Professional Development for In-Service Teachers of Programming: Evaluation of a University-Level Program

Majid Rouhani(✉), Miriam Lillebo, Veronica Farshchian, and Monica Divitini

Department of Computer Science, NTNU, Trondheim, Norway
{majid.rouhani,monica.divitini}@ntnu.no

Abstract. Professional Development (PD) organizations provide training programs for computer science teachers through teacher PD. Programming as part of a teacher's PD has grown in importance in K-12 education. As a result, an increasing number of teachers, often without sufficient training, are teaching programming in their schools. It's challenging to learn to program, and it's even more challenging to teach it. Teachers must be able to program and teach programming, which is usually done in the context of multiple disciplines. In this study, we take a closer look at a teacher's perspective on teachers' PD in programming. We focus on the PD program offered by our university, composed of two courses over one academic year, and define the following research questions: Which strengths and weaknesses of the PD program provided are suited for in-service teachers when learning to program? Learning programming takes time. How could this long learning process be supported? This is an interview study of sixteen in-service teachers who joined the program. The main findings seem to be that teachers are comfortable with how the program is set up. They prefer flexible courses so that they can adapt the implementation with their regular teaching. They also emphasize the need for a continuous learning process and a community of practice (CoP) to continue developing. The main contribution of this study is the evaluation of the PD program at the university.

Keywords: In-service teachers · K-12 · Professional development · Programming

1 Introduction

Computer science (CS) in general and programming, in particular, have long been thought to be relevant to information technology (IT) experts. However, in recent years, this viewpoint has shifted. Software and technology are increasingly playing a role in practically every element of society and life at the current rate of digitalisation [1]. The teachers' PD within digital competence will therefore become more critical in the time ahead. Effective PD must prioritise teachers' and students' needs and address various elements at both the individual and situational levels [2]. Match PD to instructors' backgrounds, link it with the course's curriculum and employ effective motivational design

E. Barendsen and C. Chytas (Eds.): ISSEP 2021, LNCS 13057, pp. 123–134, 2021.
https://doi.org/10.1007/978-3-030-90228-5_10

to promote teacher engagement are three recommendations made by Qian, Hambrusch's [3] study for building effective online PD for CS teachers. Learning programming is a difficult task, according to the scientific community [4]. Besides, teaching teachers may be more challenging and complex than other forms of adult education [5]. Therefore, it is essential to have a teacher perspective to map their needs and expectations on PD in programming.

In this study, we take a closer look at a teacher's perspective on their PD in programming and seek the answer to these research questions: (RQ.01) Which strengths and weaknesses of the PD program provided are suited for in-service teachers when learning to program? Learning programming takes time and might be challenging to learn and teach. (RQ.02) How could this learning process be supported? This is an interview study of sixteen in-service teachers who are preparing to teach programming at school. The sample includes teachers who attended a programming course at the university level. We used the thematic analysis method to analyse the qualitative data.

The following section presents the related work and positions the article in the context of the PD of in-service teachers in programming. Section 3 presents the case and research method. In Sect. 4, we present the results and discuss the main findings in Sect. 5. Conclusions and further work is presented in Sect. 6.

2 Related Work

There has been growing interest in incorporating programming into teacher education and PD in recent years [6]. This process, however, has not been easy; various studies have shown difficulties in teaching computer programming, and teachers sometimes do not know where to get proper training or how to use what they have learned in their classrooms [7]. At the same time, we know that learning programming is a time-consuming process, and in-service teachers usually do not have the required time. There are also challenges associated with teaching programming. Furthermore, the tools and technologies used when developing programs change over time and maintaining this knowledge is essential [8]. Teachers must prepare to integrate digital competencies into their teaching as computer science becomes more widely integrated into school curricula in a growing number of countries. This integration is a moving target, with new methods, tools, and applications appearing and disappearing at such a rapid pace that teachers must have the confidence to explore what is brand-new, relevant and plan their educational activities to include digital competencies independently and continuously [9]. PD is widely agreed to be critical for curriculum innovation in computer science teaching. However, in many countries, preparing PD for teachers is problematic due to instructors' lack of technological experience and abilities, pedagogical understanding, and topic knowledge of computer science [10].

Effective PD seems necessary, but what does it mean, and how do teachers view this from their point of view? The American Learning Policy Institute published a report in 2017 that outlines seven characteristics of effective teacher PD that lead to changes in teacher behaviour and improved student learning outcomes: focus on specific content, incorporation of active learning, support for collaboration, use of effective practice models, providing coaching and expert support, offering feedback and facilitating reflection, and is of sustained duration. Effective professional learning incorporates most or all of these elements [11]. Several studies have found that PD is more successful when six features are included, defined as required or sufficient circumstances, e.g. [12, 13]. Despite minor changes in terminology and disagreements at the periphery, the core statements are highly consistent, as demonstrated by a recent meta-synthesis [14]. Sims and Fletcher-Wood [15] reexamines in their research the evidence that supports this consensus, suggesting that the reviews on which it is based have significant methodological weaknesses, such as using inadequate inclusion criteria and relying on an incorrect inference procedure. As a result, the consensus is likely to be incorrect. According to the argument, researchers might make more headway in identifying characteristics of good PD if they looked for alignment between evidence from basic research on human skill acquisition and aspects of rigorously evaluated PD treatments.

According to a survey of research published in the United States between 2010 and 2014, most computer science PD programs were less than a week-long and not stretched out over time, rather than taking the form of a one-time summer workshop [10]. However, there has been discussion in recent years about the need to move away from this type of traditional intensive training proposal to create spaces where teachers play an active role in learning, requiring studies that address the 'teaching experience' in these new training settings [16]. According to numerous research, the most productive context for informal workplace learning is a school culture that supports and values collaborative learning [17]. Cascade training, a "train the trainer" technique in which the first generation of trainers receives training and then delivers the specific content to the next generation of trainers, is a very complimentary paradigm in education. This procedure can be repeated indefinitely [18]. This study's main contribution is to investigate the PD program provided by the university in programming and how they expect the long learning process to be supported from a teacher's perspective.

3 Case Description and Research Design

The program is designed for in-service teachers in grades 8 through 13. (secondary school). With web-based lectures and weekly activities, the program is online, with no physical gatherings. Students enrol in these courses with the approval of their school and agree to provide teachers with free time to complete the course. The completion rate of the program was 87%. A nationwide program funded by the Ministry of Education partially covers the higher expenditures for schools[1].

One hundred ninety-two in-service teachers from different districts of the country participated in this PD program. More than 90% of the teachers in this study work in secondary schools, with most of the teaching in upper secondary schools (In the national educational system, this corresponds to the last three years of the K-12 system). STEM-related subjects are taught by more than 80% of the teachers. Others instruct in various areas, including language, history, music, arts and crafts, and so on. In their schools, 24% of participants have already taught programming. Before beginning the first course, 61% of participants had some programming experience. The participants

[1] https://www.udir.no/kvalitet-og-kompetanse/.

all are from the same country, although they represent various schools and districts. All schools follow the national curriculum. Participants have varying levels of teaching experience, ranging from new teachers to those who have been teaching for many years. The incentive to enrol in these academic courses varies as well. The majority think they are motivated, but others are less convinced that programming should be included in their classes. The curriculum comprises two courses totalling 15 ECTS credits. The first course (7.5 ECTS) covers the fundamentals of programming (variables, operators, if- and loop statements, lists, functions and libraries). The programming language we use is Python. The textbook "Starting out with Python" [19] is used. Student activities in the first course include three mandatory exercises, a reflection note and a mini-project (two weeks duration). Students can also complete specific assignments from the book. In the second course, we focus on how to teach and apply programming to interdisciplinary subjects. Student activities are four mandatory exercises, a feasibility study and a project. We have organised this course 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.

3.1 Research Design

The Overall Method. This study is based on sixteen interviews with in-service teachers in K-12. The study uses semi-structured interviews to explore the research questions by capturing teachers' reflections on their PD program.

Interview Guide. The interview guide was constructed based on the following main elements: Part A (the PD program): (1) The professional development program at the university is divided into two programming subjects. In the first part, the focus is on learning to program, while in the second part, the focus is on teaching programming. What do you think about this form? (2) The second programming course is designed to be flexible, where students can set up their learning trajectories. What is your opinion on learning programming in a subject like this? (3) Do you think it would be necessary to have additional courses which build upon this program? Part B (PD in general): (1) What is the best way for in-service teachers to learn to program? (2) Have you ever been in a situation where you had to learn some programming concepts with your students? How has it affected your role as a teacher? (3) Learning programming takes time. How could this long learning process be supported? (Courses, seminars, CoP, school activities, peer learning, etc.)

Participants. The research team sent an invitation letter to all the participants of the 2020/2021 cohort. Sixteen teachers expressed interest in participating in the study: T1 (F, USS, Mathematics, physics, technology and research theory), T2 (F, USS, Mathematics and science), T3 (M, USS, Mathematics and science), T4 (F, LSS, Mathematics, social studies and Norwegian), T5 (F, USS, Physics), T6 (M, LSS, Arts and Crafts, Programming), T7 (M, USS, Chemistry, mathematics and science), T8 (M, LSS, Programming, mathematics, and science), T9 (M, LSS, Programming, mathematics, and science), T10 (F, USS, Mathematics), T11 (F, LSS, Programming, mathematics, and science), T12 (F, USS, Mathematics, biology, and science), T13 (F, USS, vocational subjects in business support for ICT service, user support for ICT), T14 (F, LSS, Mathematics, science, technology and design), T15 (M, LSS, Programming, mathematics, and science), T16 (M, LSS, Norwegian, English, media and information, and programming). Interviews were conducted via Zoom, using the service offered internally by our university for GDPR-compliance. The interviews were recorded with a Zoom recorder and then transcribed by the interviewer. The relevant national agency approved the research. All the participants have been informed about the study, their rights and have been explicitly given their consent.

Data Preparation and Coding. We recorded interviews with teachers over zoom. Two of the researchers divided the recordings equally and proceeded to transcribe them individually. Some teachers requested approval of transcriptions before using them in this study. The two researchers coded a few transcriptions individually to create some initial codes and categories. The research team agreed on a joint codebook. Since we only use two coders, they used the method "Percent Agreement for Two Raters[2]." Approximately 20% of transcriptions were coded by two researchers independently, and the research team calculated the inter-rater reliability. When we achieved an IRR of about 80%, we encoded the rest of the transcripts.

4 Results

Some of the study's findings are presented in this section. Due to length limits, we can only give a high-level overview of the most frequently used codes. Direct quotes from the reflection notes are translated from Norwegian by the writers.

4.1 Teachers' PD in the Context of the Programming Courses at the University (Part A)

All interviewees have given their opinion on how the program has contributed to their PD in the area of computer science. We ask teachers about their views on how this form of PD fits and support their advancement in the field. In the following sections, we explain the most frequently codes discussed by participants.

Organization/Structure. Although there was both positive and negative feedback on the organisation and implementation of the courses, most are optimistic about the way the program is set up and performed. The positive aspects are that the program focuses on both learning programming and applying it in practice, flexibility concerning the choice of topics in the course and time for completion of the activities, and focus on programming didactics. The negative aspects are about the connection between lessons and exercises; exercises must be more relevant to teachers' disciplinary knowledge; flexibility of the course is challenging for some and struggle to find the right balance of work; and desire for even more lectures/follow-ups. Fourteen teachers liked how the

[2] https://www.statisticshowto.com/inter-rater-reliability/.

program was organised or structured. Amongst these teachers, many noted that they appreciated how the program was structured throughout the year, with the fall semester having an introduction to programming and the spring semester focusing on teaching programming in school. E.g., "I think it has been excellent. It was very nice to get that run-up with learning basic programming this fall and then direct it towards teaching this spring. Especially now, considering that it is very concerned about how we should use it further in teaching. So, this spring semester has been beneficial considering that, so I think it's helpful with the division that has been. It has been beneficial to learn the basics before moving on to thinking about using them for your teaching" (T10).

The Flexibility of the Course. The course, which runs in the spring, is flexibly arranged. This means that teachers essentially choose topics that are relevant to their subject areas. Throughout the semester, they work with subject matter and exercises that build upon each other, and teachers can use that to create a teaching plan for use in their teaching later. Fourteen teachers reflect on the flexible nature of the course. Teachers feel that this allows them to balance their coursework with their work and family life and focus on relevant topics to their subjects. E.g., "… And the spring has been quite flexible in a way, which was very good. It was a bit more to learn, but you got to use it. And that was perhaps the most important thing; not just learning the code, but also using it. Yes, I think there is a good distribution like that" (T13). "Do you mean that we can choose a lot ourselves what we want to do in the exercises? Yes, I think it is essential because we are in different subjects and have different grade levels. We have other focus areas, so I think it was very nice to imagine real situations in my classes" (T4). Few teachers also mention the negative impact of flexibility because it may be confusing and overwhelming. Some teachers may have little knowledge or understanding of programming to decide what is helpful for them to learn. An introduction to each part of the course could be beneficial to make awareness when deciding which areas they have to focus on throughout the course.

Teacher's Needs and Interests. Although many (14) express that they have learned enough programming and can apply it in their subject areas, some (7) also point to more miniature courses to learn in-depth topics such as Arduino, micro: bit, raspberry pi etc. Some others suggest dividing the program into smaller parts to create classes with more homogeneous students regarding background knowledge and their teaching level. Others say that more courses are not needed now as they need time to absorb what they have learned so far. In addition, it is said that they have come far enough to build on this knowledge by themselves. Many of the teachers (8) were optimistic because they felt that the course had allowed them to learn relevant things for their classes. E.g., "I think it is essential because we are teaching different subjects, and we have different grade levels. We have other focus areas, so I think it was very nice to have the option of using relevant cases in the class" (T5). When it comes to time usage, teachers say that even though it is challenging to find enough time, more time should be allocated through school and privately, which means that one gets to develop more. Few teachers point to the technical infrastructure, and related challenges that might need a bit more attention than they currently get in the course. E.g., "Yes, we have been lucky, so we have bought everything because the principal is so fond of programming and technology. But we experience technical issues, e.g., updating the firmware etc. It would have been nice if the course focused a little more on technical challenges as well…" (T14). There is a need to support teachers with technical issues, considering that their available infrastructure might vary.

Workload. Many teachers (10) talk about the amount of work in the course and seem to be divided in the middle. Those who think there is too much to do, have not allocated enough time. In addition, it is mentioned that teachers are not aware of how much work it is to learn to program. The others think that the workload is appropriate and emphasize that the flexibility (both in terms of content and times for submitting the exercises) helps to even out the load.

Final Assessment/Collaboration. The examination form in the course is project-based, where several people can collaborate. This seems to many teachers (8) to be positive. E.g., "I really liked the shape of the final assessment, that it is a project. I think it's incredibly significant because there are a lot of people taking exams right now, and I see how stressed they are, and they memorise, memorise and memorise, and go to the exam, then the exam is finished, and they may forget everything shortly after the exam. But here, as I go through the exam, I learn at my leisurely pace, writing and noting every detail, and afterwards, I can use this project as a finished product. And if I want to repeat and come back, then I can come back, print it, and use it right away. I think this is fantastic with this project" (T12). Many teachers felt that collaboration during the course allowed them to use their strengths and support each other as a group. One teacher suggested that an additional activity in the study could be for teachers to share and present the teaching plans that they have created to their peers. Teachers also report on good collaboration between instructors/teaching assistants, and many felt a low threshold for asking questions and got responses quickly.

4.2 Teachers' Perspectives on PD (Part B)

The second part of the interviews was related to teachers' reflections on their future PD in programming after completing the university's programming courses. By completing these courses, they have gained some experience and may have thoughts and opinions about how they envision the subject area's development. We asked teachers what they think is the best way of PD in programming for in-service teachers. Learning programming may be time-consuming, and we asked them how they want this process to be supported. Furthermore, we were interested in knowing their view of the development in the subject area and the students since the learning process can extend over a more extended period.

Teachers' Perspective on How to Learn to Program. All interviewees (16) expressed their opinion on how teachers should learn to program. We have identified 170 code references in transcriptions where teachers discuss different aspects such as the community of practice, collaboration and competency development through participation in courses, seminars, and self-initiatives. Thirteen teachers mention collaboration in the interviews and consider it to be an essential tool in learning programming. They emphasize the importance of collaboration with fellow teachers to share skills and competencies and motivate each other. Thirteen teachers talk about courses or seminars as a meaningful way to learn and teach programming. Many teachers mention also practising and learning by doing as an essential part of the process. Programming is time-consuming, and teachers need to find the required time to do this. One teacher says that programing should be like a 'hobby': "Learning programming should become a hobby. One must prioritise and spend time to learn and understand the concepts well enough" (T9). Nine teachers expressed that a teacher professional environment may be an important arena for learning to program. Educating teachers could happen internally at the school through courses, or that the school would allocate time for teachers to learn together at school.

Time. When we talk about the PD of teachers in programming, we keep coming back to the time issue. It is crucial to allocate enough time to practice and improve programming skills. Fourteen teachers talk about "time" as a constraint when learning to program. E.g., "The fact that schools must give time to learn this, I think, is essential. Because it does not come by itself, and you cannot expect anyone to sit down to do this on Saturday nights because this is too big for it" (T10).

Willingness to Learn More. Sixteen teachers talk about their desire to learn more and cope with the pupil's motivation to learn to program. E.g., "I could well imagine such an additional in-depth course, but the question is what you are allowed to do as an employer. The employer wants to lift now with a minimum of programming skills. So, we get a 15 ECTS credits course now and then we are supposed to know to program" (T1). Some teachers seem to experience that their work colleagues are unwilling or unable to learn to program and report that this applies especially to older teachers who have started planning their retirement rather than learn programming.

Role as a Teacher. Some teachers emphasize the importance of active learning. Pupils need to be guided in the right directions and also learn themselves during this process. One teacher explained that she was anxious about not having as much knowledge about programming as necessary. At the same time, the teacher felt that there might be some advantages because the pupils might become more motivated by knowing that they are more knowledgeable on a subject than the teacher is. E.g., "I like knowing well what I am going to teach. So being a week in front of the students, it's not something I like. But I think it is also a considerable value because if the students discover that they know something that the teachers do not know, they become extra engaged in it and want to show it off and get even more motivated to learn even more. So I feel a little uncomfortable, but I know it may be excellent" (T10). Fourteen teachers talk about their experience when learning to program together with pupils, and those who have not experienced it but reflect on the possibility of it happening in the future. Twelve teachers described that they have been in situations where they have had to learn together with their pupils. E.g., "I think that I will come across that. Well, programming is relatively new for many pupils. Still, eventually, we will work with it throughout primary school. When they get to high school, they will probably have a completely different base of competence than what they have now, so I imagine that it will not be long before learning something with the pupils" (T10). Few teachers mention that they might have unrealistic expectations about what they need to know and put too much pressure on themselves, probably lowering their confidence. E.g., “I think, in the textbook, the author says that when we learn to read and write, none of us expects us to be world-famous as Ibsen or Byron, or who at any time. We learn to read and write because we want to write a letter to Grandma. Why do we look at programming so that if we are to learn to program, then we must be like Steve Jobs? And it helped me lower my shoulders and the way the instructor explains here in this course. Very calm with a good pace, do not expect you to have fantastic prior knowledge, we begin” (T12). Another perspective that teachers (6) have talked about is the willingness of the school administration concerning making necessary time and resources available for teachers. Some schools are extra supportive, while others teachers feel they do not have enough back cover.

5 Discussions

Teachers have shared their experiences and highlighted some essential points from their perspectives on the PD in programming at the university. They seem to be comfortable with how the program is set up, i.e. the first part (7.5 ECTS) focuses on teaching them basic knowledge in programming. In comparison, the second part (7.5 ECTS) focuses on didactics, i.e. how to teach programming. Teachers emphasize the importance of *connecting programming to the relevance of their subject areas* from the outset, even if the focus is on fundamental concepts, such as using relevant examples and assignments from the subject areas in upper secondary schools. This can contribute to increased learning outcomes and more significant engagement.

Another aspect that teachers highlight is *the flexibility* of the program. Since time is a constraint, teachers are concerned with the shortest path to the goal, i.e. learning only what is relevant to their situation in the first place to get started with teaching programming in a short time. This offers a flexible solution where participants can set up their learning paths. Several dimensions can make the program’s flexibility valuable: On the personal level, participants vary in terms of their prior knowledge level, attitudes, and motivation; on a pedagogical level, they teach different subjects, school level and focus areas; and in a school context, the support by leadership and technical infrastructure may vary. There is a need to support teachers with *technical issues*, considering that their available infrastructure might vary. This problem could be addressed with (hands-on) tutorials about the more common technologies and troubleshooting sessions.

Teachers also mention the *project-based learning approach and the bridge model* [20] used in this PD program. They develop a teaching plan that they can use directly in their teaching after completing this program. This may increase their self-efficacy in teaching programming. Regarding teachers’ PD in programming after completing the university’s programming courses, we asked them to reflect on what they think is the best way of PD in programming for in-service teachers. What seems to occupy teachers the most is *establishing a community of practice, collaboration and development through participation in courses, seminars, and self-initiatives*. They emphasize the importance of collaboration with fellow teachers to share skills and competencies and motivate each other. Continuous development through collaboration may be an essential factor that may be facilitated at several levels: between teachers, teachers and pupils, school & teachers internally and externally. Even for teachers who have attended an extensive program, most of them still feel the need to continue their learning process with additional courses, short term workshops, a CoP, collaboration at school etc. It is therefore essential to see any PD as a step in a life-long learning process.

Teachers are concerned with several challenges when learning to program, and the most important one is the *lack of time* that may have several negative aspects. Learning to program can be time-consuming, and in-service teachers need to develop their programming knowledge in parallel with other duties, making it even more challenging. The lack of time can lead to the whole PD becoming complicated and may demotivate the individual. Therefore, teachers must receive the necessary *support from the schools* (both in terms of time and financial support for purchasing equipment, etc.). Good PD is needed to achieve the right level of competence, and that this requires adequate support from the school. Teachers getting the time to attend courses at the university level is beyond what a single school can generally manage. This implies that there is a need for a higher-level commitment. In our case, schools get public support for it. We think it is important to underline that proper qualification of teachers cannot be achieved without a commitment that does not put all the weight on the shoulder and goodwill of individual teachers and schools. Another point is the *expectations of schools* for these types of PD programs. Many schools might think that PD in this area is something to do once due to changes in the national curriculum, rather than as a continuous effort that requires time, dedication, and a clear strategy.

Another practical issue is that PDs that follow the school calendar can create problems for teachers as they already follow their pupils through the same calendar. Thus, the exam may coincide with the exam that the students should have, which complicates the situation for teachers. This reinforces the need for a flexible PD program, but at the same time, we see that more teachers express that there will be challenges in finding out what to do as the scope becomes overwhelming. Instructors must therefore follow them closely. The *role of the teacher* may not be the same anymore. They point to some area that requires changes in the way they think and teach. Teachers might have *unrealistic expectations* about what they need to learn and put too much pressure on themselves, probably lowering their confidence. There might be a need to scaffold teachers' reflection on the competence level that they want/need to achieve. Another point that several teachers emphasize is that active learning should be an essential part of *the teaching process*. Pupils need to get involved, and teachers do not have to worry about not knowing enough about the topic. They will, to a greater extent, learn and develop in the subject together with the students.

6 Conclusion

This paper presented the results from the analysis of sixteen interviews of in-service teachers attending a PD program about learning and teaching programming. Our main objective was to evaluate the PD program offered by the university and asked the research questions: Which strengths and weaknesses of the PD program provided are suited for in-service teachers when learning to program? Learning programming takes time;

how could this long learning process be supported? The main findings seem to be that teachers are comfortable with how the program is set up. They also emphasize the importance of connecting programming to the relevance of their subject areas from the beginning, establishing a community of practice, collaboration and development through participation in courses, seminars, and self-initiatives. They highlight the importance of collaboration with fellow teachers to share skills and competencies and motivate each other. Also, the follow-up initiatives and the need for schools to create supportive environments for cooperative efforts and CoP are essential factors. The role of the teacher may not be the same anymore. They point to some area that requires changes in the way they think and teach. Teachers might have unrealistic expectations about what they need to learn and put too much pressure on themselves, probably lowering their confidence.

The main contribution of this study is the evaluation of the PD program provided by the university in programming. Results may be applicable in a PD program with a similar context but not be easily generalised. All participants are from the same country but teach on different school levels and subjects. The study focuses on teachers who have been given by their school the time to improve their teaching. Further, teachers' perspectives on PD in programming may be seen in the context of the university course already being provided. The findings may be affected by the highly motivated students who participated in this study and were pleased with the course.

As part of future study, it might be interesting to study teachers' perceptions from different institutions in future studies—both teachers who do not participate in any particular PD program and those who do.

Acknowledgments. We thank the teachers who participated in the interviews. The work is partly funded by the Norwegian Directorate for Education and Training (https://www.udir.no) and the Norwegian Center for Excellent IT Education (https://www.ntnu.edu/excited).

References

1. Nouri, J., et al.: Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. Educ. Inq. **11**(1), 1–17 (2020)
2. Galaczi, E., et al.: The Cambridge English Approach to Teacher Professional Development (2017)
3. Qian, Y., et al.: Who needs what: recommendations for designing effective online professional development for computer science teachers. J. Res. Technol. Educ. **50**(2), 164–181 (2018)
4. Ring, B.A., Giordan, J., Ransbottom, J.S.: Problem solving through programming: motivating the non-programmer. J. Comput. Sci. Coll. **23**(3), 61–67 (2008)
5. Mozelius, P., Humble, N.: Lessons learnt from teacher professional development in programming. In: INTED 2020. IATED Academy (2020)
6. Monjelat, N., Lantz-Andersson, A.: Teachers' narrative of learning to program in a professional development effort and the relation to the rhetoric of computational thinking. Educ. Inf. Technol. **25**(3), 2175–2200 (2020). https://doi.org/10.1007/s10639-019-10048-8
7. Hiltunen, T.: Learning and Teaching Programming Skills in Finnish Primary Schools-The Potential of Games. University of Oulu, Oulu (2016)
8. Baxter, D., Simon, B.: Clash of the timelines: lessons learned from the front lines of CS education. In: Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment (2014)

9. Nordén, L.-Å., Mannila, L., Pears, A.: Development of a self-efficacy scale for digital competences in schools. In: 2017 IEEE Frontiers in Education Conference (FIE). IEEE (2017)
10. Menekse, M.: Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014. Comput. Sci. Educ. **25**(4), 325–350 (2015)
11. Darling-Hammond, L., Hyler, M.E., Gardner, M.: Effective teacher professional development. Learning Policy Institute (2017)
12. Cordingley, P., et al.: Developing great teaching: lessons from the international reviews into effective professional development (2015)
13. Desimone, L.M.: Improving impact studies of teachers' professional development: toward better conceptualizations and measures. Educ. Res. **38**(3), 181–199 (2009)
14. Dunst, C.J., Bruder, M.B., Hamby, D.W.: Metasynthesis of in-service professional development research: features associated with positive educator and student outcomes. Educ. Res. Rev. **10**(12), 1731–1744 (2015)
15. Sims, S., Fletcher-Wood, H.: Identifying the characteristics of effective teacher professional development: a critical review. Sch. Eff. Sch. Improv. **32**(1), 47–63 (2021)
16. Reding, T.E., Dorn, B.: Understanding the "teacher experience" in primary and secondary CS professional development. In: Proceedings of the 2017 ACM Conference on International Computing Education Research (2017)
17. Webb, M., et al.: Computer science in K-12 school curricula of the 2lst century: why, what and when? Educ. Inf. Technol. **22**(2), 445–468 (2016). https://doi.org/10.1007/s10639-016-9493-x
18. Karalis, T.: Cascade approach to training: theoretical issues and practical applications in non-formal education. J. Educ. Soc. Policy **3**(2), 104–108 (2016)
19. Gaddis, T., Agarwal, R.: Starting Out with Python. Pearson, London (2015)
20. Rouhani, M., Divitini, M., Olsø, A.: Project-based learning and training of in-service teachers in programming: projects as a bridge between training and practice. In: 2021 IEEE Global Engineering Education Conference (EDUCON). IEEE (2021)