Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Per Dahl Haagensen

Supporting in-service teachers of programming in secondary schools

A design science study on the impacts of designing, implementing and evaluating an internally organized professional development course in programming

Master's thesis in Natural Science with Teacher Education
Supervisor: Monica Divitini
Co-supervisor: Majid Rouhani

June 2021

NTNU
Norwegian University of Science and Technology

Per Dahl Haagensen

Supporting in-service teachers of programming in secondary schools

A design science study on the impacts of designing, implementing and evaluating an internally organized professional development course in programming

Master's thesis in Natural Science with Teacher Education
Supervisor: Monica Divitini
Co-supervisor: Majid Rouhani
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

In modern society, technology has gained enormous importance for human life. Technology has also brought with it new disciplines people must master. This has led to several countries introducing programming into their national curricula. In Norway, programming has been added to competence goals in subjects such as mathematics, science, arts and crafts, and music. This new inclusion in the curriculum has led to a need for continuing education and professional development of in-service teachers in programming. Based on this, there will be a need to research how to perform continuous education on in-service teachers. One possible way to increase the teachers' programming competence is to organize the professional development internally at the schools and investigate what consequences an internally organized professional development courses have for external or internal stakeholders who are designing these professional development courses. This master's thesis examines which internal and external elements influence the course design for professional development courses organized internally in a school.

The research is based on a flexible and qualitative design investigating the elements of an internally organized professional development course for teachers through design, implementation, and evaluation. This starting point forms the basis for the research question in this study: *What internal and external factors influence the design process of an internally organized competence development course for teachers in programming?*. Furthermore, the research wants to identify which design choices can increase teachers' learning outcome from such a course. The findings of the study are seen in the light of previous research on professional development, seven features of effective professional development by Darling-Hammond et al. (2017) and finally, the term teacher self-efficacy.

The results indicate that there are many aspects to consider when designing such a course. However, the results also indicate that internally organized competence development courses can be a good starting point for conducting effective and relevant professional development of teachers.

Sammendrag

I det moderne samfunn har teknologien fått en enorm betydning for menneskers liv. Teknologien har også ført med seg nye fagdisipliner menneske må beherske. Dette har ført til at flere land i verden har innført programmering inn i deres nasjonale lærerplaner. I Norge har programmering blitt lagt til i kompetansemål i fag som matematikk, naturfag, kunst og håndverk og musikk. Denne nye inkluderingen i lærerplanen har ført til et sterkt behov for etterutdanning og kompetanseheving av lærer i programmering. Det vil på bakgrunn av denne være et behov for å forske på hvordan denne kompetansehevingen kan gjøres på den måten som resulterer best utbytte for lærerne. En mulig måte å gjennomføre kompetanseheving av lærer på, er å gjennomføre de internt i skolene og undersøke hvilke konsekvenser et internt organisert kompetansehevingskurs har for eksterne eller interne aktører som skal designe disse kompetansehevingskursene. Denne masteravhandlingen undersøker hvilke interne og eksterne elementer som påvirker kursdesignet til kompetansehevingskurs organisert internt i en skole.

Forskningen er basert på et fleksibelt og kvalitativt design som har sett på elementene ved et internt organisert kompetansehevingskurs for lærere gjennom design, implementasjon og evaluering. Dette utgangspunktet danner grunnlaget for forskningsspørsmålet i denne studien: *Hvilke interne og eksterne faktorer påvirker designprosessen av et internt organisert kompetansehevingskurs for lærere i programmering?*. Forskningen ønsker å identifisere hvilke designvalg som kan øke læringsutbytte lærerne får av et slikt kurs. Funnene i studien blir sett i lys av tidligere forskning innenfor kompetanseheving av lærere, syv egenskaper til effektiv kompetanseheving av Darling-Hammond et al. (2017) og til slutt begrepet self-efficacy.

Resultatene indikerer at det er mange aspekter å ta hensyn til under design av et slikt kurs. Resultatene indikerer også til at internt organiserte kompetansehevingskurs kan være et godt utgangspunkt for å drive effektiv og relevant kompetanseheving av lærere.

Preface

It is strange to sit in an empty reading room writing these words, which concludes my studies in natural science with teacher education. The last couple of months have been chaotic. However, more importantly, it has been extremely interesting, and I have definitely learned a lot. I am very grateful for being able to lay down the work required and keep the focus and motivation up to finish this project.

This master thesis would not have been introduced to the world without the excellent support and guidance I received from my supervisor Monica Divitini throughout the process. I have really enjoyed every discussion we have had, and I am thankful for you not giving up on me in times of negativity. I would also like to thank my co-supervisor Majid Rouhani for providing me with data and useful insights throughout the process.

A special thanks are also needed to give to the teachers participating in the course and the interviews. This thesis would not have come to light without you.

Trondheim. June 2021
Per Dahl Haagensen

Contents

Figures

Tables

Acronyms

DSR Design Science Research. 5, 19, 20

ICT Information and communications technology. 3

IDE Integrated Development Environment. 32

MOOC Massive Open Online Courses. 13, 70

NSD The Norwegian Centre for Research Data. 41

NTNU Norwegian University of Science and Technolog. 3

PD Professional Development. 1–4, 7–9, 16, 22, 30, 37–39, 62–65, 67–70, 73, 76, 78

PjBL Project-based Learning. 23, 31, 37

STEM Science, technology, engineering and mathematics. 70

TSE Teacher Self-Efficacy. 4, 5, 7, 8, 16, 18

UDIR The Norwegian Directorate of Education and Training. 28, 31, 32

Chapter 1

Introduction

In the last couple of decades, technology has gradually taken a more significant part of human's everyday life and is changing the society we are living in. The 21st century is categorized by the presence of technology in everyday life. The entrance of technology in modern society has led to the need to acquire new knowledge. When computers succeed in affecting most aspects of modern society, people need to be adaptable to future technology. As technology is advancing, there is an increasing need to provide an education that can educate students interacting with these computer-related instruments. Computer Science (CS), the study of computers and computational systems, is becoming a much-needed knowledge in the future. Even though most people use technology every day, there are very few of them who understand how they are made or programmed. As a result of the technological upspring in society, programming has been implemented in national curriculums worldwide. Kunnskapsløftet 2020, known as Fagfornyelsen, is the new national curriculum implemented in Norway in 2020 ('LK20', 2020) and have implemented programming as their own competence goals in the subjects mathematics and science from primary school. However, most of the teachers who are going to teach programming in the near future have no prior programming experience. A consequence of this is that there is an emerging need for Professional Development (PD) in programming for in-service teachers. How can we teach these teachers to get adequate competence to perform tuition? Who will conduct this PD?

From the literature, there is evidence of the existing need for a more broad effort of informatics educators to educate people why informatics should be

known to all, how it should be thought, what topics should be thought, and for whom the teaching of informatics is meant (Holmboe et al., 2001). The Committee on European Education also recommends that every student have access to some informatics education in the school system. The committee also points out that the teachers who educate these subjects in informatics should have a formal competence in informatics (on European Computing Education (CECE), 2017). The same report states that there is a lack of teachers with formal competence in informatics and recommends that this shortage have to be solved by performing continuous teacher training for in-service teachers and hiring more informatics teachers. The teachers who are hired as informatics teachers should be hired based on the exact requirements as other advertised teaching positions and should meet the exact requirements regarding competence (on European Computing Education (CECE), 2017).

In the initial planning process of the new national curriculum in Norway, there were many discussions about whether programming should be implemented as a freestanding subject or be baked into consisting subjects (Sevik and fl, 2016). Eventuality, there was decided that programming should be integrated into established subjects. Sanne et al. (2016) argues that embedding programming into other subjects could lead to it being down-prioritized as well as ending up with many stressed teachers that do not feel they have enough competence for teaching programming. The arguments of choosing the current implementation were that there was no interest in implementing programming as a standalone subject and because other Nordic countries have implemented a similar model (Bocconi et al., 2018).

1.1 Motivation

We can not change the fact that programming eventually ended up being embedded into mathematics and science. The way forward must consist of finding the right strategy of achieving adequate programming competence for the teachers who shall teach programming in the future. Many of these teachers will have to apply to continuous education to get prepared for teaching. Today, the conversion phase from the old curriculum to the new curriculum is still in the early days. To explore how to prepare these teachers to teach programming in their classroom, literature on PD can help to get an overview of the research field. PD is in the Cambridge dictionary as training that is given to managers and people working in professions to increase their knowledge and skills (in this case, training in-service teachers in programming).

The addition of programming in the Norwegian curriculum led to many teachers needing programming competence in order to teach it in the future. Studies from other countries show that the teachers did not feel confident teaching

programming at the time programming was implemented in the curriculum (Waite, 2017). The teachers expressed they had a lack of sufficient knowledge of computers and programming and that they have had too little formal teacher training.

In the newly revised Norwegian curriculum, there is included a chapter called *professional development and school development* ('Professional environment and school development', 2021). The chapter explains hoe schools should be professional environments where there reflect around common values and develop their practice. Every school community is different, so the way schools implement PD is dependant on e.g., prioritization from the administration, teachers attitude and the pupils at the school. As research show, *collaboration* between teachers utilized properly within schools, can contribute to a more active learning environment and can positively contribute to pupils' achievements (Darling-Hammond et al., 2017). Because of this, to research the affects of designing a internally organized PD course for in-service teachers would be a interesting aspect to elaborate on. This could give indications of how a internally organized course can be designed in the context of specific schools, and which affects, both positive and negative, this can have on the teachers' outcome of the course.

With the motivation presented, this study will support the educational system and teachers on the road to teaching programming in the future. As there is little previous literature on internally organized PD programs of in-service secondary teachers in programming, a paper representing a designed PD course design would be highly relevant for future teachers and other involved stakeholders. The design focuses on the competence goals found in the Norwegian curriculum in the subjects mathematics and science. The thesis aims at identifying challenges regarding designing, implementing and evaluation such a course and get an understanding of which factors in the design, implementation and evaluation phase is affected by the course being organized internally. The study also acknowledges that the results are taken from six teachers' experience and that the results may differ from the number of participating teachers or other external factors.

1.2 Context

For the last five years, I have been studying Natural Science with Teacher education. My main field of study is Information and communications technology (ICT). This research will mark the end of my studies and will be submitted as my master's thesis in Informatics at the Department of Computer Science at the Norwegian University of Science and Technolog (NTNU). This thesis is written under the supervision of Monica Divitini and co-supervisor Majid

Rouhani.

The study is designing, implementing and evaluating a PD course in programming for in-service teachers at a secondary school in Norway. The course consisted of four lessons and was completed over a two-month period internally at the participating school. The teachers who lectured mathematics and science, the science section, were the participating teachers.

1.3 Problem and Research questions

In the next couple of years, teachers' from many countries will use and teach programming. One of the significant challenges with this addition in the curriculum is that many teachers who now will teach programming do not have any formal competence in teaching programming. The possible solution for this problem is to perform PD courses to continue the education for teachers. As the new curriculum in Norway was implemented for most teachers in 2020, there are still many unanswered questions regarding this transition. What is the most effective way of performing continuous training in programming on the teachers? What role does every individual school have to ensure the teachers obtain adequate knowledge to teach programming? Many questions come to mind, and this study will investigate these questions in the light of the research questions presented below.

Main research question: *Which internal and external factors affects the design process of a internally arranged professional development programs in programming for in-service teachers*

This question will be further investigated through the following sub-questions:

RQ1 Which elements should be considered when designing a programming teacher course?

RQ2 Which course design can help reaching the learning objectives of such a course?

RQ3 How did the course participants experience the designed course?

The sub-questions discuss elements that arose through the designing, implementing, and evaluating phase of the course. These elements will be discussed in light of seven features of effective PD presented by Darling-Hammond et al. (2017), and explore how an internally organized course can be used to promote these features. The seven features of effective PD can be found in section 2.1.2. The findings will be discussed in light of Teacher Self-Efficacy TSE. TSE is a term which is important to discuss in the domain of PD in programming

as teachers sense of TSE can result in teachers implementing new strategies in their practice and manage a high-quality classroom environment(Zee and Koomen, 2016). A further elaboration on the term TSE can be found in section 2.2.

1.4 Method

The study is based on a flexible design, using a Design Science Research (DSR) approach. DSR creates innovation through analysis, design, and implementation (Hevner et al., 2004). The study is qualitative, and the data collection of this study is retrieved from semi-structured interviews with six in-service teachers. The data were analyzed using the constant comparative method, retrieved from grounded theory (Glaser and Strauss, 1967). The data were analyzed by the researcher using Nvivo.

1.5 Ethical aspects

When conducting real-world, there are many ethical concerns the researcher must consider. This is also especially important when researching in school with its teachers. The ethical aspects are not only occurring in the starting phase of the research but are relevant throughout the whole process (Robson and McCartan, 2017). It is important to avoid bringing any harm, stress, or anxiety to the participants. The Norwegian National Research Ethics Committees in Norway have made guidelines for research ethics in the social sciences ('Guidelines for Research Ethics in the Social Sciences, Humanities, Law and Theology', 2019).

The purpose of research is to pursue what is the truth, but it is not possible to fully achieve the truth ('Guidelines for Research Ethics in the Social Sciences, Humanities, Law and Theology', 2019). In studies in the discipline of social sciences, there is much interpretation in the research process, and different interpretations and theoretical backgrounds from researchers can result in a different interpretation of the same material ('Guidelines for Research Ethics in the Social Sciences, Humanities, Law and Theology', 2019). This is also present in this study, as the researcher has analyzed and interpreted the study's data material in terms of his values. There has therefore been considered important for the researcher to reflect on how his attitudes and values could affect the interpretation of the data material. The researcher cannot be completely impartial, and it is, therefore, crucial for the researcher to have a thought about these aspects in the research (Postholm, 2010).

This study has also conducted six interviews with teachers who attended the course. When dealing with individuals, the researcher must protect the individual's human dignity, privacy, consent, and confidentiality. Another aspect of ethics worth mentioning is the confidentiality agreement each teacher has with the school. This agreement prohibits teachers from giving out personal information of any individuals related to the school to outsiders who do not have a connection with the school. In other words, the data retrieved from the interviews had to be carefully examined to secure that all the information was in line with the ethics. Furthermore, under the interviews, the teachers could also give statements towards school administration or other school management's political aspects, which the researcher must be careful of using. This is especially important in this study, as other teachers could recognize which person said what under the interviews.

1.6 Results

The results from this study indicate that various internal and external factors affect the design process of an internally organized professional development course in programming. The study indicates, for example, that elements such as *time* and *teachers' previous knowledge and experience* are vital factors that influence the design of a teacher training course.

The study's results indicate that course designs emphasize the usage of active-learning approaches as a design that contributes to increased learning. However, the teachers also indicate that the course activities must have a clear content focus and make it easy for them to see connections with the activity and their teaching practice.

The teachers' experience of the course was varied. The teachers indicate the main reasons for this as the course's *level of difficulty* and the *duration* of the course. On the other side, the teachers were positive towards organizing the course internally. The results also indicate that a project-based activity approach could contribute to increased teacher collaboration.

Overall, the study indicates that there are many elements to consider in continuous education in the programming of in-service teachers and that internally organizing the education has positive effects on the teachers' outcome.

1.7 Outline

The outline is added to help the reader get an overview of the structure of this thesis. The thesis consists of seven chapters. Chapter 2 will introduce the reader to the term Professional Development, identified effective features of Professional Development and look at some research with similar approaches to this study. A definition of the term Teacher Self-Efficacy (TSE) will also be presented.

Chapter 3 presents the research method used in this study, followed by a presentation of the course design in chapter 4.

Chapter 5 presents the researcher's analysis on the conducted study interviews with six teachers. This chapter includes both the method of analysis and a presentation of the results. The findings from this study will be discussed in light of relevant in chapter 6. Chapter 7 will answer the research questions, reflect on this research's limitations, and present recommendations for further work.

Chapter 2

Theory

This chapter aims to give the reader an overview of relevant literature used in the research. First, the concepts of Professional Development (PD) is presented, followed by introducing the seven features of effective PD from Darling-Hammond et al. (2017). Then, an overview of previous literature on PD in the domain of programming will be presented. At last, the term Teacher Self-Efficacy (TSE) will be presented.

2.1 Professional development

Professional Development, also known as staff development, in-service training, or continuing education, has been a term researchers have had an increasing interest in studying PD and its effects on teachers further. Cambridge's online dictionary defines the term PD as "training that is given to managers and people working professions to increase their knowledge and skills" ('Professional development', 2021).

There are several occupations where Professional Development is prioritized, e. g. doctors, lawyers, teachers, and engineers, to mention a few. The variety of occupations participate to PD to learn and apply new skills which are required or beneficial to use in the future (Mizell, 2010). Education defines PD as; schools' and districts' used strategy to ensure the educators have a continuous strengthened practice throughout their career (Mizell, 2010). PD in education is a formal process and embraces conferences, seminars, workshops, or other

arranged courses. While this is the most common view of PD, as a specialized type of training offered by some specialists at a limited time and location, Han et al. (2015) refers to PD as a continuous process that occurs all the time at schools everywhere. This PD is in terms of teachers that improve their quality of instructions. This development will only happen if the teachers' are willing to self-reflect on their teaching practices. Studies have shown that only a few teachers are willing to modify the design of their instruction, and as a result of this, mandatory PD is recommended (Guskey, 2003).

One of the most significant challenges school management face towards the teachers is motivating them to incorporate their best teaching practice. The schools' issues regarding motivating the teachers result from teachers who constantly ignore implementing innovations added to the curriculum and a reluctance to change their practice, and uninterested in implementing the newly added competence goals in the curriculum to their teaching practice (Semadeni, 2009). There are several reasons why it is difficult to increase the teachers' effectiveness. First of all, most teachers are already overwhelmed in terms of social and academic responsibilities (Hester et al., 2020; Semadeni, 2009). Second, some teachers have previous experience with newly added additions to the curriculum come and go, resulting in a resistance towards implementing new programs in their teaching (Semadeni, 2009). Third, earlier arranged workshops and other training formats have ignored adult learning principles, resulting in limited learning outcomes for the teachers. Finally, the teachers receive the same recognition and pay as other colleagues, resulting in enthusiastic teachers losing their motivation to improve and adjust their teaching (Capraro, 2014; Darling-Hammond and Youngs, 2002; Hester et al., 2020; Semadeni, 2009).

2.1.1 Effective known professional development practices

Garet et al. (2001) presented a paper discussing features that contribute to effective PD. In the last couple of decades, several studies on PD and teacher training have emerged. The studies have tried identifying core features that make PD effective. An "effective professional development program is structured to provide content knowledge, explicit modeling, opportunities and collaborative planning time" (Cateté et al., 2020, p. 2). The most effective PD duration is proven to be consistent throughout the year and changes as the teachers' needs also change (Desimone and Garet, 2015). In 2001, Garet et al. (2001) published the first large-scale empirical study that compared the effects of having different characteristics of PD of teachers' learning. The study is trying to identify features that significantly change the teachers' classroom practice, knowledge, and competence. The features were divided further into distinct groups; sponsorships, structural features, core features, and teacher outcomes. The study's data collection was retrieved by receiving responses from 1027 German teachers. Figure 2.1 shows the model by Garet et al. (2001), after

analysing the survey answers. The model illustrates every feature within each category. The model uses quantitative data retrieved from teacher interviews. The paths in the model indicate which features are influenced by another feature. The paths are direct, the numbers relating to each path represent the standard regression coefficient (β).

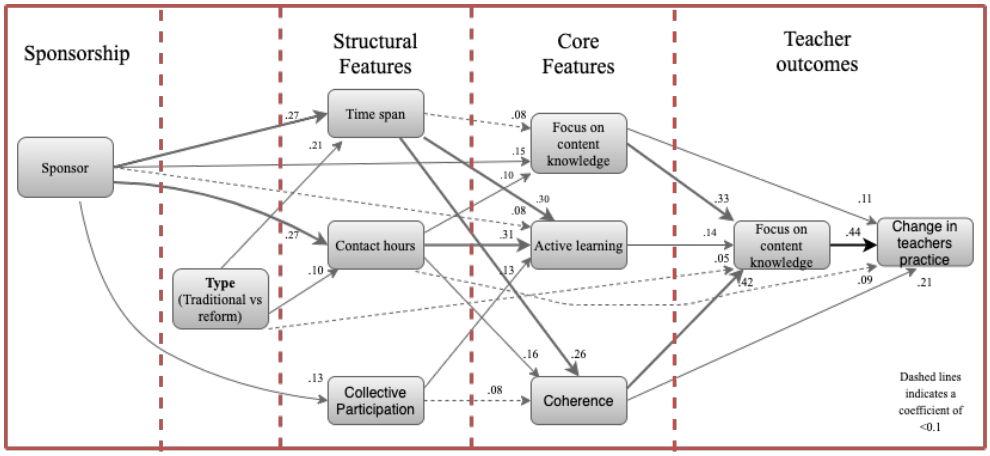


Figure 2.1: The model presented by Garet et al. (2001). Based on results from a questionnaire

The study observed three core features which met the established requirements mentioned above, **focus on content knowledge**, **opportunities for active learning** and **coherence with other learning objectives** (Garet et al., 2001). With these features in mind, Garet et al. (2001) also argues that also other structural conditions can affect the teacher's learning in PD. The contributed features address the three aspects; a form of activity, collective participation of teachers, and the duration of the activity. The same features were identified by Desimone (2011) some years later as well. Desimone presented a conceptual framework containing five features that should be in focus in any program of PD. The three core features, presented by Garet et al. (2001), contribute to the introduced conceptual framework. Compared to the core features presented by Garet et al. (2001), Desimone (2011) added *duration* and *collective participation* to the core feature list. The conceptual framework aims to offer a foundation for studying whether or not a PD program is effective. In practice, Desimone (2011) suggests using the framework to test three outcomes: Do teachers learn? Do they change their practice? Does the students' achievement increase as a result? (Desimone, 2011)

2.1.2 The seven features of effective PD

After Garet et al. (2001) paper was published, further studies looking into features of PD have been conducted. The presented features from Garet et al. (2001) and Desimone (2011), are used as a basis and have been further developed. Darling-Hammond et al. (2017) have performed a systematic review of PD studies. The study identified seven key features that can increase teacher knowledge as well as improving student outcomes. These seven factors are: 1) `content focus`, 2) `active learning`, 3) `collaboration` 4), `use of model and modeling`, 5) `coaching and expert support`, 6) `feedback and reflection`, 7) `sustained duration`. Compared to the three features Garet et al. (2001) presented sixteen years before, the similarities are present.

Content focus is the first feature mentioned by Darling-Hammond et al. (2017). A content-focused PD program includes activities and material directly connected to what the teachers teach. By embedding a content-focused PD program, it lays the foundation to addressing pupils' different needs in different settings (Darling-Hammond et al., 2017). The key features of having content-focused PD models "illustrates teacher professional learning opportunities designed for teaching content to specific student populations with targeted strategies to support their achievement" (Darling-Hammond et al., 2017, p. 7). In other words, it is important to connect the theory to the teachers' teaching practice.

Active learning is the second category of effective teacher PD programs. This feature focuses on the teachers' involvement and collaboration during a PD program. In such courses, teachers should be allowed to get involved, such as observe other teachers, give and receive feedback and presentations, contrary to other passive learning strategies (Desimone, 2011). In addition, the PD participants are more likely to engage with the content if the learning is meaningful to them (Wei et al., 2009).

Collaboration is the third presented feature. PD implements collaboration in several ways, from small one-to-one groups to schoolwide collaboration. In Darling-Hammond et al. (2017), 32 of the 35 reviewed studies, incorporated some elements of collaboration. A PD program that utilizes effective collaborative structures for teachers to learn and explore together can positively contribute to the students' achievements (Darling-Hammond et al., 2017). The characteristic considers the nature of collaboration and the way it supports professional learning (Bates and Morgan, 2018).

Use of models and modeling is the fourth presented feature and addresses the models and modeling used in PD. There are several different models to use, e.g., video lesson and curriculum materials including sample assessments and student work samples, and aims to help the teachers to obtain a vision of practice on which to improve and adjust their learning and growth (Darling-

Hammond et al., 2017).

The fifth feature, *Coaching and exert support*, and indicates the importance of coaching and exert control from the PD instructor. In education, previous literature has concluded that there usually are other educators or employed teachers at the respective school who prepare and lecture the PD content (Darling-Hammond et al., 2017). Previous research provides evidence that teachers who receive coaching are more likely to explore their desired teaching practices and implement them more in their practice than other traditional PD approaches. Involving participating teachers to support effective implementation of new curriculum or tools can be a supporting factor of effective PD (Gallagher et al., 2017).

Feedback and reflection is the third presented feature of effective teacher PD. Overall, feedback and reflection are critical components of adult learning theory. When effective teaching PD programs implement practices of generating feedback and reflection from PD participants, it allows them to share positive and constructive instances of their practices. Furthermore, the feedback and reflections can be concerning the models used in the PD program (Gallagher et al., 2017). In conclusion, implementing a PD program witch focuses on feedback and reflection can contribute to creating richer environments for teacher learning (Darling-Hammond et al., 2017).

The last presented feature of effective PD is *Substained duration*. The literature shows that a PD program with a short duration does not lead to the participating teachers changing their practice. Regarding what is considered the threshold for the duration of PD models, previous research has not identified concrete guidelines regarding adequate PD duration (Darling-Hammond et al., 2017). The PD program should promote learning over time, both within and between the PD sessions, and can contribute to the participants getting more hours of learning as a result of participants doing further research in between the PD sessions (Darling-Hammond et al., 2017).

Other studies have also identified similar features, and *"Addressing the challenges of a new digital technologies curriculum: MOOCs as a scalable solution for teacher professional development"* by Vivian et al. (2014), is an example of such a study. The features presented in this study focused on helping teachers link new knowledge presented in PD to their existing knowledge, job responsibilities, and the need to practice new knowledge in a job-related context (Vivian et al., 2014). Other studies do also emphasize the importance of a collaborative learning environment in PD that provides an active listening experience (Knapp, 2003). Other studies call attention to the importance of reflective learning. The importance of reflective learning is emphasized as "intellectual and effective activities in which individuals engage to explore their experiences in order to lead to new understandings and appreciations" (Boud et al., 2013,

p. 19). In Bond's research, he claims that when new ideas are brought to consciousness, teachers can evaluate them and decide whether or not they will adopt them in their future courses. A possible consequence of introducing new and unknown materials that go against the teachers' prior knowledge and beliefs without acknowledging it could contribute to limiting learning gains of the PD participants (Boud et al., 2013).

2.1.3 PD research in the domain of programming

The entry of programming in curriculums worldwide has led to an increased need for PD programs educating in-service teachers in programming. The search of how to utilize the potential of the computer in education became a central focus point in research after its availability opened for the public (Solomon et al., 2020). As long back as the 1960s, research on developing programming tools and languages for children existed. The programming language *Logo* was invented by Seymour Papert, Wallace Feurzeig, Daniel Bobrow, and Cynthia Solomon in 1966 (Solomon et al., 2020) and became at its release the first-ever programming language designed for children. The language was invented as a tool for pupil's learning and exploration and was designed primarily with mathematics in mind. The invention of Logo was an ambitious project and had features which were impressive at the time, such as compatibility to build and program LEGO robots using the Logo language (Solomon et al., 2020).

"*A report from 2018 on the Nordic countries approach to introducing computational thinking and programming in compulsory education*", (Bocconi et al., 2018), they review each country's approach of introducing programming in the curriculum. The study call attention to two main areas; algorithmic thinking and programming of computers and physical objects and programming in visual and text-based environments (Bocconi et al., 2018, p. 21. The Nordic countries implemented programming in existing subjects instead of making a new one. Therefore, the content and tools used within the in-service teacher courses are very similar.

Finland was one of the first European countries to implement algorithmic thinking and programming as a mandatory, interdisciplinary activity from grade 1 of school (as of 2016). One of the main aspects of Finland's implementation is to promote pupils to develop problem-solving skills in the context of real-life problems (Bocconi et al., 2016). In 2017, a design-based research study by Partanen et al. (2017) organized a programming Massive Open Online Courses (MOOC) for in-service teachers in Finland, aiming to support Finnish teachers learning programming with material prepared by more experienced teachers or peers. The organized course was based on learning teachers' programming using the programming language, *Racket*, which is designed es-

pecially for teaching purposes. The study presents results from two individual conducted teacher training courses conducted during Autumn 2015 and Spring 2016. The study explores what the researchers learned from organizing the course, how the participating teachers experienced the course, and how the course affected the teachers' self-efficacy in teaching programming (Partanen et al., 2017).

Partanen et al. (2017) further explains how the teachers were willing and positive towards learning programming. The teachers stated they especially appreciated the course's exercises tailored to fit in authentic classroom settings (Partanen et al., 2017). The feedback contributed by the participating teachers Partanen et al. (2017) received on the course was more favorable compared to a previously performed beta-course. The researchers reflect that this may result from establishing a more reasonable level of difficulty and workload than the last course. Up to 369 teachers participated in the conducted beta-course and based on the teachers' feedback on the course, the next course was adjusted accordingly. The teachers' feedback revealed a too high workload in the first course, as one of the implemented tools was more challenging than anticipated. Partanen et al. (2017) express how the topics used in the course have to be organized based on difficulty; the exercises must have a connection with the introduced topics and adequately start simple and then proceed with more advanced topics later. The teachers' made artifacts were by the teachers considered a motivational and useful task the course implemented. Overall, the study suggests the course improved the teachers' self-efficacy, thus amplifying the importance of researching the course's long-term effects. "In the effort to provide effective in-service training, the improvements of the learning platform and fine-tuning the course material should be continuous" (Partanen et al., 2017, p. 12).

The mixed research study "*An investigation of the effects of programming with Scratch on the pre-service IT teachers' self-efficacy perceptions and attitudes towards computer programming*" by Yukselturk and Altiok (2017) conducted in 2016 a research study regarding PD programming programs, exploring how using Scratch, a block-programming language, affects the participating pre-service teachers' perceived self-efficacy and attitudes towards programming. The study was conducted in a research university in Turkey and aimed to understand basic programming and develop games using Scratch for the course participants. Thus, the study's research question explores how obtaining experience in Scratch affects pre-service teachers' perceived self-efficacy and attitude towards programming. The course had a duration span of five weeks, and as a part of the course requirements, the students were given two assignments, a mid-term, and a more extensive project focused on design, develop and present a game using Scratch. The study's data collection includes qualitative and quantitative data retrieved from interviews and survey responses,

respectively. Results from the study indicate that the Scratch environment significantly affects the pre-service teachers' attitude regarding some activities, promoting a feeling of enjoyment and comfort in the course and activities where the students feel capable of writing long and complex programs (Yukselturk and Altiok, 2017).

Furthermore, the negative attitudes some students had regarding programming before the course decreased after their experience with Scratch in the PD program (Yukselturk and Altiok, 2017). The teachers describe Scratch as an exciting tool that made them feel more enthusiastic about programming than other programming courses they had attended, using different programming languages. In addition, the use of Scratch and block-programming eliminated the syntax errors the teachers previously had experienced with other programming environments, and it gave more visually pleasing feedback to the user is making programs. On the other side, according to the teachers, the teachers perceived the Scratch environment to be designed towards beginner programmers and, as a result, not a beneficial environment to use after the pupils have achieved a certain level of competence (Yukselturk and Altiok, 2017). Furthermore, during the interviews, the teachers also stated that the Scratch interface not showing any lines of text-based code might result in teachers facing challenges when adapting their acquired learning from the course to new and unused programming tools later (Yukselturk and Altiok, 2017).

In 2020, Rouhani et al. (2020), published the study "*Teaching Programming in Secondary Schools: Stepping and Stumbling Stones*". The study is motivated by identifying challenges secondary schools' in-service teachers face that negatively affect their teaching and the elements that promote learning and motivation, respectively referred to as stumbling stones and stepping stones by the researchers. The in-service teachers participating in the study were attending a university-level programming teacher training course. The connected in-service teacher training program is offered to in-service teachers who soon implement programming in their teaching. As a final assessment of the teacher training program, the course participants delivered reflection notes, which were used as the data collection of the study (Rouhani et al., 2020). The study contributes by identifying the so-called stumbling and stepping stones of the conducted course. As other studies have shown, Rouhani et al. (2020), identifies stumbling stones like *time*, *interdisciplinary*, *Varying level of knowledge and motivation among students*. When comparing the identified stumbling stones to the effective features of PD by Darling-Hammond et al. (2017), the similarities are obvious. The similarities could indicate that even though effective PD features are identified, they are challenging to implement thoroughly in practice. The stepping stone elements retrieved from the data analysis were motivation and collaboration, and community participation. The course attendants connected their source of motivation to the excitement of teaching something new, and

they sense a feeling of perceiving knowledge about a topic which they perceive as highly relevant for society and for the future job market (Rouhani et al., 2020). The stepping stone *collaboration and community participation* includes both cooperation with peers and students, as well as the use of resources from different resources as a starting point for their projects (Rouhani et al., 2020).

The results from the presented study will be compared to the findings of this research and define the common and the uncommon themes they share. Furthermore, as these studies do not use the same design or research methods, it could give indications of which elements from each study contribute to making PD programs high-quality and effective.

2.2 Teacher self-efficacy

Teacher Self-Efficacy (TSE) is a term that has been a scorching topic in cognitive psychology research since the work of the Rand Corporation in the late 1970s (Zee and Koomen, 2016. A few years later, in 1977, Bandura published the paper *"Self-efficacy: Toward a Unifying Theory of Behavioral change"*. In this paper, Bandura is defining perceived self-efficacy as a person's belief in one's capabilities to organize and execute the courses of action required to produce a given result (Bandura, 1977). In his paper, Bandura separates outcome expectancy from efficacy expectancy. He defines outcome expectancy as a person's estimate that a given behavior will lead to a certain outcome (Bandura, 1977). Banduras reasoning for why these are differentiated is that even though individuals believe that a particular action will produce a particular outcome, this information does not influence their behaviors if they are in doubt whether they can perform the required actions (Bandura, 1977). A visualization of the difference is visualized in 2.2 Even though Bandura's context was in terms of treatment; the term self-efficacy is still an important term in every human endeavor. Therefore, the definition of self-efficacy is also valid in terms of teacher's self-efficacy and is defined as a teacher's beliefs in his/her capabilities to perform certain actions to produce a given outcome.

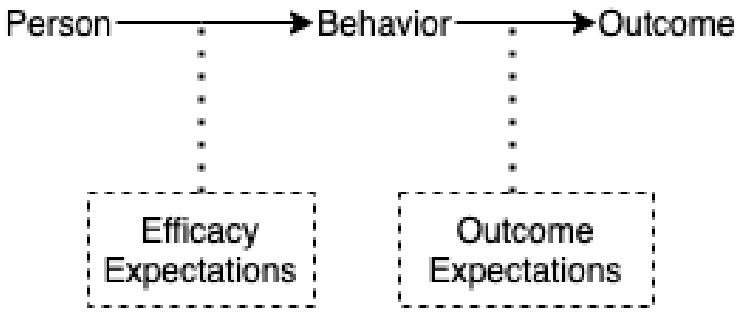


Figure 2.2: Visualization of the difference between Efficacy Expectations and Outcome Expectations

Historically, the foundational tenets of TSE have primarily been based on Rotters' (1966) attribution-based theory of locus, and Bandura's research on TSE (Bandura, 1977; Rotter, 1966) has made a significant impact on the TSE research (Zee and Koomen, 2016, p. 193). Rotter assumes in her work that individuals generally differ in their perceptions of whether outcomes result from external factors or a result of internal factors. Examples of external factors are factors such as sheer luck, fate, or others, while internal factors deal with the individual's actions (Rotter, 1966, p. 25). Even though Bandura's research was built strongly on Rotter's theory, he disagreed that control expectancies only is influenced by individuals' behavior. Bandura also believed that individuals' perceived capabilities or self-efficacy to perform those behaviors in particularized domains (Bandura, 1977, 1986, 1997; Zee and Koomen, 2016). As a result of this, Bandura made the distinction between response-outcome expectancies and self-efficacy expectations (Bandura, 1977). The response-outcome expectations refer to every individuals' estimate that a given behavior will lead to specific outcomes(Bandura, 1977, 1986, 1997). Bandura is not exclusively looking at the perceived environmental contingencies when he defines self-efficacy expectations. Bandura explains that a teacher's lack of self-efficacy could result in the teacher not implementing new behaviors as they lack the belief to produce such actions (Zee and Koomen, 2016).

Marjolein Zee and Helma M. Y. Koomen published in 2016 the paper "Teacher Self-Efficacy and Its Effects on Classroom Processes, Student Academic adjustment, and teacher well-being: A Synthesis of 40 years of research". In their paper, they are stating that the increase in research can be connected with the belief that TSE, or teacher's self-referent judgments of capability, are relevant for a range of adjustment outcomes at different levels of classroom ecology (Zee and Koomen, 2016). When using various measures and definitions, previous studies imply that teachers with a high sense of self-efficacy will manage a high-quality classroom environment because those teachers are planning lessons that advance students abilities and making efforts to involve them in a

meaningful way as well as effectively manage student misbehavior (Zee and Koomen, 2016)

The paper written by Zee and Koomen is integrating 40 years of TSE research. The main goal was to give an up-to-date critical review of 4 years of research and explore what consequences TSE has for the quality of classroom processes, students' academic adjustments, and the teachers' psychological well-being. The results from the study suggest that Teacher TSE shows positive links to students' academic adjustment, patterns of teacher behavior, and practices related to classroom quality. Zee and Koomen are also discussing how TSE shows positive links to teachers' psychological well-being and personal accomplishments. (Zee and Koomen, 2016). These factors include higher levels of job satisfaction, commitment and lower levels of stress and burnout.

As mentioned in section 1.3, the term TSE is an important aspect to evaluate when doing teacher training as it had shown to contribute to positive effects on the teachers' self-esteem and the teachers' willingness to develop and change their practice. Therefore, even though TSE is not explicitly mentioned in any of the research questions, the term will be discussed in later chapters and will be occurring throughout this thesis.

Chapter 3

Research method

3.1 Chapter overview

This chapter presents the research approach and the data collection used in this study. This study uses a Design Science Research (DSR) approach. The data collection in this study contains of different types of data collected through the design, implementing and evaluating phase of this study. The methods presented in this chapter are chosen as they intend to contribute to answering the study's research questions:

RQ1 Which elements should be considered when designing a programming teacher course?

RQ2 Which course design can help reaching the learning objectives of such a course?

RQ3 How did the course participants experience the designed course?

3.2 Research approach

3.2.1 A qualitative and flexible research design

For the study's research questions, a qualitative research design was chosen. The formulation of the questions makes them difficult, giving quantified answers, and as exploration is required to answer them, the study is positioned as a qualitative research design. The qualitative research paradigm is being used when analyzing qualitative data. Qualitative data is generally categorized as non-numerical data. The data is often represented in forms of text and is commonly collected through interviews or surveys. The researchers of qualitative studies focus on the participants' views and opinions when they collect data (Robson and McCartan, 2017). These data are differentiated from quantitative data, which consists of numbers.

The study design can be categorized as flexible as this study inherits characteristics of such a design. Flexible designs are categorized by commonly using multiple qualitative data, having evolving study designs, and having a single idea or problem the researcher seeks to understand (Robson and McCartan, 2017).

3.2.2 Design science research process

A Design Science Research approach was used in this research. Design science is a research approach that is relevant and commonly design to support information systems research. DSR is fundamentally a problem-solving paradigm, as it "*seeks to create innovations that define the ideas, practices, technical capabilities, and products through the analysis, design, implementation, and use of information systems can be effectively and efficiently accomplished*" (Hevner et al., 2004, p. 11). It is suited to use a design science approach when a study tries to understand learning processes through design, exploration, enactment, redesign, and evaluation (Wang and Hannafin, 2005). In DSR studies, the researchers are actively involved and maintain collaboration with participants. There are five fundamental characteristics of DSR: 1)it refines theory and practice, 2) it happens in a real-world setting and is grounded in relevant contexts, 4) it uses mixed methods in accordance with potential new needs and emerging issues, and 5) it is contextual. DSR is categorized as contextual as the research findings are connected with the design process (Wang and Hannafin, 2005).

DSR methodology can be in many different forms. Peffers et al. (2007) have presented a methodology design model from which this study has taken inspiration. The model can be found in figure 3.1. The model presented by Peffers et al. (2007) is visualizing the whole DSR methodology and how different re-

search entry points will affect the study. The model also gives a representation of DSR methodology as an iterative process. How this study implemented the DSR methodology model is explained in chapter 4.3.1.

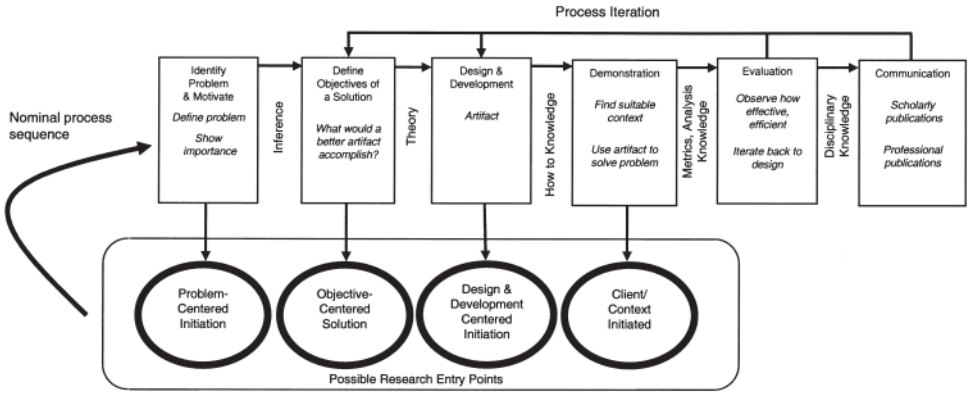


Figure 3.1: DSR methodology process model from Peffers et al., 2007

As mentioned earlier, the DSR process used in this study have taken inspiration of the models presented by Hevner et al. (2004) and Peffers et al. (2007), shown in figure 3.2 and figure 3.1. In terms of the model from Hevner et al. (2004), the model illustrates how environment and knowledge base affect DSR and creates three research cycles that affect the study. In this design, the environmental part is connected to the course design and consists of national curriculum analysis, the researchers' own experiences with teaching, and analysis of the course context(e.g., school needs, constraints, technical aspects, etc.). The knowledge base is consisting of previous literature and related work to PD. The environment and knowledge base are mainly connected and contribute to RQ1 of this study, as the question is regarding which design elements affect a PD program.

Based on the environmental and knowledge base analysis, the course design was defined. This is shown in the DSR pillar in figure 3.2, where the rigor cycle and relevance cycle will contribute to inputs for design adjustments. As this pillar is based on the design, it is connected to RQ2.

Hevner et al. (2004) have presented seven guidelines for DSR within the discipline of information systems. The presented guidelines are: 1) design as an artifact, 2) problem relevance, 3) design evaluation, 4) research contributions, 5) research rigor, 6) design as a search process, 7) communications of research (Hevner et al., 2004, p. 83). Every DSR projects have three design science cycles, as shown in figure 3.2.

Figure 3.2 borrows from the information system framework Hevner et al. (2004) presents. The figure illustrates that every design science study's inherent three

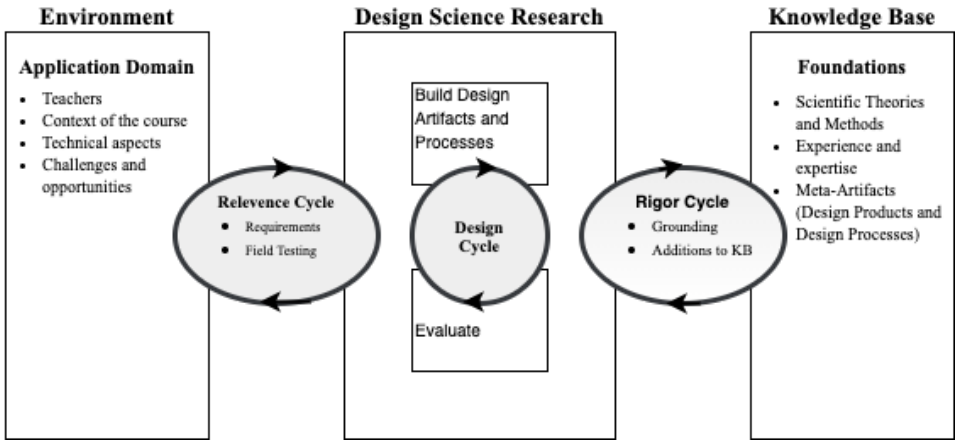


Figure 3.2: Visualization of DSR cycles

research cycles are relevance, design, and rigor. It is a requirement that these three cycles are presented and identifiable in DSR studies (Hevner et al., 2004). The relevance cycle connects the designed course activities with the contextual environment of the study, while the rigor cycle connects the course activities with related scientific foundations and experience. The internal design cycle is the most important part of a DSR project (Hevner et al., 2004). This cycle iterates between designing the course activities and these activities' evaluation and feedback, leading to design refinements. This cycle intends to generate design alternatives and evaluate them consecutively against the requirements until a satisfactory design is achieved (Simon, 1996).

This study's approach is based on all the above. In addition, the study uses theory and applies methodology from design science to guide the course design iterations. In more detail, the study uses a design science methodology as it looks into the complexity of an educative context (in this case, PD of in-service teachers) and is grounded in relevant theory (in this case, related work with PD). Using a design science approach was also appropriate to fulfill the needs of this study, allowing for the continuous design, evaluation, and redesign.

Chapter 4

Course design

4.1 Chapter overview

This chapter describes the course design implemented in this study. The course used in this study was designed as an in-service teacher course at a secondary school in Norway. The course aimed at helping the teachers develop basic programming competence and develop the competencies needed to teach programming. The course presented in this chapter consists of 4 lessons. The two first lessons were designed to give the teachers knowledge and competence of programming principles and give the teachers an overview of helpful resources and tools they could use in their future planning. The two last lessons were designed for the teachers to use their gained knowledge from the first two lessons and create a teaching plan they could use in their class and used a PjBL approach. This chapter will first describe the course's context before presenting the implemented design to rationale the choices made in this design.

This chapter also describes the process and method used in this study. The data used in this study were collected from semi-structured interviews done with selected course participants.

4.2 Context of the course design

In this section, the context of the design will be presented. This section first describes the school where the course was arranged and a description of the course participants. Then, early in the planning process, the teachers who were attending the course answer a questionnaire. The results from this survey will be presented. Finally, the design process will be explained.

4.2.1 School

The designed course is a teacher teaching course for in-service teachers at a secondary school in Norway. Finding a school that was interested and had time to contribute to the study was a high priority in the beginning phases of the study. After getting in contact with one school and having some meetings with the school management, they managed to set time for a course. The researcher had previously held a teacher training course in programming for the school. This course was not mandatory for the schools' teachers and had fluctuating attendance. The school is a secondary school and has 500 pupils and 70 teachers. The classes at the school go from eighth to tenth grade. Every school level is divided into three classes, and they contain around 50 pupils. The classes are further divided into two groups when they have tutoring.

4.2.2 Context of the course

The school managed to set off time for four lessons over four months. The researcher asked the school management if it was possible to compress the duration. The school indicated that it had to be that way, as other ongoing projects at the school claimed a lot of the teachers' time. The first lesson was at the end of January, and the last lesson was at the end of April. The teachers who were attending the course were the school's scientific section, the teachers lecturing scientific subjects at the school. This course was implemented as a part of these teachers' weekly section meetings. Due to the covid-19 pandemic, the school management could not instruct their teachers to attend the course physically. A consequence of this was that the course had to be designed on the basis that it should be able to perform physically as well as digitally. The course lessons had a duration of 120 minutes and were held at the end of the teacher's working day.

4.2.3 Course Participants

The teachers at the respective school where the course is being implemented are divided into different sections, based on which subjects the teachers' lecture.

For this course, all the participants were from the science section of the schools, which means most teachers who are attending are lecturing at least mathematics or science. The figure 4.1 shows an overview of what subjects the teachers were teaching at the time of the course start. The majority of the teachers had formal competence in mathematics and science, while one of the attendants lectured mathematics only. One of the teachers who attended the course did not have any competence in any scientific studies but was educated as a special needs teacher. Even though this teacher will not lecture programming directly to a class, it will be advantageous for the teacher to have basic programming knowledge when assisting students one-to-one. Other demographic information like age has not been used in this study due to ethical reasons and relevancy. The science section at the school consists of 15 teachers.

4.2.4 Survey 1

Before the first course, the teachers who were going to attend were asked to answer a questionnaire (see appendix D). The answers from this questionnaire created a picture of the course participants' expectations and prior knowledge. The answers from this survey formed the basis of many of the choices made for the first lesson. The survey contained 12 questions, and the questions were distributed into four sections; background, previous knowledge, motivation and attitude towards programming, and expectations to the course. The rationale behind including these as categories were that these categories could give the researcher an indication of what content should be added, and adjust the content accordingly, both in terms of relevant activities related to the subjects the teachers teach and the activities' level of difficulty. In the background section, the course attendants were asked which subjects they were lecturing and which topic is relevant for their classes in the upcoming weeks. In this section, the teachers answered the questions by writing the answers in the text field.

The second section of the survey asked about the teachers' previous knowledge. The following statements were given to the teachers in this section:

In order to answer these statements, the teachers were asked to answer with a score from 1 to 7. These values have represented a value from a Likert scale where all the values are mapped from 1 = "strongly disagree" to 7 = "strongly agree."

Table 4.1 shows all questions and statements given to the course participants. Out of the fifteen teachers of the science section, ten teachers answered the survey. The teachers indicate through their answers that they have minimal prior experience with programming. Only one out of the ten teachers indicates that he/she has been programming much earlier and three of the teachers indicate that they have no clue how to implement programming in their teaching. When

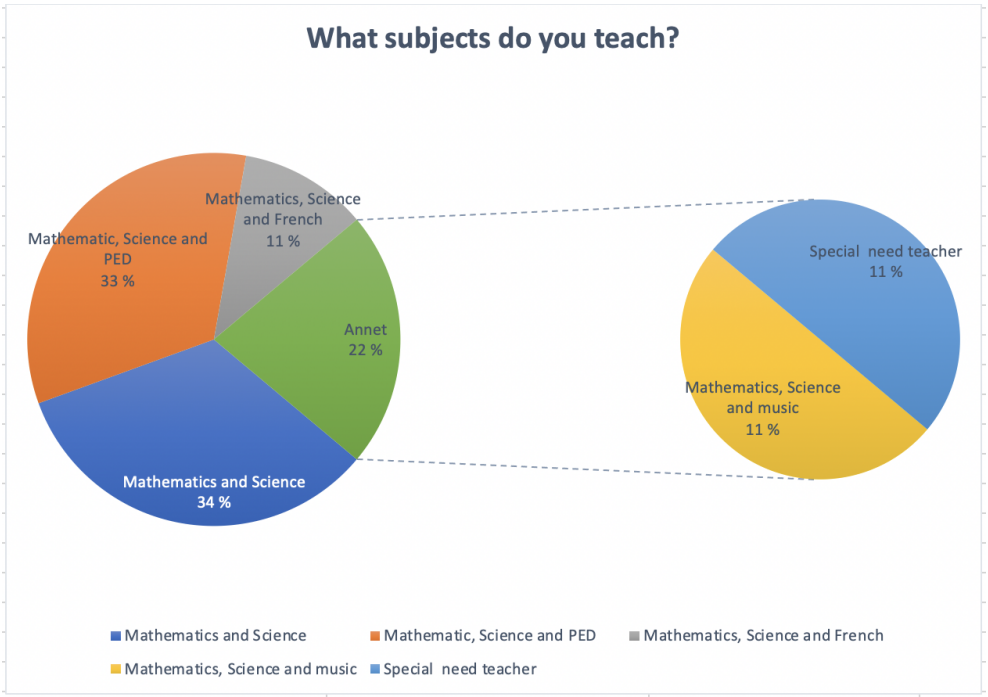


Figure 4.1: Overview over what subjects the teachers are teaching

Statement	Coded Variable	Mean	Median
I consider my digital literacy skills as	*DigitalLiteracy*	4.2	4
In what degree have you been doing programming earlier?	*ProgrammingExperience*	2	1
I have an overview of which competence goals are relevant for my subjects in terms of programming	*CompetenceGoalsOverview*	3.3	4
I consider my motivation to learn programming as	*ProgrammingMotivation*	4.2	5
I see a benefit of everyone learning a little programming	*BenefitialToLearn*	4.2	5
I think programming can be a useful tool for some students in mathematics	*BenefitialMath*	4.4	5
I think programming can be a useful tool for some students in Science	*BenefitialScience*	3.9	4
I think programming is hard to learn	*ProgrammingHardToLearn*	4.5	5
I have an idea of how I can implement programming in my teaching	*HowToImplement*	3.6	4

Table 4.1: Overview over relevant questionnaire statements and questions

the teachers are asked about their course expectations, they emphasize that they want to acquire basic programming competence and see direct examples of how programming can be used in their classroom. The teachers also state that they want a list of resources and teaching plans, which could help them

get new ideas.

4.3 Course Design

This section describes the implemented course design used in this study. This chapter will first describe the course structure and the learning objectives formed for the course, followed by a presentation of the implemented course design.

4.3.1 Course planning process

The course planning process was iterative. The flexible nature of the study allowed the researcher to adjust the course design along the course period. The first course lesson was planned based on the context of the course, national competence goals, and answers from the questionnaire. The last ten minutes of every lesson included giving the teachers feedback to give feedback to the course leader. This feedback was used in the planning phase of the next course. It is believed that it is crucial to have a high interest and commitment among the participants. As the course supervisor, it is crucial to lay the foundation for the teachers to give constructive feedback along the course timeline. By being active in the supervisor role, after each lesson, the researcher gets an intuition of what the teachers have accomplished, thus can adjust the next lesson accordingly. A model visualizing the design process can be found in figure 4.2.

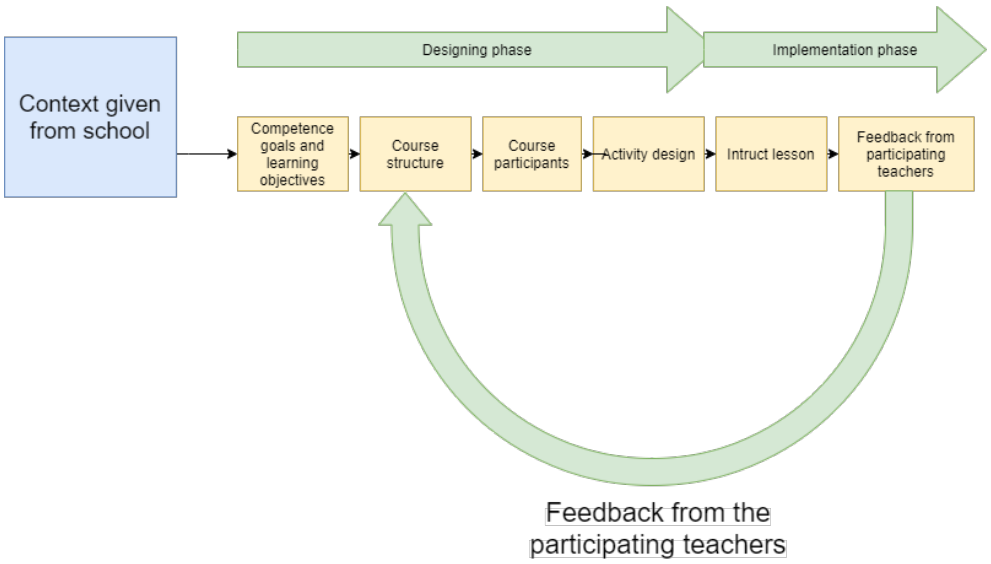


Figure 4.2: A visualization over the design process of the course

Figure 4.2 is visualizing the different phases in the design and implementation phase of the research and aims to illustrate how the received feedback from the course's lesson required a reflection around the course structure, activities, and participants in light of the received feedback. The main argument of doing this process iterative is the belief that this could contribute to the course being thoroughly designed with the participants in mind, and therefore could result in a course that is adapted to the environment at the respective school.

4.3.2 Learning Objectives and Competence Goals

In the planning phase of the teacher training course, the starting point was to get an overview of the implemented competence goals from the national curriculum. The relevant competence goals were obtained from The Norwegian Directorate of Education and Training (UDIR) website ('Curriculum for Mathematics Year 1–10', 2020) and is shown in table 4.2.

Subject	Competence Goal	Competence aims and assessment
Mathematics	Explore how algorithms can be created, tested and improved by means of programming	Aims to reach this goal by end of 8^{th} grade
Mathematics	Simulate outcomes in random trials and calculate the probability that something will happen by using programming	Aims to reach this goal by end of 9^{th} grade
Mathematics	Explore mathematical properties and relationships by means of programming	Aims to reach this goal by end of 10^{th} grade
Science	Use programming to explore scientific phenomena	Aims to reach this goal by end of 10^{th} grade

Table 4.2: Overview over relevant competence goals

As the current implementation of programming in the Norwegian curriculum is partially established and because the course participants mainly educated mathematics and science, the focus was on the goals presented in table 4.2. To get an overview of the competence goals contributed to formulating the main learning objectives for the course. Many aspects should be considered when working out the learning objectives. First, the learning objectives have to correspond to the competence goals found in the curriculum. Second, external factors like time available, the time between each course, and participants' prior programming knowledge and experience were considered. At the end of the course, the teachers were going to create individual teaching plans where programming is applied. The first two lessons will support the teachers in defining and evaluating different activities they can use in the classroom and give them education in basic programming concepts and principles. In the last two courses, the teachers use their acquired knowledge from the first part of the course to create their own teaching plan. The following learning objectives were formulated as shown in 4.3.

There are the main categories that separate the learning objectives. The category programming knowledge considers the teachers' obtained knowledge about basic programming terms, available tools, and different programming languages they can use in their teaching. The learning objectives in the category programming skill had to be realistically formulated with regard to the course's duration and the teachers' prior programming knowledge. Learning objectives with a more pedagogical approach can be found in the generic competence category.

After the course learning objectives were formulated, planning the individual lessons and finding material and high-quality activities were prioritized. When designing the lesson activities, there was a focus on making them as relevant

Category	Learning objectives
Programming knowledge	• have basic knowledge of basic programming principles like variable, condition and loop • have basic insight over available resources and learning tools
Programming skill	• develop and run basic program • be able to explain basic programming examples in teaching • understand how cooperation can be utilized to promote learning of programming
Generic competence	• convey basic programming to pupils, both oral and written • discuss and reflect on programming solutions • plan a teaching plan in programming in their respective teaching subjects

Table 4.3: Learning outcomes: A list over the course's main learning objectives

for the teachers as possible and at the same time introduce them to basic programming principles. The seven factors of effective PD presented of Darling-Hammond et al. (2017) were taken into account in the activity planning.

The literature also revealed *teachers self-efficacy* as an important term connected to evaluating the effects a PD program had on the teachers. The overall aim should be to design a course that positively affects the teachers' self-efficacy towards both their programming skill and teaching programming. As previous literature reveals teachers self-efficacy as a factor which affects teachers' well-being at work and their willingness to incorporate new additions in their practice (Zee and Koomen, 2016).

4.3.3 Course structure

As mentioned earlier, the course structure was divided into two sections, including two lessons in each. The two first lessons were designed to give the teachers knowledge and competence of programming principles and give the teachers an overview of helpful resources and tools they could use in their future planning. The two last lessons were designed for the teachers to use their gained knowledge from the first two lessons and create a teaching plan they could use in their class. Thus, the aim of dividing the course into two sections was to have one part focusing on the technical aspects of programming and then use the acquired technical knowledge and connect it and adjust it into their practice. A similar approach has been tried by Rouhani et al. (2021), where the second section aimed to make the teachers bring disciplinary knowledge and pedagogy combined with traditional and instructional methods to create an artifact (in our case, a teaching plan) that they could use directly in their practice. In other words, the teaching plan they make will "act as a

bridge between training and practice" (Rouhani, 2020, p. 1).

Project-based learning

It is essential to let the students practice and do activities in programming to grow knowledge and experience. Hence, in computer science, educations programming courses have project-based activities and assessments implemented in them, which ask enrolled students to build larger programming projects (Sindre et al., 2018). Correspondingly, the researcher decided to separate the course into two sections; one focused on traditional education and one focused on a specified project given to the teachers. Furthermore, recent research points out that Project-based Learning (PjBL) can improve creativity and teamwork competence, which could correlate to students creating more unique projects, enhancing their overall academic results, and encouraging teachers to obtain a deeper understanding of computers' role in modern society. Consequently, the researcher decided to implement elements from project-based learning theory into the last section of the course. The project-based theory is designed based on the view from constructivism theory, a theory concerning teacher and learning approach towards contextual teaching and learning (Krajcik et al., 1994). The learning approach focuses on supervising and assisting the students in an individual or collaborative project. The project should combine various subjects or material in the curriculum (Efendi and Sanjaya, 2017). For example, during the second section of the course, the teachers started a project to design and compose teaching plans in one of their teaching subjects with programming as its central theme. The teachers were also allowed to work in groups.

The researcher decided to use a teaching plan as the project for the second section. The rationale behind this choice was that the project could encourage the teachers to work with content they wanted and do it relevant to their practice. In addition, the limited time available made the project suitable for the course. The project aims to let the teachers explore programming in their setting and develop a tool they can utilize later. At the same time, the researcher considered the project's relevance to teaching practice as a motivating factor for the teachers.

4.3.4 Technical considerations of the course design

A plan for the first lesson was drafted after the main learning objectives were formulated. The first two courses mainly focused on the learning objectives from the categories programming knowledge and programming skill [fig:4.3]. UDIR published in February 2021 examples of exam assignments that the new curriculum into account. The assignments found here are examples of assignments that could appear at the final exam for Norwegian pupils graduating

secondary school. UDIR's publication contains eight example assignments. Two out of the eight assignments are focused on programming and computational thinking. One of the presented assignments can be found in figure 4.3.

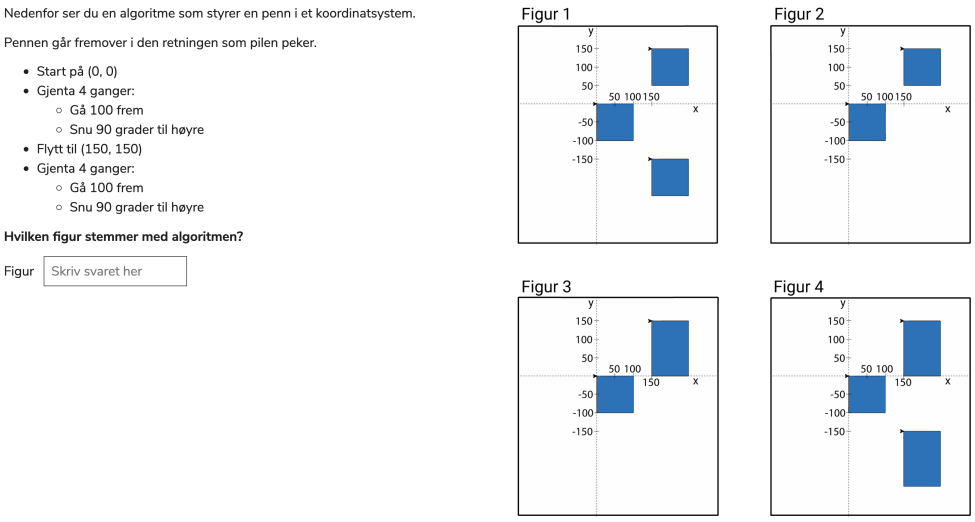


Figure 4.3: A programming assignment retrieved from an example mathematics exam from UDIR ('Eksempeloppgaver i matematikk for 10. trinn', 2021)

As the course duration was limited to four lessons, the first two lessons were only focused on one programming language. The programming language Python was eventually chosen as the language in focus in the first part of the course. Python also has syntactical similarities to the pseudo-code presented in the assignment from figure 4.3. There were also technical factors that had to be considered. The teachers at the school had computers that used the Linux-based *Chrome OS*. As Chrome OS won't have any Integrated Development Environment (IDE) available, the online IDE Replit were chosen as the programming environment[1].

Figure 4.4 shows the replit programming environment. The environment runs off a virtual machine and will work on any computer connected to the internet. Another benefit of using repl is the possibility of making team environments. All the participating teachers were invited and had accepted the invitation before the first course lesson. The replit team's environment has two roles implemented: teacher and student. The teacher's role has permission to create team projects and assignments. The teacher decides if the projects allow for student collaboration. This makes it easy to do pair programming later during the course. Another benefit of using replit teams is that it allows the course supervisor to monitor and review students' work during class and afterward.

[1]Repl IDE can be found here: https://replit.com/

Figure 4.4: Overview over the Replit programming environment

Figure 4.5 shows how the environment looks when a teacher monitors a student's progress during class. This feature makes it easier to provide continuous feedback to the teachers during courses and may help the supervisor provide the teacher's information to improve their learning.

4.3.5 Lesson 1

The first two course lessons aim to show the teachers different use-cases of programming in their teaching. The programming language in focus was Python. Learning objectives were formulated for the lesson to help evaluate each course lesson. The learning objectives formulated for the first lesson can be found in figure 4.4

Category	**Learning objectives**
Programming knowledge	• knows about the programming principles variable, condition and loop • explore the Python programming environment • explain the difference between different data types in programming (integer, string, array)
Programming skill	• develop and run basic program in Python • be able to explain basic programming examples in teaching
Generic competence	• convey basic programming to pupils, both oral and written

Table 4.4: Learning outcomes: A list over learning objectives of the first course lesson

Before the first lesson, the researcher had prepared a structural overview of relevant and valuable programming resources for the course participants. The

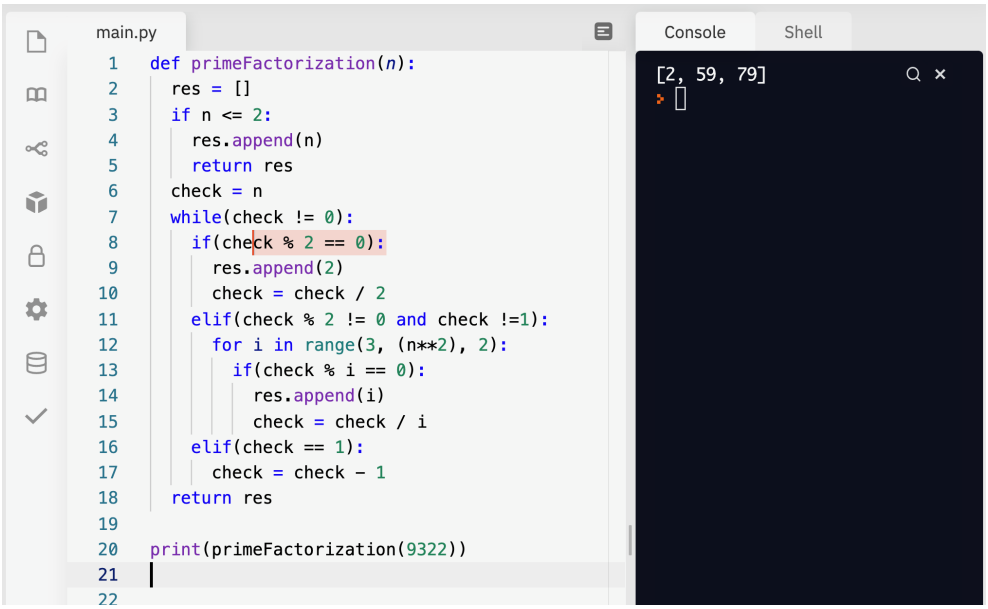


Figure 4.5: Teachers view over students progression in replit teams

resources should be appropriately selected to support the teachers during the course and their upcoming teaching and learning related to programming. The resources were both resources and teaching materials from the government and other independent resources that the researcher had previous experience with teaching. In the first lesson, Python was used to demonstrate the use of variables, conditions, and loops. The overall theme of the first lesson was an introduction to Python and fundamental programming principles. The activities in lesson 1 used a design based on the first competence goals from table 4.2.

The responses from the survey indicated that the teacher had little prior programming experience activities. As a result, easy activities with a low entering threshold were designed for the first lessons to promote the teachers' feeling of mastery. The teachers were continuously given assignments through the first lesson, and the assignments were discussed in plenary. This approach was chosen as it was intended to promote both active learning and collaboration among the teachers (Darling-Hammond et al., 2017). When the teachers are working with activities during the first lesson, the course instructor consistently assists the teachers and provides feedback on their work. Table 4.5 presents four of the assignments given to the teachers during the first lesson. As the first lesson intended to give the teachers a basic and relevant introduction to programming, the assignments were designed towards mathematics. The tasks vary in difficulty, and it was not anticipated that the teachers would solve all the problems. The program, which asked the teachers to decide if a number is a prime number, was only added to have some more complex tasks to give to

teachers that advance quickly or have more prior knowledge than others.

Assignment category	**Assignment**
Variables and basic programming knowledge	Create a program that saves two variables **num1** and **num2** and saves the sum of these variables to a new variable **result**. The program should print out the variable result.
Programming operators, variables and conditions	Create a program that asks the user for a number, and decides whether that number is an odd or an even number. The program should print out a text including the number the user entered and if it is an odd or even number.
Variables and loops	Create a program that prints out all numbers up to 10000 that is divisible with 4
Variables, condition, loops and problem solving	Create a program that decides whether or not a number is a prime number

Table 4.5: List over course assignments from first lesson

4.3.6 Lesson 2

The last ten minutes of every course lesson are designed for the teachers to give feedback on the course content and to discuss eventual challenges or other findings they met during the lesson. The following feedback was received in the feedback session after the first lesson:

- The terms variable, condition, and loops were understandable concepts according to the participants, but the teachers indicated that they did not see a clear connection to how they could use it in their teaching
- The teachers indicated that text-programming using Python and receiving feedback in a console was demotivating and "not fun"
- Teachers indicate that they could explain the programming principles variable, condition, and loop to their pupils after the lesson
- The teachers said they wanted to see a finished made teaching plan in the next lesson

The feedback the teachers gave also fits with what previous literature has stated, by Darling-Hammond et al. (2017) as a key feature of PD or Partanen

et al. (2017), by having positive results after adjusting a programming PD based on participants' feedback. As the course design is learner-centered, the feedback caused adjustment to the course plan. Therefore, the feedback is environmental factors that result in an adjustment in the course design. This process can be seen in figure 3.2 and is a part of the relevance cycle from the design science research cycles by (Hevner et al., 2004).

The teachers expressed in the feedback session that text-based programming with Python was tiresome because the feedback they received when programming was intriguing. In other words, they wanted to use something more engaging than receiving print statements in a console line. As a result, the Python library Turtle was chosen as the primary tool in the second lesson. Turtle is a Python library that allows the user to draw pictures on a digital canvas. Drawing pictures and shapes also added possibilities to design a lesson around mathematics and geometry. The rationale behind choosing Turtle was that it was a library the researcher had previous knowledge of in addition to being visual and more engaging than the console line traditional Python offers. The lesson was designed as a regular class of mathematics at 10^{th} grade. A document containing the teaching plan was made and sent to the teachers at the beginning of the lesson. The formulated learning objectives for the second lesson can be found in table 4.6. Some of the learning objectives were retained from the first lesson, as they were relevant for this lesson as well.

Category	**Learning objectives**
Programming knowledge	• knows about the programming principles variable, condition, loop and function • solve problems with variables, conditions and loops using Turtle
Programming skill	• develop and acquire basic knowledge of Turtle • be able to explain basic programming examples in teaching • draw simple figures with the Turtle library
Generic competence	• convey basic programming to pupils, both oral and written

Table 4.6: Learning outcomes: A list over learning objectives of the second course lesson

This lesson focuses on the fundamental programming principles introduced in lesson one, but instead of using plain Python, the library Turtle is introduced. The lesson structure remained the same as the previous one in terms of teaching approach and systematically introduced the relevant programming principles for the lesson. Each principle was introduced explanation and exemplification individually by the supervisor. In addition, the teachers were given principle-specific tasks after each introduction to encourage active learning. Finally,

the lesson's last minutes were dedicated to summarizing the lesson's content, retrieving feedback, and giving information about the following lessons.

4.3.7 Lesson 3 & 4

In the third and fourth lessons, the learning approaches changes to a more exploring approach and incorporates elements from PjBL.

Before the third lesson, the researcher made a document aimed as a supporting tool for the teachers during the project phase of the course. The document can be found in Appendix E. The document included references to already made teaching plans, additional programming resources, and a list of questions to support the planning. The questions were included to support the teachers in reflecting on their design and supporting them in the starting phase. The questions were categorized into two sections: one to support the teachers in identifying their teaching plan's learning path and supporting the plan's activity design. The document aimed to help the teachers establish their learning trajectories and learning objectives in the initial planning phase. The chosen resources included materials the researcher considered helpful in regards to what the course content had been. There were also added some resources towards teaching Science, as this was something the teachers had indicated they wanted to see from lesson 2. The questions aimed to support the teachers in establishing their learning objectives referred the teachers to look up topics in their subjects where programming could be used as a tool. The questions also wanted to challenge the teachers to reflect on the chosen topic they wanted to implement and whether that is suitable to include in a programming lesson.

The second part of the guide includes questions that challenged the teachers to formulate competence goals and learning objectives, reflect on which programming principles are relevant for this lesson, and on how this lesson could be assessed. Lastly, the documents aim to get the teachers to reflect on the development of their practice and what possible benefits and advantages the teachers observe by implementing programming in the respective teaching plan they are designing.

The last phase of this course was designed to involve the course participants in their PD. The lessons were designed for teachers to individually or collaboratively work with their plans. Even though the focus had been on Python in the previous lessons, the teachers could design a teaching plan covering another tool or language. The programming microcontrollers and tools the school had been acquiring were also available for the teachers to explore.

Chapter 5

Analysis of the teacher interviews

5.1 Collecting data using semi-structured interviews

The data used in this study is retrieved from six interviews with selected course participants. Since this study follows a qualitative research approach, interviews with the course participants were considered a good approach to collecting data to answer this study's research questions. Interviews are considered a suitable method of obtaining data about the teachers' motivation, values, and attitudes towards the course (Robson and McCartan, 2017). The interviews were designed as semi-structured interviews. The questions prepared for the interviews were designed to contribute to answering the research questions of this study. The questions in the interview guide are divided into four main categories. The four categories were background, programming, and attitudes toward programming, course, and feeling of mastery, and course and PD. In a semi-structured interview, the interview guide narrows the interview down to formulated topics. The interview guide can also contain suggestions of possible follow-up questions to ask (Kvale et al., 2015). The interviews were conducted the week after the final course lesson. The teachers were interviewed individually. Personal interviews allow the researcher to try to understand the interviewee's perspective. The questions used in a personal semi-structured interview should seek descriptions, experiences, reflections, and especially interpretations regarding the study's research questions (Kvale et al., 2015). In a semi-structured interview, there is a focus on finding a balance between asking prepared questions from the interview guide or asking questions freely in an open conversation (Kvale et al., 2015). It is not unusual that the question order or interview themes do not follow the interview guide as the interview

progresses (Robson and McCartan, 2017).

5.1.1 Interview guide

Question category	Sample of question
Background	• What subjects are you teaching?
Programming and attitudes	• What prior experience did you have with programming before the course? • Can you describe your attitude towards programming?
Course and feeling of mastery	• Did you felt comfortable teaching programming before the course? • To what degree has the course shown you how to adapt assignments to your pupils? • How did the course affect your sense of self-efficacy towards teaching programming?
Course and professional development	• What did you learn from the course? • What did you think about making your own teaching plan during the last part of the course? • To what degree can teacher training courses like this affect teacher collaboration at the school?

Table 5.1: Sample questions retrieved from the interview guide. See A

The interview questions were formulated with the research questions in mind and were divided into four main aspects: *Background*, *Programming and attitude*, *course and feeling of mastery* and *course and professional development*, as shown in table 5.1. The questions also took into account the outline and content of the course. The used interview guide used in this study can be found in the appendix A.

The interview, as a whole, was designed in a way that could contribute to answering this study's research questions. As the formulation of the research questions is of such an open nature, the questions were based on topics found in related work and expected respondents' answers. Regarding RQ1, questions regarding aspects of PD in general and directed towards the study. Questions about the teachers' attitude towards programming, prior knowledge, and programming skills were also formulated as they could contribute to answering RQ1. Regarding RQ2, the interview included questions that conduced the teachers to reflect on the course's activities and their apprehension of viable activities that could have been implemented. Ultimately, questions pointed towards the teachers' course experience were formulated and contributed to answering RQ3. When connecting all the pieces and seeing them in light of the literature, the main RQ will be answered.

There was performed one test interview before conducting the first actual interview. This interview was conducted with a teacher working for another school. The test interview contributed to adjusting the wording of some questions and indicating possible answers the respondents could give. After the test interview, the ordering of the questions was also adjusted.

5.1.2 Interview respondents

The course participants were informed that this study required teachers who would participate in a course evaluating interview after the course. At the end of the course, six teachers indicated they were interested in participating as interview respondents. Two of the teachers were women, and the remaining four were men. An overview of the interview participants, which subjects they teach, and which teaching subject they expect to use programming in, in the future can be found in table 5.2.

Participant Number-gender	**Teaching grade**	**Teaching subject(s)**	**Programming subject(s)**
1-Male	8^{th} grade	Math, science and music	Math and science
2-Female	9^{th} grade	Math and science	Math and science
3-Male	8^{th} grade	Math, science, physical education and optional subjects	Math and science
4-Male	10^{th} grade	Math, science, physical education and optional subjects	Math and science
5-Female	9^{th} grade	Math, science and physical education	Math and science
6-Male	8^{th} grade	Math and science	Math and science

Table 5.2: Overview of the interviewed teachers' teaching grade, teaching subjects and which subjects they expect to use programming in

As the individual course lessons' had a various amount of teachers attending, obtaining six teachers for the interviews was satisfactory. The amount of interview respondents in such studies varies a lot. A draw of having few respondents is the possibility of not being presented with all the interesting data. On the other side, having too many respondents could lead the data collection "saturated." This means that the recent interviews' content add no new information (Kvale et al., 2015; Robson and McCartan, 2017). All teachers were at least teaching mathematics and science, which were the subjects they expected to use programming in the future. Other attributes like age were not collected due to ethical reasons.

5.1.3 Conducting the interviews

All six interviews were conducted the following week after the last course lesson. It was desirable to conduct the interviews this quickly after the final lesson

to ensure the respondents not forgetting the course elements. The interview respondents were given information in advance of the main themes the interview touches. There are several methods of documenting the interview content, including video recording, voice recording, or written notes (Kvale et al., 2015). To prevent transcribing the interviews continuously as they were progressing, they were recorded using a voice recorder. Using a voice recorder cause protection of privacy concerns and required the project to be registered to The Norwegian Centre for Research Data (NSD). When using a voice recorder to record the interviews, more time during the interviews were open to write observational notes and observe the dynamics of the interview, the respondent's choice of words, pauses, or other verbal attributes (Kvale et al., 2015). The approval of the application to NSD can be found in appendix C.

Five of the interviews were conducted physically at the teachers' school, while the last was conducted using the online video-communications service Google Meet.

After the interviews were conducted, the content was transcribed by hand by listening to the recordings. Transcriptions of interviews involve structuring the interviews to make them more suitable for (Kvale et al., 2015). The interviews were transcribed in two rounds. The first round included transcribing the interviews word by word, excluding filler words. After the first round, the transcriptions are proofread and appropriately structured. Lastly, the transcription was examined while listening to the respective recording.

5.1.4 Method of analysis

The method used to analyze the transcribed interviews is a method that is developed to support grounded theory studies. The method is developed by Glaser and Strauss in 1967 and is called *the constant comparative method of analysis* (Glaser and Strauss, 1967). Even though this analysis method is developed with grounded theory studies in mind, it can still be used in studies with other qualitative study designs. The method splits the analysis into three parts, open coding, axial coding, and selective coding (A. Strauss et al., 1990; A. Strauss et al., 1998).

The qualitative data analysis program Nvivo was used in the analysis process presented in the next section ('Nvivo 12', 2021). Nvivo has many implemented features that made the analysis process more structural and efficient than using other software.

Coding process

Open coding is the part of the analysis where the researcher names and categorizes phenomena through an intense and close inspection of the data material (Postholm, 2010). In this coding process, parts of the data are separated and allocated a name, or a *code* (A. Strauss et al., 1990; A. Strauss et al., 1998). The size of the codes could vary, from an utterance to a paragraph. A piece of data could also be labeled with several codes if it may be considered to fall between more than one conceptual category (Robson and McCartan, 2017). The initial coding phase is done to develop possible concepts that can link to parts of the data material (Postholm, 2010). After doing the initial round of open coding, 300 codes were made. To check whether some of the codes that emerged from the latest interviews were appearing in the previous interviews, they were checked once more. During the process of open coding, there were some potential categories that emerged, and the codes were grouped. The categories emerged as some of the codes seemingly covered the same phenomenon. After the initial round of open coding, the codes were compared to the researcher's supervisor's codes.

During the process of open coding, there were seven categories that emerged when looking for common themes and answers from the interviews. A list of the categories and the category description can be found in table 5.3.

Category	Description
Course specifics	The teachers' feedback or reflections regarding specific activities or the course in general
Teaching programming	Reflections around teaching programming
Attitude towards programming	The teachers reflections and statements regarding their attitude towards programming
School community and collaboration	Reflections regarding the school community and teacher collaboration
External challenges with programming	Reflections around the external challenges around the implementation of programming
Planning a programming lesson	The teachers reflecting around challenges connected with planning a programming lesson
Teacher stereotype	The teachers reflect whether or not there exists a teacher stereotype

Table 5.3: A list over the emerging categories after open coding and categorization

The next phase of the analysis is called axial coding(also called theoretical coding) (Postholm, 2010; Robson and McCartan, 2017). In the process of axial coding, the categories made from the open coding are linked together. The goal of this process is to specify the categories by looking at what circumstances made them (Postholm, 2010). These circumstances contribute to the model and define sub-categories. Questions about *when*, *why* and *under what conditions* can accentuate the relation between the categories and their sub-categories (A. Strauss et al., 1990; A. Strauss et al., 1998). The defined sub-categories after the axial coding can be seen below in table 5.4.

Category	Sub-Category
Course specifics	Course impact on self-efficacy
	Course content
	Contextual aspects
	Course structure
	Degree of difficulty
	Corona aspects
	Positive feedback
	Negative feedback
	Proposed changes
Teaching programming	Pedagogical aspects
	Adaptive teaching
	Pupil competence level
	Questions from pupils
	Technical challenges
	Pupil competence level
	Student participation
	Computational thinking
	Positive aspects
	Planning a programming lesson
Attitude towards programming	Fear and skeptisicm
	Difference in teacher attitudes
	Positive towards programming in school
	Mixed feelings about current implementation
	Can improve teaching
	Attitude will change over time
School community and collaboration	Positive towards teacher collaboration
	Pedagogical and pedagogical forum
	Challenges
External challenges with programming	Time
	Political aspects
	Teacher discipline and attitude
	Different prioritization at schools
	Programming resources and tools
	Lack of teacher training
Programming skill	More training is needed
	Basic programming competence
	Teacher feel a lack of competence
	Big difference in teachers' programming competence

Table 5.4: Categories and their respective sub-categories after axial coding

After the axial coding, the category "*teacher stereotype*" was removed from the analysis as it did not contribute to the research question. The category "*Planning a programming lesson*" was removed as a category and replaced as a sub-category under teaching programming. Table 5.5 presents all sub-categories, frequency of code for each category, and sample sentences corresponding to the categories.

Sub-category	Freq	Sample sentence
Course specifics		
Course impact on self-efficacy	170	I have tried several things myself now. And, I have been sitting trying and failing, so I feel more comfortable with the entire issue yes. Undoubtedly (Teacher 6)
Course content	78	Code.org. It was fun! I thought I could actually use it in teaching (Teacher 2)
Contextual aspects	67	I especially do not think it was an advantage in this course that it was such a long time between each lesson (Teacher 2)
Course structure	25	And then I think it, the last session, the one in the math room. I think that lesson was easier to relate to and understand (Teacher 3)
Degree of difficulty	29	When we were working with Python, I thought that I never would have came up with this myself (Teacher 3)
Corona aspects	3	Corona makes everything difficult, right. But, the teachers are way more focused when attending physically (Teacher 2)
Positive feedback	18	I this course we have tried formulating our own learning objectives towards programming. That will make it easier to plan the teaching later (Teacher 2)
Negative feedback	17	When we started, I thought it were very hard (Teacher 3)
What do teachers want to see in teacher training	57	Teachers want to see completed teaching plans or other examples and proposals of how things can be arranged in the classroom (Teacher 3)
Proposed changes	25	You can start by showing us one tool only, and then introduce us to the learning materials later in the course (Teacher 4)
Teaching programming		
Pedagogical aspects	55	My focus has almost always been on how we can manage to solve it pedagogically, how we can present it in a pedagogical way (Teacher 1)
Questions from pupils	31	I don't feel comfortable teaching a class and tell the pupils that a don't know the answer to any of the questions they're asking(Teacher 3)
Adaptive teaching	14	I think it will be very challenging the weakest pupils (Teacher 5)
Technical challenges	11	Difficulties reading instructions, reading a task or another technical issue. I think 90% of the challenges will be technical challenges (Teacher 6)
Pupil competence level	6	So, we have to create differentiated tasks to the pupils as in other regular subjects. The challenge is the knowledge of the pupils (Teacher 5)
Challenges	10	I'm afraid I will receive more questions when teaching programming than in a regular math class (Teacher 3)

Table 5.5 continued from previous page

Student participation	15	By inviting the pupils in to our research project regarding programming, can we obtain the knowledge together. It becomes our project (Teacher 1)
Computational thinking	8	But, people often forget to think about computational thinking, so programming is so much more than writing lines of code in Python (Teacher 5)
Positive aspects	10	I were surprised over how excited the other teachers were (Teacher 2)
Planning a programming lesson	23	I almost think the most challenging when planning a lesson in programming is choosing the right tools (Teacher 5)
Attitude towards programming		
Fear and skepticism	43	Yes, because teachers have a fear towards programming (Teacher 1)
Difference in teachers' attitude	29	There must be a big difference among teachers also. Both the teacher's willingness of learning something new and prior knowledge of the course (Teacher 1)
Positive towards programming in school	29	Its about time. If we think the pupils should obtain an understanding of the world they are living in, it's absolutely about time. (Teacher 6)
Mixed feelings about current implementation	18	It is frustrating when new goals are added to the curriculum, without it being offered any kind of professional development course from above (Teacher 3)
Important skill for the future	5	That part have I meant being most important for the future (Teacher 1)
Can improve teaching	3	But, now I see how the different tools and materials can be used, and give the pupils a bigger perspective in my teaching (Teacher 6)
Attitude will change over time	2	But, after a while, I definitely think it will become more natural (Teacher 4)
School community and collaboration		
Positive towards teacher collaboration	37	You have to have someone to play ball with, in a collegium (Teacher 1)
Pedagogical forum	28	But, what I miss the most on the weekdays, is someone to talk pedagogy with (Teacher 1)
Challenges	28	There is too little meeting point in the subject sections (Teacher 3)
External challenges with programming		
Time	22	I'm not ready to teach programming yet. I will need more time (Teacher 3)

Table 5.5 continued from previous page

Teacher discipline and attitude	12	The biggest factor is the teachers you are going to teach, that they are not only motivated, but also disciplined (Teacher 6)
Political aspects	11	The implementation feels very politically forced for the teachers. We already have a lot to do (Teacher 2)
Different prioritization at schools	10	I got friends who work on other schools nearby, and they haven't even touched programming yet (Teacher 4)
Programming resources and tools	8	Actually. that jungle. To navigate that jungle of resources takes a lot of time (Teacher 2)
Lack of teacher training	4	So, I would rather have better support of the teachers, but that are what they probably are trying to do now (Teacher 2)
Programming skill		
More training is needed	4	I look dark on teaching programming now, without further teacher training (Teacher 3)
Basic programming competence	11	So I kind of have that basic computer knowledge in place (Teacher 1)
Difference in teachers' programming competence	8	Well, there was someone there who sat there to know more, and then it might be boring for them (Teacher 4)
Teacher feel a lack of competence in programming	6	Good question. So, I would say that the competence is equal to zero (Teacher 3)

Table 5.5: Sub-categories, frequency of codes, and sample sentences

From figure 5.5 all the emerging codes from the coding phase are listed. As a result of the researcher's open approach to the coding, codes that do not directly contribute to answering the research questions are also included. The codes are also included to give the reader a clear understanding of the interviews' content.

The qualitative data management software Nvivo creates various visualization of the emerged categories from the analysis. A method for displaying hierarchical data using nested figures is called treemapping. Figure 5.1 shows the tree-map drawn using draw.io[1] during the analysis process. The map is drawn based on a generated tree-map created by Nvivo. The tree map is reconstructed for readability.

[1]See more: https://app.diagrams.net/

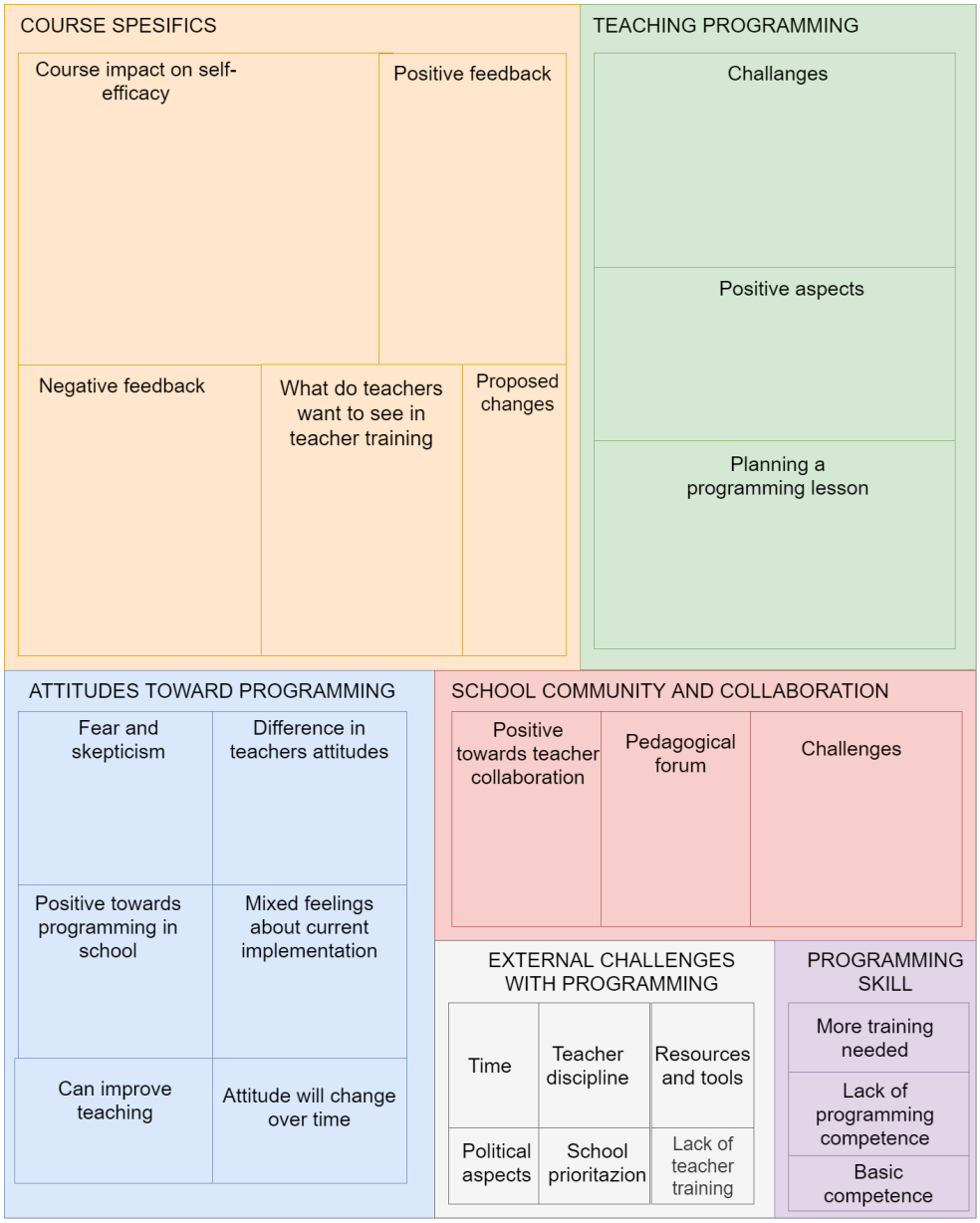


Figure 5.1: Treemap of the categories and their sub-categories.

5.2 Results from the teacher interviews

This section presents the results from the analysis of the interview transcriptions from the study. The section is divided into sub-chapters which present quotes retrieved from the main categories and sub-categories from the analysis.

The presented quotes are translated from Norwegian to English. In addition, some codes were premade before the interview. These were made to easier understand the context and emotions of the interview when reading through the transcription after some time. The list of the premade codes can be found in Appendix G.

5.2.1 Programming skill

All the teachers expressed that they felt they had general digital literacy skills. Four of the teachers indicated they had a lack of competence in programming. When the teachers were asked to describe their own programming competence, these were some of their answers:

> My competence in programming? That borders to zero, yes. So there, I am reasonably blank. (Teacher 3)

> Yes, good question. I will say my competence equals zero. (Teacher 4)

On the contrary, the rest of the teachers described their own programming skills as basic. These teachers also had prior programming experience. For example:

> I have some experience with programming from my teaching education. I have programmed in C#, Python, Matlab, and HTML. So I knew a lot of basic programming in advance. (Teacher 2)

> Well, I started studying computers at the university in [year] and tested different communication systems. So I have some experience and knowledge about the most basic programming principles. (Teacher 1)

5.2.2 Course specifics

Through the interviews, the teachers reflected on the course content. The teachers gave both feedback, positive and negative, on the course activities and discussed possible changes they would have implemented in such a course.

Course impact on self-efficacy

In general, the teachers indicated that the course had a varied effect on their sense of self-efficacy. Two of the teachers say they feel more relaxed toward teacher programming after participating in the course. When one of the teachers get asked if he feels the course has made him more comfortable towards teacher programming, he says:

> Let's just say I have become more relaxed towards teaching programming. So, I have been working along with Scratch, Python, and other stuff, and I am

> aware of the potential. I guess I know the resources are there, and I have the logic and pieces together to make it interesting. (Teacher 1)

The other teacher indicates that the course has made programming more harmless for him than earlier. When he gets asked what he has learned from the course, he answers:

> The most important thing is that it is not as difficult as I had imagined. I have always believed that I will make it happen. It is just I have to experience it that it works here. (Teacher 3)

The fourth teacher indicates that the course has given him the belief that he eventually will become prepared to teach programming, but he needs more time training. He says:

> I will say that the course has given me the faith that I will make it. It certainly has. But I'm not there now. But, yes, it should be possible to do, I mean. (Teacher 4)

Two of the teachers indicated the course had minimal effect on their perceived self-efficacy towards teaching programming or their self-efficacy towards their own programming skill.

> I: Could you try to explain how the course affected your perceived self-efficacy in solving problems using programming?
> R: I do not think there had been any particular change there. I think that what we have been doing, on this course and other courses before this, is so intuitive for me that I understand it quite quickly (Teacher 6)

Positive feedback on course

During the interview, the teachers were asked questions regarding the course they had participated in. During the interview, there were given some positive feedback on the course. The feedback was divided into three sub-categories; course content, contextual aspects, and course structure.

Course content

Related to the course content, the teachers seemed really happy with the resources they were introduced to during the course. One teacher indicates that having available resources is important for him.

> No. First of all, been reassured. That, and finding resources. I know I will find resources. That is the most important thing for me as a teacher in daily life. (Teacher 1)

In the case of activities, there was especially one activity the teachers were positive towards. The turtle activity from lesson two the teachers explicitly mention in the interview. One of them said:

> I am most excited about the coding. I think that part is most fun, right. And that smiley assignment you gave us, that was something I liked to do. I think drawing in programming is very fun. And I started thinking about how I could use this in arts and crafts as well. (Teacher 5)

Contextual aspects

All teachers were very positive towards planning and executing such teacher courses internally at their school. One of the teachers said during the interview:

> I think it is very reasonable to have it at the house. You can compress the course a bit, and don't spend unnecessary time traveling and such. I would absolutely prefer to have it at the house. That way, all participating teachers can sit together and discuss. I see that as a great strength. (Teacher 4)

The teachers indicated that one of the positive aspects of having the teacher training course internally at the school is the possibility of collaborating with colleagues and seeing other teachers at work which you can ask if a challenge occurs regarding programming. A teacher says:

> I: What do you think about having the course internally at the school?
>
> R: Yes, that is positive, very positive. You can see a similarity in the college, and everyone is laying their cards at the table. And here at [school name], it is a culture of being yourself. There is no one judging you for anything. (Teacher 5)

Negative Feedback

Their teachers also expressed some dissatisfaction towards different aspects of the course. A lot of the negative feedback was generally pointed towards activities from the first half of the course. The teachers also pointed out some conceptual aspects that could have been improved.

Degree of difficulty

All the interviewed teachers indicated that either themselves or another teacher at the course had experienced the activities from the two first lessons too difficult. When one of the respondents gets asked what his feelings were after the first lesson, he says:

> No, I thought that I have a long way to go before I am able to teach this stuff and, I have fear towards teaching this as of now. Because I thought it looked awfully difficult. (Teacher 4)

Contextual aspects

The contextual aspects the teachers mentioned in the interviews that negatively impacted the course quality were mainly focused on the time aspects of the course. The first thing the teachers mentioned was that there was too much time between each course.

> Well, the lessons should have been arranged for four consecutive weeks. But, in school, that is not possible, with so many other things going on all the time. There are so many processes going on in parallel in the common time that it is very difficult arranging intensive courses. (Teacher 3)

The second thing they mentioned was the time at the day the course was performed.

> On Tuesday, the teachers have a common time from half-past one to half-past three. If you could have rotated, and let the teachers start later on the day, and let the teachers have common time at the start of the day instead. Then some of the teachers would not have been so tired at the course lessons. (Teacher 4)

5.2.3 What do teachers want to see in teacher training

The first interviewed teacher stated the thing he wants in such teacher training courses is to get an overview of quality resources. For example:

> I: How does your attitude towards programming change since your participation in the course?
>
> R: Well, for my part, the main focus of such courses is to get an overview of which resources are available (Teacher 1).

The teachers indicated that the relevancy of the activities and their connection to the teachers practice were the most important aspect to consider when designing activities. For example:

> But, the last part where you connect the topic to a specific subject, that was useful.(Teacher 5).

> I: What do you expect of the content of such teacher training courses?
>
> R: Something that has could be directly transferred to my classroom. This could be for example showing an already made teaching plan (Teacher 6).

5.2.4 Teaching Programming

The teachers' reflection about using programming in their teaching and the challenges they will meet in the future is presented in this section. When talking about teaching programming, the teachers were very focused on the challenges.

Challenges

All of the teachers' indicated that there is a lot of pedagogical aspects they have to figure out before being able to teach programming to the pupils. One of the teachers said:

> Some pupils struggle with reading comprehension. Also in mathematics as well as programming. It will take a longer time and creates more impatience. That is, there are so many small pedagogical things we have to solve in order to arrive at a satisfactory pedagogical plan. (Teacher 1)

Another challenge regarding teaching programming the teachers are mentioning is whether or not they are able to answer pupil questions or not. The teachers mentioned that they think they will receive more questions in a programming class compared to a regular math class. One of the teachers say:

> I do not feel I can walk around in a class and tell the pupils that, "*I don't know*", to every question they ask. That can not at least be my main answer. (Teacher 4)

Even though a lot of the teachers indicated it was uncomfortable not being able to answer the question from pupils, all the teachers answered that the solution was, to be honest towards the pupils.

> I: Do you feel exposed to a lack of competence?
>
> R: You have revealed yourself anyway, I think. It is worse to try to hide it. Then you have to be honest that you are unsure and wonder if you can figure it out together instead. (Teacher 1)

Adaptive teaching was also mentioned as a challenge in four of the six interviews. The teachers say that it is hard to differentiate assessments and education as there is a huge gap in the pupils' competence.

> I: What challenges do you think you will meet when teaching programming?
>
> R: That has to be the differences in the pupils' competence because some people know a lot. (Teacher 5)

One of the teachers is concerned about how the weakest pupils will handle the implementation of programming in the curriculum.

> And then I think it can be challenging in relation to the weak students. For that, if you are going to make a program for arithmetic on Pythagoras, then you must also understand Pythagoras. And, there may be many who do not. (Teacher 5)

The last challenges the teachers mentioned under the interviews were technical challenges. The teachers that had previous experience with teaching programming indicated that technical problems were the main time-consuming aspect of teaching programming. One of the teachers says:

> Difficulties with reading instructions or reading a task or another technical challenge. I think 90% of the challenges will be related to the technical bit. (Teacher 6)

Positive aspects

Under the interviews, most of the teachers expressed thoughts about the positive aspects of implementing programming in their teaching. The aspects stated from the teachers regarding programming were that it could lead to beneficial student participation and promote problem-solving skills (Computational thinking).

Student participation

As mentioned above, the teachers expressed they thought implementing programming in their teaching could lead to beneficial student participation. As the implementation of programming is in an early phase, the teachers stated that student participation is applicable and necessary as programming is relatively new to them as well. Some exampled from the interviews can be found below:

> I: How would you react if a student asked a question you do not know the answer to in class?
> R: Yes, it's really the rule of thumb here then: "I'm not a world champion, so here we have to find out things as a team." It can, on the other side be turned into something that is very positive for the teaching.
> I: How?
> R: Yes, by involving the student in the whole process, and forward find the knowledge together. It kind of becomes our project. But, it is important for the teachers to present the facts and explain that they are in line with the pupils (Teacher 1)

> I: How would you react if a student asked a question you do not know the answer to in class?
> R: [...] And, then it is just to hear if some of the pupils know the answer, and if they do, great, then they can give an objective of trying to teach the class. In that sense, I think it is an advantage. (Teacher 6)

> I am so lucky that I have three experts in my class who are interested in programming. They got tested now on Friday, and they may not have been adequately pedagogical, but I think they thought it was fun. (Teacher 2)

5.2.5 Attitude towards programming

Regarding the teachers' attitudes towards programming, their opinions are divided. Some teachers said that they would have implemented programming in the curriculum many years ago, while other teachers said that they would like to drop implementing programming in their teaching. However, all teachers generally positive towards programming being a part of the curriculum. The current implementation, however, is making the divided opinions.

Fear and skepticism

All the teachers mentioned in the interviews that either they or some teachers they know fear teaching programming. One of the teachers said:

> Because I think a lot of the teachers here think it is frightening to teach programming, something they don't know very well. Maybe some of the teachers are not used to throwing themselves into new things. (Teacher 4)

One of the teachers expressed that one of the reasons for the teachers having a fear towards programming and skeptical attitude is the way it is implemented in the curriculum. The teacher said:

> I: What do you think about programming being implemented in the curriculum?
>
> R: [...] I feel that it is a reform that is, not very much from the top-down, except with programming. For that is the only point that has been inserted from a political point of view that the grassroots has not asked for. That is one reason the teachers are skeptical, I think. (Teacher 4)

In the fourth interview, the teacher expressed that he was afraid of looking "stupid" in the classroom.

> As a teacher, you are always afraid of standing in the classroom looking stupid towards the pupils. That is natural, and I think it eventually is going to work out. (Teacher 4)

The fear the teachers related to were mostly related to the lack of training and experience, as well as skepticism towards the implementation in the curriculum. In three of the interviews, however, the teachers stated that they thought it was normal for teachers to be skeptical in the first period of implementation. They expressed this as they had experienced new additions to the curriculum

before. In the excerpt below, the teacher said that the same fear and skepticism towards innovation in the curriculum also was present when Geogebra, a software used in mathematcs[2] was introduced.

> I: Do you think your colleagues share your views on programming?
> R: Yes, I think so. I think many have to want to drop teaching programming. And then it's probably the case that the older you are, the more you will let go of it. [...] But, I think there is some skepticism among the teaching staff about new things anyway. Regardless of whether it was, as we are talking about here programming, you have a certain resistance in it and skepticism on a par with what we had when Geogebra was introduced too. (Teacher 2)

Difference in teacher attitude

When talking about the teachers' attitudes, it became clear that there was a wide range of different attitudes. The second teacher said:

> I think there are some who find it very exciting, who think it is positive and who see that you can use it as a tool too. For example, making mathematics feel more relevant. And then I think many people think it's scary. But I am only thinking, not really knowing (Teacher 2)

> Yes, the attitudes are very varied. Some teachers have a positive attitude towards programming and have prior experience as well, who think this will be unproblematic. [...] And then, some teachers are as negative as before, and think they will get everything served on a silver platter, and have little interest in laying down the effort needed themselves. So, we have teachers all across the spectrum (Teacher 2)

Positive towards programming in school

All teachers were generally positive towards implementing programming in the curriculum. None of the interviews teachers said that they explicitly had a negative attitude towards programming. The teachers expressed that they see the relevance of learning programming in today's society and that they are positive about the skills programming can promote. One of the teachers said.

> I: What do you think about the fact that programming is now included in the curriculum?
> R: [...] I think there is a lot of negativity connected to word programming. That it is difficult, incomprehensible, right? A computer, no one knows what is inside a computer. There is a lot of people thinking that. But then you forget thinking about the computational thinker as well. Programming is much more than only writing lines of codes with Python (Teacher 5).

[2]Read more about Geogebra here: https://www.geogebra.org/about

Three of the teachers expressed that the explosion in technology use in the last decades will lead to a need for computer and programming knowledge. Two of the teachers stated:

> I: What do you think about programming being implemented in the curriculum?
> R: I think it is good. It is actually about the time it was being implemented. If you think about that the general education should give pupils an insight and a picture of the world they are living in today, it is absolutely about time.

> I: Has your attitude towards programming changed since attending the course?
> R: [...] And I overall think it is very good that it is now a part of the curriculum. What did I tell you ten years ago? It has been too difficult to pull the load by myself earlier. So I'm just grateful that this is happening. So I think myself is quite open in relation to the new curriculum. Or that part of the curriculum at least (Teacher 1)

Mixed feeling about current implementation

When talking about the implementation of programming the curriculum, the teachers discussed whether implementing programming as a stand-alone subject would have been a better solution. Under are some of the statements:

> I am afraid the current implementation will lead to varying quality in the programming teaching. That it will be like, "okay, lets do something random." (Teacher 1)

> One thing that is not so good by implementing it into subjects like mathematics or science is that then it is only up to the math teachers and science teachers, and not many other teachers at the school, teaching different subjects, get involved. Then you lose a bit of the overall feeling that you see the usefulness of programming in other subjects as well. (Teacher 2)

> But, the feeling I got when I heard programming was implemented into my subjects was, "oh, shall we use time on something new, without anything being taken away and without the teachers getting more resources or time." (Teacher 4)

5.2.6 School community and collaboration

Another category that emerged from the analysis phase was school community and collaboration. Under this category, teachers' statements regarding their attitude towards teacher collaboration, creating a pedagogical forum for discussing programming, as well as challenges regarding teacher collaboration.

> I think the most important thing we can do, is to focus on discussing the pedagogical challenges around programming and not focusing that much on the actual tool. (Teacher 1)

Positive towards teacher collaboration

In the case of the course, all teachers were very positive towards having the course internally at their school. All the interviewed teachers expressed that such courses promote the teachers to collaborate more than usual.

> I: How do you think the course impacted the collegial cooperation regarding programming?
> R: I think it impacted the cooperation quite a bit. It was also nice to see which other teachers that have prior experience with programming. For example, I did not know [teacher name] had much experience. Such things, I get to know through the course right and to become aware of who sits on the competence, then you know whom to ask if you are stuck or have any questions. (Teacher 3)

One of the teachers stated that the designed course contributed to the teachers getting an overview over which teachers had prior experience and knowledge and that this could be a factor that could increase collaboration and teamwork. The teacher also states that seeing other teachers in the same situation as themselves contributes to making the whole situation more harmless. For example:

> Yes, it is very positive arranging the course internally at the school. It contributes to the teachers seeing equality, and everyone is laying their cards on the table right, which I think is unproblematic. Now we have a bunch of teachers who felt like a bunch of dinosaurs before the course that now sees that programming is actually not that bad and thinks that it is achievable. We plan together and test out together (Teacher 5).

Pedagogical forum

> I: What factors do you think must be the basis for designing an effective professional development course?
> R: [...] The thing I miss the most in the everyday work life is someone to talk pedagogy with (Teacher 1)

The quotation above is retrieved from the first interview. The teacher expressed how he feels there has been a lack of pedagogy talk related to programming at the school. He proposes to make a didactic and pedagogical forum at the school, where the teachers can collaborate and discuss programming. There were also other teachers who mentioned the need to discuss the pedagogical aspects of programming as well. For example:

> And I think it would have been beneficial, if we, the science teachers at the school, could observe each other for example and shared experiences that way, that would be nice. (Teacher 2)

> And, when there as so many teachers who are in the same situation regarding programming, we have to stick together and try out new tools together.

(Teacher 1)

Challenges

The teachers were all positive towards teacher collaboration and expressed they thought their school environment was friendly and helpful. Even though the teachers mainly were positive towards teacher collaboration, one of the teachers expressed that getting all teachers to collaborate is challenging.

> I: How do courses like this affect the teachers' collaboration towards teaching programming?
>
> R: Well, it is hard to say. There are people who always are interested in collaboration, while others have zero interest in that and run their own race. Even though we try to coordinate and plan, there will always be someone that will not bother. It will always be that way, I think (Teacher 6)

5.2.7 External challenges with programming

The teachers brought up a lot of external challenges regarding teaching programming. The common features that reoccurred were: lack of time, teacher discipline and attitude, political aspects, different prioritization at schools, and available programming resources.

In all the interviews, the teachers indicated that the lack of time was they faced towards teaching programming. Two of the teachers said they were comfortable teaching programming after the end of the course, while the rest of the teachers indicated more training is needed to reach a comfortable level of knowledge to teach. The two teachers who said they were comfortable teaching also indicated that more training would be beneficial and wanted, as they were only comfortable teaching very basic programming principles. Quotes under this sub-category can be found below:

> I: Do you feel the course has made you capable to individually plan a programming lesson?
>
> R: No, not yes. I still need more time. (Teacher 4)

> I: You mentioned the computational thinker. Do you think your colleges have that term in mind when they shall teach programming?
>
> R: No, I do not think so. And, the reason for that, the way I see it, is that the teachers have been given too little time and too little national training has been provided. (Teacher 3)

> But, the feeling I got was, "oh, we are going to use time on something new without being supplied with more resources or more time." (Teacher 4)

Another aspect some of the teachers brought up was that the difference in teacher programming competence eventually could lead to a difference in teaching quality.

> It will be, do not call it unfair, but a consequence of the teachers' different competence levels will lead to a difference in teaching as some are very experienced and know a lot of programs and tools. I think those teachers will perform higher quality teaching to their students compared to the teachers who are skeptical and inexperienced. (Teacher 4)

> I: Did you meet any challenges during the course?
> R: For my own part, I met no challenges. It was okay writing code with Python and figuring out new stuff. But, the greatest difficulty has to be, that we as a collegium, would not necessarily want to start at the same place but eventually ending at the same place of competence. (Teacher 6)

It also was pointed out in the interviews that the teachers missed official national resources, which provided guidelines to what is expected of the teachers, in more detail than the competence goals.

> I: What challenges do you think you will meet in the future when planning a programming lesson?
> R: I think the most challenging aspect is choosing the right tools. If you want to do something, there always exists a pre-made lesson you can take into account when planning, but choosing the right tools is difficult. And as any other planning process, and especially us following the new curriculum, because there exists almost no complete national material. (Teacher 6)

Chapter 6

Discussion

As mentioned in the introduction (section 1.1), the implementation of programming in the Norwegian curriculum is in the early days, and the situation could look quite different shortly. The designed PD program of this study, were designed to get the participating teachers, with little experience with programming, started on the path to becoming comfortable in their teaching of programming. The course was also designed based on the given context from the school. These factors were out of the researcher's control to change. This chapter will discuss the course in its given context in light of previous research and the study's research questions. The discussion aims to shape a picture of the number of obstacles and puzzles to solve when designing a PD program and discussing good PD practices to handle these elements properly. As a result, based on the argumentation from previous sections, a proposal of general good PD practices when designing a programming teacher course.

6.1 Chapter overview

The codes that emerged from the analysis and statements retrieved from the interview phase presented in the two previous chapters. This chapter will discuss the findings and discuss them up against the research question for this study. The following chapter takes a deeper look into the research questions and discusses them, considering previous relevant research.

6.2 Elements of consideration in professional development designs

After analyzing the interviews, the results indicate that there are contextual aspects to consider when designing a teacher training course regarding programming. This section discusses the teachers' external considerations during the interviews, which were not related to their pedagogical or programming skills. The mentioned points in this chapter aim to give an overview of issues that are outside of the teachers' control.

6.2.1 Time

All the teachers mentioned issues regarding time during the interviews. The common theme amongst the teachers is that they express that there are hardly any planned teacher training from above and a general sense of lack of time which is a big challenge towards teaching programming. A lot of the challenges regarding time are also outside of control to the individual teachers. The teachers expressed concerns about the lack of allocated time to learn programming from the respective school. From the literature, time or duration is one of the most critical aspects of PD, and the most effective form of PD is consistent in duration and adjusts as the teacher needs changes (Capraro, 2014). In Darling-Hammond et al. (2017) list over seven key features of effective PD, sustained duration is the last listed feature. The designed course from this study implemented four lessons over three months. In the given context, it was not reasonable to expect that a course with such a short duration was the preparation needed for the teachers to implement programming in their teaching. The teachers also reported that the time between each lesson and the course's total duration was too short. They also expressed how time is an issue all schools struggle with in general, not only related to training teachers in programming. In Norway, the delegated mandate the schools have received from the government is increasing is in such a scale that it is impossible to satisfy all needs.

On the contrary, sustained PD, over a more extended period, will instead contribute to higher quality in-depth discussions between the teachers and a more thoroughly designed course, as the course supervisor can adjust the course according to obtained feedback from participants (Garet et al., 2001). The teachers, however, expressed they would have liked short, intense courses rather than courses with a long sustained duration. They explained that this decision with a shorter, more intense course would motivate and engage more teachers, as some lose focus if a project is too long. Even though the teachers indicated that shorter and more intensive courses would be more appropriate for some teachers, they stated that it is the nature of their profession to be responsible

for their personal development of competence. In other words, teachers have an individual responsibility to teach the content found in the national curriculum. The teachers also expressed that they thought there would be a more significant focus on programming in the science section in the time forward, indicating that the teachers are self-reflecting on their teaching practices. According to Guskey (2003), this could eventually lead to teachers improving their classroom instructions. Studies have also shown that teachers' who perceive less pressure from work are more likely to be self-determined toward teaching and will implement instructions that are more pupil-directed and increases the pupils' learning freedom (Pelletier et al., 2002).

6.2.2 Location of the course

In this study, some benefits of arranging the course internally at the school came to light. Firstly, by arranging a course internally at school for a selected group of teachers, gathering information regarding the teachers' prior knowledge, expectations, and their attitude is easier compared to PD programs for teachers from different schools. By arranging the course internally at schools, it could be the premise of what Rouhani et al. (2021) refers to as turning a *stumbling stones* into a *stepping stones*. The presented *stumbling stones* presented by Rouhani et al. (2021), are *time*, *interdisciplinary* and *varying level of knowledge and motivation among students* and are mentioned similar aspects mentioned by Darling-Hammond et al. (2017) as effective features of *pd*, e.g. *time*, *collaboration* and *content focus*. The gathered information could be used to design course activities and possibly predict challenges the teachers will face during these activities and adapt the activities accordingly, in terms of difficulty and interdisciplinary.

Another benefit of arranging PD internally at schools is also the possibility of actively creating a professional community(in this case, a programming community for teachers) where the teachers could share ideas, give feedback and observe each other. Four of the interviewed teachers reflected on the benefits of creating such a community. Their argumentation of why they see the creation of such a community could be beneficial to teachers is in line with what Darling-Hammond et al. (2017) states as an element that could contribute to teachers learning and exploring together, which in the long run, positively can affect the teachers' achievements. A possible aim for the course instructor is to set the basis for the teachers to collaborate in course lessons and continuous professional development. As the literature also states, the role of coaching and exert control the course instructor hold is an essential factor of the outcome of such courses, both in terms of promoting collaboration among teachers and the overall teachers learning outcome (Darling-Hammond et al., 2017).

Finally, arranging PD courses internally at schools could also contribute to

an increase in the teachers' involvement during the course, as they know each other. The interviewed teachers all stated that they felt their school had an open and non-judgmental environment. This was also observed during the course lessons, as the teachers gave consecutive feedback on the course content and were active in classroom discussions. The willingness of the teachers to interact in discussion and contribute by giving feedback could be an effect of the course being arranged internally. One of the interviewed teachers also reported they had started sharing the teaching plans they made during the course with other colleagues. By arranging a course internally, it could indicate that the threshold of giving feedback and reflect around course activities is lower in internally arranged PD programs because a learning environment is already established as the teachers know each other. Darling-Hammond et al. (2017) refers to *feedback and reflections* as the fifth feature of effective PD, and emphasize that adjusting a course's content concerning the received feedback could lead to a more prosperous learning environment for the teachers

The location of a PD course arrangement is indicated through the interviews as a factor to consider when designing the course. All the teachers were positive towards arranging teacher training courses internally at their employed school. The main benefits reported by the teachers were collaboration and efficiency. Three teachers reported that assembling all the teachers from the science section to participate in a teacher training course at their workplace had various beneficial effects. First, they report a sense of ease by seeing other colleagues in the same shoes as themselves. Secondly, it also contributed to the teachers get to know which experiences the other teachers have in programming. Finally, each and all reported that arranging collective PD programs for teachers internally at their school positively increases the collaboration amongst them. Some teachers also mentioned that arranging the course internally at school contributed to more teachers showing up.

6.2.3 School, community and cooperation

Five teachers reported positive attitudes toward collaborating with other participating teachers and eventually collaborating with their pupils when they start teaching programming in the future. Three teachers also expressed that the course would have been much harder if collaboration was prohibited. These statements are being confirmed by the literature as well. Schools have increasingly focus on structured teaching as a collaborative community, and this articulates that teacher collaboration is an essential feature of PD (National Commission on Teaching and America's Workforce, 2016).

After the end of the third course lesson, one of the participating teachers approached the course instructor. The teacher had experience with a learning resource she wanted to demonstrate to the other teachers. Even though this

resulted in using the time the teachers could have used to plan their teaching plan, the teachers referred to the situation as a positive contribution to the last lesson. The teacher demonstrated a mathematics teacher resource, called *Kikora*, which has implemented block-based programming tasks for secondary school pupils. The situation was also explicitly mentioned in the interviews. One of the teachers reports he would never have known *Kikora* existed if it was not introduced during the course and that he would use *Kikora* in the first programming lesson he lectures. The resource the teacher demonstrated was a resource the researcher did not have access to, as it was licensed and paid for by the school to the mathematics teachers. To get an overview of what kind of experience the teachers have and which tools they have used could be an element to consider gathering information about in the planning phase.

6.2.4 Available school resources, tools, and teaching materials

It is essential to have a focus on adding high-quality content and activities in the course planning process (Darling-Hammond et al., 2017; Garet et al., 2001), and when revising the course design after each iteration of the design circle shown in figure 3.2. When preparing the course's activities, learning objectives and competence goals is the main focus. Other aspects like available school resources, tools, teaching materials, and other technical aspects are also essential to consider. The teachers participating in the study's course used a computer installed with Chrome OS. As a result, the researcher had to do research regarding which tools were compatible with Chrome OS. All teachers' expressed the need to have an overview of available resources but had various opinions regarding the number of resources that should be made available at the beginning of the course. External resources online have often been reported as an essential source of inspiration for teachers (Rouhani et al., 2020).

Three of the teachers expressed during the interview that they found it challenging to choose the right tools and resources to use when they planned the programming lesson for the last two lessons. Under the interviews, two teachers said there were introduced too many resources at the beginning of the course. The proposal of making fewer resources and tools available to the teachers at the beginning of the course would fall in line with the first feature of effective PD; content focus (Darling-Hammond et al., 2017; Garet et al., 2001). Contrary to the others, two teachers expressed that it is impossible to receive too many tools or resources during a PD course.

Teacher professional learning that is context-specific and context-based is a crucial factor in addressing the diverse needs of the participating teachers (Darling-Hammond et al., 2017). Therefore, the designed course activities should show the teachers activities they can use themselves in their teaching. This element was pointed out by the interviewed teachers during the interviews as well.

The resources and teaching materials delivered to the teachers should only be related to the course content and include teaching plans and other resources the teachers can use in their classes. As the relevancy of the content increases, it allows the teachers' to study the particular element or pedagogy or student learning in the outcome area (Antoniou and Kyriakides, 2013), test out new additions to the curriculum with the pupils (Penuel et al., 2011), and study their pupils work in class (Doppelt et al., 2009).

6.2.5 Difference in teachers' programming competence and prior experience

Another factor that will affect the course design is the participating teachers and their prior experience and knowledge. For example, as experienced in the first two lessons, some of the attending teachers (4) expressed that the two first lessons' content was too advanced and presented too quickly by the course supervisor. In addition, the teachers report that their lack of experience with programming in general, particularly with text-based programming, was one determinant that provoked a feeling of insufficient competence to reach the full learning potential of the training. On the other side, the remainder of the interviewees expressed that they faced no challenges during the course and examined the inclusion of Python as an encouraging factor to them. The teachers reporting this were also the teachers who had experience with programming of some kind earlier. In summary, the course supervisor should gather information about prior programming knowledge and experience of a PD program participants ahead of the course to concretize and shape the course content according to the responses. Having this information enables the course designer to implement activities that have a link with the course content and its learning objectives, in addition to adequately select activities with a suitable degree of difficulty (Partanen et al., 2017).

6.3 Which course design elements promote reaching the learning objectives?

The section will present the design elements which affected the course's learning outcomes for the participants. The design elements include the implemented course structure and activities. In addition, factors like teaching materials and technical choices are also presented in the section. This section will discuss different elements in PD programs that promote learning and help the teachers reaching the learning objectives of the course and will be connected to RQ2

found in section 1.3.

6.3.1 Course structure

As explained in section 4.3.3, the researcher decided to implement the course design into two parts; one section which aimed to give the teachers competence of basic programming principles, and one section where the teachers work with a project, in this case, designing teaching plans. As a result, the consensus amongst the teachers was mainly positive towards the course structure. Furthermore, five of the teachers indicated they especially appreciated the project-based section of the course. Their rationale of feeling more comfortable towards the section the teachers indicated could be a consequence of the section's collaborative nature. Four of the teachers also indicated a contributing factor to the second section's satisfaction connected to let the teachers work with subject matters they find interesting and meet their level of competence. It could be argued that these statements indicate that a project-based learning approach in such a course can promote active learning and collaboration, two features of effective PD (Darling-Hammond et al., 2017). Furthermore, the methodology of project-based learning has shown to increase students' commitment and excitement and results in more bottomless and efficient maintenance of new ideas (Rouhani et al., 2021).

The teachers were also mainly positive towards dividing the course into two sections in the overall course structure but emphasized potential issues with the approach. In general, the teachers thought the structural approach was a good idea, but that some aspects have to comply, e.g., having a reasonable amount of time to disposal; ideally, regular lessons over a sustained period (Darling-Hammond et al., 2017; Garet et al., 2001). Two of the teachers reported that they would have liked expanding the first section, as their prior programming experience was too low. The remaining four teachers reported they would have preferred to only work with teaching plans. Overall, the teachers were positive towards the structure but reported that the lack of time-restricted its potential.

How the course instructor designs the course structure can significantly affect the teachers' outcome of the course. When designing the structure, the context of the course must be considered. How many hours do you have? How many teachers are attending? What is their prior knowledge and competence? After the literature review, it became clear that previous PD programs that used traditional learning approaches had a negligible effect on the learning outcomes for the teachers (Cateté et al., 2020; Garet et al., 2001), in contrary to choose activities that promote active learning and are inquiry-based. By using a traditional learning approach, the method of teachers directs students to learn through memorization and learn techniques to solve tasks. In this study's course, designing a course divided into two parts was the initiative done by the

researcher to promote active learning. In this case, the course's second part was based on a project-based learning approach, where the teachers were given the task of designing their teaching plan. The researcher's rationale behind this choice was that the following approach would be perceived as highly relevant for the teacher in terms of relevancy to their work and at the same time promote the teachers to collaborate. Through observations in the course's lessons by the researcher and statements from the interviewed teachers, the implementation of project-based learning contributed to an increase in teacher collaborating when planning was considered highly relevant for their future teaching.

Overall, the way a PD course is structured has an impact on the teachers learning outcomes. With this in mind, a course structure focusing on both traditional learning approaches and inquiry-based learning was well received by the interviewed teachers. However, it is crucial to implement a realistic structure in terms of the course context, e.g., time and the participants' prior knowledge. In this case, it could be argued that implementing a project-based approach after only having tutored two lessons was too early. A possible solution, in this case, could be instead of implementing the project-based approach as a separate section, implement it as a part of all four lessons instead. This way, the instructor could introduce a topic followed by exploring how the topic can be used in their teaching. This could contribute to increasing both the activity of the teachers, as well as make the content focus more visible (Darling-Hammond et al., 2017).

6.3.2 Course activities

Relevancy of activities

The teachers report the relevancy of the activities in light of teaching as the most important factor of the designed course's activities. This is also confirmed in previous literature, and includes the feature *content focus* presented by Darling-Hammond et al. (2017) and Garet et al. (2001). In other words, the teachers have to see the use case of activities and understand how the introduced principle can be implemented in their teaching. When designing such PD programs, the activities should be designed based on the goals retrieved from the respective curriculum. A challenge regarding making relevant activities arise when the participating teachers lecture different subjects. In this study's case, all teachers were lecturing similar subjects, so this was not considered. However, the teachers had some remarks regarding the relevance of certain activities. The teachers reported that some of the course focus should have been on programming activities focusing on Science. The teachers indicated the challenges of using programming as a multidisciplinary tool. When programming was introduced to the Norwegian curriculum, it was introduced

as *programming and technology for everyone*. It was intended to become a tool that could be used in several subjects and multidisciplinary. The teachers reported that focusing primarily on one subject resulted in a lack of a general overview of how programming could be used multidisciplinary. This was also found by Rouhani et al. (2020), where the teachers in the study indicated prioritizing the STEM subjects in programming PD courses could cause a misunderstanding of how it is supposed to be a multidisciplinary tool used in several subjects.

Activities which promote active learning

When designing a course with a time frame of 4 lessons, the chosen activities had to be chosen appropriately in terms of motivating the course participants as well as having a high focus on the activity content, active learning, and collaboration (Darling-Hammond et al., 2017).

The teachers well received the activities, which promoted the teachers to explore a principle and actively contribute to their learning process. As the second-mentioned feature of effective PD, *active learning* is highlighting the importance of including the teachers in the planning process and use activities which promote active learning rather than traditional learning approaches (Darling-Hammond et al., 2017). In-house arranged PD programs make it easier for the course instructor to implement active learning activities compared to MOOCs because of the physical involvement with the teachers.

In the introductory section of this course, the researcher introduced the relevant programming principles individually, followed by a task the teachers were going to solve. The tasks were implemented to prevent the lecturing from the first course section, including activities where the teachers statically received information from the instructor. There were mixed responses to the tasks given in this section regarding the level of difficulty, but they were positive towards letting the teachers try out the principles for themselves. There was delegated much focus to include activities where the teachers could work individually or in groups when designing the course. The reasoning behind adding these tasks was explained from the researcher's personal experience of learning programming, where "*learning by doing*" is considered a good way of learning to program. All teachers said that they need more mass training in programming to be comfortable with teaching it.

6.3.3 Facilitating feedback and reflections

As mentioned in section 4.3.1, the design process of the course was iterative and adjusted throughout the course period. This was also a result of the course

being arranged internally at the school. The course lessons were lectured with all the teachers in the same room, making it easy to use the last minutes of the lessons to let the teachers reflect and give feedback on the lesson. For example, the feedback regarding Python, and its lack of motivational feedback given to the teachers during activities, resulted in changing the plan for the second lesson. The teachers also perceived this as a positive change towards lesson 2 compared to lesson 1. With this in mind, it could be argued that it is reasonable not to design and plan the course in much detail in the initial planning phase. When a course facilitates the possibility for the teachers to give feedback and reflect, the feedback received could contribute to a change of course design, which will eventually increase the course's learning outcome. An example of a previous course that also implemented changes as a result of teachers' feedback is the study of Yukselturk and Altiok (2017). By lowering the degree of difficulty and some course activities, a change in the teachers' perceived attitude and self-efficacy towards programming occurred considerably compared to the previous course.

Another important factor of letting the teacher reflect over the course and their learning outcome. As Han et al. (2015) report, teachers who are willingly and actively self-reflecting around their practice, will develop them to give improved instructions in their teaching and eventually be willing to modify the design of their instructions.

6.3.4 Elements to consider during course lessons

During the lessons, the course instructor role plays an essential role in the course's teachers' comprehension. Under the interviews, the teachers reported they were optimistic about the amount of help they could receive from the instructor. It is worth mentioning that this study's designed course at the most had 14 participating teachers. In other studies, the amount of attending teachers was usually higher. The low participation in the course resulted in the teachers receiving more guidance and assistance than larger-scale courses, housing more teachers. The importance of the instructor role is also emphasized as the fifth feature of effective (Darling-Hammond et al., 2017).

In terms of this study's course design, three of the teachers responded that they would have liked the course to get feedback on their work during the lessons or on their own time. The teachers would have liked to deliver their work at the end of each lesson and receive feedback on their work. This could most definitely be argued as a reasonable change to make to the original design. As we know from the literature, teachers who receive coaching and feedback are more likely to implement the topics into their teaching (Darling-Hammond et al., 2017; Garet et al., 2001). The feedback given during a course is beneficial for both the teachers and the course instructor (Garet et al., 2001)

6.4 How was the course experienced by the teachers?

6.4.1 Course specifics

Selection of programming language

The teachers reported varied feelings towards the course, but the overall consensus among the interviewed teachers was that the course increased self-confidence towards teaching programming. Even though they feel more relaxed, they indicated they still have to practice more to have such knowledge to feel safe when teaching programming. When the teachers reflect on why they feel more relaxed, they report that they feel they understood the logical way of thinking when they solved the tasks during the course, but that they still have to practice more to be comfortable with the syntax of a programming language. Choosing the correct programming language is, therefore, a challenge to decide when designing the course. The researcher experienced under the first two lessons that the teachers faced many challenges regarding the syntactical rules of Python. As a result, a lot of the time, supporting and exerting the teachers included mainly consisted of basic syntax mistakes, such as missing a colon or using the wrong indentation in their code. Other studies have experienced using programming languages that are block-based to be a possible solution of eliminating all the syntax errors during the course (Yukselturk and Altiok, 2017). Yukselturk and Altiok (2017) also reports that the teachers participating in their course thought the visual environment of Scratch and the feedback it gives the user would be more motivating to both teachers and students in teaching, compared to other text-based languages. With this in mind, changing the first lesson to use Scratch instead of Python could be a change that increases the teachers' learning outcome from the course and increase their perceived self-efficacy.

Teachers creating their own teaching plan

In the starting phase of the teachers creating their teaching plan, they faced challenges. During the lesson and the interviews, the first challenge indicated by the teachers was the challenge of getting started with the planning. The reported challenges were related to making plans that are engaging to the pupils while also being adapted to each student's knowledge level. Even though the guideline document they had received included many examples of previously made teaching plans and other learning materials, they reported facing challenges connecting it and making it relevant in their teaching. It is therefore also crucial for the course instructor to offer support for the teachers in this process (Rouhani, 2020).

On the other side, the results from the interviews indicate that the teachers liked the openness of the project, both in terms of constraints regarding collaboration and what tools they were allowed to use. This resulted in a variety of different made teaching plans, planned both individually or in groups. After the course, the researcher received teaching plans using Scratch, Python, and Micro-bits in focus. Therefore, it could be argued that creating their teaching plan increases the level of collaboration amongst the colleague internally at a school. Furthermore, the variety of tools used in the teaching plan indicates similar to Rouhani et al. (2020), the importance of making tools, resources, and other teaching material that can inspire and motivate the teachers available. Another positive element the teachers reflected on regarding the activity was how it promoted the teachers to look up competence goals, create learning objectives, and find design quality activities suited for their class and context. This resulted in the teachers felt the activity was highly relevant. The teachers also stated they planned to use the teaching plan later in their class from the interviews. This may also indicate an increase in the teachers' perceived self-efficacy (Thorsnes et al., 2020).

6.5 Implication of the research

After designing, implementing, and evaluating the PD program used in this study, the results indicate there are many aspects to consider during the planning process and that internally arranging PD programs could affect the outcome of the course in several directions. For example, many teachers have little experience with programming and report fear of teaching it. This study indicates that short, internally arranged courses could impact the teachers' attitude and self-efficacy towards teaching programming. Internally PD courses could therefore be a suitable and effective way of preparing the teachers to teach programming.

The most important advantages of internally arranged courses are indicated through the study as aspects of collaboration and facilitating feedback and reflections. However, the teachers also reported they miss a community where they can discuss didactic and pedagogical aspects and share experiences. The researcher argues that internally arranged courses could be used as such a community. A further investigation of how such courses can promote the creation of such communities would be helpful to assist PD program instructors and teachers in the future.

The interviews show that teachers want to see activities they directly can transfer over to their teaching. This is another aspect worth having an increased focus on in the future. The researcher agrees with adding relevant activities during the course but emphasizes the need for tasks that challenges the teach-

ers' critical thinking skills and promote exploring. Even though there are many resources on programming, there is still a lack of quality material that is relevant for the teachers. Therefore, a further look into what type of activities is effective is recommended.

The competence goals related to programming in the Norwegian curriculum do not limit what programming language should be used in the teachers' education. The lack of specified tools to use and the overwhelming amount of resources and tools available, the teachers reported as a challenging aspect of the implementation. Even though arguments could be made of how the open competence goals gives the teachers more room to play with, the researcher recommends to at least list a few recommended text-based languages to use. The only proposed language is the block-based language Scratch (Sevik and fl, 2016). The researcher argues that proposing different choices can decrease the teachers' sense of the overwhelming amount of resources.

Chapter 7

Conclusion

This chapter concludes the study. First, a summarizing of the research will be presented, followed by giving a conclusion of this research in light of the formulated research questions. Then some points regarding further research will be presented, and finally, the limitations of this study will be discussed.

7.1 Answering the research questions

This master's thesis has explored which external and internal factors affect the design process of an internally arranged professional development course in programming for in-service teachers by designing, implementing, and evaluating the designed course. The main research question of the study is:

Which internal and external factors affect the design process of internally organized professional development programs in programming for in-service teachers?

This research question was further explored through the following sub-questions:

RQ1 Which elements should be considered when designing programming teacher course?

RQ2 Which course design can help reaching the learning objectives of such a course?

RQ3 How did the course participants experience the designed course?

The course was designed in the context given by a participating secondary school in Norway. The course design, observations, feedback from teachers, and six teacher interviews have been analyzed in this study and represent the data collection. The following section will answer each research question individually.

RQ1 Which elements should be considered when designing a internally organized programming teacher course?

The result from the study indicated several internal and external elements which should be considered when designing a PD course. The elements have been discussed in light of the seven features of effective PD by Darling-Hammond et al. (2017). The results show that internally organizing Professional Development (PD) programs can positively influence the school community and increase collaboration among the teachers. The simplicity of gathering relevant information regarding the teachers' prior knowledge and experience is also indicated through the results as an advantage. Even though the study indicates positive aspects of internally organizing PD courses, there are still some elements that were challenging to wield. The element of *time* and *available resources and teaching materials* is one of the occurring challenges of organizing PD courses in general. With this in mind, there was an overall consensus amongst the teachers that they prefer organizing the teacher training in-house as it will result in a course adapted to their community and learning environment. When designing PD courses around already established teacher environments, the study indicates it can be a good basis for the creation of didactic and pedagogical communities for teachers inside the school. In the long run, this can arguably increase the teachers' self-efficacy.

RQ2 Which course design can help reaching the learning objectives of such a course?

The study indicates different design elements which contribute to the teachers obtaining the learning objectives. In terms of what the teachers want to see during teacher training, *content relevancy* is reported as an essential factor for the teachers. These activities could be where the teachers transfer their practice or concrete examples from their course to their practice. The results also indicate that the teachers were positive towards the inquiry-based activities, as this resulted in exploring the tools more in detail. The study also indicated that using programming tools with a more engaging user interface is more motivating for the teachers and increases their willingness to explore them by themselves. The course design in this study was divided into two sections, one traditional and one project-based section. The study's results indicate that letting the teachers make their teaching plan contributed to increased collaboration amongst the teachers, and the teachers reported the activity as highly

relevant. The teachers did indicate they felt the course should include some feedback on the teachers' work during the course. To implement this in the course design would have been an addition worth considering on the overall course design.

RQ3 How did the course participants experience the designed course?

The results indicate that the teachers had mixed opinions about the course. The teachers reported the time frame the course was designed within. All teachers indicated shorter duration between each course and more courses would have been more effective. The results also indicated that the teachers report a lack of skill and experience with programming. The teachers had mixed opinions towards the difficulty of the course. Some teachers were very positive about using Python and text-based programming, while the other half proposed block-based programming as more suitable. The teachers report that internally organized courses contribute to an increase in collaboration amongst the teachers. There was also an indication that internally organized courses could make every teacher's competence visible to each other, which could turn out to teachers knowing which colleagues have experience and knowledge and can address them in the future if they need support.

In conclusion, the teachers indicate that the course's primary outcome is a sense of relief. In addition, they report that the course made them see they are capable of teaching programming in the future and could indicate a positive change in their perceived self-efficacy towards teaching programming.

7.2 Limitations of the research

There are several limitations worth addressing in this study. When conducting research, it is vital to present the results reliable. In this work, reliability was essential to reflect on by the researcher. For example, when transcribing the teacher interviews, the researcher had to avoid interpreting two similar statements from teachers differently due to how it was transcribed. Though, the reliability in transcriptions has shown to be better when it is the same person conducting and transcribing the interviews (Kvale et al., 2015). Another aspect worth mentioning is the researcher's active role in this research. The researcher knew the participating school and participating teachers before the study, as the researcher had performed teacher practice there through his study program. Therefore, the researcher had to be attentive to his active role in the research and avoid adding information about the school or teachers which could harm them in any way. In addition to this, it is also, as mentioned earlier, impossible for the researcher to assume his attitudes and opinions while researching, especially under the analysis of the transcribed interviews. To ad-

dress this, comparing findings in light of other similar research could enhance the reliability of the research (Robson and McCartan, 2017). The chapters, including information about the school, teachers, and discussion of the study's findings, were also sent to the respective school and interviews teachers for validation.

Another limit of this study is the number of participants. On average, ten teachers were attending each course lesson. There were also only interviews with six teachers. Therefore, the results can only give a slight indication of the effects of internally organizing a teacher training course in programming. Even though the results are pretty consistent with other research's results, there will be a need for more research on the topic to confirm this study's findings.

It is also worth mentioning that this study is conducted in an early phase of implementing programming in the Norwegian curriculum. The situation will look quite different in the next couple of years. Therefore, it is important when evaluating this research to evaluate it in the context and time it is conducted and use it as a guideline to identify the faced challenges when designing such a course.

7.3 Further work and research

The entry of programming in curricula worldwide is still in the beginning phase, and there are many unanswered questions. Nevertheless, some elements emerged from the results and discussion of this study, which can be relevant to further research.

The results indicate that the teachers were positive towards organizing the course internally at their school as the results indicate that this can positively affect various elements of professional development, like teacher collaboration and feedback and reflection. The researcher considers the positive effects it had on the teacher community and collaboration as very relevant. This is also mentioned in the section "*professional development and school development*" in the Norwegian core curriculum ('Professional environment and school development', 2021). Further research which confirms these findings and takes a closer look at the effects of internally organized PD courses is recommended.

Another aspect that might be interesting to research in the future is to connect the data to demographic information, like age and gender. This was primarily avoided in this study due to privacy concerns. As the teachers were positive towards the project-based activity of designing their teaching plans, research into how this activity can be conducted didactically and pedagogically is another aspect worth further investigating.

Bibliography

Antoniou, P. & Kyriakides, L. (2013). A Dynamic Integrated Approach to teacher professional development: Impact and sustainability of the effects on improving teacher behaviour and student outcomes. *Teaching and Teacher Education*, *29*, 1–12.

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Advances in Behaviour Research and Therapy*, *1*(4), 139–161.

Bandura, A. (1986). *Social foundations of thought and action : a social cognitive theory*. Prentice-Hall.

Bandura, A. (1997). *Self-efficacy : the exercise of control*. Freeman.

Bates, C. & Morgan, D. (2018). Seven elements of effective professional development. *The Reading Teacher*, *71*, 623–626.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P. & Punie, Y. (2016). Developing computational thinking in compulsory education. implications for policy and practice. *EUR - Scientific and Technical Research Reports*. https://doi.org/10.2791/792158

Bocconi, S., Chioccariello, A. & Earp, J. (2018). The Nordic approach to introducing Computational Thinking and programming in compulsory education. *Report prepared for the Nordic@ BETT2018 Steering Group*.

Boud, D., Keogh, R. & Walker, D. (2013). *Reflection: Turning experience into learning*. Routledge.

Capraro, R. (2014). Impact of Sustained Professional Development in STEM on Outcome Measures in a Diverse Urban District. *The Journal of Educational Research*.

Cateté, V., Alvarez, L., Isvik, A., Milliken, A., Hill, M. & Barnes, T. (2020). Aligning theory and practice in teacher professional development for computer science. *Koli Calling '20: Proceedings of the 20th Koli Calling International Conference on Computing Education Research*.

Curriculum for Mathematics Year 1–10. (2020). https://www.udir.no/lk20/mat01-05?lang=eng

Darling-Hammond, L., Hyler, M. E. & Gardner, M. (2017). Effective teacher professional development. *Learning Policy Institute*.

Darling-Hammond, L. & Youngs, P. (2002). Defining "Highly Qualified Teachers": What Does "Scientifically-Based Research" Actually Tell Us? *Educational Researcher*, *31*, 13–25.

Desimone, L. (2011). A Primer on Effective Professional Development. *The Phi Delta Kappan*, *92*, 68–71. https://doi.org/10.2307/25822820

Desimone, L. & Garet, M. (2015). Best Practices in Teachers' Professional Development in the United States. *7*, 252–263.

Doppelt, Y., Schunn, C., Silk, E., Mehalik, M., Reynolds, B. & Ward, E. (2009). Evaluating the impact of a facilitated learning community approach to professional development on teacher practice and student achievement. *Research in Science*, *27*, 339–354.

Efendi, R. & Sanjaya, R. (2017). Implementation of project based learning: Research overview. *Applied Science and Technology*, *1*(1), 279–285.

Eksempeloppgaver i matematikk for 10. trinn. (2021). https://www.udir.no/eksamen-og-prover/eksamen/eksempeloppgaver/eksempeloppgaver-i-matematikk-grunnskolen/#

Gallagher, H. A., Arshan, N. & Woodworth, K. (2017). Impact of the National Writing Project's College-Ready Writers Program in high-need rural districts. *Journal of Research on Educational Effectiveness*, *10*(3), 570–595.

Garet, M., Porter, A., Desimone, L., Birman, B. & Yoon, K. S. (2001). What Makes Professional Development Effective? Results From a National Sample of Teachers. *American Educational Research Journal - AMER EDUC RES J*, *38*, 915–945.

Glaser, B. G. & Strauss, A. L. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine de Gruyter.

Guidelines for Research Ethics in the Social Sciences, Humanities, Law and Theology. (2019). https://www.udir.no/lk20/overordnet-del/?lang=eng

Guskey, T. (2003). Analyzing Lists of the Characteristics of Effective Professional Development to Promote Visionary Leadership. *Nassp Bulletin*, *87*, 4–20.

Han, S., Yalvac, B., Capraro, M. & Capraro, R. (2015). In-service Teachers' Implementation and Understanding of STEM Project Based Learning. *Eurasia Journal of Mathematics, Science and Technology Education*, *11*, 63–76.

Hester, O. R., Bridges, S. A. & Rollins, L. H. (2020). "Overworked and underappreciated": special education teachers describe stress and attrition. *Teacher Development*, *24*(3), 348–365. https://doi.org/10.1080/13664530.2020.1767189

Hevner, A., R, A., March, S., T, S., Park, Park, J., Ram & Sudha. (2004). Design Science in Information Systems Research. *Management Information Systems Quarterly*, *28*, 75–.

Holmboe, C., McIver, L. & George, C. E. (2001). Research agenda for computer science education. *PPIG*, *13*.

Knapp, M. S. (2003). Chapter 4: Professional development as a policy pathway. *Review of research in education*, *27*(1), 109–157.

Krajcik, J. S., Blumenfeld, P. C., Marx, R. W. & Soloway, E. (1994). A collaborative model for helping middle grade science teachers learn project-based instruction. *The Elementary School Journal*, *94*(5), 483–497. http://www.jstor.org/stable/1001838

Kvale, S., Brinkmann, S., Anderssen, T. & Rygge, J. (2015). *Det kvalitative forskningsintervju*. Gyldendal akademisk.
LK20. (2020). https://www.udir.no/laring-og-trivsel/lareplanverket
Mizell, H. (2010). *Why Professional Development Matters*. ERIC.
National Commission on Teaching and America's Workforce. (2016). What matters now: A new compact for teaching and learning (National Commission on Teaching and America's Future, Ed.).
Nvivo 12. (2021). https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home
on European Computing Education (CECE), T. C. (2017). *Informatics Education in Europe: Are We All In The Same Boat?* (Tech. rep.). New York, NY, USA, Association for Computing Machinery.
Partanen, T., Niemelä, P., Mannila, L. & Poranen, T. (2017). Educating computer science educators online - a racket mooc for elementary math teachers of finland, 47–58. https://doi.org/10.5220/0006257800470058
Peffers, K., Tuunanen, T., Rothenberger, M. A. & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, *24*(3), 45–77.
Pelletier, L., Séguin-Lévesque, C. & Legault, L. (2002). Pressure from above and pressure from below as determinants of teachers' motivation and teaching behaviors. *Journal of Educational Psychology*, *94*, 186–196.
Penuel, W. R., Gallagher, L. P. & Moorthy, S. (2011). Preparing Teachers to Design Sequences of Instruction in Earth Systems Science: A Comparison of Three Professional Development Programs. *American Educational Research Journal*, *48*(4), 996–1025.
Postholm, M. (2010). *Kvalitativ metode: en innføring med fokus på fenomenologi, etnografi og kasusstudier*. Universitetsforlaget.
Professional development. (2021). Retrieved April 7, 2012, from https://dictionary.cambridge.org/dictionary/english/professional-development
Professional environment and school development. (2021). https://www.udir.no/lk20/overordnet-del/3.-prinsipper-for-skolens-praksis/3.5-profesjonsfellesskap-og-skoleutvikling/?lang=eng
Robson, C. & McCartan, K. (2017). *Real World Research, 4th Edition*.
Rotter, J. (1966). Generalized expectancies for internal versus external control of reinforcement. *Psychological monographs*, *80 1*, 1–28.
Rouhani, M., Farshchian, V. & Divitini, M. (2020). Teaching Programming in Secondary Schools: Stepping and Stumbling Stones. *Interaction Design and Architecture(s)*, *47*, 48–68.
Rouhani, M., Olsø, A. & Divitini, M. (2021). Project-based learning and training of in-service teachers in programming: Projects as a bridge between training and practice.
Rouhani, M. (2020). Programming for teachers: Supporting participants in defining their learning path in a flexible online learning trajectory course.
Sanne, A., Berge, O., Bungum, B., Jørgensen, E., Kluge, A., Kristensen, T. & Voll, L. (2016). Teknologi og programmering for alle. En faggjennomgang med forslag til endringer i grunnopplæringen. *Oslo*, *19*.
Semadeni, J. H. (2009). Taking charge of professional development : a practical model for your school.

Sevik, K. & fl, M. (2016). Programmering i skolen, 1–32. https://www.udir.no/globalassets/filer/programmering__skolen.pdf

Simon, H. A. (1996). *The Sciences of the AFrtificial* (3rd ed.). MIT Press.

Sindre, G., Giannakos, M., Krogstie, B. R., Munkvold, R. & Aalberg, T. (2018). Project-based learning in it education: Definitions and qualities. *Uniped*, *41*(02), 147–163.

Solomon, C., Harvey, B., Kahn, K., Lieberman, H., Miller, M. L., Minsky, M., Papert, A. & Silverman, B. (2020). History of logo. *Proc. ACM Program. Lang.*, *4*(HOPL). https://doi.org/10.1145/3386329

Strauss, A., Corbin, J. & SAGE. (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. SAGE Publications.

Strauss, A., Strauss, A., Corbin, J. & Corbin, J. (1998). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications.

Thorsnes, J., Rouhani, M. & Divitini, M. (2020). In-service teacher training and self-efficacy. https://doi.org/10.1007/978-3-030-63212-0_13

Vivian, R., Falkner, K. & Falkner, N. (2014). Addressing the challenges of a new digital technologies curriculum: MOOCs as a scalable solution for teacher professional development. *Research in Learning Technology*, *22*.

Waite, J. (2017). Pedagogy in teaching Computer Science in schools: a literature review (After The Reboot: computing education in UK Schools). *24*, 2019.

Wang, F. & Hannafin, M. (2005). Design-based research and technology-enhanced learning environments. Educational Technology Research and Development, 53(4), 5-23. *Educational Technology Research and Development*, *53*, 5–23.

Wei, R. C., Darling-Hammond, L., Andree, A., Richardson, N. & Orphanos, S. (2009). Professional learning in the learning profession: A status report on teacher development in the us and abroad. technical report. *National Staff Development Council*.

Yukselturk, E. & Altiok, S. (2017). An investigation of the effects of programming with scratch on the pre-service it teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, *48*(3), 789–801. https://doi.org/https://doi.org/10.1111/bjet.12453

Zee, M. & Koomen, H. M. Y. (2016). 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*, *86*(4), 981–1015.

Appendix A

Interview guide

Intervjuguide

Bakgrunn

1. Hvilket trinn underviser du på?
2. Hvilke(t) fag underviser du i?
3. I hvilke av disse fagene skal du bruke programmering i undervisningen?

Programmering og holdninger

4. Hvilke erfaringer har du med programmering før deltagelse på introduksjonskurset?
    a. Har du undervist i programmering før? Hvis ja:
        i. Hvilket fag?
        ii. Hvordan gjennomførte du økten og hvordan følte du at det gikk?
    b. Har du tatt noen kurs tidligere som inneholdt noen form for programmering?
    c. Er det noen andre erfaringer du ønsker å dele?
5. Kan du beskrive din kompetanse i programmering før kurset?
6. Hva tenker du om at programmering er inkludert i lærerplanen?
7. Hvordan har dine holdninger til programmering endret seg siden deltagelsen på kurset?
8. Føler du at kurset har påvirket din interesse for programmering?

Kurs og mestringsevne

9. I hvilken grad følte du deg rustet til å undervise i programmering før kurset?
10. Hvilke utfordringer tror du at du nå kommer til å møte på når du skal planlegge programmeringstimer fremover?
11. Hvilke utfordringer tror du at du møter i programmeringsundervisningen?
12. I hvilken grad føler du at kurset har gjort deg i stand til å planlegge egne undervisningstimer i programmering?
13. I hvilken grad har kurset forberedt deg å lage oppgaver til elever i programmering?
14. Føler du at kurset har gjort deg mer komfortabel til å undervise programmering?
15. Hvordan ville du håndtert en situasjon der en elev spør deg om noe du ikke kan om programmering?

16. Føler du at kurset har gitt deg en oversikt på hvordan du kan tilpasse undervisningen?
17. Hvilke utfordringer møtte du på underveis på kurset?
18. Var det noen spesifikke deler av kurset som endret en interesse for programmering hos deg?
19. Hvis du skulle ha opparbeidet deg mer kunnskap i programmering etter kurset, har du en tanke på hvordan du ville ha gjort dette?
20. Hvordan påvirket kurset din følte mestringsevne i å undervise programmering?
    a. På hvilken måte?
21. Hvordan påvirket kurset din følte mestringsevne i å løse problemer med programmering?

Kurs og kompetanseheving

22. Hva sitter du igjen med av lærdom etter kurset?
23. Er det noen ting du skulle ønsket hadde blitt gjort annerledes?
    a. Måten kurset er lagt opp på
    b. Innhold
    c. Egenlæring
    d. Oppfølgning fra kursholder
    e. Forbedringer?
24. Hva tenker du om at de to siste kursene ble satt av til arbeid med undervisningsplaner?
25. Hva tenker du om kursets oppdeling i 2 deler (En undervisningsdel/En workshop/prosjekt)
26. Føler du at du fikk videreført noe fra del 1 over til planleggingen?
27. Hva tenker du om å planlegge kurs på denne måten?
28. Hvordan tror du man kan utføre kompetanseheving av lærere i programmering på en god måte?
29. Hvilke holdninger tror du dine kollegaer har til programmering?
30. Tror du at kurs som dette, kan være med på å påvirke hvordan lærerne jobber sammen frem mot å undervise programmering?
    a. Hvis ja/nei
        i. På hvilken måte?

Avslutning

Takke læreren for at han/hun kunne stille på intervjuet. Spør læreren om det er noe mer han/hun vil tilføye etter lydopptakeren er skrudd av.

Appendix B

Form of consent

Vil du delta i et forskningsprosjekt tilknyttet min masteroppgave?

Anmodning om tillatelse til bruk av lydopptak av intervju.

Jeg er student på lektorprogrammet i realfag ved NTNU. Dette semesteret skal jeg skrive min masteroppgave i informatikk. Jeg ønsker å finne ut av hvordan skoler effektivt kan drive kompetanseheving av lærere innad i skolen når det kommer nye krav til lærerplanen. I mitt tilfelle, er disse nye kravene programmering i matematikk og naturfag. Den nye lærerplanen, fagfornyelsen, innfører programmering som pensum i mange fag fra og med grunnskolen, og jeg har derfor en interesse å forske på hvordan vi kan etterutdanne lærere til å få tilstrekkelig programmeringskompetanse for å utføre egen undervisning.

For å få så godt dokumenterte data som mulig, er det ønskelig å gjøre lydopptak av eventuelle intervjuer av lærere. Derfor ber jeg om tillatelse til å kunne gjøre lydopptak, samt samle inn materiale produsert av lærere. Det er snakk om intervjuer av ca. 10 lærere. Forutsetningen for tillatelsen er at alt innsamlet materiale blir behandlet med respekt og blir anonymisert, og at prosjektet ellers følger gjeldende retningslinjer for etikk og personvern. Det er helt frivillig å delta og man kan til enhver tid trekke seg fra deltakelse uten å måtte oppgi noen grunn til det.

Dine rettigheter

Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke personopplysninger som er registrert om deg, og å få utlevert en kopi av opplysningene,
- å få rettet personopplysninger om deg,
- å få slettet personopplysninger om deg, og
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

Jeg skal gjennomføre et innføringskurs i programmering til lærerne, og ønsker å intervjue dem i forkant og etterkant av kurset. Her vil jeg få en oversikt over forventningene lærerne har i forkant av kurset, og hva de sitter igjen med etter kurset. Materialet vil kun bli hørt av meg og mine veiledere. I det som presenteres fra prosjektet vil involverte personer bli anonymisert. Innsamlede data vil bli slettet etter at prosjektet er avsluttet, senest 14. Juni 2021.

Hvis dere vil vite mer om dette, eller hva det innsamlede materialet skal brukes til, så er det bare å ta kontakt med meg på telefon eller epost (se øverst på første side for detaljer).

Faglig ansvarlig ved NTNU er Monica Divitini. tlf.: +47 73594462; epost: divitini@ntnu.no

NTNUs personvernombud er [anonymisert]: tlf. [anonymisert]: epost [anonymisert].

Hvis <du/dere> har spørsmål knyttet til NSD sin vurdering av prosjektet, ta kontakt med:

- NSD – Norsk senter for forskningsdata AS på epost (personverntjenester@nsd.no) eller på telefon: 55 58 21 17.

Jeg håper dere synes denne forskningen er av verdi, og at dere er villig(e) til å være med på den. Jeg ber om at svarslippen på neste side fylles ut om hvorvidt dere gir eller ikke gir tillatelse til deltakelse i prosjektet.

På forhånd takk!

Vennlig hilsen

Prosjektansvarlig	Student
[anonymisert]	[anonymisert]

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet «Professional development in programming for in-service teachers at secondary schools.» og har fått anledning til å stille spørsmål. Jeg samtykker til:

- ☐ å delta i intervju som vil bli tatt opp med lydopptaker

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

---

(Signert av prosjektdeltaker, dato)

Appendix C

Approval from NSD

Vurdering (1)

17.02.2021 - Vurdert

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg den 17.02.2021, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

DEL PROSJEKTET MED PROSJEKTANSVARLIG
Det er obligatorisk for studenter å dele meldeskjemaet med prosjektansvarlig (veileder). Det gjøres ved å trykke på "Del prosjekt" i øvre venstre hjørne av meldeskjemaet.

MELD VESENTLIGE ENDRINGER
Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke type endringer det er nødvendig å melde:
https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema
Du må vente på svar fra NSD før endringen gjennomføres.

TYPE OPPLYSNINGER OG VARIGHET
Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 14.06.2021.

LOVLIG GRUNNLAG
Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake.
Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

PERSONVERNPRINSIPPER
NSD vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om:

• lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
• formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke behandles til nye, uforenlige formål
• dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet
• lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet

DE REGISTRERTES RETTIGHETER
Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: åpenhet (art. 12), informasjon (art. 13), innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18), og dataportabilitet (art. 20).

NSD vurderer at informasjonen om behandlingen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER
NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

Ved bruk av databehandler (spørreskjemaleverandør, skylagring eller videosamtale) må behandlingen oppfylle kravene til bruk av databehandler, jf. art 28 og 29.

For å forsikre dere om at kravene oppfylles, må dere følge interne retningslinjer og/eller rådføre dere med behandlingsansvarlig institusjon.

OPPFØLGING AV PROSJEKTET
NSD vil følge opp ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet.

Lykke til med prosjektet!
Tlf. Personverntjenester: 55 58 21 17 (tast 1)

Appendix D

List over survey questions

Programmeringskompetanse

0 %

Obligatoriske felter er merket med denne stjernen *

Spørreskjema før programmeringskurs på Rosenborg skole

Hei matematikk- og naturfagsseksjonen på Rosenborg skole. 15. mars er det klart for å gjennomføre første del av programmeringskurset jeg skal holde for dere denne våren. I den anledning har jeg et ønske om at dere besvarer denne spørreundersøkelsen. Dataen fra denne spørreundersøkelsen vil bli brukt i tilknytning min masteroppgaveskrivning. Jeg forsker for tiden på hvordan et lærerkurs som benytter seg av Prosject Based Learning(PjBL) kan være med på å øke lærerens mestringstro og motivasjon.

Denne undersøkelsen vil derfor ta for seg hovedtemaene **bakgrunn, holdninger, forkunskaper og forventninger.**

Spørreundersøkelsen består av til sammen 12 spørsmål, og vil ta deg i underkant av 5 minutter å besvare.

Neste side

Programmeringskompetanse

20 %

Obligatoriske felter er merket med denne stjernen *

Bakgrunn

Hvilke fag underviser du i? *

Hva er tema dere gjennomgår i ditt fag i løpet av de neste ukene? *

Forrige side

Neste side

Programmeringskompetanse

40 %

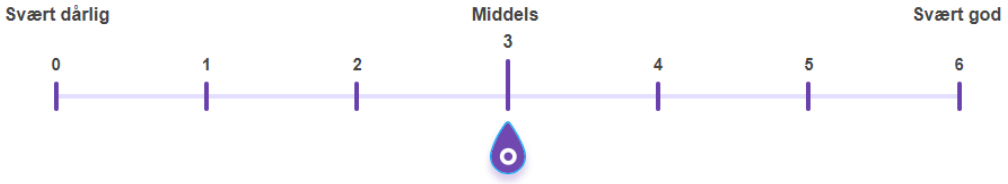
Obligatoriske felter er merket med denne stjernen *

Forkunnskaper og holdninger

I denne delen ønsker jeg å vite litt om dine forkunnskaper innenfor prgrammering. I denne delen gir jeg dere enkelte påstander. Jeg vil at dere skal svare et tall mellom 1-7 om dere er 1 (svært uenig) eller 7 (svært enig).

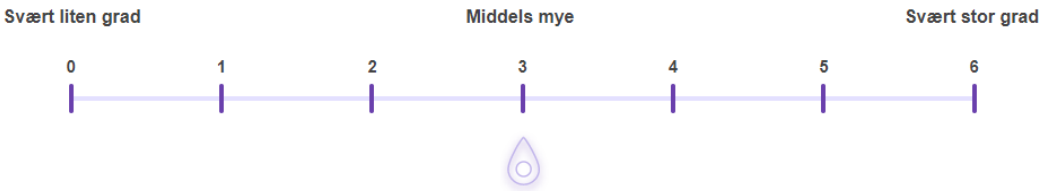
Forkunnskaper

Jeg anser min egen digitale kompetanse som *



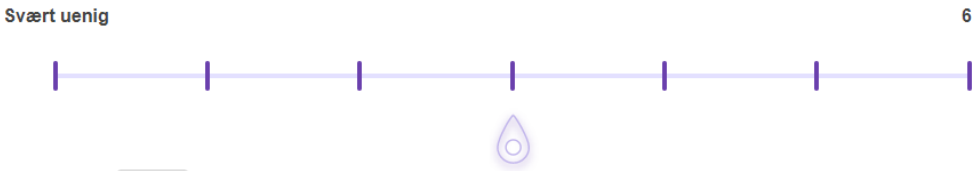
Verdi 3

I hvilken grad har du holdt på med programmering tidligere? *



Verdi

Jeg har oversikt over hvilke kompetansemål som er aktuelle for meg i mitt fag med hensyn på programmering *



Verdi

Forrige side

Neste side

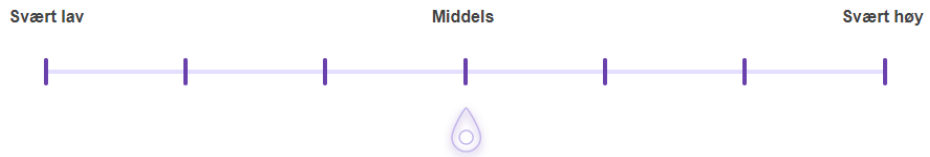
Programmeringskompetanse

60 %

Obligatoriske felter er merket med denne stjernen *

Motivasjon og holdninger

Jeg anser min motivasjon til å lære seg programmering som *



Verdi

Jeg ser en nytte i at alle elever skal lære seg litt programmering *

Verdi

Jeg tror at programmering kan være et nyttig verktøy for enkelte elever i undervisning i matematikk *

Verdi

Jeg tror at programmering kan være et nyttig verktøy for enkelte elever i undervisning i naturfag *

Verdi

Jeg synes eller tror programmering er vanskelig å lære seg *

Verdi

Jeg har en tanke om hvordan jeg kan implementere programmering i min undervisning *

Verdi

Forrige side

Neste side

Programmeringskompetanse

80 %

Obligatoriske felter er merket med denne stjernen *

Forventninger

Hvilke forventninger har du til programmeringskurset?

Forrige side

Send

Appendix E

Planning guide given to the teachers

Planleggingsguide

Dette dokumentet skal være med på å hjelpe deg til å strukturere og planlegge en undervisningstime i ditt respektive fag. Spørsmålene i denne planleggingsguiden er kun ment for å være veiledende og skal hjelpe deg på vei i planleggingen. Under finner du en tabell som lister opp en del nyttige ressurser jeg anbefaler å se på. Her finner du alt fra ferdige undervisningsopplegg til oppslagsverk rundt syntaks etc.

Ressurs	Kilde	Kommentar
Kidsa koder	https://oppgaver.kidsakoder.no/python	Her vil dere finne mange programmeringsopplegg og oppgaver hentet fra hele verden. Alle oppgavene er oversatt på norsk, og alle oppgavene inneholder også en lærerveiledning som fungerer som god hjelp i planleggingsfasen.
Knightlab - Pythonprosjekter	https://knightlab.northwestern.edu/2014/06/05/five-mini-programming-projects-for-the-python-beginner/	Her finner du en liste over mulige Python-prosjekter du kan gjennomføre i klasserommet. Merk at disse prosjektene ofte krever en del timer for å gjennomføre, men de kan være nyttige for å hente litt inspirasjon innledningsvis.
W3Schools - Syntax	https://www.w3schools.com/python/?ref=hackernoon.com	Husker du ikke hvordan du lagde en for-løkke eller enn if-setning? Frykt ikke. På W3Schools finner du en oversikt over Python-syntaks.
Espen Clausen sin blogg	https://espenec.wordpress.com/	Espen har lagt ut mange undervisningsopplegg som passer fint å bruke på

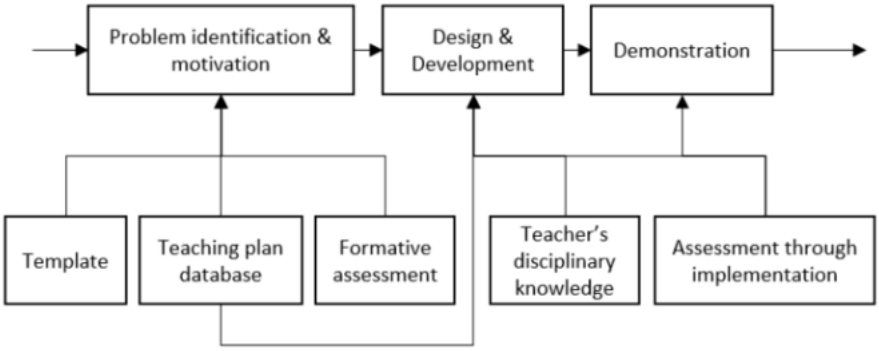
		ungdomsskolen. Oppleggene han har listet opp på sin side, benytter seg ofte av programmeringsspråket Scratch.
Lokus – Programmeringskurs	https://aunivers.lokus.no/fagpakker/realfag/programmering	Her finner dere mange kurs som gir en grunnleggende innføring i flere programmeringsprinsipper. Kursene er veldig visuelle og enkle å følge
Ressurser som er nyttige for naturfagslærere		
Eksempel på en lærerplan for en Naturfagsklasse på Skauen skole	https://skauenskole.no/wp-content/uploads/sites/6/2020/08/Naturfag-Årsplan-8.-klasse-2020_21-1.pdf	Her er der satt opp diverse undervisningsopplegg for en Naturfagsklasse på en ungdomsskole i Norge.
Måle luftfuktigheten i jord ved bruk av Micro-bit	https://makecode.microbit.org/projects/soil-moisture	Et opplegg som bruker Microbit med litt ekstrautstyr for å måle luftfuktigheten i jord. Et opplegg som absolutt passer fint til kompetansemålene som er satt opp i Naturfag på ungdomsskolen
Hvordan lage kompass med Microbit	https://oppgaver.kidsakoder.no/microbit/pxt_kompass/README	Opplegg som lager et fungerende kompass ved å bruke MicroBit.

Planlegging av undervisningstime

Du skal i de neste to kursdagene i programmering, planlegge ditt eget undervisningsopplegg. Du skal ta utgangspunkt i et av fagene du underviser i og planlegge en time på **60** minutter. Undervisningsøkten skal ta utgangspunkt i et kompetansemål du har hentet fra Lk20 (**Fagfornyelsen**). Det er du selv som har ansvar for innholdet i undervisningsøkten. Jeg

anbefaler og se nøye over ressursene jeg har gitt ut ovenfor. Her finner dere mange gode opplegg det kan være nyttig å ta utgangspunkt i.

Under her finner dere en liste med spørsmål det er lurt å ha i bakhodet når dere planlegger. Disse spørsmålene er kun ment for å være veiledende. Har dere noen spørsmål? Spør meg da!



Spesialiseringsfase

Kursfasen vi nå beveger oss inn kalles for en realiseringsfase. I denne fasen starter kursdeltakerne (dere), å utvikle deres egne undervisningsopplegg. I denne fasen er målet at dere skal lage en kobling mellom dette kurset, og deres egne pedagogiske praksis. Denne fasen vil dra dere igjennom flere steg som skal hjelpe dere på vei i planleggingen:

Del 1 – Problem og motivasjon

I denne delen skal dere finne et problem/tema i et av deres fag som dere ønsker å planlegge en undervisningsøkt til. Her er det viktig å tenke på følgende:

1. Hvordan kan dette problemet løses med programmering?
2. På hvilken måte kan programmering belyse en problemløsningsstrategi ved dette problemet?
3. Hvorfor velger jeg å bruke programmering på dette temaet vs. andre tema?
4. Hva kan mulige undervisningsmål være for denne økten?
5. Hvilke andre nettressurser kan hjelpe meg i planleggingsarbeidet?

Del 2 – Design og utvikling

1) **Undervisningsmål**
    a. I hvilket fag gjennomføres denne undervisningsøkten i?
    b. Hvilke kompetansemål fra lærerplanen er relevante for undervisningsøkten?
    c. Hva skal elevene lære i denne timen? Formuler minst et læringsmål.
2) **Sammenheng**
    a. Hvordan kan programmering brukes som et verktøy i denne undervisningsøkten?
3) **Programmeringselementer**
    a. Hvilke programmeringsprinsipper vil denne timen ta for seg?
        i. Hvordan skal du introdusere og forklare disse begrepene?
4) **Elevforutsetninger og andre rammefaktorer**
    a. Beskriv klassemiljøet og de forutsetningene du er sentrale for læringsprosessen
    b. Hvilke programmeringsspråk vil bli benyttet i undervisningsøkten?
        i. Hvorfor har du valgt dette språket over andre?
5) **Vurdering**
    a. Hvordan kan du finne ut hva elevene har lært i løpet av timen?
        i. Vis til læringsmålene du har satt opp ovenfor
    b. Formativ/Summativ vurdering
6) **Utvikling av egen praksis - refleksjon**
    a. Hvilke utfordringer tror du at du kommer til å møte på underveis i undervisningsøkten?
    b. Hvilke muligheter gir bruk av programmering som verktøy deg i denne undervisningsøkten?

Tidsramme	Hva skal læreren gjøre?	Hva skal eleven gjøre?	Hvordan?	Hvordan?

-

Appendix F

Example of a teaching plan designed by a course participants

Undervisningsøkt - Programmering

1. Tema på undervisningsopplegg i programmering

- Hva er tema for denne undervisningsøkten?

  Repetisjon av programmet scratch, og hvordan det kan brukes. Elevene har tidligere jobbet med å lage spill hvor en av oppgavene var at de skulle ta for seg sannsynlighet i spillet. Mange lagde spill som ikke hadde sannsynlighet i seg, så vil derfor ha fokus på sannsynlighet i opplegget her.
- Hvilke programmering prinsipper vil undervisningsøkten ta for seg?

  Variabler, løkker, betingelse/vilkår

2. Fag

- Hvilke fag gjennomføres denne undervisningsøkten i?

  Matematikk

3. Læreplanmål

a. Gjengi de aktuelle kompetansemålene for denne timen

   Elevene skal kunne simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe, ved å bruke programmering.

b. Finner du noen mål fra den overordnede delen av lærerplanen som er aktuell for denne timen?

   1.4: Skaperglede, engasjement og utforrskertrang:

   - *Barn og unge er nysgjerrig og ønsker å skape.*
   - *Elevene skal lære og utvikle seg gjennom sansning og tenkning, estetiske uttrykksformer og praktiske aktiviteter*

4. Læringsmål

- Lag 2 læringsmål som henger sammen med kompetansemålene ovenfor.

  Elevene skal programmere terningkast som belyser "det store talls lov", ved at det stimulerer mange nok terningkast til at fordelingen blir mest mulig lik sannsynligheten

Elevene skal bruke ulike blokker som variabler, løkker og betingelser

5. Forkunnskaper

- Gi en beskrivelse på elevenes forkunnskaper innenfor tema.

  Elevene har hatt en lengre periode med sannsynlighet og programmering. De skal ha kunnskap om bruken av scratch, og hva de ulike kommandoene gjøre. I tillegg har de jobbet med sannsynlighet i en lengre periode som gjør at de skal ha god kunnskap om hva sannsynlighet er.

- Gi en beskrivelse av elevenes forkunnskaper innenfor programmering.

  Elevene har jobbet med scratch i en lenger periode, både ved å lage ulike figurer, gi kommandoer til figurer og lage egne spill fra bunn av. Mange elever opplevde programmering som utfordrende, og trengte mye hjelp og støtte for å få til et spill som kunne la seg gjennomføre.

Undervisningsopplegg

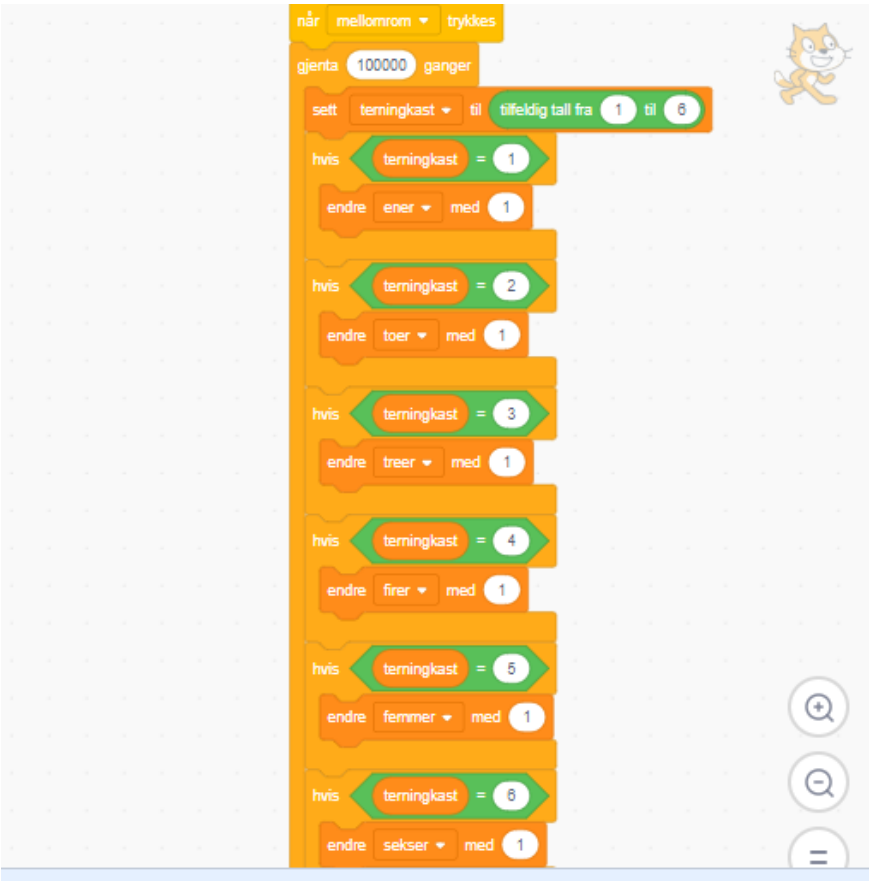
Elevene har jobbet med sannsynlighet og programmering over lengre tid, og har blant annet jobbet med “Det store talls lov”. Det vil derfor være interessant for elevene å se hvordan resultatet av 100 000 terningkast vil bli, kontra 20 terningkast.

Vil det være slik at fordelingen av de ulike tallene vil bli mer jevn når terningen kastes 100 000 ganger?

Elevene får følgende oppgave:

> "Det store talls lov" sier at jo flere tilfeller man har av en hendelse, jo nærmere sannsynligheten vil man komme. I dag skal dere bruke en terning for å utforske dette. I et terningkast vil dette si at desto flere ganger du kaster en terning, jo nærmere vil resultatet være sannsynligheten for de ulike tilfellene.
>
> 1. Du skal kaste en terning 30 ganger. Noter i boka di hvor mange enere, toere, treere, firere, femmere og seksere dere får. Ut i fra dette forsøket, finn sannsynligheten for å få de ulike tallene. Stemmer dette?
> 2. Du skal bruke scratch og lage et program som simulerer hvor mange enere, toere, treere, firere, femmere og seksere du får når en terning kastes 100 000 ganger. Hva tenker du om resultatet?

Resultat av programmering:

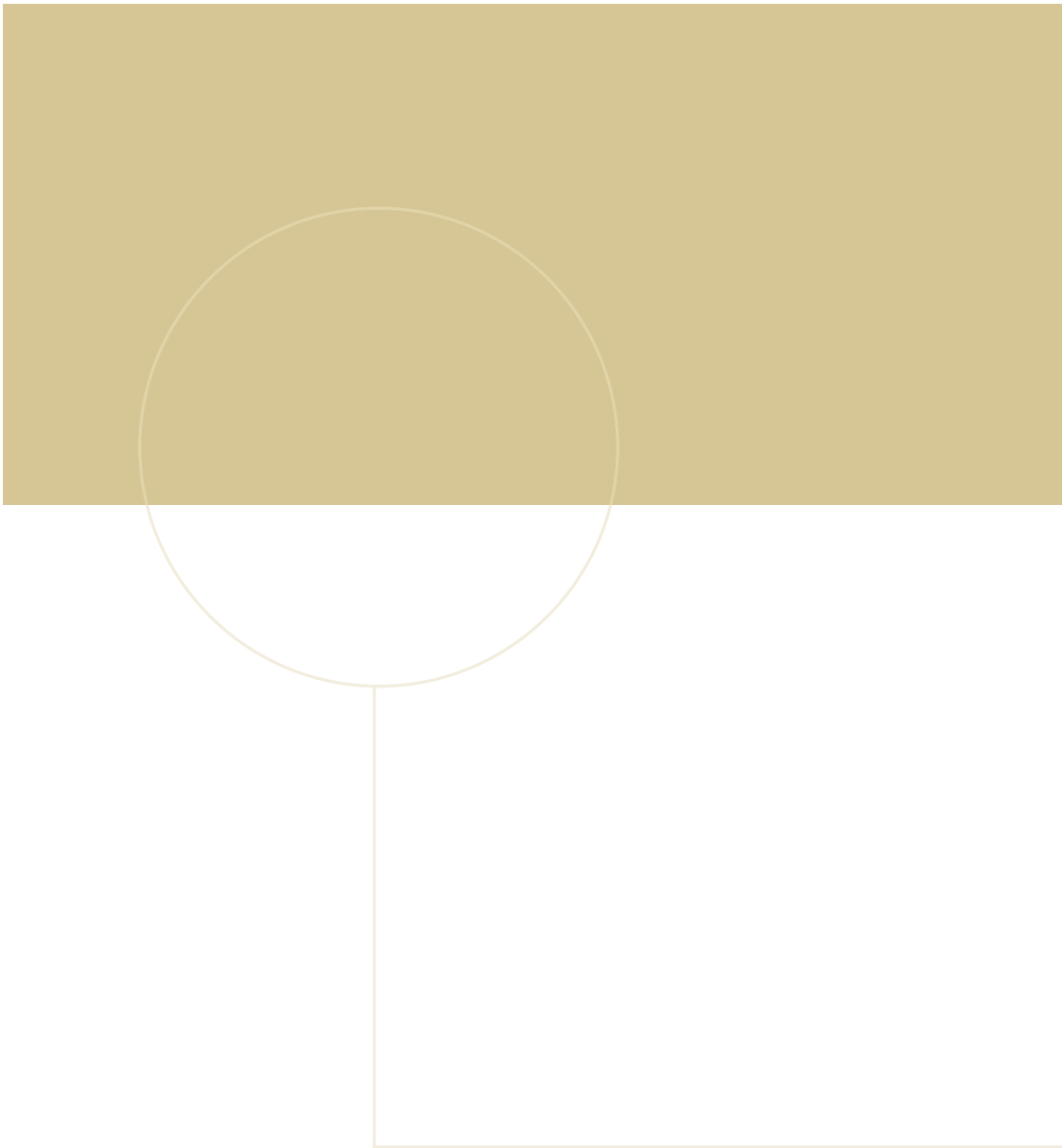


Appendix G

List over used, premade codes for the analysis phase

Transcription codes

…	Pause up to three minutes.
..	The respondent is hesitant
_	Interruption
[…]	Omitted statement
Cursive	Pressure



NTNU

Norwegian University of
Science and Technology