# Agile Projects to Foster Cooperative Learning in Heterogeneous Classes

Petra Kastl and Ralf Romeike
Friedrich-Alexander-Universität Erlangen-Nürnberg

***Abstract*— Teaching programming novices in secondary education is often described as a challenge, because student groups are often quite heterogeneous. Teachers react with special methods e.g. with individualized teaching, group work or project-based learning. As such a method for computer science education, that supports project-based learning agile projects are being discussed more recently. In the context of a design-based research work, experienced teachers have adapted and applied a theory-derived agile model in a wide range of contexts. In this qualitative case-study, we analyze 11 interviews with 6 teachers on their observations from 20 agile projects with over 400 students. The aim is to gain insight into how agile practices assist individual learning processes and how they help teachers to design and organize projects in order to support students individually. A structured content analysis shows that agile teams face similar obstacles as teams in plan-driven projects, but that they can overcome them better in agile projects. Additionally, the analysis indicates that in agile projects the quantity and quality of interactions increases, which has positive effects on the construction of sustainable skills and that students furthermore are frequently involved in feedback processes and reflections, which makes their learning more goal-oriented. Moreover, the data shows that teachers can identify their students' strengths and weaknesses better and also observe their individual learning processes better throughout agile project. Based on that knowledge they can design their agile project in a way that each student is challenged systematically and purposefully.**

***Keywords—agile methods; computer science education; K-12; project–based learning; individual differences; cooperative learning; heterogeneity;***

## I. INTRODUCTION

Diversity can be taken as a chance to enrich learning. However, handling heterogeneity can also become one of the biggest challenges in teaching. Computer science (CS) teachers often observe this challenging heterogeneity when they teach programming novices. As a teacher, you may consider differences in your students' previous knowledge, perseverance, or self-reliance irrelevant – it is not uncommon to simply teach to the so-called "average student". The students' feedback to such lessons usually reflects their heterogeneity and ranges from "way too fast" to "way too slow", or from "wow, interesting" to "just boring". This is particularly true because most programming novices consider it very difficult to gain their first programming skills [1]–[3]. Thus, it is rather likely that less-skilled students are discouraged whereas students with prior knowledge become disinterested, if the learning opportunities do not match individual prerequisites. Alternatively, teachers can provide individualized instructions. However, individualization in classes dissolves groups, whereas creating software usually is a cooperative process. Software engineers mainly work in collaboration and therefore more recently higher education also puts more emphasis on the "softer" skills, which are needed for software development (SD) [4], [5]. In order to offer the necessary collaborative learning opportunities, methods such as agile projects have been becoming increasingly popular in higher education during the last decade and are being discussed in secondary computer science education (CSE) more frequently. They are based on an iterative process that emphasizes collaboration and communication and agile practices and artifacts help the teams organize and structure their work. First studies with adapted agile models in secondary CSE show promising results [6]–[8]. Teachers successfully conducted agile projects even early on in their students' learning process [9]. However, a well-discussed aspect of cooperative learning methods such as PBL is that adequate structuring and organization is needed to activate all learners alike. The aim of this qualitative case-study is, to investigate various aspects of agile projects in order to gain a clearer insight into how they activate and encourage the individual student within agile teams in heterogeneous classes. Thus, the following questions are addressed: (RQ1) What is the influence of group formations in agile projects on the individual learning process? (RQ2) How do agile practices and artifacts help teachers to support students individually and purposefully? (RQ3) How do agile projects activate and support the individual student's learning processes?

The paper is structured as follows: In section 2, we provide a short overview on PBL as a method for cooperative learning. Furthermore, we discuss related research work about agile projects in education and situate the study within the long ongoing discussion about how to handle heterogeneity. This is followed by a discussion of related literature and research work in section 3. Thereby, we focus on the discussion of five issues, which we consider to be relevant for the design, structuring and organization of agile projects as a method for handling heterogeneity. In this context, we also detail the research questions related to the issues. In section 4, we describe the methodology of our qualitative study and outline the category system, which was developed based on the previously identified issues. This is followed by a brief overview over the different agile projects and the respective contexts in section 5. In section 6, we present the results of a structured content analysis of the data, which we discuss in section 7, followed by the conclusions.

## II. RELATED WORK AND THEORETICAL BACKGROUND

### *A. Project-Based Learning in General Education*

Project-based learning is a cooperative learning approach in the classroom which aims at engaging students in explorative problem-solving activities in authentic contexts. The idea of learning in projects goes back to the 16th century and was picked up and enriched with various pedagogical aspects in the last century, both during the American progressive education movement (e.g. Dewey and Kilpatrick) as well as in the movements of educational reform in Europe [10]. Project-based learning (PBL) draws on interests and demands of the students who work together in groups to strive for a common and complex result, often a product [11]. This includes planning, problem solving, analysis of different solutions and the evaluation of the process and its product. However, there is a difference between PBL in other subjects and in CS education. As professional SD is mostly done in projects, there are scientific process models that are applied. These models are adapted to educational projects in order to give students insights into ‘real world work’ in SD. Adaptations of plan-driven process models such as the waterfall model to general education in the 1980s and 1990s in Germany [12], [13] describe e.g. student activities along the software life cycle. These adaptations showed conflicts with objectives of PBL in general education such as social learning vs. best product efficiency, cooperation and communication vs. documentation and efficient work split, generalists vs. specialists or flat team structures vs. hierarchical ones, which had to be solved as much as possible. On the other hand, characteristics of PBL (e.g. [14]) show similarities to characteristics of professional agile projects (e.g. [15]): the emphasis on self-organized and focused work, a complex and not precisely defined goal, and good and effective collaboration. Furthermore, social learning and knowledge transfer combined with a collective responsibility for the outcome are important in PBL and in agile projects. Additionally, in PBL, students aim to acquire, apply and enhance a variety of subject-related, methodological and social competencies [11], which at the same time characterize successful software engineers [16], [17].

### B. *The Agile Approach in Higher Education*

During the last decade, many authors report on their successful effort to design courses combining agile approaches and PBL in higher education. Maurer and Anslow [18] point out that teaching group-based agile software development (SD) courses is difficult. Based on their experience, they provide recommendations and techniques on how to bring agile methods successfully into the curriculum, as they consider it important that today’s CS students understand agile practices. McKinney and Denton [19], [20] designed an agile CS1 lab in order to enable students to gain team experiences very early in their studies, and they conclude that it is important to provide “concrete definitions about team skills”. They report benefits such as deeper learning, as well as higher retention, fun and achievement. Rico and Sayani [21] also recommend introducing agile practices as early as possible and they found that appropriate coaching, mentoring of the teams, and meaningful feedback are key factors for success. Kropp et al. [22] elicit specific collaboration and communication skills needed in agile teams and design a teaching concepts based on constructionism and “established psychological-pedagogical perspectives on learning”. Schroeder et al. [4] report on the successful setup of two SD labs, which are based on Scrum and focus on human interactions, social skills and permanent collaboration and feedback. Hazzan [23] presents a study in which he focuses on the conflict students face, “between their urge to express personal skills, and the unavoidable need to cooperate”. He found that students prefer to distribute the major part of the reword equally among the team members. He suggests considering the students’ personal attitude to this topic when forming groups. Layman, Cornwell and Williams [24] present a case study in which they examine a course layout with personality types and learning styles in mind and found that agile approaches make it possible to design courses that appeal to a wide range of students. Beside these contextualized, qualitative studies, there is a large number of quantitative studies discussing the effectiveness of pair programming (PP) as a pedagogical tool. A meta-analysis shows that PP was effective in improving students’ grades on assignment and their satisfaction in an academic context, and that the students’ skill level was the factor that affects PP’s effectiveness the most. However, these results from higher education can only be transferred to secondary education in a very limited way. The perspectives and objectives, the used tools, the selected agile practices, and the social structures are different. Additionally, the projects in secondary CSE usually take place in the classroom only, students work 2 lessons per week on their project and the teacher is always present.

### C. *The Agile Approach in General Education*

Only a few papers discuss agile methods in the K-12 CSE context so far. Meerbaum-Salant and Hazzan identified several challenges teachers cope with in plan-driven projects in Israeli high schools [6]. From a teacher’s perspective, they design a theory-based agile mentoring methodology (ACMM) and evaluate its application. In these teaching interventions, each student individually works on his / her own project-subject and product, i.e. there is no particular emphasis on team work. The findings of the study show that the methodology addresses each of the challenges teachers are faced with during the mentoring process in the given context. Another small-scale study explores the effectiveness of pair programming (PP) [8]. Romeike and Göttel present an agile model for projects in CSE (AMoPCE), which has a profound theoretical basis, highlights the learners’ perspective, and emphasizes cooperation [25]. Based on this paper, further studies follow, which connect theory and practice. First results indicate that the agile practices and artifacts support the objectives of PBL in particular students’ self-organization and subject-specific, social and management skills [7], [9]. In [26] the authors discussed first steps of an approach for investigating agile projects as a method to teach heterogeneous classes.

### D. *The Agile Model for Projects in CSE*

The Agile Model for Projects in Computing Education (AMoPCE, see fig. 1) [25], on which this study is based, includes adapted agile practices and artifacts known from professional

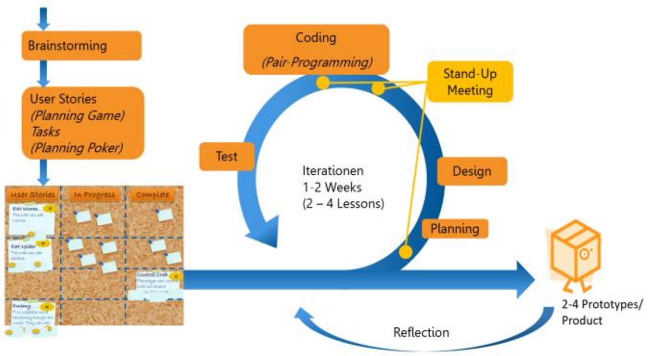


Fig. 1. Model for agile projects in CSE (AMoPCE)

agile SD. They provide clear lines of action that can be followed by the students, as well as easy-to-use hands-on activities (e.g. generating user stories, planning poker, defining tasks). For the design, the authors discussed these practices and artifacts from a pedagogical perspective.

Similar to more recent curricula [27], [28], the authors suggest computer program design as an iterative and collaborative process. Students are supposed to solicit, evaluate and integrate peer feedback to develop or refine solutions, and to use techniques such as pair programming, refactoring and code reuse [6], [27]. Also, pedagogical objectives, e. g. the enhancement of soft skills, especially communication, self-organization and the ability to work in teams, are emphasized [11], [26]. This theory-based model describes how agile projects can be structured, but it does not contain practice-proven design principles that, e.g., show how agile projects in heterogeneous classes can be carried out successfully.

### *E. Handling Heterogeneity in General Education*

The question of how to handle individual differences among learners is neither new, nor specific to CSE. Many studies in various school subjects, as well as meta-studies have been published over the past 50 years [29]–[32] and different approaches to handle heterogeneity have been discussed. In this study, we address an approach, known e.g. from cooperative learning [31], [33], [34], or discussed in the context of ability grouping [35]–[37]. It includes any temporary educational, methodical and / or organizational measure using any form of within-class grouping, thereby offering opportunities for individual and cooperative learning. The aim is to challenge and support the different learners purposefully and systematically. While treating each student as an individual, there are common objectives and competences all students are expected to meet after the activity. PBL is one of various ways to organize such cooperative learning environments.

## III. How we learn – psychological and pedagogical issues

In order to examine and evaluate agile projects as a method, to activate all students and enable teachers to challenge and support each student individually and purposefully in a cooperative learning arrangement, we firstly explored related literature. The aim was to identify aspects relevant for learning in general, as well as for cooperative learning and ability grouping in particular, similar to the approach of Brown and Palincsar [38]. Thereby, we selected five issues that we consider most relevant for the design, structuring and organization of agile projects in heterogeneous classes. These issues are the basis for the development of a category system, which is used to analyze the collected data. In the following, we provide a brief rationale for each issue, the associated research literature and how it is related to the research questions.

### *A. Forming Groups and Obstacles in Cooperation*

In general, there are four different ways to form groups: students' self-selection, random selection, selection by topic choice and selective forming based on criteria (i.e. homogeneous vs. heterogeneous ability grouping). For projects in higher education Redmond [39] suggests that students who are assigned to groups are more likely to be exposed to other students with different abilities from which they can learn new things than when groups are formed by students' self-selection. Hazzan [23] suggests considering the students preferences with respect to the reward structure, too. In general education, there are numerous studies and meta-studies in various subjects that investigate how learning processes in cooperative learning scenarios are influenced by the kind of group forming in general, and by heterogeneous vs. homogeneous ability grouping in particular [36], [40]. However, there are no consistent results. In this context, we address the question how homogeneous vs. heterogeneous group forming in agile projects influences the performance of the individual students. (SQ1.1)

A second aspect is that cooperative learning requires specific skills that students often must learn first. Particularly good students e.g. face, as Hazzan describes it, *"a conflict between their urge to express personal skills, and the unavoidable need to cooperate with their teammates"* [23]. In this context, we address the question as to which obstacles are observed in agile teams that hinder individual learning processes and how agile projects support students in overcoming them. (SQ1.2)

### *B. Teachers' perspective*

#### *1) Objectives and Prerequisites*

**O**bjectives can challenge and stimulate students, if they fit the learners' prerequisites [29]. Otherwise, students may tend to either lose their interest or to become discouraged. An assessment of the students' prerequisites, however, refers to a variety of subject-specific skills as well as to various skills necessary for cooperative learning, e.g. being self-reliant, creative, open-minded, being able to stand one's ground, or to create shared success. We therefore address the question of how agile projects help the teachers to identify the individual strengths and weaknesses of their students with respect to such skills and to provide them with meaningful individual goals that fit their prerequisites. (SQ2.1).

#### *2) Guidance vs. Coaching*

The general consensus is that students develop mental representations and strategies through a slow process of accumulating experience [41], [42]. In their first steps especially, less-skilled learners benefit from strong and structured guidance, initially simple tasks and quick feedback. With increased expertise, guidance can be faded away [38], [43]. Thus, an appropriate method to teach everyone ideally provides each student with the guidance or coaching he / she needs at the appropriate time. Similar to Meerbaum-Salant and Hazzan [6], we pursue the sub-question of how agile projects support teachers to provide adequate individual guidance or coaching throughout projects that start early on in the students' learning process (SQ2.2).

### *C. Students' perspective:*

#### *1) Interaction*

Studies conducted in the 1980s in the context of cooperative learning in general education found that the nature of the learners interactions determines, whether they contribute to improving cognitive abilities [37], [44], and that certain kinds of interactions are typical for successful teams [45]. As agile projects emphasize communication and cooperation we investigate how they foster the kind of interactions that support individual learning-processes (SQ3.1).

*2) Feedback and Reflection*

Self- and peer feedback can have positive effects on learning [46]. However, there is evidence that feedback messages are complex and that students have to learn how to decipher them and how to make use of them [47], [48]. The same applies to reflectiveness and self-reflectiveness, which is an important skill for life-long and self-regulated learning [49]. Thus, we are interested in how agile projects foster the ability of the individual student to reflect and provide, receive, and implement feedback (SQ3.2).

## IV. METHODOLOGY

This case-study is embedded in a design-based research process, which holds the potential to investigate various aspects of an intervention within a context in order to develop empirically-derived design principles [50]. Design-based research (DBR) makes it possible to avoid narrow measurements and makes several aspects accessible to the investigation: dispositions, key competences for later success, and social interactions. Thus, not only correlations between variables recognizable in the surface structure of the lessons can be shown, but also the contextual factors on a deeper level that influence this correlation. The answers on the "how and why" of the success or failure of a lesson are particularly valuable, as they can lead to the extension of scientific knowledge and to more sustainable practical innovations. Reed and Guzdial [51] also suggest a DBR approach for CSE in order to avoid polarities such as discussions about "the only path to success". DBR is characterized by the motivation to improve teaching and learning. It is based on an initial theoretical design and an iterative process of practical implementation phases and validated revisions of the pedagogical design. Therefore, all activities are performed in close collaboration with practitioners. The research design of an DBR process is flexible, as the experience shows that studies in classrooms often lead to unexpected findings that raise surprising or interesting new questions.

Our DBR process started in 2013 with an initial work shop, in which the theoretically-derived agile model AMoPCE was presented to the teachers and discussed with them. They were explicitly encouraged to adapt the model to their specific needs and their context. Since then, we have been closely working together with 6 experienced teachers on an equal footing. Each phase of implementation of adapted agile projects in their classroom was followed by another workshop in order to reflect and evaluate their experiences. As of the time of writing, they have conducted 20 agile projects, based on AMoPCE, with more than 400 students aged between 13 and 18. Data and material such as pictures of project boards, the software of prototypes, project diaries of the students and presentations of their work from the individual projects are available. Further data was collected in the workshops as well as in semi-structured interviews lasting 2 – 3 hours each. As experts in their field and based on their pedagogical content knowledge, the teachers detailed and shared their insights into the deep structure of their cooperative agile learning arrangements. We asked them to describe their experiences with non-agile and agile projects, the context and individual adaptations of their agile projects, their motivation and goals and their observations and experiences in detail. The interviews were repeated in each iteration of the DBR process. Due to personal circumstances, some teachers are currently in the second phase of practical application while others already perform agile projects in the fourth year. In the present case-study, roughly 20 hours of interview material was investigated. For the analysis of the data, a structured content analysis approach was applied [52]. The category system (see table 1) was developed deductively on basis of the literature study, which we detailed in chapter 3.

TABLE I. CATEGORY SYSTEM FOR STRUCTURED CONTENT ANALYSIS

| Main category | Sub-Categories |
|---|---|
| Forming groups and obstacles in cooperation | Information on the kind of group forming, how it took place and the corresponding pedagogical reasoning |
| | Statements about homogeneous or heterogeneous groups with respect to individual learning |
| | Obstacles caused by individual team members that hinder others learning process |
| | Characteristics of agile projects, that allow students to overcome these obstacles |
| Objectives and prerequisites | Characteristics related to the learners differences in personal, professional, management, and social skills |
| | Characteristics of agile projects that allow the teachers to recognize the strengths and weaknesses of students |
| | Individual objectives provided by the teachers |
| Guidance vs. Coaching | Characteristics of agile projects that facilitate the decision, what kind of support the individual student needs at a given moment |
| | Characteristics of agile projects that facilitate the provision of support |
| | Guidance/ Coaching provided by agile practices and artifacts |
| | Characteristics of agile projects that stimulate peers to provide guidance/ coaching |
| Interaction | Characteristics of agile projects that foster interactions, which the teachers consider valuable |
| | Kinds of interactions such as giving explanations, summarize or discuss possible solutions, which the teachers observe in agile projects |
| | Explanations of the teachers as to why a certain interaction supports individual learning processes |
| Feedback and reflection | Kinds of reflections, which teachers observe such as reflection of the product or of the performance in agile projects and how they are related to agile practices and artifacts |
| | Explanations of the teachers as to why a certain kind of reflection supports individual learning processes |
| | Feedback from the product, the peers or the teacher, provided in agile projects and how it is related to the agile practices and artifacts |
| | Explanations of the teachers as to why a certain feedback supports individual learning processes |

## V. THE PROJECTS

Projects are traditionally considered relevant in CSE. The teachers participating in our research project had conducted plan-driven, sequential projects in their classes that followed the software lifecycle, for three to ten years, before moving to the agile approach. They reported that projects in CSE are challenging for the students, as they require a variety of professional skills on many different levels. Therefore, they used to schedule their plan-driven projects late in the students' learning process, when the students were expected to be confident in the necessary skills. The objectives of said projects mainly focus on the application of existing professional skills. With the change to the agile approach, 16 projects were carried out early on in the students' learning process, while 4 were still conducted at the end of the school year, according to the curriculum guidelines.

The agile projects have been conducted in three different federal states of Germany. 7 projects were conducted with students, who had been programming for at least one year (2-3 lessons per week) and 13 with programming novices. The students were aged between 13 and 18 and the classes had a size of 16 to 32 students. In one class, most students' mother tongue was not German, and two projects were conducted within a special program for gifted students, who were programming novices.

The students developed games, such as action games or arcade games, some even with basic artificial intelligence or animated graphics (see fig. 2). Others developed simulations or software with a graphical user interface and a database, which includes patters such as the model view control or the observer pattern. One team e.g. developed software, to support the process of distributing textbooks owned by the school to all students at the beginning of the school year and of collecting them at the end of the school year. To accomplish this, the team worked with an "external" customer during the project and voluntarily offered assistance when the software was first used.

One class used visual programming with scratch, 3 classes used processing, 15 programmed in java, with integrated development environments such as Greenfoot[1], BlueJ[2], or eclipse. One project was conducted with Excel in a class, which was taught to model and implement data flow diagrams. Two teachers introduced repositories and two teams decided to use a repository even though it was not introduced by the teacher. The other teams integrated their code “manually”. The project boards were haptic and basically had the three columns “to do”, “in progress” and “done” (see fig. 3).

The teachers' motivation to conduct agile projects has been very diverse. One teacher wanted his students to experience that SD is a collaborative process. Another teacher wanted her students to reflect the agile process. Two teachers’ motivation was to give the students time to develop an interesting product in a self-organized process and enable them thereby to learn soft skills and organizational skills. The fifth teacher was inspired by the prototypes as running software, which can be reflected by the students and allow to provide frequent feedback. The sixth teacher was motivated by the chance to avoid problems which a late integration of a three-tiered architecture usually caused in his plan-driven projects.

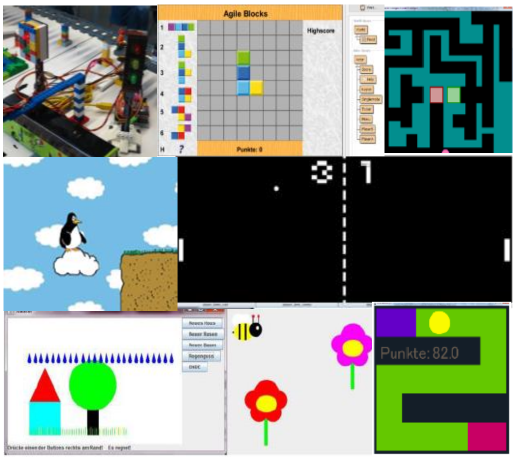


Fig. 2. Examples of games, animated graphics and physical computing

TABLE II. TEAM PROPERTIES AND THE PROJECT DURATION

| Kind of group forming | Total number of teams | Students age | Team size | Project duration |
|---|---|---|---|---|
| By topic choice (heterogeneous) | 16 | 13-16 | 8-12 | 7-8 months, 2 lessons per week |
| Random (heterogeneous ) | 8 | 15-16 | 6 | 6-7 months, 2 lessons per week |
| Random (homogeneous, only high performers) | 1/3 | 16-17 | 4/6 | 6 months, blocks of 4 hours in different intervalls/ 2 lessons per week |
| Students’ self-selection (homogeneous and heterogeneous) | 40 | 14-18 | 2-6 | 3-8 weeks, 2 or 3 lessons per week |
| Ability grouping (homogeneous) | 6 | 16-18 | 6 | 10-12 weeks, 3 lessons per week |

The teachers adapted AMoPCE by e.g. introducing

- mind maps to visualize the project goals
- customer meetings to provide guidance
- a truck factor to make sure students take on the roles of the driver and navigator seriously
- student-stories at the project boards where the teacher provides learning assignments
- a problem-area at the project board that helps the teacher to categorize issues and plan support
- very small task-like user stories, which make the need to refine user stories with tasks superfluous (see fig. 3).

As can be seen in Table 2, the teachers used all kinds of group forming, whereby they obtained the consent of their students before and / or after group formation. In one class, students with very good theory-knowledge, but little programming experience, asked for the formation of mixed teams, in which they can collaborate with more experienced programmers. Ability

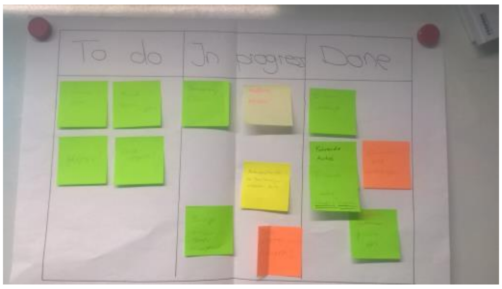


Fig. 3. Project board with small task-like user stories

[1] Interactive Java development environment designed for educational purposes at the high school and undergraduate level and allows easy development of simulations and interactive games. Developed and maintained at King’s College London with support from Oracle.

[2] Java interactive development environment, developed mainly for educational purposes, but also suitable for small-scale software development. Maintained by the BlueJ Team.

grouping, resulting in homogeneous groups was applied in two projects, which were conducted with students, who had more than one year's worth of programming experience. These projects were scheduled late in the year.

In all projects, pair-programming was applied and the pairs were all formed by students' self-selection. In one project with gifted students, the pair-constellation was changed regularly and the students welcomed this opportunity to spread the knowledge. In all other projects, pairing did not change regularly, with the students being free to change it if they wanted. Despite one disrespectful student all students were able to overcome possible obstacles and succeeded in reaching good cooperation in their pairs and teams in the end. Only one student out of more than 400 behaved very disrespectfully and could not improve his behavior, so that the teacher finally took him out of the team and left him to work alone on a project.

The mentoring was similar in all projects. The students had access to supportive material such as master examples from the introductory period. In the beginning of the project the teachers participated in the team meetings and helped the students to substantiate their ideas, to identify initially achievable goals and to write the first small and manageable user stories. After they had guided their students through their first 2-4 short iterations, the students were essentially able to self-organize their work. Now the teachers were able to identify obstacles in their students' cooperation and helped them, to overcome these issues. Particularly in the long projects the teachers took care to maintain all students actively involved throughout the project. They fostered reflections, and supported their students in developing a feedback culture based on agile values. This approach is similar to the agile competence pyramid suggested by Kropp and Meier [5].

## VI. RESULTS

### A. *What is the influence of group formations in agile projects on the individual learning process? (RQ 1)*

#### *1) How does homogeneous vs. heterogeneous group forming in agile projects influence the performance of the individual students? (SQ1.1)*

In line with existing studies, **teachers do not make consistent statements** about how heterogeneous or homogeneous groups influence individual learning processes. A teacher sums up his experience as follows:

> *"In my opinion, homogeneous groups communicate faster than extremely heterogeneous ones, who always try to profit from the freak first. But, it is not always like that."*

This teacher then described how he **supports the teams to gain skills to cooperate effectively**. Teachers report that students need more cooperation skills in agile projects than in usual short group work and that they have to gain them first. All statements about teams with good skills in cooperation however, point out that students benefit from heterogeneous grouping. Therefore, we consider the second question more relevant.

#### *2) Which obstacles are observed in agile teams that hinder individual learning processes and how do agile projects support students to overcome them. (SQ1.2)*

Similar to non-agile projects and independent of the way groups were formed, teachers reported typical problems caused by **individuals such as *chief programmers, dominant managers, man of few words*** (students who do not communicate much) **or *graphic painters*** (students who prefer drawing to coding). However, the teachers also reported that the students were more likely to overcome these problems in agile projects. They argued that **frequent team meetings** throughout the project led to the student feeling more uncomfortable and offered the team members opportunities to directly address such problems:

> *"Your graphic is nice, but we expect you to provide the corresponding code in the next iteration!"*

Another argument was the **visibility of personal responsibilities** on the project board as well as the **visibility of (missing) results** in the prototypes.

> *"I saw that a task was 'in progress' quite long, so I asked the programming pair directly."*

> *S: "We need your code now!"*
> *J: "We have no code."*
> *S: "Okay, then the prototype doesn't have the functionality, when the 'customer' comes today."*

In order to overcome the problems, the team and / or the teachers provided **individual advices, objectives and frequent feedback** to aid the student.

> *"He was an expert, but he often was rude to others. When I talked to him he said 'Well, I explain it, but they don't listen to me!' We discussed this and he struggled to become knowledgeable about the team members and to give them understandable explanations."*

### B. *The teachers' perspective: How do agile practices and artifacts help teachers to support students individually and purposefully? (RQ2)*

#### *1) How do agile projects help the teachers to identify the individual strengths and weaknesses of their students and to provide them with meaningful individual goals, which fit their prerequisites? (SQ2.1)*

Compared to previous non-agile projects and regular lessons, the teachers noted that they had a clearer and more comprehensive picture of the individual strengths and weaknesses of their students throughout the project. Based on that knowledge, they were able to supplement and refine the general learning objectives by individual ones and thus could challenge the individual students purposefully. As a reason, they mention the **frequent interactions and conversations** in agile teams during regular stand-up meetings, plannings and reflections of the prototypes or PP.

> *“At the moment one student is programming and then he informs the team in a meeting. The problem is this dominant student. I will talk to him.”*

> *“In the plannings she spreads confidence. This helps the team a lot.”*

Additionally, by self-organizing their work students were involved in a great variety of **different activities and tasks,** which required varied skills.

> *"In the beginning, one student was the moderator, but over time, the other students alternately took on the role.”*

> *“Only if the task was complex and difficult, he somehow took the leadership.”*

Teachers also mentioned the fact that they had **more time to observe** these activities and to listen to the conversations, as the students were able to self-organize their work soon with the help of concrete agile hands-on practices, agile artifacts and the helpful structure.

> *“I can just listen to them, because they have this structure and organize their work by themselves now. This is great!”*

*2) How do agile projects support teachers to provide adequate individual guidance or coaching throughout projects, which start early in the students’ learning process? (SQ2.2)*

In comparison to non-agile projects, the teachers noted that it became easier to provide adequate and individual guidance. Reasons they gave are, that the students’ **frequent conversations** in the teams and programming pairs and the **information provided by the project board and the prototypes** allowed them to observe the students’ individual learning progress throughout the projects.

> *“When they discussed bugs in their programs, I noticed that over time they realized in which phase it happened, i.e. in the planning, design, or coding.”*

> *“As they put their names on the tasks, I can see, which programming tasks a certain pair can solve at the moment.”*

**Adaptations of the agile framework,** such as the introduction of customer meetings, a cell for problems at the project board or student stories helped provide adequate and individual guidance.

> *“If a post-it note was pinned in the problem cell, I either wrote down a cue on it, passed the problem on to the whole team, or I scheduled an input, and noted the date on the post-it note.”*

Furthermore, the structure of agile projects itself supported individual and accumulative learning: by **tailoring up their product,** the students started with simple initial tasks, and gradually solved tasks that were more complex at their own pace.

> *“The weaker students often chose tasks, which were similar to already solved ones and the more experienced students helped them to identify such tasks.”*

> *“He then did maybe five of the small tasks while others did one.”*

However, teachers also observed that weaker students do choose difficult and huge tasks too:

> *“Firstly I helped them split the task. But still I doubted that they can do it. But I was surprised. They really wanted the feature and with another hint, they succeeded. Afterwards, they proudly explained it to interested teams.”*

Additionally, **peers offered appropriate help**: Whereas teachers initiated and coached the provision of help in the teams especially in the beginning, the students in agile projects soon agreed that it was important to look after each other and learned how to do it.

> *“In my long projects, the good students learned faster and the gap increased. But I expected that every student can explain any feature of the prototype to a customer. Before meetings, I saw that they always organized a short briefing.”*

> *“They then distributed the tasks in a way that the girls, who had very good knowledge but less programming abilities, got programming tasks too.”*

### *C. The students’ perspective: How do agile projects activate and support the individual student’s learning processes?(RQ3)*

*1) How do agile projects foster the kind of interactions that support individual learning-processes? (SQ3.1)*

Teachers reported that in agile projects, the quantity and quality of the interactions with peers, which support individual learning processes, increased compared to previous plan-driven projects. At the same time, “questions” such as “It’s not working.” ceased and the few questions addressed to the teacher were task-related. The teachers reasoned that students more **frequently interacted with specific subject matters and exchanged and discussed** them with their peers.

> *“In a pair it would be rude, not to answer a direct question.”*

*"If they have a problem the pair starts discussing. If they do not find a solution they call the team together. If they do not get any further, they explain their thoughts to me and what they have already done."*

*"In the planning they identified and named similarities to features they had already implemented."*

As further examples, they outlined situations in which students **discussed possible approaches** to realize user-stories in their planning and **agreed on an approach** or situations, in which they tested the prototype and analyzed and fixed errors.

*"They prune their ideas down to a feasible size and they change approaches, where they realize they were at dead ends."*

*"Of course they discuss possible solutions. Otherwise the parts wouldn't fit together in the end."*

This kind of **verbalizations** helped each student to recognize gaps in his / her own understanding and created opportunities to **ask for concrete help** and to **give and receive explanations**, which led to a deeper understanding and more sustainable knowledge.

*"Meanwhile, they know about some problems, there is specialist XY, and they ask him for help."*

*"Whenever she was asked for help, she programmed the solution quickly. When I asked, she says 'Yes, that's pretty exhausting.' We discussed possible ways and she then instead gave explanations and hints. The other team members now had to solve their problems more on their own and became good at it."*

Furthermore, teachers argued that it was also helpful for the individual knowledge construction that students repeatedly took **different perspectives** in their interactions in agile projects. For example, some students spoke in the role of a customer or user, while others spoke as programmers or testers during stand-up meetings, plannings, or retrospectives.

*2) How do agile projects foster the ability of the individual student to reflect and provide, receive, and implement feedback? (SQ3.2)*

In terms of feedback and reflection, teachers report that the students developed better skills in agile projects than in non-agile projects. **Regular tests and presentations of concrete prototypes** stimulated and fostered an active and focused involvement in evaluation and assessment processes.

*"An adequate evaluation of their own product and its degree of production in the regular presentations of the prototypes, was part of the assessment."*

*"'Done' means to the students that the product meets the team's expectations and that's a good thing, I think."*

In each iteration, the students **verbalized and visualized** team goals and individual goals in the beginning and **tested and discussed** the results of the team's and their own work at the end. Particularly user stories, tasks, the running software of the prototype and the structuring framework assisted them with this.

*"The students presented every second prototype in the plenum. They explained, what they had done so far and how they came to their results, compared them with the mind map, where they had visualized their goals and defined their next sub-goals."*

Teachers also observed that students were regularly engaged in developing the ability to **self-regulate their own learning**. As mentioned before, slow learners implemented less tasks than quick learners implement in the same time and they advanced less quickly from simple to complex tasks.

However, in order to reflect, students need pre-defined criteria. A teacher described a typical situation in which students failed due to missing criteria: A skilled student had worked on a refactoring and additional features and the team integrated his work. In the face-to-face talk at the end of the project, these students complained: *"He came up with his solution and big parts of our work ended up in the trash. This caused great frustration and we did not continue to work hard."* The students did not realize that they should have addressed the problem in the team when it occurred. They had no suitable criteria to reflect the situation without the support of the teacher.

But if students are provided with objectives and respective criteria for the evaluation, they can succeed in an agile project:

*"He had a goal and analyzed how he can reach it. He realized that the criterion is not (only) to have an amazing product but also to have good cooperation and harmony in the team. And he put it into reality."*

*"He could have done it all by himself but he knew that he was expected to integrate into the team."*

These feedback and reflection activities helped each learner to identify gaps in his/ her work as well as their learning progress and to become knowledgeable about their own skills and about the team.

Furthermore, it was helpful for the students to **receive differently formulated feedback** from several team members and not only from the teacher. This made it easier for them to decipher the feedback message.

> *"Students often formulate things in ways I would never do, and their classmates understand that much easier."*

Together with the iterative structure of agile projects this allowed students to develop strategies to identify and close individual gaps and made their learning processes more goal-oriented.

## VII. Discussion

This qualitative case-study was designed to investigate 'how and why' agile projects can be an appropriate method to handle heterogeneity in CS classes with programming novices. Therefore, underlying structures of a variety of projects were analyzed in order to filter out appropriate design principles for successful teaching practices. For the individual projects, the teachers were explicitly encouraged to adapt the agile framework. As shown in section 5, this resulted in a large number of very different projects in various contexts. The adaptations are characterized by the teacher personality and their motivation. Additionally, the process was adapted to the needs of the student group, primary goals and further basic conditions. Data from these highly contextualized projects was analyzed with a structured qualitative content analysis. The underlying category system was based on theoretical considerations and literature studies.

The results illustrate that agile projects can indeed be an effective method to foster self-managed cooperative learning in heterogeneous classes. They offer opportunities for all learners to acquire a large variety of skills regardless of their individual prerequisites. While students are well-practiced in short group work and in taking care of themselves, now they can learn how to self-organize collaboration even for a long period gradually and at their own pace. Step by step and with assistance from the teacher and peers the students gain skills such as creating common ground and making decisions, being knowledgeable about the team and striving for a common goal, or seeing the forest for the trees and handle complexity. All these skills are necessary to perform well as a team, to learn from and with each other and to define common achievable sub-goals, which little by little lead to the final goal. The iterative structure of agile projects allows both weaker and excellent students to accumulate experience as they start with simple initial tasks and tailor up their product. The prototype at the end of each iteration allowed them to validate their learning. Furthermore, agile practices and artifacts foster the kind of interaction that benefits all learners. This enables all learners to construct flexible, generalized cognitive structures and to acquire fluid and crystallized abilities, irrespective of their prior knowledge and abilities. Finally, the frequent planning activities followed by an active engagement in feedback and reflection, fosters the development of skills, which make longer periods of self-organized cooperative learning and self-regulated learning more goal-oriented.

Furthermore, analyzing the data indicates that an appropriate guidance by the teacher in the beginning, which gradually turns into coaching, positively influences if and how individual students perceive these learning opportunities. The transparency of agile projects enables teachers to support each student the way he / she needs it during the project and to fade away guidance gradually and appropriately for the individual student throughout the project. In addition, teachers may emphasize feedback and reflection. They therefore pre-define a limited number of transparent common objectives and related criteria. For heterogeneous groups, teachers can successfully supplement them by individual objectives and criteria during the project and thereby design individually-tailored learning opportunities without knowing each students' prerequisites in advance.

## VIII. Conclusion and future work

The results can be generalized to a certain degree as they describe conditions and relationships and resulting design principles for such interventions. They show how agile methods support the teams in mastering the challenge to integrate different experiences, perspectives, knowledge, and personalities of the individuals in order to develop shared understanding and common ground by rich interaction, focused discussions and negotiation.

The following year, a teacher observed that the students, who had participated in agile projects in CS lessons, performed better in a project-seminar in physics. These students encouraged their team e.g., to define small sub-goals, to plan for the next sub-goal, to determine tasks, to visualize personal responsibilities for tasks, to call for the partial results, and to keep an eye on the big picture, before planning the next sub-goal. Inspired by a student, who said to another, *"You learned that last year, that's a real advantage!"* he wrote down his observations of the performance of the individual students in a form. The results of an evaluation suggest that the students, who participated in earlier agile projects were mainly able to apply their social and organizational skills in the other subject, to transfer the learned methods beneficially, and to spread the agile values. He observed this regardless of how well the students' professional achievements in the agile project had been. Therefore, a future cooperation in research on agile projects with teachers from other subjects can be beneficial for education.

The teachers who conducted projects that lasted more than 5 months report an enormous individual increase in professional, social, management and personal skills, which they consider notably higher and more sustainable than in comparable projects without agile methods. In these projects, students moved beyond dependence and independence to interdependence. A teacher, who has been conducting projects in CS-education for more than 10 years, reflected his experience with this learner-centered cooperative method in his final statement:

> *"You just have to believe in them a lot more. It is amazing how well this works."*

## References


[1] A. Robins, J. Rountree, and N. Rountree, "Learning and Teaching Programming: A Review and Discussion," *Comput. Sci. Educ.*, vol. 13, no. 2, pp. 137–172, 2003.

[2] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *ACM SIGCSE Bull.*, vol. 37, no. 3, pp. 14–18, 2005.

[3] 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," *ACM SIGCSE Bull.*, vol. 38, no. 3, pp. 118–122, 2006.

[4] A. Schroeder, A. Klarl, P. Mayer, and C. Kroiß, "Teaching agile software development through lab courses," in *Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON)*, 2012, pp. 1–10.

[5] M. Kropp and A. Meier, "Collaboration and human factors in software development: Teaching agile methodologies based on industrial insight," in *2016 IEEE Global Engineering Education Conference (EDUCON)*, 2016, pp. 1003–1011.

[6] O. Meerbaum--Salant and O. Hazzan, "An Agile Constructionist Mentoring Methodology for Software Projects in the High School," *ACM Trans. Comput. Educ.*, vol. 9, no. 4, pp. 1–29, Jan. 2010.

[7] P. Kastl and R. Romeike, "Now they just start working, and organize themselves" First Results of Introducing Agile Practices in Lessons," in *ACM International Conference Proceeding Series*, 2015, vol. 09–11–Nove.

[8] M. Missiroli, D. Russo, and P. Ciancarini, "Learning Agile Software Development in High School: An Investigation," in *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, 2016, pp. 293–302.

[9] P. Kastl, U. Kiesmüller, and R. Romeike, "Starting out with projects - Experiences with agile software development in high schools," in *ACM International Conference Proceeding Series*, 2016, vol. 13–15–Octo.

[10] M. Knoll, "The project method: Its vocational education origin and international development," 1997.

[11] P. C. Blumenfeld, E. Soloway, R. W. Marx, J. S. Krajcik, M. Guzdial, and A. Palincsar, "Motivating project-based learning: Sustaining the doing, supporting the learning," *Educ. Psychol.*, vol. 26, no. 3–4, pp. 369–398, 1991.

[12] K. Frey, "Die sieben Komponenten der Projektmethode - mit Beispielen aus dem Schulfach Informatik.," *Log*, vol. 3, no. 2, pp. 16–20, 1983.

[13] S. Schubert and A. Schwill, *Didaktik der Informatik*. Springer, 2011.

[14] J. L. R. T. Markham, J. Larmer, "Project Based Learning Handbook: A guide to standards focused project based learning for middle and high school teachers," *Buck Inst. Educ. Novato, CA*, 2003.

[15] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*. Addison Wesley, 2004.

[16] P. L. Li, A. J. Ko, and J. Zhu, "What makes a great software engineer?," *Proc. - Int. Conf. Softw. Eng.*, vol. 1, pp. 700–710, 2015.

[17] R. E. Kelley, "How to be a star engineer," *IEEE Spectr.*, vol. 36, no. 10, pp. 51–58, 1999.

[18] C. Anslow and F. Maurer, "An experience report at teaching a group based agile software development project course," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 2015, pp. 500–505.

[19] D. McKinney and L. F. Denton, "Affective assessment of team skills in agile CS1 labs: the good, the bad, and the ugly," in *ACM SIGCSE Bulletin*, 2005, vol. 37, no. 1, pp. 465–469.

[20] D. McKinney and L. F. Denton, "Developing collaborative skills early in the CS curriculum in a laboratory environment," in *ACM SIGCSE Bulletin*, 2006, vol. 38, no. 1, pp. 138–142.

[21] D. F. Rico and H. H. Sayani, "Use of agile methods in software engineering education," in *Agile Conference, 2009. AGILE'09.*, 2009, pp. 174–179.

[22] M. Kropp, A. Meier, M. Mateescu, and C. Zahn, "Teaching and learning agile collaboration," in *Software Engineering Education and Training (CSEE&T), 2014 IEEE 27th Conference on*, 2014, pp. 139–148.

[23] O. Hazzan, "Computer science students' conception of the relationship between reward (grade) and cooperation," *ACM SIGCSE Bull.*, vol. 35, no. 3, pp. 178–182, 2003.

[24] L. Layman, T. Cornwell, and L. Williams, "Personality types, learning styles, and an agile approach to software engineering education," *ACM SIGCSE Bull.*, vol. 38, no. 1, pp. 428–432, 2006.

[25] R. Romeike and T. Göttel, "Agile Projects in High School Computing Education – Emphasizing a Learners ' Perspective," in *Proceedings of the 7th Workshop in Primary and Secondary Computing Education (WiPSCE'12)*, 2012, pp. 48–57.

[26] P. Kastl and R. Romeike, "Agile Projects as a Method in CSE to Teach Heterogeneous Classes" 2017. [Online]. Available: http://issep2017.cs.helsinki.fi/files/short/agile-projects-as-a-method-in-cse-to-teach-heterogeneous-classes.pdf.

[27] "CSTA K-12 Computer Science Standards, Revised 2017." [Online]. Available: https://www.csteachers.org/page/standards.

[28] P. Bell, "On the theoretical breadth of design-based research in education," *Educ. Psychol.*, no. June, pp. 1–36, 2004.

[29] E. G. Cohen and R. A. Lotan, *Designing Groupwork: Strategies for the Heterogeneous Classroom Third Edition*. Teachers College Press, 2014.

[30] L. Corno and R. E. Snow, "Adapting teaching to individual differences among learners," *Handb. Res. Teach.*, vol. 3, no. 605–629, 1986.

[31] D. W. Johnson, R. T. Johnson, and M. B. Stanne, "Cooperative Learning Methods: A Meta-Analysis Methods Of Cooperative Learning: What Can We Prove Works," *Methods Coop. Learn. What Can We Prove Work.*, pp. 1–30, 2000.

[32] D. W. Johnson and R. T. Johnson, *Learning together and alone. Cooperative, competitive, and individualistic learning*. ERIC, 1994.

[33] R. E. Slavin, "Cooperative learning and student achievement," *Educ. Dig.*, vol. 54, no. 6, p. 15, 1989.

[34] R. Hertz-Lazarowitz, S. Kagan, S. Sharan, R. Slavin, and C. Webb, *Learning to cooperate, cooperating to learn*. Springer Science & Business Media, 2013.

[35] J. E. Rosenbaum, "Social Implications of Educational Grouping," *Rev. Res. Educ.*, vol. 8, pp. 361–401, 1980.

[36] R. E. Slavin, "Ability Grouping in the Middle Grades: Achievement Effects and Alternatives," *Elem. Sch. J.*, vol. 93, no. 5, pp. 535–552, 1993.

[37] N. M. Webb, "Peer interaction and learning in small groups," *Int. J. Educ. Res.*, vol. 13, no. 1, pp. 21–39, 1989.

[38] A. L. Brown and A. S. Palincsar, "Guided, cooperative learning and individual knowledge acquisition," *Knowing, Learn. Instr. Essays Honor Robert Glas.*, pp. 393–451, 1989.

[39] M. A. Redmond, "A computer program to aid assignment of student project groups," in *ACM SIGCSE Bulletin*, 2001, vol. 33, no. 1, pp. 134–138.

[40] D. W. Johnson, R. T. Johnson, and E. J. Holubec, *Circles of Learning: Cooperation in the Classroom*. Interaction Book Company, 1993.

[41] P. A. Kirschner, J. Sweller, and R. E. Clark, "Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching," *Educ. Psychol.*, vol. 41, no. 2, pp. 75–86, 2006.

[42] H. Aebli, *Zw{ö}lf Grundformen des Lehrens: eine allgemeine Didaktik auf psychologischer Grundlage*. Klett-Cotta, 1983.

[43] S. Kalyuga, P. Ayres, P. Chandler, and J. Sweller, "The expertise reversal effect," *Educ. Psychol.*, vol. 38, no. 1, pp. 23–31, 2003.

[44] B. C. Fletcher, "Group and individual learning of junior school children on a microcomputer-based task: social or cognitive facilitation?," *Educ. Rev.*, vol. 37, no. 3, pp. 251–261, 1985.

[45] A. King, "Verbal interaction and problem-solving within computer-assisted cooperative learning groups," *J. Educ. Comput. Res.*, vol. 5, no. 1, pp. 1–15, 1989.

[46] D. Boud, R. Cohen, and J. Sampson, "Peer learning and assessment," *Assess. Eval. High. Educ.*, vol. 24, no. 4, pp. 413–426, 1999.

[47] R. Higgins, P. Hartley, and A. Skelton, "Getting the message across: the problem of communicating assessment feedback," *Teach. High. Educ.*, vol. 6, no. 2, pp. 269–274, 2001.

[48] J. Hattie and H. Timperley, "The Power of Feedback," *Rev. Educ. Res.*, vol. 77, no. 1, pp. 81–112, 2007.

[49] D. Boud, "Sustainable assessment: rethinking assessment for the learning society," *Stud. Contin. Educ.*, vol. 22, no. 2, pp. 151–167, 2000.

[50] Design-based Research Collective, "Design-Based Research: An Emerging Paradigm for Educational Inquiry," *Educ. Res.*, vol. 32, no. 1, pp. 5–8, 2003.

[51] D. Reed and M. Guzdial, "The Power of Computing; Design Guidelines in CS Education," *Commun. ACM*, vol. 55, no. 4, pp. 8–9, 2012.

[52] P. Mayring, "Qualitative content analysis: theoretical foundation, basic procedures and software solution," 2014.