consumers but also providers of learning material and expertise. This is important because it might motivate teachers and, at the same time, help with sustaining communities. Training should promote participation in local communities, i.e. with teachers sharing same national curricula, language, and school culture, as well as in global communities to confront different perspectives and promote a reflective practice. It is however important to point out that some studies show that, despite the documented benefits of collaborative learning, many online instructors and students avoid it due to the time and commitment it requires [35]. It is therefore important to create among teachers awareness of the potential benefits and help them to develop the right skills for a fruitful collaboration.

Our data show the importance to promote self-efficacy, i.e. the belief of a teacher in their capabilities to teach the subject [36]. Improving teachers' self-efficacy requires working on teachers' competencies and giving them better control of external challenges. We know that learning to program and teaching programming is demanding [16]; therefore, preparing a teaching plan may affect teachers' self-efficacy. By developing teaching plans supervised by course instructors, teachers may be better equipped to start teaching programming for their students. In addition, it might be useful to make teachers aware of challenges that they might face and heuristics to address them. This requires however a good understanding of the specific needs of the teachers. For example, as already mentioned, the possibilities of cooperation with peers might be very different in small rural schools from large urban schools. As another example, the teachers in our cohort have not reported major issues with availability of resources. This might however be a major challenge in other contexts, as reported e.g., in [37].

There are two missing stones in the data, despite we have been explicitly looking for them. Gender issues are only marginally mentioned by a very limited number of teachers. Considering the gender gap in computing education [38], it is worrying that this is an issue that is not high in the teachers' reflections. Assessment is another issue that is appearing rarely in the analysed documents. Again, given the importance and challenges connected to assessment in computing education [39, 40], it is a concern. These two aspects should be addressed more explicitly in teacher training to increase their awareness.

7 Conclusion

This paper presented the results from the analysis of the reflection notes delivered by 173 in-service teachers attending a university-level computing education program. In this study, we aimed at identifying challenges and enablers connected to teaching programming in secondary schools. Though some of these challenges and enablers have already been identified in previous literature, our main contribution consists of providing a holistic approach that looks at stumbling and stepping stones together and how they might influence policies and teacher training. This approach also underlines the importance of looking at teachers individually and as part of a complex ecosystem that affects how they operate, bringing opportunities and challenges.

We are fully aware that the study focuses on a particular group of teachers, i.e., the ones who have been given by their school the time to improve their teaching. Therefore, their perception of challenges and possibilities is impacted by the course. On one side, this is an interesting group because it allows identifying stepping and stumbling stones core to learning and teaching programming, even when this happens in optimal conditions, i.e., with allocated time and expert support. Simultaneously, regular in-service teachers not receiving any training in programming might face additional challenges that are not identified in a study with this specific population. As part of future work, we intend to study the broader population of in-service teachers to investigate the extent to which the identified stumbling and stepping stones might be generalised. It should also be noted that we have not connected the reflection notes to demographic information, mainly for privacy reasons. In future studies, it might be interesting to study how the teacher's profile, e.g., age and experience, might influence the results. It might also be interesting to study another didactical issue that teachers discovered during the implementation of their teaching plan: the importance of collaboration through pair programming. Another area that might need more attention in future research is challenges with pupil' homework when parents cannot help due to lack of programming knowledge. The same issue may apply when learning other subjects where parents are not proficient and cannot help children. Do teachers consider the difficulties with learning programming to be more challenging than those met when learning languages?

Another limitation of the work is that we analysed reflection notes where teachers decide what to focus on and the time they have to complete the task. This offers the advantage of providing a direct perspective on the teachers' point of view and what they value as relevant. Simultaneously, the fact that something is not mentioned does not necessarily imply that teachers do not think about it. To overcome this limitation, we are planning interviews with the teachers to collect a complementary data set. Finally, studies across countries would be relevant to understanding how the educational system impacts teachers' experiences.

Acknowledgments. We thank all the teachers who attended the course and participated in the study. The work is partly funded by Excited, The Norwegian Center for Excellent IT Education (https://www.ntnu.edu/excited).

References

1. Jaskiewicz, T., et al., Hacking the hackathon format to empower citizens in outsmarting "smart" cities. Interaction Design and Architecture(s) Journal 2019. 43: p. 8-29.
2. Poplin, A., L. Shenk, and U. Passe, Transforming Pervasive into Collaborative: Engaging Youth as Leaders with GIS through a Framework that Integrates Technologies, Storytelling, and Action. Interaction Design and Architecture(s) Journal 2017. 35: p. 182-204.
3. Clements, D.H. and D.F. Gullo, Effects of computer programming on young children's cognition. Journal of Educational psychology, 1984. 76(6): p. 1051-1051.
4. Duncan, C., T. Bell, and S. Tanimoto. Should Your 8-Year-Old Learn Coding? New York, NY, USA: Association for Computing Machinery.
5. Jeannette, W., Computational Thinking - Jeannette Wing. 2008.
6. Dağ, F., Prepare pre-service teachers to teach computer programming skills at K-12 level: experiences in a course. Journal of Computers in Education, 2019. 6(2): p. 277-313.
7. Motschnig, R., et al., When Kids are Challenged to Solve Real Problems – Case Study on Transforming Learning with Interpersonal Presence and Digital Technologies. Interaction Design and Architecture(s) Journal 2017(34): p. 88-111.
8. Cools, S., et al., The Diorama Project: Development of a Tangible Medium to Foster STEAM Education Using Storytelling and Electronics, in Citizen, Territory and Technologies: Smart Learning Contexts and Practices 2017, Springer. p. 169-178.
9. Gal-Ezer, J. and C. Stephenson, A tale of two countries: Successes and challenges in K-12 computer science education in Israel and the United States. ACM Transactions on Computing Education (TOCE), 2014. 14(2): p. 1-18.
10. Santos, P.S.C., L.G.J. Araujo, and R.A. Bittencourt. A mapping study of computational thinking and programming in brazilian k-12 education. in 48th Annual Frontiers In Education Conference. 2018. IEEE.
11. Kjällander, S., et al., Makerspaces across settings: Didactic design for programming in formal and informal teacher education in the Nordic countries. Journal of Digital Learning in Teacher Education, 2018. 34(1): p. 18-30.
12. Dagiene, V., et al. Students' performance on programming-related tasks in an informatics contest in Finland, Sweden and Lithuania. in ITiCSE '14: Proceedings of the 2014 conference on Innovation & technology in computer science education. 2014. ACM.
13. CECE, Informatics Education in Europe: Are We All In The Same Boat? 2017, ACM.
14. Cooke, L., et al., Can Everyone Code?: Preparing Teachers to Teach Computer Languages as a Literacy. 2020, IGI Global. p. 163-183.

15. Hiltunen, T., Learning and teaching programming skills in Finnish Primary Schools--The potential of games. University of Oulu, Oulu, 2016.
16. Cheah, C.S., Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review. Contemporary Educational Technology, 2020. 12(2): p. ep272-ep272.
17. Jiau, H.C., J.C. Chen, and K.-F. Ssu, Enhancing self-motivation in learning programming using game-based simulation and metrics. IEEE Transactions on Education, 2009. 52(4): p. 555-562.
18. Winslow, L.E., Programming pedagogy - A psychological overview. ACM Sigcse Bulletin, 1996. 28(3): p. 17-22.
19. Robins, A., J. Rountree, and N. Rountree, Learning and teaching programming: A review and discussion. Computer science education, 2003. 13(2): p. 137-172.
20. Jenkins, T. Teaching programming--A journey from teacher to motivator. in 2nd Annual conference of the LSTN Centre for Information and Computer Science. 2001.
21. Liu, W.C., J.C.K. Wang, and R.M. Ryan, Understanding motivation in education: theoretical and practical considerations. 2016, Springer. p. 1-7.
22. Rouhani, M., et al. Design of a programming course for teachers supporting flexible learning trajectories.
23. Du Toit, E., Constructive feedback as a learning tool to enhance students' self-regulation and performance in higher education. Perspectives in Education, 2012. 30(2): p. 32-40.
24. Heritage, M., Formative assessment: What do teachers need to know and do? Phi Delta Kappan, 2007. 89(2): p. 140-145.
25. Ovando, M.N., Constructive Feedback. International Journal of Educational Management, 1994.
26. Braun, V. and V. Clarke, Using thematic analysis in psychology. Qualitative Research in Psychology, 2006. 3(2): p. 77-101.
27. Thorsnes, J., M. Rouhani, and M. Divitini. In-Service Teacher Training and Self-efficacy. in International Conference on Informatics in Schools: Situation, Evolution, and Perspectives. 2020. Springer.
28. Saldaña, J., The coding manual for qualitative researchers. 2015: Sage.
29. Wang, X.-M., et al., Enhancing studentsâ□™ computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt. Journal of Educational Technology & Society, 2017. 20(4): p. 58-68.
30. Hotaman, D., The teaching profession: knowledge of subject matter, teaching skills and personality traits. Procedia-Social and Behavioral Sciences, 2010. 2(2): p. 1416-1420.
31. Guzdial, M., Learner-centered design of computing education : research on computing for everyone. Morgan & Claypool Publishers. 147-147.
32. Savec, V.F., Use of ICT and innovative teaching methods for STEM. Innovative Approaches to the Application of Digital Technologies in Education and Research SLET-2019, Stav-ropol-Dombay, Russia, 2019: p. 1-10.
33. Jenkins, T. Teaching programming--A journey from teacher to motivator.
34. Ferk Savec, V., Use of ICT and innovative teaching methods for STEM. 2019.
35. Zygouris-Coe, V., Collaborative learning in an online teacher education course: Lessons learned. ICICTE Proceedings, 2012: p. 332-342.
36. Zee, M. and H.M.Y. Koomen, Teacher self-efficacy and its effects on classroom processes, student academic adjustment, and teacher well-being: A synthesis of 40 years of research. Review of Educational research, 2016. 86(4): p. 981-1015.
37. Sentance, S., L. Singh, and P. De Freitas. Challenges Facing Computing Teachers in Guyana. in SIGCSE '20: Proceedings of the 51st ACM Technical Symposium on Computer Science Education. 2020. ACM.
38. Klawe, M., T. Whitney, and C. Simard, Women in Computing---Take 2. Commun. ACM, 2009. 52(2): p. 68-76.

39. Klawe, M., T. Whitney, and C. Simard, Women in Computing---Take 2, in Commun. ACM. 2009, Association for Computing Machinery: New York, NY, USA. p. 68-76.
40. Grover, S. and R. Pea, Computational thinking in K--12: A review of the state of the field, in Educational researcher. 2013, Sage Publications Sage CA: Los Angeles, CA. p. 38-43.