published as: Thomas Staubitz, Ralf Teusner, Christoph Meinel: MOOCs in Secondary Education - Experiments and Observations from German Classrooms, In Proceedings of 10th Annual Educon Conference (Educon2019), 9-11 April, 2019, Dubai, VAE

MOOCs in Secondary Education - Experiments and Observations from German Classrooms

Thomas Staubitz, Ralf Teusner, Christoph Meinel
Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
{thomas.staubitz, ralf.teusner, christoph.meinel}@hpi.de

***Abstract*—Computer science education in German schools is often less than optimal. It is only mandatory in a few of the federal states and there is a lack of qualified teachers. As a MOOC (Massive Open Online Course) provider with a German background, we developed the idea to implement a MOOC addressing pupils in secondary schools to fill this gap. The course targeted high school pupils and enabled them to learn the Python programming language. In 2014, we successfully conducted the first iteration of this MOOC with more than 7000 participants. However, the share of pupils in the course was not quite satisfactory. So we conducted several workshops with teachers to find out why they had not used the course to the extent that we had imagined. The paper at hand explores and discusses the steps we have taken in the following years as a result of these workshops.**

Keywords—MOOC, Secondary Education, School, Teamwork, K-12, Programming course, Java, Python.

I. Introduction

In the paper at hand we attempt to answer the question if MOOCs are a proper instrument to support computer science teachers in schools and what needs to be done to optimize the courses for this purpose. We ran a row of experiments at different scale over the last 4 years[1]. In 2014 we ran our first *regular*[2] MOOC, particularly targeting pupils. In 2015 we ran this course in a second iteration. We then conducted a couple of workshops with teachers to figure out if and how they used these courses and what they would need to better integrate them in their classroom education. One of the major findings in these workshops has been that the runtime of the courses is too short and the weekly workload is too high to make use of them in school classes. We found two teachers that were willing to test a *stretched*[3] version of the Python MOOC in their computer science class in 2016. As this experiment was quite successful, we scaled it and ran the stretched course again in 2017, this time with about 1000 pupils in schools all over Germany. In 2018, we offered the same stretched Python course again and we additionally offered a second stretched course: *Introduction to Object-Oriented Programming in Java*, which had been conducted a year before as a *regular* course on our platform.

First of all, we wanted to learn more about the pupils and the teachers: their age, background, previous knowledge, etc. We, therefore, conducted a series of surveys, one of them targeted only the teachers, the others targeted the course participants. The teacher survey was only conducted once during the first large-scale iteration of the stretched Python course. The in-course surveys have been conducted in similar form in all of the stretched courses.

To answer the general question if and how MOOCs can be employed in schools and classes, we attempt to answer the following subquestions:

- How can we adjust the courses to better fit the needs of the teachers?
- How do the teachers employ these courses in their classes?
- How do the teachers use our grading mechanisms, and how do the points the pupils receive in the course affect their actual school grade?
- In which classes and age groups do the teachers use the courses in school?
- What is the motivation of teachers and pupils to join the course?
- Is the course appropriate for boys and girls alike?

Our main focus in this paper is to find out how the teachers employ these courses in their classrooms or in extracurricular school settings. This is not because we do not consider the learners point of view to be important. The motivation is rather that we first need to satisfy the teachers' needs so that they are willing to/can employ these courses in their classrooms. Only when we have accomplished that we can have a deeper look in the special needs of pupils as the actual target group of these courses.

Additionally, we examined the applicability of a particular feature of the Java course in the school context. This course includes a task that has to be solved by teams. The results of the teamwork are peer assessed by the other teams. In this context the following questions are in our focus:

- Is (virtual) teamwork (across schools) possible in a classroom setting?
- Are there differences between (virtual) distributed teams and (face-to-face) local teams?
- Is peer assessment a suitable tool to grade such type of work in the classroom context?

The remainder of the paper is structured as follows: In Section II we briefly present some related work and literature.

[1] 2014-2018
[2] With *regular* we mean courses that are visible in the course list of our platform and have a runtime of about 4 weeks.
[3] Same content, more time ⇒ less weekly workload

Section III provides some basic information about the examined courses: number of participants, subjects, length, etc. In Section IV we examine and discuss the data that we collected in several surveys amongst teachers and course participants. This data contains both quantitative and qualitative elements. Finally, in Section V and Section VI, we will outline our next steps and conclude our findings.

II. Related Work

Experiments with MOOCs for pupils have been conducted in several countries. In Finland, Kurhila and Vivahainen [8] have reported about a MOOC that corresponds to the CS1 course of the University of Helsinki. Several schools in Finland have included the course as an optional offer in their curriculum and provided formal school credits for completion. They noticed that several high school students throughout Finland were keen on learning how to program in such a way. Pupils who successfully complete the MOOC are offered an interview with the option to study at the University of Helsinki. Thus, the MOOC serves as a prolonged entrance exam for potential university students. Aksela, Wu, and Halonen [1] have examined a MOOC on Sustainable Energy targeting adolescents. In Austria, Khalil and Ebner [6] examined a course "Mechanics in everyday life" on high school level. They made use of their platform's learning analytics tools to determine how the participants interacted with the platform. Their results have been generally positive and they recommend to develop further MOOCs for the classroom. EdX offers a full program[4] to prepare high school students for college and university.

In our programming courses we set a strong focus on learning-by-doing. For this purpose, we have developed a tool that allows students to program in the browser without having to install any additional software. The tool also allows us to automatically grade the work of the students by running tests on the students' code to check wether their solutions fulfil the given requirements. We could mention a long list of research on auto-graders as related work here. This is not the focus of the paper at hand, however. Please refer to e.g. [2] or [12] to explore the topic of auto-graders in more depth. Furthermore, we could list many programming MOOCs that have been conducted before on many of the major MOOC platforms[5]. Again, programming MOOCs are not the focus of this paper. As we are focusing mainly on the usage of MOOCs in schools, we have restricted ourselves to the few listed examples dealing with the usage of MOOCs in schools. This list makes no claim of being exhaustive.

A major element of our Python courses builds on the work of Papert and Solomon [11] by including Turtle graphics as a playful element into the offered exercises.

While hands-on programming is an important element in our Java courses as well, we set a second focus there on social learning. Discussions among the participants to deepen their understanding are strongly encouraged. Hereby, we follow a social constructivist approach basically following Vygotsky's theory of proximal development [17].

III. The School MOOCs

A. Prehistory

In 2014 and 2015, we conducted the first programming MOOCs on our platform that have been explicitly addressing children, pupils, and adolescents with two iterations of the course *Playfully learning to program*[6][7]. Back then these courses attracted about 7-8,000 participants, many of them were members of the actual target group (9-17 year olds), but the the majority of the participants recruited themselves from our typical user spectrum (20-80 year olds). The courses have been conducted by a professor of computer science, who is a main contributor to the Python language [10].

The courses consist of videos and quizzes, same as our other MOOCs, but additionally come with 3 hands-on programming exercises per video that had to be solved with our then new online programming tool (see [13] for details) that allows to run, test, and grade code in the browser without requiring any installations on the participant's computer.

The exercises are designed according to the first three levels of Bloom's taxonomy [4]. The first exercise, directly following a video is on the *knowledge level*. Basically, the participants just have to follow the video step by step to solve the exercise. Exercises 2 and 3 then cover the *comprehension* and *application level* of the taxonomy[8].

Some of the exercises adapt Seymour Papert's turtles, which have already been used decades ago to teach children logic, mathematical ideas, and programming [11]. Our main focus with these exercises is not to test what the pupils' have learned. The idea is to encourage them not only to watch and listen but to get their hands dirty and start to actually write code. Therefore, there is neither a time limit nor are the attempts limited in all coding exercises in these courses.

As we had designed these courses with schools and teachers in mind, one of our main questions was: why didn't more teachers join them together with their classes? To answer this question we conducted several workshops with school teachers and principals in different settings and conducted some informal interviews with the few teachers who did use the courses in class [5]. One of the major results of these workshops was rather simple to solve: many of the teachers stated that the format doesn't fit the reality in school. The courses are too short and too dense. With two teachers that had participated in one of the workshops, we, therefore, conducted a prototypical pilot in 2016.

B. Prototype

While the original MOOCs ran for about 4 weeks, the idea that had evolved in the workshops was to stretch the courses to a duration of about half a school-year, while keeping the exact same content, to significantly reduce the required time

[4]https://www.edx.org/high-school

[5]e.g. An Introduction to Interactive Programming in Python by Rice University on Coursera, Programming for Everybody (Getting Started with Python) by the University of Michigan also on Coursera, Introduction to Java Programming by UC3M on EdX, and many more

[6]https://open.hpi.de/courses/pythonjunior2014

[7]https://open.hpi.de/courses/pythonjunior2015

[8]In Anderson and Krathwol's [3] revision of Bloom's taxonomy, these levels correspond to Remembering, Understanding, and Applying

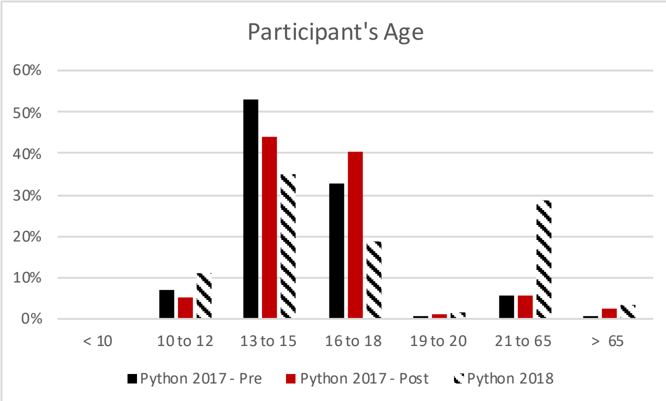

Fig. 1. The age distribution of the participants in the Python MINT-EC 2017 pre (n=562) and post (n=209) course survey and the Python MINT-EC 2018 survey (n=146)

per week. Two of the teachers, who participated in one of the workshops, asked us if we were interested to conduct a stretched version of the Python course with them and their classes. A few months later, in February 2016, the course was up and running. Even in this small setting, preparation required some time and effort, mostly on the side of the teachers and their school.

The pupils' (and/or their parents') consent had to be obtained, forms had to be signed, the course had to be scheduled, some additional technical equipment had to be purchased [5]. 23 pupils had enrolled, so it was rather a SPOC (Small Private Online Course) than a MOOC. We had a fantastic show-rate of 100% and 17 of the 23 pupils (74%) achieved a Record of Achievement (RoA)[9]. The experiment was quite successful: the teachers asked for a new iteration in 2017. This time we decided to go for scale.

C. Mint-EC MOOCs

The MINT-EC[10] is the national excellence cluster for schools with a strong profile in mathematics, informatics, natural sciences and technology in Germany. As the workshop that incubated the prototype was held at their annual meeting of school principals and we mainly used their network to market the MOOC, we called the following courses the *MINT-EC MOOCS* to distinguish them from our regular courses. Nevertheless, they were open for every school to participate.

1) Python MINT-EC 2017: Despite the decision to go for more participants, we kept the course hidden from the general public. The main reason was not that we did not want to have them in there. It was more that we felt uneasy to add these courses with their very special time frame amongst our regular courses with a two to six week duration. Nevertheless, with only little advertisement, the course achieved about 1000 enrollments. In terms of content, the course was identical to the previous iteration of the regular MOOC in 2015 and the prototype in 2016. 431 of the participants who had enrolled before course middle received a Record of Achievement (RoA), 581 at least got a Confirmation of Participation (CoP)[11]. Again, the show-rate[12] was fantastic (96%). All this considered, the resulting completion rate is at 42% (RoA)[13] or 57% (CoP)[14]. When we write about completion rates in the following, we will use the form XX(YY)%, whereas XX is the completion rate in terms of RoA and YY is the completion rate in terms of CoP.

2) Python MINT-EC 2018: In 2018 we ran the course again under exactly the same settings. The only difference was that the course was a little more exposed to the public this year.

It still was hidden on our main platform, but we had openly linked to it from our white label platform[15], where we are currently establishing a channel[16] with more school-oriented material[17]. In this year we had 312 participants. The show-rate was 91%, 87 participants received a RoA, 131 participants received a CoP, resulting in a completion rate of 30(46)%.

Figure 1 shows the age distribution in the two Python MINT-EC courses. In the 2017 iteration, we asked some of the questions in both: the pre-course survey and the post-course survey. We will discuss the differences in the numbers of the pre- and post-courses survey in Section IV, for now we just want to point to the difference between the 2017 and 2018 iterations of the course. While in 2017 the major part of the participants was younger than 18 years, we find a large group of 21-65 year olds in 2018. In 2017, the few participants that were older than 18, have almost exclusively been teachers. In 2018, the course attracted a few participants of some of our other (*regular*) courses, as it was openly listed on the new channel on our white label platform. We would particularly like to mention one participant, who recently had retired from work and did a great job in answering all questions in the discussion forum before we even had a chance to read them. We will discuss the usage of the discussion forum in Section IV in more detail. The lower number[18] of participants was partly due to a too late announcement of the course, mostly, however, due to the fact that this time we ran a second course in parallel: *Introduction to Object-Oriented-Programming in Java*.

3) Java MINT-EC 2018: Some teachers had asked for Java courses for a while, as they often fit better in their curriculum. So we applied the same stretching to our Java course that ran in 2017 as a regular MOOC with close to 10,000 participants.

[9]The Record of Achievement is our higher ranking certificate, which requires active contribution of the participant and contains the points the participant has reached

[10]https://www.mint-ec.de/

[11]The CoP is the lower ranking certificate that just states that the participant has seen half of the course content.

[12]Participants who have at least seen one of the course items while the course was running.

[13]Generally we calculate the completion rate as the *number of certificates / (number of participants at course middle - no-shows at course middle)*. We consider this calculation to be more realistic than the simplistic *certificates / participants* as it only takes those into account that had a realistic chance to earn a certificate.

[14]Normally, we do not work with the number of CoPs to calculate the success rate. In this particular case it actually could be considered as some teachers considered the CoP to be sufficient to pass the class.

[15]https://mooc.house

[16]Channels on our platform are basically groups of thematically related courses.

[17]https://mooc.house/channels/schul-cloud

[18]Compared to the 2017 iteration.

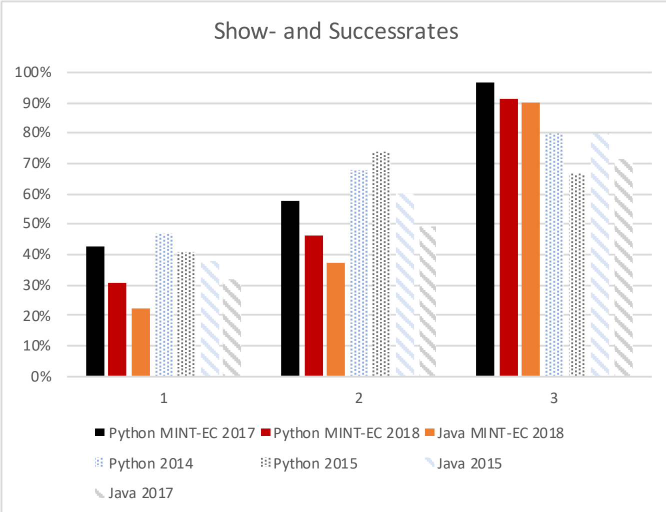

Fig. 2. Comparison of success- and show-rates: 1-Records of Achievement — 2-Confirmations of Participation — 3-Show-rate

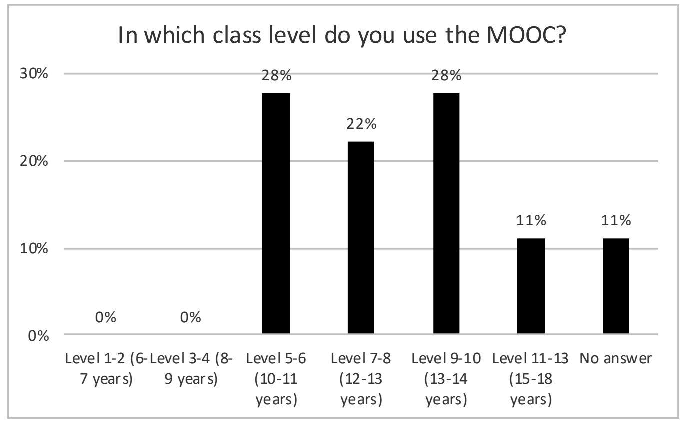

Fig. 3. Most of the teachers participating in the survey used the MOOC with ten to fourteen year-olds in class levels 5 - 10 in high school. (n=18)

The 2017 version of the Java course had been developed with some of our on-campus students in a project based learning (PBL) setting during an on-site seminar[19]. A previous version of this course had been developed under very similar circumstances two years earlier with a different group of students[20]. The Java MINT-EC course in total attracted around 500 participants. From the 413 that already had registered at course middle, about 90% actually started the course by visiting at least one course item. 83 participants received a RoA, 139 a CoP, resulting in a completion rate of 22(37)%. This course also contained a task that had to be solved and submitted by teams. We will discuss this also in more detail in Section IV. Figure 2 shows the success- and show-rates in comparison and also compares these numbers of the MINT-EC courses to the numbers in the corresponding regular courses. The term *success-rate* has to be handled with care, however, as some teachers only embedded portions of the course in their class and enriched it with material and activities of their own as the following example shows.

One of the teachers asked her pupils to create some videos on their own to explain the solutions of some exercises to the others or to explain a certain Java feature. At the end of the class she organized a Java party. The students were asked to bring songs about Java, find out why the language is called Java, write a Story using Java terminology, create a ranking of programming languages according to their potential salary chances, find out how to find a job as a programmer, find and present other resources to learn programming, etc. We interviewed her and learned about what can go wrong. They started a bit late as she learned too late about the course. Then all the computers in the school were to be replaced with newer models. It took 8 weeks from the day the old computers had been removed to the day the new computers had arrived and all pupils were able to log-in again. In the end only one week was left before summer break[21]. So the teacher had to improvise. The pupils were asked to watch the videos at home, they tried to work with mobile phones, etc. As they had to watch the videos at home their workload increased. Although many complained about this in a survey conducted by the teacher at the end of the course, the overall experience was perceived positive by most pupils. In this survey, many pupils also expressed the wish for more collaboration and discussions within the class. Despite the less than optimal circumstances, 10 of 16 pupils in the class completed the MOOC with a RoA and most of them completed the class by working on the alternative tasks offered by the teacher.

[19]https://open.hpi.de/courses/javaeinstieg2017

[20]https://open.hpi.de/courses/javaeinstieg2015

IV. Evaluation

A. *Teacher Survey in Python MINT-EC 2017*

In the first scaled-up iteration of the Python MINT-EC course, we conducted a survey among the teachers who attended the course with their classes. 18 teachers responded our call, representing about 200 participants[22]. All of them were high school teachers, most of their classes have been in levels 5-6, 7-8, and 9-10. The pupils were between ten and fourteen years old (Figure 3). The size of the classes ranged from less than 10 to 30 pupils, in a few cases even more than 30 pupils (Figure 4). The course has been employed in regular classes as well as in in-school and off-school extracurricular activities (Figure 5). 25% of the teachers in the survey used exclusively the course's/platform's grading features to grade their pupils, another 20% added tasks and exercises of their own, and about a third didn't grade the course participation at all.

Of those who graded the course, three teachers added these grades to their pupils' half year report, another two added them as a bonus. Some just used the points to collect points for the

[21]In Germany school summer break starts at different months depending on the federal state. The earliest schools are on vacation in June/July the latest in August/September, which makes it somewhat difficult to schedule countrywide programs.

[22]This number is calculated as the amount of teachers times the size of their classes.

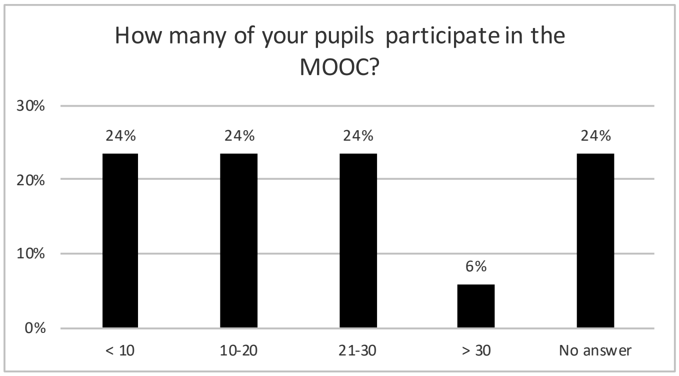

Fig. 4. The size of the classes ranged from less than ten to more than 30 pupils. 65% of the teachers accompanied the class alone, 11%(2) just let them work on the task alone, two more worked in teams of two (n=18).

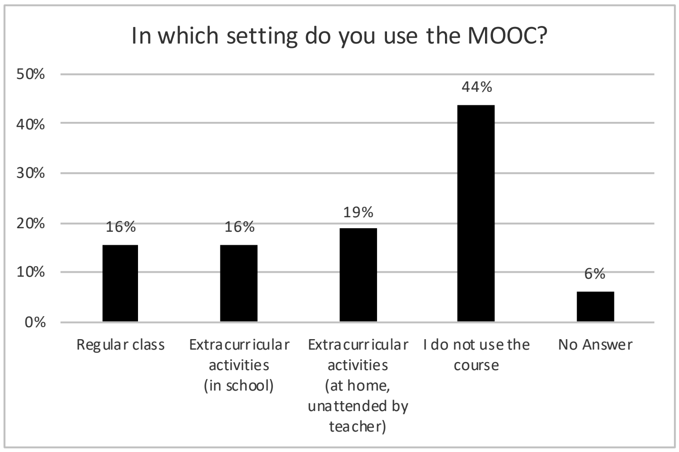

Fig. 5. The course has been employed in regular classes as well as in in-school and off-school extracurricular activities (n=32).

MINT-EC-certificate[23].

Most of the teachers in the survey have been teaching this class alone, one teacher stated that s/he teamed up with another teacher, one teacher just passed the link to the course to selected students who s/he thought might be interested. Most of the teachers originated from the German federal state of Northrhine-Westphalia, which doesn't offer regular mandatory computer science classes in schools, but there have also been teachers from schools in federal states that have basic and advanced computer science courses in their regular curriculum. The teachers liked about the course that it...[24]

- "offers a good possibility to differentiate and was integrated without problems in their class."
- "has a straight forward structure that was easy to follow for the pupils."
- "ran in a longer time frame, taking away the time pressure so we had time to discuss the contents."

[23]The MINT-EC certificate is awarded to pupils of MINT-EC member schools together with their university-entrance diploma (A levels). Throughout their time at school, they can collect points for this certificate for extra efforts in the MINT area

[24]The original German statements have been translated by the authors

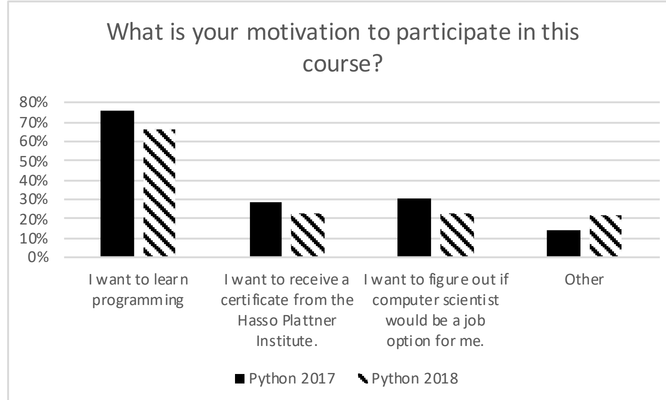

Fig. 6. The majority of the participants had joined the course with the declared goal to learn programming. The figure shows the results for the same survey in the 2017 (n=562) and the 2018 (n=146) iterations of the Python MINT-EC course.

- "is individually applicable, doesn't require any preparation, provides the pupils with tools of self-control and allows me to offer interesting projects for completely different pupils."
- "offers certificates."
- "offers quizzes and the hands-on programming exercises in CodeOcean. The pupils have a lot of time to acquire the necessary skills and knowledge on their own, and can hand in whenever they are ready."

One teacher forwarded a statement of a mother whose two twelve and fourteen year old children participated in the course:

> Both of my kids said the same. They liked the course, particularly the part with the turtle graphics. They said that the quizzes are too easy. The exercises sometimes are pretty hard to solve by just watching the videos but they actually liked that, so that they had to investigate further and solve things on their own.

B. *User Surveys in Python MINT-EC 2017*

In the same course we also asked the participants about their background, their motivation and their experience. In the pre-course survey, which was offered right at the course start, 562 participants responded[25]. In the post-course survey, 209 participants responded[26]. 75% of the participants in the pre-course survey stated that their motivation to participate in the course is to learn the Python programming language, about 30% were mostly interested in receiving a certificate, another 30% stated that their main interest was to figure out if becoming a computer scientist would be an interesting option for a future career[27]. Figure 6 shows that the answers in the 2017 and the 2018 iterations of the Python MINT-EC course were almost identical.

[25]This represents about 50% of the course participants

[26]This approximately represents a similar percentage of those who were still following the course at that point

[27]Multiple answers were possible

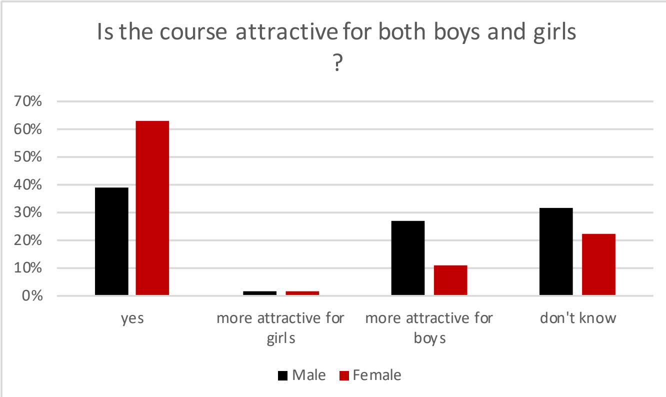

Fig. 7. Female participants perceived the course better suitable for a female audience than male students. (n=209)

Fig. 8. The difficulty of the given tasks and exercises were perceived as just right by the participants of the Python 2017 post course survey. (n=209)

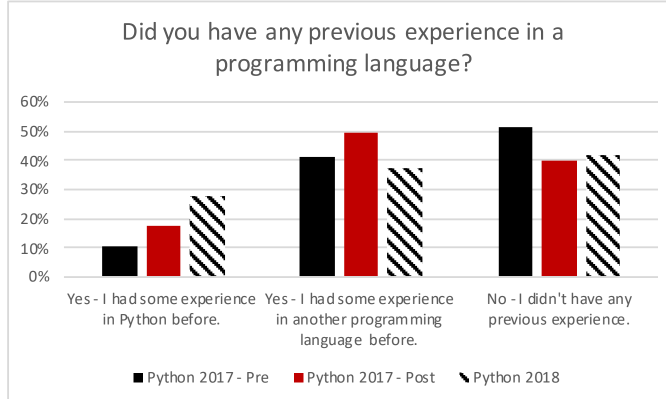

Fig. 9. The previous programming experience of the participants. Python 2017 pre (n=562) and post course survey (n=209) and Python 2018 survey (n=146)

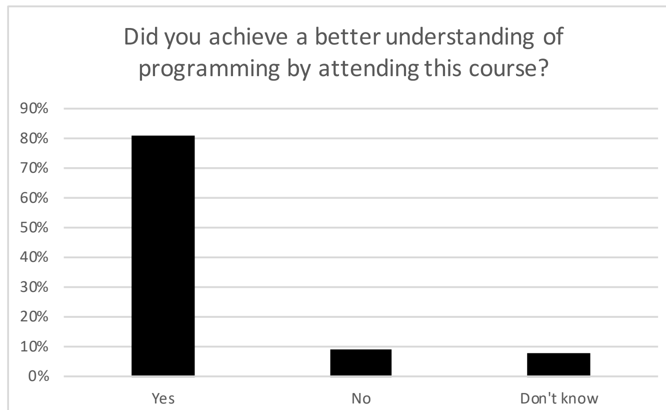

Fig. 10. The vast majority of the participants stated that they achieved a better understanding of programming. Python 2017 post course survey (n=209)

In terms of gender we had about 70% male and 30% female students. We asked the pupils if they see the course equally fit for boys and girls. Interestingly, only 40% of the male participants saw the material fit for both genders, while more than 60% of the female participants did. Close to 30% of the male participants stated that the course is more attractive for boys while only 10% of the female participants did. Close to nobody found the course more attractive for girls (see Figure 7). More than 50% of the pupils found the difficulty of the exercises just right, about 20% found them very difficult, another 20% found them very easy (see Figure 8). About 70% of the participating pupils considered videos, self-tests and quizzes appropriate learning materials for adolescents. Only about 30% considered the discussion forum and a Python book as appropriate for adolescents.

We repeated some of the pre-course survey questions in the post-course survey. In terms of age distribution we observed a shift from younger (13-15 years) to older (16-18 years) pupils from the pre-course to the post-course survey. This indicates that rather the younger students dropped-out during the course (see Figure 1 in Section III). In a few cases, however, we had very young participants in the top ranks of the course. An 11-year old participant was among the Top-5 in Python MINT-EC 2017, a 12-year old was among the Top-20 in Java MINT-EC 2018. In terms of gender the pre- and the post-course surveys did not differ significantly. Figure 9 shows that also in terms of previous experience the numbers shifted slightly between pre- and post-course survey, indicating that those participants who had previous programming experience were more likely to complete the course than those who did not. Finally, Figure 10 shows that more than 80% of the course-end survey participants achieved a better understanding of programming by attending the course.

C. *User Surveys in Python MINT-EC 2018*

In this course we only conducted a pre-course survey. 146 (about 50%) of the course participants submitted the survey. It showed that in comparison to the previous iteration we had a group of older (see Figure 1 in Section III) and more experienced participants (see Figure 9). In terms of gender the distribution was identical to the year before.

D. *User Surveys in Java MINT-EC 2018*

213 (about 50%) of the course participants submitted the pre-course survey. More than 60% of them took the course in the context of a computer science class in school, another 4% took the course in the context of an extracurricular activity in school. The others took the course out of general interest. 30% of the participants stated that their main motivation was the

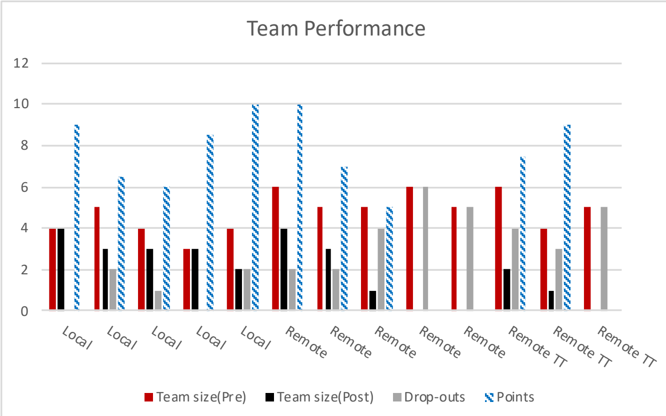

Fig. 11. Team size (Post): team members who succeeded in both parts of the peer assessment (submission of work and evaluation of the work of others). — Team size (Pre): number of participants that have been assigned to this team at the beginning. — Drop-outs: participants that have not received points for the team work task as they did not evaluate the work of others. — Points: the amount of points the team received for their work. — Local: all team members are from the same school and can meet face-to-face. Remote: team members are from different schools all over Germany and have to use the communication and collaboration features of the platform and other online tools. — TT: These are teams with *grown-ups* only, mostly teachers of the participating pupils.

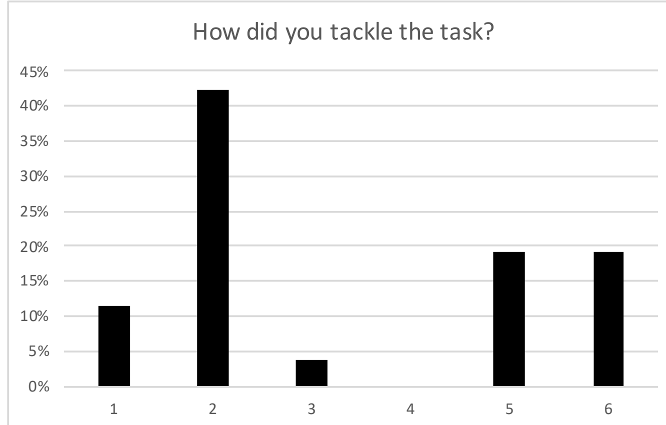

Fig. 12. 1. We worked on all tasks collaboratively. — 2. We discussed the assignment and then split the tasks (Glossary, diagram, code, ...) — 3. We discussed the assignment and then split the tasks (Bathroom, kitchen, library, ...) — 4. I mostly worked on the task alone. (Some of) The others just provided feedback. — 5. I solved the task completely on my own. — 6. Someone else did most of the work, I just contributed minor parts. (n=26)

Record of Achievement, for 50% learning the course content was more important than the record. The rest said that they just wanted to shop around or were only interested in certain course items or didn't answer. Our focus of interest in this course, however, was on a new experiment. As in the previous *regular* version of the course, we included a task that had to be solved by teams. The teams had to model a given scenario in an UML-like diagram. The tasks focus was not so much on proper UML-notation but rather on the experience to work on a program in a top-down way: planning the program before writing code. There was no directive how the models had to be created. Drawing by hand was as good as using UML-editors. Next to the diagram, they had to hand in a glossary explaining the most important terms in their program, and a program scaffold, generated from the UML either manually or automated; given the diagram editor provided that functionality. After the pupils had handed in their work, they had to grade and review the work of two to four other teams. Furthermore, they had the option to grade the contribution, organizational and social skills of their team mates and the quality of the received reviews. Only those who succeeded in both parts of the peer assessment–submitting the solution and reviewing the solutions of the peers–received points. 62 participants had registered for the team task, 15 of them have been teachers or other grown-ups, another five of them have been adult pupils of a vocational school. 26 of the 62 have successfully finished both components of the peer assessment. 12 of the 36 *drop-outs* have been teachers, who only wanted to see what the students were supposed to do and how the process works. The team task survey also was answered by 26 students, 21 of them had successfully submitted the task, five of them had dropped-out of the team task. 21 of the survey participants were pupils, three teachers and two others. Four of the five drop-outs who answered the survey were pupils, one other. Up to here we have examined the data from a *single participant* point of view. We will now switch and examine the data from a team perspective. We had 13 teams in total.

Three of them failed completely, one of the failed was a teachers' team. Two ended *dysfunctional*[28], again one of them was a team of teachers.

We have developed a tool that supports us to match the teams by a set of criteria that is defined by the teaching team of a course [14] . In this case, two parameters were essential for the team matching: Given the participant had provided her/his school name when registering for the task, we created local teams. If not, the teams were formed based on the amount of time that the participants have specified they want to spend on the task.

In the end we had five local teams in three different schools and five remote teams. Additionally, we had three teams whose members all were teachers. All five local teams succeeded, two of the five remote teams succeeded as well, one remote team was *dysfunctional*, two dropped-out completely. See Figure 11 for an overview of team sizes before and after the task and the points the teams achieved for their work. The teacher teams basically were inactive except for a single teacher in one of the teams and one teacher and one of the older non-teacher participants[29] in another team. The number of teams in this experiment is too small to make decisive statements about remote vs. local teams.

In a similar setting examined by the authors [16], substantial differences between local and remote teams in terms of cohesion and success have not been observed. Kizilcec [7] also did not observe substantial differences between local

[28]We call teams with less than two participants dysfunctional. This doesn't mean they're not successful, they're just not a team.

[29]Generally, we try to achieve teams that are heterogeneous by age and gender. In this case, as several teachers had expressed the wish not to be teamed up with their pupils, and as we had no means to differentiate between teachers and pupils than the age, we sorted them by age. For the older non-teacher participants this resulted ending up in basically inactive teams. One of these participants, however, was able to talk one of the teachers into solving the task together.

and remote teams in terms of learning outcomes. The best result in this experiment, considering both, achieved points and team cohesion, has been achieved by a remote team. Next to remote/local there are other factors that influence the success of a team. Possibly, the local teams took the course in-class, while the remote teams rather took the course in a less formal setting. Therefore, the motivation of the local teams to succeed might have been higher as the pupils' grades were affected. Obviously, local teams have certain advantages as they need less technical support than the remote teams and many of the tasks to be done can be accomplished with less overhead. Another interesting aspect is that (except for the teachers' team) all teams that have not submitted anything, have been all-male teams, while all teams that have managed to submit something have been mixed teams. 21 of the 33 male participants (64%) dropped out of the team task, while only 4 of the 14 female participants (29%) dropped out of the team task. So, even when it looks as if local teams perform better in this example, we still think that–viewed from the right angle–it rather supports the theory that local and remote teams can be performing equally well. In the survey we asked how many team members actively contributed to the task[30]. Six participants stated that five members have been actively working on the task, one participant stated that six team members have been actively working on the task. As the survey data is anonymized, we do not know how any of those who made these statements have been in the same team[31]. In the course data, however, we only have teams with maximal four members successfully finishing (both steps of) the task. So either they wanted to cover inactive team members or more team members have actively contributed than those who have received the points, in other words some of those who actively contributed to the task have missed to do the evaluation part of the task. This might have happened deliberately (more than a third of the survey participants had stated that they are only interested in learning how to model and were not interested in the points) or accidentally (maybe they missed the deadline or did not realize that the evaluation part was mandatory as well). Another interesting question was **how** the teams tackled the task. We allowed both: a collaborative, discussion oriented solution or a cooperative, divide and conquer oriented solution. Three of the survey participants had chosen a collaborative approach[32], the majority, however, had chosen a cooperative approach and split the tasks (someone did the diagram, someone else the code scaffold, yet another one the glossary). Only one team split the tasks by program parts (One participant did Class 1, the next Class 2, etc.). Five participants claimed to have done the work completely on their own, another five admitted that they had only contributed minor parts while someone else did most of the work. Figure 12 shows the results for this question in percentages.

The next topic we've been interested in was to find out if grading by means of peer assessment, in general, is applicable in the school context. About 60% of the participants found peer assessment either "very suitable" or "well suitable", more than 20% found it "sort of ok". Only about 14% found it "not suitable" or "not so suitable".

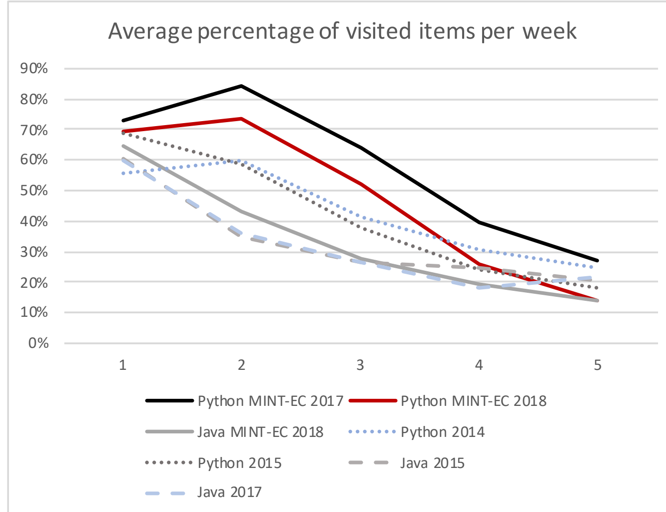

Fig. 13. Percentage of average visited items per course section (no-shows have been removed) Python MINT-EC *: [1: Welcome; 2-5: Week1-4] — Java 2015: [1,2,4,5: Programming 1-4; 3: Modeling] — Java 2017: [1,2,3,5: Programming 1-4; 4-Modeling] — Java MINT-EC 2018: [1,2,3,5-Programming 1-4; 4-Modeling]

We then asked if the participants perceived the grade that they received by their peers as just. 80% stated that the grade was just right, two participants stated that they received more, another two stated that they received less points than they deserved. The given grades were equally distributed from five to ten[33] points.

Finally, we wanted to know if the course in total and the team work task in particular enriched the class. The course in total was perceived as enriching by almost 60% of the participants, another 20% found it "so/so", 20% did not like it. The team task was perceived as less enriching (30% positive, 40% neutral, 30% negative).

As the number of teams was manageable, we tested if more support from the teaching team, such as announcements with instructions how to approach the task or about approaching deadlines, etc., were perceived helpful by the teams. 50% found these messages helpful and sufficient, another 20% found them helpful and would have liked more. 20% didn't find them helpful and the rest claimed that they never have received such messages. This result is encouraging enough to plan for a better platform support for team announcements, so that it will be possible to keep-up with this practice in large-scale courses.

E. Comparison to Regular MOOCs

To complete the image we finally compared several key indicators between the stretched School-MOOCs and the corresponding regular MOOCs. The data has been drawn from the platform's reporting system and represents all registered

[30] Some of the answers to this question have to be handled with care as they are inconsistent with the answer to other questions in the survey. Particularly, those who answered "I worked on the task alone" or "I did not submit" have answered differently in other questions, which offered the same answer options.

[31] But as six have stated that five were active in the team, at least two of the teams were affected.

[32] Again, due to the anonymized data we do not know if they have been in the same team or not

[33] The possible maximum

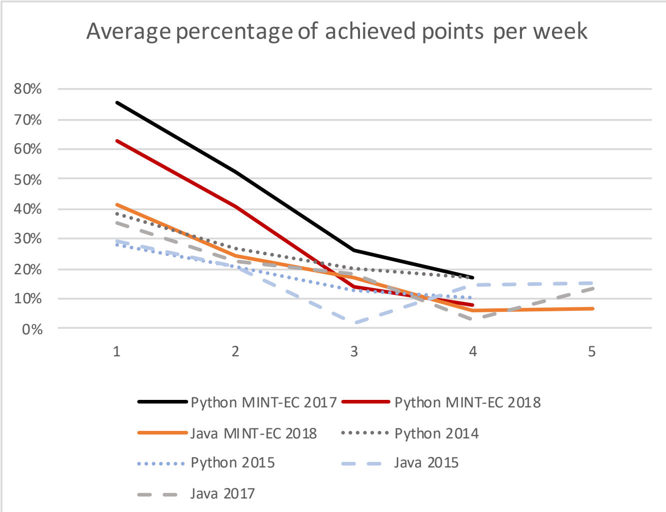

Fig. 14. Comparison of the average percentages of points achieved per course section (no-shows have been removed) Python MINT-EC *: [1-4: Week1-4] — Java 2015: [1,2,4,5: Programming 1-4; 3: Modeling] — Java 2017: [1,2,3,5: Programming 1-4; 4: Modeling] — Java MINT-EC 2018: [1,2,3,5: Programming 1-4; 4: Modeling

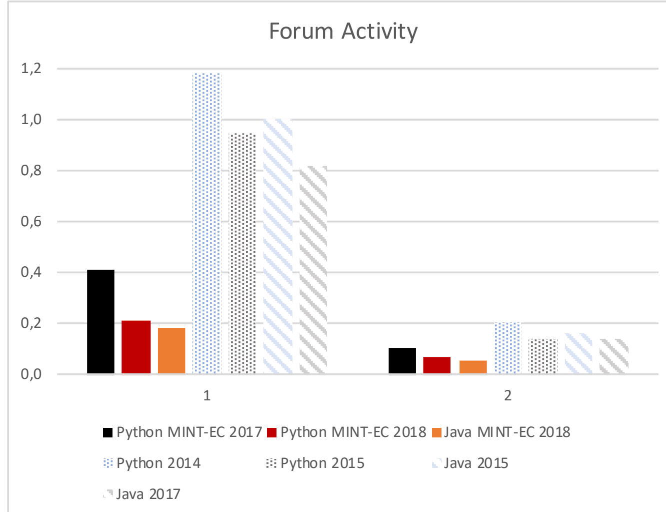

Fig. 15. Comparison of general discussion forum activity 1: average posts per user — 2: average threads per user

participants. Where appropriate, the no-shows have been removed from the evaluation to remove noise. In these cases, this will be indicated in the image caption.

The most common factor are the success- and show-rates. Figure 2 in Section III shows that the course Python MINT-EC 2018 had a 10% smaller completion rate than Python MINT-EC 2017. The completion rate of Java MINT-EC 2018 was even smaller. Let's first analyze the difference between the iterations of the Python MINT-EC course. The content has been absolutely identical. The show-rate in 2018 was a little lower but as we removed the no-shows from the equation already this doesn't matter. In comparison to the regular courses, the completion rate measured in RoAs Python MINT-EC 2017 was even a little higher than in Python 2014 and came close to Python 2015. Python 2015 features videos and exercises identical to the Python MINT-EC * MOOCs. Taking a more detailed look at the visited items and the achieved points per week, we can see that here as well we can find an about ten percent higher average through out the course. Figures 13 and 14 show that the curves are running almost in parallel for both visited items and achieved points. So far no big surprise, the pupils have visited in average 10% less items, therefore have achieved 10% less points and finally 10% less of them received a certificate. Seems to be straightforward and obvious. The question is why. The major differences between the two courses are to be found in the user structure. There might have been a difference in the way the teachers have employed the courses: in-class vs. extracurricular. We do not have data denying this hypothesis, but there is also no evidence that supports it. If we step back to Section III and have a look at Figure 1 we can see that in 2018 a rather large group of grown-up participants had joined the course. Many of them revealed in the survey that they are teachers and want to figure out if they might use the course in a future iteration. So we examined if this group might distort the overall results. We, therefore, removed this group from the overall dataset and the success rate measured in Records of Achievements increased from 30% to 40% and is thereby identical to the previous iteration. Riddle solved. Figure 13 shows that all of the courses start with a very similar percentage of visited items, somewhere between 60 and 70%. In the four Python MOOCs the curve is actually a little distorted, as the first section is not a real week but just an introductory section running in parallel to the first week. Therefore, the curves of the Python courses might as well be shifted left, resulting in almost identical curves for all courses. Figure 14 shows that the Python MINT-EC courses start-off with a significant higher percentage of achieved points. About week three, however, this levels out with the other courses. We assume that during the first two weeks there is more teacher support (or pressure). We've also had reports from teachers that only asked for a *Confirmation of Participation* as a proof of completion. As the CoP only requires that the participant has seen more than 50% of the course content, the pupils who only joined the course because they were *forced* by their teachers, often turned inactive when they had reached that percentage after Week 2. Finally, we take a look at the activity of the participants in the discussion forums. Figure 15 shows that there are significantly less threads and posts in the MINT-EC courses than in the regular courses. This is easily explained as these pupils generally participate in groups and sometimes have the support of their teachers, therefore, the discussions in the forum are much less essential to them. Nevertheless, some discussions pop-up from time to time, leaving the forum completely unattended should therefore be avoided. In our regular courses, monitoring the forum is the teaching teams' main task while the course is running, in the first iteration of the MINT-EC edition, we paid a student to monitor and trigger discussions in the forum. In the second edition we just checked-in once or twice a week to prevent the worst. Very few questions have been asked and most of them have been answered very quickly by one other participant, a recent retiree, before we even saw them.

Löwis et al. [10], have made a similar observation during the first iteration of the regular Python course. They also showed that the younger (¡19) participants mainly asked questions while the older participants mainly answer them. In our regular courses we often trigger discussions, by starting introductory threads, quizzes with disputable answers, or offering exercises, which animate the participants to upload a solution in the forum to be discussed there [15]. Particularly in the programming courses, which always feature hands-on exercises, we have a comparatively high forum activity. Many participants need help to get started or get stuck at some detail, so they ask for help in the forum. Particularly, in the Java courses we try to follow a social constructivist approach. The discussions in the forums play an essential role in understanding the material. Therefore, we see the very low forum activity with some concern.

V. Future Work

We plan to re-run the courses next year with a broader marketing. Furthermore, we plan to intensify our collaboration with the teachers, visiting classes where the courses are used, to get a better understanding for the needs and problems of the users. Particularly, we are interested in if and how the lack of activity in the forum is replaced by classroom discussions. For the Java course we need to collect more basic data in surveys to make it better comparable to the Python courses. As for the paper at hand our main focus was on the way the courses are employed by the teachers, we need to switch the perspective and examine the pupils' perspective in more detail.

The top feature on the teachers' wish list, is the possibility to monitor the progress of their pupils. We are convinced that the interest of the vast majority of the teachers in such a feature is driven by the possibility to recognize their pupils' weak points early enough while it is not yet to late to provide special support to the struggling ones. Nevertheless, this kind of monitoring comes with certain privacy implications that need to be analysed in a lot of detail. Finally, we are interested in a deeper examination how to integrate tests and exercises following the SOLO-taxonomy (see e.g. [9]) in these courses.

VI. Conclusion

We have shown that it is possible to employ MOOCs in classrooms if they are only slightly modified in terms of the given time frame. Teachers have successfully used these courses in *in-class* as well as in *in-school and at-home extracurricular* settings. Pupils and teachers as well have received these courses very positively and see them as an element that enriches their education. The courses can be used as a substitute for computer science education, in case there is no such offer in school, which unfortunately still often is the case in Germany. They can also be used to add a different flavor to existing computer science classes. The age ranges that we have targeted for the courses (10-15 for the Python course and 15-18 for the Java course) appear appropriate, but as some pupils have shown, also younger participants can be very successful if they come with previous skill or talent. The courses are also perceived to be well suited for both genders. Although the numbers are a bit too low to make bold statements, we are particularly, positive about the success of the teamwork task and how the pupils perceived the method of peer assessment to evaluate such tasks. The equal distribution of grades also shows that they took this task seriously and didn't just give everyone the best possible grade. We are looking forward to future courses and an intensified collaboration with teachers.

References

[1] M. Aksela, X. Wu, and J. Halonen. Relevancy of the massive open online course (mooc) about sustainable energy for adolescents. *Education Sciences*, 6(4), 2016.

[2] K. Ala-Mutka. A Survey of Automated Assessment Approaches for Programming Assignments. *Computer Science Education*, 15(2):83–102, 2005.

[3] L. W. Anderson and D. R. Krathwohl, editors. *A Taxonomy for Learning, Teaching, and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon, New York, 2 edition, December 2001.

[4] B. S. Bloom. Taxonomy of educational objectives. *Cognitive Domain*, 1, 1956.

[5] C. T. Grella, T. Staubitz, R. Teusner, and C. Meinel. *Can MOOCs Support Secondary Education in Computer Science?*, pages 478–493. Springer International Publishing, Cham, 2017.

[6] M. Khalil and M. Ebner. A stem mooc for school children; what does learning analytics tell us? In *2015 International Conference on Interactive Collaborative Learning (ICL)*, pages 1217–1221, Sept 2015.

[7] R. F. Kizilcec. Collaborative learning in geographically distributed and in-person groups. In *Proceedings of the Workshops at the 16th International Conference on Artificial Intelligence in Education AIED 2013, Memphis, USA, July 9-13, 2013*, 2013.

[8] J. Kurhila and A. Vihavainen. A purposeful mooc to alleviate insufficient cs education in finnish schools. *Trans. Comput. Educ.*, 15(2):10:1–10:18, Apr. 2015.

[9] R. Lister, B. Simon, E. Thompson, J. L. Whalley, and C. Prasad. Not seeing the forest for the trees: Novice programmers and the solo taxonomy. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ITICSE '06, pages 118–122, New York, NY, USA, 2006. ACM.

[10] M. Löwis, T. Staubitz, R. Teusner, J. Renz, C. Meinel, and S. Tannert. Scaling youth development training in it using an xmooc platform. In *2015 IEEE Frontiers in Education Conference (FIE)*, pages 1–9, Oct 2015.

[11] S. Papert and C. Solomon. Twenty things to do with a computer. *Educational Technology*, 12(4):9–18, 1972.

[12] T. Staubitz, H. Klement, J. Renz, R. Teusner, and C. Meinel. Towards practical programming exercises and automated assessments in massive open online courses. In *International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 23–30. IEEE, 2015.

[13] T. Staubitz, H. Klement, R. Teusner, J. Renz, and C. Meinel. Codeocean - a versatile platform for practical programming excercises in online environments. In *2016 IEEE Global Engineering Education Conference (EDUCON)*, pages 314–323, April 2016.

[14] T. Staubitz and C. Meinel. Collaboration and teamwork on a mooc platform: A toolset. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, pages 165–168, New York, NY, USA, 2017. ACM.

[15] T. Staubitz and C. Meinel. Collaborative learning in moocs - approaches and experiments. In *IEEE Frontiers in Education Conference (FIE) 2018, San Jose, USA*, 2018.

[16] T. Staubitz and C. Meinel. Team based assignments in moocs: Results and observations. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, L@S '18, pages 47:1–47:4, New York, NY, USA, 2018. ACM.

[17] L. S. Vygotsky. *Mind in society: The development of higher psychological processes*. Harvard University Press Cambridge, Mass., 1978. (Original manuscripts [ca. 1930-1934]).