Agile Projects as a Method in CSE to Teach Heterogeneous Classes

Petra Kastl and Ralf Romeike

Computing Education Research Group, University of Erlangen-Nürnberg, Germany
{petra.kastl,ralf.romeike}@fau.de

Abstract. To teach and reach all programming novices in classes with 20 to 30 students is challenging due to the students' great heterogeneity. Therefore, teachers often apply cooperative learning methods such as project based learning. As such, agile projects are being discussed in general CS education more recently. In a case study, interviews with six teachers about their observations from 22 agile projects are analyzed with respect to aspects of cooperative learning and heterogeneity. Therefore, a qualitative content analysis is performed in order to find first answers to the question how certain design elements in agile projects influence students' individual learning processes.

1 Introduction

Teachers in CSE often reporta challenging heterogeneity in classes with programming novices. They observe great differences especially in students' aptitude, their previous programming abilities, the contexts they are interested in or their self-reliance and self-confidence. As a teacher, you may consider these differences to be irrelevant and ignore them. By doing so, it is rather likely that less-skilled students' are discouraged whereas students with prior knowledge become disinterested. This is particularly true because most programming novices consider it very difficult to gain their first programming skills (e.g.[13]). Alternatively, teachers can provide individualized instructions for each student. However, if the students only work for themselfs and alone, they cannot experience that software development mostly is a cooperative process. In order to offer collaborative learning opportunities, methods such as, agile projects are being discussed in CS education more recently.

With this work, we pursue the question how agile projects support each students individual learning process in heterogeneous CS-classes. In section 2 we situate the case study within studies about agile projects in education and within the discussion about handling heterogeneity. In particular, we refer to related work with regard to five issues, which we identified and consider to be relevant for the design, structuring and organization of such agile projects. In section 3, we describe the methodology and present the research questions followed by first findings in section 4.

2 Background and related work

Numerous publications discuss the use of agile practices in higher education, especially the effectiveness of pair programming (PP). A meta-analysis [20] of studies about PP identified lacks of research work, which investigates relevant context-factors and which considers larger parts of the software development (SD) process. In the K-12 context, only a few papers discuss agile methods so far. Meerbaum-Salant and Hazzan identified several issues in plan-driven SD process [15] and present and apply an agile mentoring model for high school projects. There is also a small-scale study about the effectiveness of PP [17]. Another paper presents an adapted agile model for projects in secondary computing education (AMoPCE), which has a solid theoretical basis and emphasizes a learners perspective [19]. It outlines an iterative and incremental project design, provides a set of adapted agile practices and artifacts and encourages team communication and collaboration. This theory paper is followed by research work, which connects teaching theory and teaching praxis and is situated in the aforementioned gap [9, 10]. These studies suggest that agile projects may also be a suitable measure in general CS education for teaching heterogeneous classes.

In general, the question of how to handle individual differences among the learners is neither new nor specific to CS education. Many studies in various school subjects as well as meta-studies have been published over the past 50 years [3, 7, 8, 22]. Basically, there are different approaches to handle heterogeneity. In this study we address an approach, known e.g. from cooperative learning (e.g. [8]). In order to evaluate agile projects as a method for fostering cooperative learning in heterogeneous CS-classes, we firstly explored related literature in order to identify issues relevant for learning in general, as well as for cooperative learning in particular. Thereby, we selected five issues, which we consider to be most relevant for the design, structuring and organization of appropriate agile projects. In the following, we provide the rationale for each issue in terms of handling heterogeneity as well as associated research literature and outline, how these aspects are connected to the agile model.

(I1) Differences in students' prerequisites: Any difference among the learners, which is beneficial with respect to intended objectives, can enrich learning. In general, there are differences e.g. in students' interests, motivation, prior skills and abilities, aptitude and performance (e.g. [3, 7]). The differences in prior skills and abilities refer to subject specific skills, cooperative learning skills as well as management skills and influence design decisions regarding the issues detailed below. As agile projects have a flat hierarchy, emphasize communication and follow the idea, that a project plan will evolve, while the product development progresses iteratively [16, 19] they may offer a variety of activities that challenge and stimulate different learners with respect to the aforementioned skills.

(I2) Forming groups: In general, there are four different ways to form groups: students' self-selection, random selection, selection by topic choice and selective forming (e.g. homogeneous vs. heterogeneous ability grouping). As group forming may influence the effectiveness of learning processes, there are numerous studies and meta-studies (e.g. [21]) regarding homogeneous vs. heterogeneous ability grouping in schools. However, there are no consistent results. In general CSE there are no studies, which investigates ability grouping in SD projects with respect to the students' individual learning progress.

(I3) Guidance: It is agreed that students develop mental representations and strategies through a process of accumulating experience [12, 1]. In their first steps especially, less-skilled learners benefit from strong and structured guidance. In their next steps, students may solve various problems and thereby construct, reconstruct and restructure their knowledge until they have a fluid and crystallized ability as well as a flexible mental representation of the concepts. With increased expertise, guidance can be faded out. Thus, a teacher ideally provides each student with the guidance he/ she needs at the appropriate time. Whereas several issues in mentoring plan-driven projects have been identified, studies indicate that agile models help teachers to overcome these issues [15, 10]. However, there are no studies, which detail aspects of guidance in agile school projects with respect to heterogeneity.

(I4) Interaction: Studies conducted in the 1980s shown that certain forms of verbal interactions such as provide or receive explanations are more likely to lead to improved cognitive abilities [22]. In addition, instructions to verbalize decision-making processes or to reach consensus have positive effects on problem-solving abilities [5]. Another study shows that asking task-related questions and discussing problem-solving strategies is typical for successful groups [11]. Agile projects stimulate, structure and support various verbal interaction [19] by providing concrete hands on practices and agile artifacts. Studies and practical reports so far indicate that these agile elements are beneficial [9, 10], but do not consider heterogeneity in particular.

(I5) Feedback and reflection: There is a considerable number of studies illustrating effects of self and peer feedback on learning (e.g.[2]). However, there is also evidence that feedback messages are complex and that students have to learn how to decipher them and how to make use of them [6]. These skills have to be developed during an accumulative process. In iterative agile projects there are regular self-organized plannings at the beginning and concrete prototypes in the end of each iteration as well as frequent reflections. They may provide more opportunities to practice such skills, than reviews of abstract requirements or design models in plan-driven projects do. The authors are not aware of studies, which analyze this aspect in agile projects in general CSE.

3 Methodology and research questions

This work is embedded in an iterative design-based research process, which was started in 2013 [10]. Such an approach holds the potential to investigate various aspects of the design in different contexts in order to develop empirically derived design principles [4]. Doing so helps to avoid discussions about "the only path to that success" but to instead find ways that describe interventions in CSE with respect to their conditions that "lead to significant learning gains", as Reed and Guzdial argue [18]. During the process, we closely work together with six experienced teachers and integrate their pedagogical content knowledge. As of the time of writing, they have conducted around 22 agile projects with more than 400 students aged between 13 and 18. In semi-structured interviews, they detailed their teaching practices, observations and experiences. During the research process, the question of how the teachers handled heterogeneity in their agile projects arose. Thus, 11 interviews are investigated to find answers to the following research questions:

(1) **How do agile methods support the issues (I2) – (I4) with respect to heterogeneity?** We are interested in teachers observations, which explain how the structure, the practices and the artifacts of agile projects supported the different students in their various learning processes.

(2) **How do teachers handle the issues (I1) – (I5) with respect to heterogeneity?** The design and organization of agile projects is flexible and teachers ideally adapt them to their context. Therefore, we are interested in design considerations and teaching practices, which were used to challenge and support the different learners purposefully and systematically.

For the analysis of the data, a structured content analysis approach is applied [14] and a category system has been developed deductively. In the following, we briefly sketch the main categories (italic), which correspond to the issues. In all categories only observations, which refer to heterogeneity are coded. *Differences in learners prerequisites* comprises data related to the learners differences in personal, professional, management and social skills and to the definition of learning goals. Statements related to group forming and group performance are summarized in the category *social component*. Observations regarding the guidance by teachers, the agile method or supportive material are bundled in the category *guidance*. The category *interaction* encompasses the various aspects of communication and cooperation, such as decision-making or discussing problem-solving. The category *feedback and reflection* comprises feedback received from the product, the teacher and the peers, its evaluation and implementation and the reflection of the product and of the individual and the teams performance.

4 First Findings

All teachers observed the aforementioned **differences in students' prerequisites (I1)**. However, when they described concrete situations we identified attributes that are more specific, such as being self-reliant, productive, or knowledgeable about team members, being able to handle complexity, to stand their ground or to ask for help. These specific attributes resulted from observations of students' interactions and were used to add individual objectives during the project in order to support the individual students purposefully. The teachers in this case study used all four ways of **forming groups (I2)** resulting in homogeneous as well as heterogeneous groups. Irrespective of the group forming, they observed typical issues such as chief-programmers, dominant managers, or graphic-painters as well as their negative influence on team performance. However, they also report that these issues could be solved more effectively during the iterations of agile projects than in previous projects, by observing students' interactions and providing individual guidance, objectives and feedback. Concerning **guidance (I3)**, teachers agree that agile projects support accumulative learning better than plan-driven projects. For instance, they point out that by tailoring up their product iteratively in agile projects, students started with very simple initial tasks and gradually and at their own pace solved tasks that were more complex. By listening to their students' discussions teachers were able to observe the individual progress throughout the project. Due to that transparency they described it as easier to offer each student the guidance he/she needed at the appropriate time and to fade it out gradually. In long projects teachers observed that the knowledge gap between quick and slow learners even increased towards the end of the project but that the students agreed at this time that it was important to look at each other and that they knew how they could do it. Regarding **interactions (I4)** teachers welcome agile practices as they offer concrete instructions. They agree, that students more frequently interacted with subject content and transformed and discussed it with peers than in previous projects. The verbalization helped each student to identify gaps in his/ her understanding. As examples the teachers outline situations, in which students discussed and agreed possible solutions or when they recapped the prototypes behavior with respect to their initial approach in order to elaborate it. In such discussions, some talked in the role of a customer or user whereas others brought in their programming expertise. Concerning **feedback and reflection (I5)** teachers value that students were involved more frequently and actively in monitoring gaps between planned goals and the results of their work than in previous non-agile projects due to the regular tests and presentations of the prototypes. They observed that the students referred to the concrete prototype and the corresponding visible user stories and tasks on the project board during the processes of evaluation and judgment. Teachers report that most teams and students rather quickly developed abilities to identify and verbalize the strengths and weaknesses in their work and to use and implement feedback messages from the product, the teacher and their peers.

In general teachers value agile projects as they made longer periods of self-managed cooperative learning more goal-oriented. Especially those teachers who conducted long projects report an enormous individual increase in professional, social, management and personal skills, in particular with respect to students' previous individual weaknesses. They consider students' individual progress notably higher and more sustainable than in comparable non-agile projects.

References

1. Anderson, J.R.: Cognitive psychology and its implications. WH Freeman/Times Books/Henry Holt & Co (1990)
2. Boud, D., Cohen, R., Sampson, J.: Peer learning and assessment. Assessment & evaluation in higher education 24(4), 413–426 (1999)
3. Cohen, E.G., Lotan, R.A.: Designing Groupwork: Strategies for the Heterogeneous Classroom Third Edition. Teachers College Press (2014)

4. Collective, T.D.B.R.: Design-based research: An emerging paradigm for educational inquiry. Educational Researcher pp. 5–8 (2003)
5. Fletcher, B.C.: Group and individual learning of junior school children on a microcomputer-based task: social or cognitive facilitation? Educational Review 37(3), 251–261 (1985)
6. Higgins, R., Hartley, P., Skelton, A.: Getting the message across: the problem of communicating assessment feedback. Teaching in higher education 6(2), 269–274 (2001)
7. Johnson, D.W., Johnson, R.T.: Learning together and alone. Cooperative, competitive, and individualistic learning. ERIC (1994)
8. Johnson, D.W., Johnson, R.T., Stanne, M.B.: Cooperative Learning Methods: A Meta-Analysis Methods Of Cooperative Learning: What Can We Prove Works. Methods Of Cooperative Learning: What Can We Prove Works pp. 1–30 (2000)
9. Kastl, P., Kiesmüller, U., Romeike, R.: Starting out with projects - Experiences with agile software development in high schools. In: ACM International Conference Proceeding Series. vol. 13-15-Octo (2016)
10. Kastl, P., Romeike, R.: Now they just start working, and organize themselves" First Results of Introducing Agile Practices in Lessons. In: ACM International Conference Proceeding Series. vol. 09-11-Nove (2015)
11. King, A.: Verbal interaction and problem-solving within computer-assisted cooperative learning groups. Journal of Educational Computing Research 5(1), 1–15 (1989)
12. Kirschner, P.A., Sweller, J., Clark, R.E.: Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. Educational psychologist 41(2), 75–86 (2006)
13. Lahtinen, E., Ala-Mutka, K., Järvinen, H.M.: A study of the difficulties of novice programmers. ACM SIGCSE Bulletin 37(3), 14–18 (2005)
14. Mayring, P.: Qualitative content analysis: theoretical foundation, basic procedures and software solution (2014)
15. Meerbaum-Salant, O., Hazzan, O.: An Agile Constructionist Mentoring Methodology for Software Projects in the High School. ACM Transactions on Computing Education 9(4), 1–29 (jan 2010)
16. Meyer, B.: The ugly, the hype and the good: an assessment of the agile approach. In: Agile! pp. 149–154. Springer (2014)
17. Missiroli, M., Russo, D., Ciancarini, P.: Learning agile software development in high school: an investigation. In: Software Engineering Companion (ICSE-C), IEEE/ACM International Conference on. pp. 293–302. IEEE (2016)
18. Reed, D., Guzdial, M.: The power of computing; design guidelines in cs education. Commun. ACM 55(4), 8–9 (2012)
19. Romeike, R., Göttel, T.: 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). pp. 48–57. ACM New York, NY, USA (2012)
20. Salleh, N., Mendes, E., Grundy, J.: Empirical studies of pair programming for cs/se teaching in higher education: A systematic literature review. IEEE Transactions on Software Engineering 37(4), 509–525 (2011)
21. Slavin, R.E.: Ability grouping in the middle grades: Achievement effects and alternatives. The Elementary School Journal 93(5), 535–552 (1993)
22. Webb, N.M.: Peer interaction and learning in small groups. International Journal of Educational Research 13(1), 21–39 (1989)