Doing versus Being Agile in K-12 Computer Science Education: Moving from Methodology to Mindset

Konstantin Oltmann[0009-0008-3035-6960] and Ralf Romeike[0000-0002-2941-4288]

Freie Universität Berlin, 14195 Berlin, Germany
{konstantin.oltmann,ralf.romeike}@fu-berlin.de

Abstract. Agile methods from software development have increasingly been adopted into K-12 (primary and secondary) computer science education. In this position paper, we explore how lessons learned from industry about the distinction between simply applying agile practices (Doing Agile) and genuinely embedding agile values (Being Agile) can enrich K-12 computer science education. Our literature review shows generally positive impacts of various agile-based educational approaches on student engagement and project outcomes. However, explicit teaching and sustained integration of agile values have so far received little to no attention in research. We argue that explicitly fostering an agile mindset with a focus on adaptability, collaboration, and continuous reflection significantly enhances computer science education, equipping students to navigate a complex and uncertain future. Finally, we outline essential conditions for successfully embedding agile values and highlight critical areas requiring further empirical research.

Keywords: Computer science education · K-12 · Agile methods · Value orientation

1 Introduction

Today's rapidly changing world, characterised as VUCA (volatile, uncertain, complex, and ambiguous), a classification that originated in military and economic contexts but now aptly describes the world around us, brings significant challenges to educational systems [2]. Schools are expected to equip students not only with traditional competencies, but also with the ability to successfully navigate through this unpredictable and challenging environment. Managing VUCA conditions effectively requires, among other competencies, agility, understood as the capacity to respond flexibly to rapid changes [10]. This competency significantly influences decision-making and behaviour, which highlights the importance of integrating it early into education.

Similar challenges in software development with rapidly changing project requirements and very complex environments led to the introduction of agile methodologies [1]. Agile project management includes a strong product vision, iterative prototyping and a culture of simplicity and collaboration. Given their success in industry, agile methods like Scrum or Kanban have been increasingly adopted in different educational contexts [27], including project-based learning within K-12 ((primary and secondary) computer science (CS) classes.

However, industry experience shows that agile initiatives often fail when procedures (Doing Agile) are privileged over underlying values and mindsets (Being Agile) [3]. Without these deeper values, implementations become superficial, less adaptable, and unsustainable. The same risk applies in education: if schools adopt agile mainly as procedures without explicitly promoting values, they may repeat industry mistakes. The question therefore needs to be whether schools currently take these agile values into account at all, or whether they miss a significant opportunity to foster meaningful skills and competencies beyond technical abilities and methodological knowledge, if agile values and mindsets are not explicitly taught and reinforced in classrooms.

While large-scale frameworks like eduScrum [33] often integrate agile practices across subjects and seek system-wide change, we focus on agile methodologies and values within K-12 CS classrooms in this position paper. By reviewing existing literature, we address the following questions:

1. How are agile methods (Doing Agile) implemented in K-12 CS classrooms and how is their effectiveness assessed?
2. How are agile values and an agile mindset (Being Agile) promoted in these educational settings?

Based on this analysis, we subsequently discuss our guiding question:

3. Why does explicitly fostering agile values enhance the quality and impact of K-12 CS education?

and highlight the importance of establishing appropriate conditions and conducting empirical research to assess and support the long-term effectiveness of these approaches.

2 Understanding Agile: History, Challenges in Industry and Transition into Education

2.1 Origins and Lessons Learned

Agile methodologies emerged in the field of software development as a flexible approach to manage rapidly changing and complex project requirements as an alternative to traditional linear methods such as the Waterfall project management model. These classic approaches were too rigid and not responsive enough to meet evolving project needs. The new agile approaches were formalised by the Agile Manifesto [1] in 2001 by emphasising four core values and twelve principles that encourage flexibility, effective team collaboration, and continuous improvement. Agile methods as the technical implementation of the aforementioned values and principles are characterised by iterative cycles of planning, execution, and reflection with a strong focus on collaboration. Agile teams frequently prototype to collect user feedback and keep results aligned with customer needs. Among many approaches, Scrum and XP are the most recognised [3], yet empirical studies show that teams seldom follow a single framework and instead blend practices to fit context [19].

Despite widespread adoption, many agile transformations in industry face persistent challenges. Studies show that insufficient emphasis on agile values contributes to struggle and failure [3, 4, 17]. Procedural adherence to agile practices (**Doing Agile**) without internalised values limits team adaptability, reduces motivation, and leads to unsustainable outcomes.

2.2 From Doing Agile to Being Agile

To move beyond a purely procedural stance of **Doing Agile**, organisations must internalise agile values and cultivate an agile mindset [4]. To clarify these concepts, we consider established definitions.

Values, as described by Schwartz [29], are deeply held principles or standards that guide individuals' behaviours, decisions, and interactions within social contexts. In agile environments, the Agile Manifesto sets out four guiding statements that constitute the agile value system: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan [1].

Similarly, a **mindset** is a relatively stable yet malleable lens through which individuals interpret situations and respond to challenges [5]. In the agile domain, two complementary perspectives are helpful. From a values first view, the agile mindset is the coherent bundle that forms when core values are internalised. From a mindset first view, it is the foundational orientation that enables the adoption and sustained practice of those values. This relates to Dweck's growth mindset, which holds that ability develops through effort, learning strategies, and above all feedback that informs subsequent action [5]. The agile mindset shares this learning orientation and the central role of feedback and improvement, while extending the focus to team collaboration and responsiveness to stakeholders. In both perspectives, it is the foundational expression of agile ideas and connects them to everyday practices and interactions.

Organisations often support internalisation through values-oriented training and workshops, supportive leadership, and a culture of trust. These measures help teams embed agile principles deeply into their daily practices and organisational culture (**Being Agile**), which significantly enhances innovation, responsiveness, and increases the long-term effectiveness and sustainability of the organisation as a whole [6, 31].

2.3 Agile Approaches Beyond Software – Adoption to Education

Agile methods have become increasingly widespread across various sectors for managing complex projects, encountering changing requirements with collaboration and responsiveness. Consequently, educational institutions and practitioners are exploring agile principles to redesign learning environments that foster collaboration and responsiveness to diverse and evolving student needs, as seen in initiatives such as eduScrum [33].

However, in education, aims and constraints differ from industrial contexts, and this also affects how agile values are framed, as we show as follows. We interpret values in the sense of the original Agile Manifesto as relative priorities expressed as "X over Y" [1]. The right side remains present but has lower priority. We use the *Manifesto for Teaching and Learning* by Krehbiel et al. [15] as the baseline for educational contexts and tailor the pairs to project-based CS education with a particular focus on school software projects:

Students and collaboration over individual accomplishment

Prototypes and demonstrable progress over verbose documentation

Shared vision and student-driven inquiry over teacher-assigned task lists and lecturing

Responsive iteration and creative problem-solving over rigid plans and fixed solutions

Continuous and sustainable improvement over scheduled standardised examinations

Compared to Krehbiel et al., we retain continuous improvement as a separate pair and interpret demonstration and application through working prototypes and regular demonstrations. Adaptability is enacted within responsive iteration and problem-solving. In the following literature review, we examine how these values and their operational practices have been implemented in K-12 CS education and what evidence supports their effectiveness.

3 Exploring Agile Methods in K-12 CS Education: A Literature Review

To provide ground for the detailed discussion on the potential contribution of agile values in K-12 CS education, this chapter presents a literature review summarising the current state of research on the application of agile methods and value orientation in this context. A distinction between teaching agile as course content ("what is taught") and using agile as a pedagogical approach ("how it is taught"), as suggested by Sharp and Lang in [30], has not been made here, given the expected natural integration of both aspects within project-based learning environments.

We employed a narrative literature review, an approach well-suited to exploring emerging topics, especially given the focused scope and anticipated number of relevant publications in this specific context. The literature search has been performed within established academic databases, including IEEE Xplore, ACM Digital Library, ScienceDirect, and Google Scholar. The search query used was:

(high school OR secondary OR primary OR K-12) AND (agile OR scrum OR XP) AND (computer science education OR information systems education).

A snowballing approach facilitated by Litmaps [18] was additionally applied to explore relevant publications through citation networks, further ensuring a comprehensive representation of existing research. Fig. 1 visualises the identified literature, clusters research groups by colour and arranges publications based on citation frequency and recency.

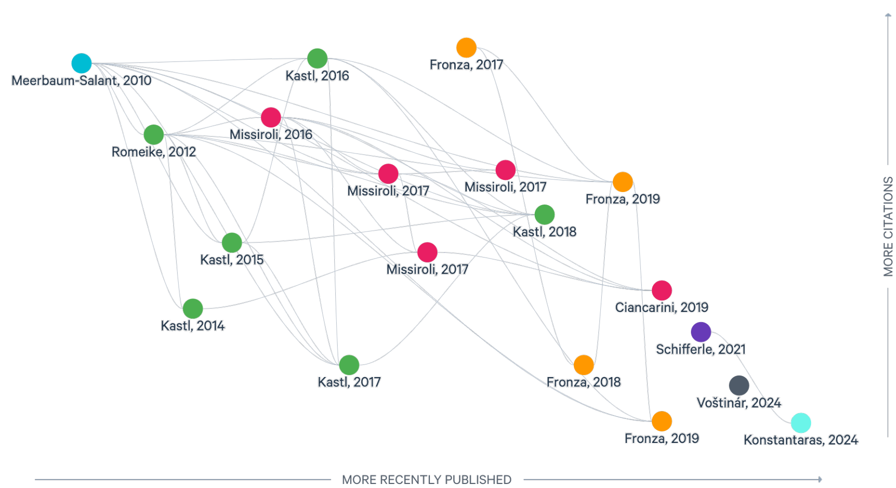

Fig. 1. Relevant literature clustered by research group (colour) arranged based on frequency and recency

As also indicated by Salza et al. [27], the result of the literature review shows that research on agile methodologies within K-12 CS education is limited, especially in comparison with similar studies for university level. The existing literature primarily consists of contributions from a select few research groups, each with multiple publications that build upon each other and some isolated studies, especially from the recent past.

3.1 How Are Agile Methods (Doing Agile) Implemented in K-12 CS Classrooms and How Is Their Effectiveness Assessed?

The following section provides an overview of identified literature with a focus on how agile methods have been practically applied and the empirical evidence supporting their effectiveness.

Meerbaum-Salant and Hazzan introduced the Agile Constructionist Mentoring Methodology (ACMM) [20], which combines agile development, constructionism, and pedagogical frameworks to mentor high school software projects.

Unlike team-based agile approaches, ACMM targets mentoring of individual student projects. Using classroom observations, interviews, and focus groups, they iteratively refined the framework and reported gains in iterative development and feedback culture.

Göttel and Romeike proposed the Agile Model for Projects in Computing Education (AMoPCE) [26], incorporating elements of Scrum and Extreme Programming (XP). Building on this, Kastl and Romeike applied the approach in practice via design-based research, qualitative analyses, and case studies [11, 12]. Findings suggest Scrum- and XP-based methods improve computational thinking, collaboration, and motivation.

Fronza et al. [7, 8] examined agile practices for teaching computational thinking and software engineering using experiments, code analyses, surveys, and interviews. User stories, short iterations, and pair programming increased engagement, problem-solving, and self-management. A two-phase progression was observed, but long-term effects require further validation.

Missiroli et al. [21, 22] studied Scrum and Test-First Development (TFD) in high school, reporting higher motivation, cooperation, and creativity from experimental work, code analyses, and qualitative feedback. Sustaining practices such as TFD proved difficult, indicating the need for further empirical study.

Schifferle and Kollegger [28] found that agile sprints and iterative prototyping improve problem-solving, offering initial insights that need further validation. Voštinár [32] reported better collaboration and self-management with eduScrum in mixed-method studies. Konstantaras et al. [14] showed that combining agile methods with robotics and coding effectively supports algorithmic thinking and collaborative skills using quantitative and qualitative data.

In summary, the reviewed literature primarily focused on directly adopting established agile methods from software development into K-12 CS education. Central questions included identifying suitable agile practices and evaluating their impact on student competencies such as computational thinking, collaboration, and motivation. Common practices were user stories, short sprints, pair programming, TFD with continuous testing, structured reviews and retrospectives. These practices were associated with higher engagement, stronger collaboration and self-management, improved problem-solving, gains in computational thinking, and frequent adaptation to local conditions rather than strict fidelity to a single framework. However, across the reviewed studies, robust longitudinal evidence on sustained effects remains limited.

3.2 How Are Agile Values and an Agile Mindset (Being Agile) Promoted in These Educational Settings?

Following the analysis of how agile practices have been implemented, we examine the extent to which agile values have been considered, how specific practices have been linked to these values and the empirical investigation of their impacts.

Meerbaum-Salant and Hazzan [20] offer a strong example of value orientation with their Agile Constructionist Mentoring Methodology (ACMM). They explicitly map agile practices such as pair programming, continuous integration and reflective processes to specific agile values like effective communication, reflection, openness to change and willingness to learn. Qualitative empirical evidence supports these explicit connections, demonstrating enhanced student skills and attitudes related to these values. However, fostering agile values in collaborative student projects differs significantly from the teacher-to-student dynamics inherent in mentoring approaches. Future research should therefore clearly distinguish between these contexts when investigating the impact of agile value orientation. The Agile Model for Projects in Computing Education (AMoPCE) [26] and its practical implementations by Kastl et al. [12] explicitly foster agile values such as self-organisation, reflection and collaboration through iterative cycles, reflective activities and feedback loops. Their qualitative studies indicate positive impacts on student collaboration and mindset, supporting explicit value integration. Similarly, Fronza et al. explicitly identify agile values such as collaboration, iterative improvement and problem-solving capabilities. Their empirical findings [8] validate that agile practices including pair programming, iterative cycles, and reflective feedback positively influence these values and students' mindsets. In summary, existing studies explicitly link agile practices to specific agile values, with qualitative evidence indicating positive impacts on student skills. Although the direct role of agile values in educational outcomes has been explored only in a limited number of studies, initial findings suggest that explicitly fostering agile values contributes meaningfully to effective learning processes in K-12 CS education.

3.3 Research Gap: Need for Further Empirical Research on Agile Methods and Values in K-12 CS Education

Our review has shown promising benefits of agile methods in K-12 CS classrooms, such as increased student motivation and stronger problem-solving skills. Fostering agile values like openness to change, collaboration, and continuous reflection also appear beneficial, although current evidence relies heavily on anecdotal findings from a select few initial studies. Existing studies provide only limited evidence on the long-term persistence of agile practices and values among students. Moreover, they rarely specify clearly which agile methods are most effective at different grade levels and school types. To meaningfully integrate agile practices and value orientation into classrooms, it is essential to investigate more precisely which approaches work best for various educational contexts. As previously noted, significant issues with superficial agile implementations in industry became clear only through empirical research. This highlights the need for empirical studies in educational contexts to identify potential challenges early as an essential step if agile values are to be sustainably integrated into schools.

4 Fostering Agile Values in K-12: A Critical Discussion

By drawing upon analogies and experiences from industry, we discuss in the following sub-section why explicitly fostering agile values can enrich K-12 CS education. We also subsequently highlight key challenges, identified through discussions with educators, which emphasise aspects that currently limit the effective integration of these values into classroom practice.

4.1 Creating Fertile Ground: Why Explicitly Fostering Agile Values Could Enhance the Quality and Impact of K-12 CS Education

An agricultural analogy can be used to show the connection between implementing agile methods and internalising agile values: if the soil is not fertile enough, even the most sophisticated farming equipment and techniques (which represent agile methodologies) cannot ensure a successful harvest. Fertile soil in this context represents deeply ingrained agile values and mindsets with a strong focus on collaboration, adaptability, and continuous reflection. As discussed earlier, it is important to remember that in industry precisely these patterns with superficially implemented agile practices without a solid grounding in agile values have led to significant problems. This insight is highly relevant for K-12 CS education, highlighting the importance of explicitly embedding agile values to create fertile conditions for successful teaching and learning.

Integrating value orientation into educational concepts is not new [9]. However, the explicit alignment of agile values with concrete methods represents the distinctive strength and opportunity provided by agile approaches. Explicit values education requires continuous and deliberate reflection and practice [29]. To extend the interplay between practices from the software industry and educational concepts, the practice of Continuous Integration (CI) provides a helpful illustration. Software developers use CI to test, evaluate, and improve their code on a regular basis, thereby increasing the quality of their work. Given this, continuous integration and frequent reflection in classroom practice may also help students internalise agile values in a meaningful and organic way. Agile structured project-based K-12 CS classrooms provide ideal conditions with iterative project cycles, where students continuously create and refine meaningful prototypes. This continuous creation of artifacts perfectly matches the constructionist approach [24], which emphasises learning through the continuous process of creating, reflecting upon, and refining meaningful artifacts. Repetition of interaction with evolving prototypes allows students not only to produce artifacts, but also to continuously deepen their understanding of agile processes. This iterative and reflective approach thus supports a holistic learning experience aligned with concepts of reflective learning, such as Kolb's Experiential Learning Theory [13] or Resnick's Kindergarten learning approach [25].

Moreover, explicitly fostering agile values significantly strengthens CS education's contribution to general education. Agile competencies help students develop into capable, accountable, and critically involved citizens, effectively preparing them for a world that is becoming more complex as described by the VUCA concept. This aligns with OECD priorities that emphasise future-ready education and the development of transferable, cross-cutting competencies such as adaptability, collaboration, responsibility, and reflective judgment for navigating uncertainty and shaping societal outcomes [23]. The relevance of these competencies is further demonstrated by industry, as companies across a variety of sectors heavily invest in professional development programmes and agile training to ensure their staff members are adequately prepared for the future [3]. Thus, far from temporary trends, competencies such as adaptability and flexibility appear increasingly critical and enduring for mastering current and future challenges.

In summary, our discussion clearly demonstrates, in response to our guiding question, that explicitly fostering agile values enhances the quality and impact of K-12 CS education. Agile values enhance educational experiences and successfully prepare students for navigating an increasingly complex and uncertain world by emphasising flexibility, teamwork, and ongoing reflection. However, our literature review made it clear that these bold claims require stronger empirical support.

4.2 Climate and Care: Essential Conditions for Sustainable Growth of Agile Values

Drawing again on the farming analogy, fertile soil alone does not guarantee the best results. To fully realise its potential, fertile soil requires a supportive climate and consistent, dedicated care.

Climate: In educational terms, the climate denotes the structural conditions. Lessons from software practice show that skipping values-oriented practices under time pressure, such as retrospectives, undermines communication and continuous improvement [16]. Likewise, securing time for retrospectives or for comprehensive agile projects within school curricula is difficult given competing priorities and rigid structures. This tension was voiced by a German high school teacher with many years of experience using agile methods in classroom projects, a view commonly shared among the educators we spoke with: *"At the end of a typical lesson, there's simply no time left for a proper retrospective. It would be fantastic if we could have enough time to experience everything that makes a real iteration valuable - ideally in one longer, uninterrupted block"*.

Mandatory assessment and grading requirements intensify this challenge, as they favour clearly measurable technical outcomes over process-oriented learning. At the same time, generative AI increases the urgency to rebalance assessment, since producing working software is easier and less indicative of underlying competencies. Assessment should therefore focus more on the development process, including how decisions are made, how feedback is integrated, and how quality improves over time. Iterative workflows support this focus because intermediate results and revisions become visible and thus assessable. Explicitly fostering agile values requires not only reconsidering time allocated to technical activities such as coding but also adapting assessment practices and reserving sufficient space for reflection and collaboration.

Care: The aspect of care reflects teachers' enthusiasm, commitment, and first-hand experience with agile methods. Industry studies show that when management does not commit to agile values or articulate a shared vision, teams struggle to internalise them [4]. The same applies in education, where teachers set direction and expectations. A German high school teacher noted how difficult it is to convey these values without having lived through agile projects or personally embraced them: *"During my own teacher training, I never experienced this kind of project setup firsthand. Of course, I understand the value of an iterative approach, but it would have been so helpful to have actually lived through it myself and actually feel the benefits"*.

To overcome this, teacher training programmes must include explicit opportunities for teachers to gain hands-on experience within agile projects. The effective implementation of agile methods relies on educators first internalising the core values, as only then can these be modelled authentically in practice.

5 Conclusion

In this position paper, we have emphasised the importance of explicitly fostering agile values and mindsets within K-12 CS education. Agile values with a focus on adaptability, openness to feedback, and continuous reflection shape students' attitudes and fundamentally influence their mindset, including how they approach challenges throughout life. In doing so, they prepare them effectively for an increasingly complex and uncertain world. The advantages of agile approaches for student motivation and teamwork have been highlighted by our literature review. Few studies, nevertheless, have methodically investigated the genuine cultivation and maintenance of agile values. Agile approaches run the risk of remaining surface-level if agile values are not explicitly fostered.

Genuinely embedding agile values requires supportive school conditions, sufficient time for reflection, and teachers who themselves authentically embody these values, which highlights a critical area for teacher education. Future research must therefore empirically examine how agile values can be effectively integrated and sustained, ultimately strengthening CS education and contributing meaningfully to students' broader educational development.

6 Acknowledgment

We used Gemini 3 and DeepL Write to support language refinement in this paper. We take full responsibility for its final form.

References

1. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for Agile Software Development (2001)
2. Bennett, N., Lemoine, G.J.: What VUCA really means for you. Harvard Business Review **92**(1/2) (2014)

3. Digital.ai: The 17th State of Agile Report (2024), https://info.digital.ai/rs/981-LQX-968/images/RE-SA-17th-Annual-State-Of-Agile-Report.pdf?
4. Dikert, K., Paasivaara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software **119**, 87–108 (2016). https://doi.org/10.1016/j.jss.2016.06.013
5. Dweck, C.S.: Mindset: The New Psychology of Success. Mindset: The New Psychology of Success, Random House, New York, NY, US (2006)
6. Eilers, K., Peters, C., Leimeister, J.M.: Why the agile mindset matters. Technological Forecasting and Social Change **179**, 121650 (2022). https://doi.org/10.1016/j.techfore.2022.121650
7. Fronza, I., El Ioini, N., Pahl, C., Corral, L.: Bringing the Benefits of Agile Techniques Inside the Classroom: A Practical Guide. In: Agile and Lean Concepts for Teaching and Learning, pp. 133–152. Springer Singapore (2019). https://doi.org/10.1007/978-981-13-2751-3_7
8. Fronza, I., Pahl, C., Susanj, B.: Agile Methods Make It to Non-vocational High Schools. In: Computer Supported Education. vol. 1220, pp. 355–372. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-58459-7_17
9. Halstead, M., Taylor, M.J. (eds.): Values in Education and Education in Values. Routledge, London (1996). https://doi.org/10.4324/9780203973554
10. Johansen, B.: Get There Early: Sensing the Future to Compete in the Present. Berrett-Koehler Publishers (2007)
11. Kastl, P., Kiesmüller, U., Romeike, R.: Starting out with Projects: Experiences with Agile Software Development in High Schools. In: Proceedings of the 11th Workshop in Primary and Secondary Computing Education. pp. 60–65. WiPSCE '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2978249.2978257
12. Kastl, P., Romeike, R.: Agile projects to foster cooperative learning in heterogeneous classes. In: 2018 IEEE Global Engineering Education Conference (EDUCON). pp. 1182–1191 (2018). https://doi.org/10.1109/EDUCON.2018.8363364
13. Kolb, D.A.: Experiential Learning: Experience as the Source of Learning and Development. Prentice-Hall, Englewood Cliffs, N.J (1984)
14. Konstantaras, A.J., Maravelakis, E., Giannakakis, G., Doitsidis, L., Spanoudakis, N., Duca, E.: K-12 Agile Learning with Educational Software and Robotics Technology. In: 2024 5th International Conference in Electronic Engineering, Information Technology & Education (EEITE). pp. 1–5 (2024)
15. Krehbiel, T.C., Salzarulo, P.A., Cosmah, M.L., Forren, J., Gannod, G., Havelka, D., Hulshult, A.R., Merhout, J.: Agile Manifesto for Teaching and Learning. Journal of Effective Teaching **17**(2), 90–111 (2017)
16. Kropp, M., Meier, A.: Collaboration and human factors in software development: Teaching agile methodologies based on industrial insight. In: 2016 IEEE Global Engineering Education Conference (EDUCON). pp. 1003–1011 (2016). https://doi.org/10.1109/EDUCON.2016.7474675
17. Kuchel, T., Neumann, M., Diebold, P., Schön, E.M.: Which Challenges Do Exist With Agile Culture in Practice? In: Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing. pp. 1018–1025. SAC '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3555776.3578726
18. Litmaps (2025), https://app.litmaps.com/
19. Masood, Z., Hoda, R., Blincoe, K.: Real World Scrum A Grounded Theory of Variations in Practice. IEEE Transactions on Software Engineering **48**(5), 1579–1591 (2022). https://doi.org/10.1109/TSE.2020.3025317

20. 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 (2010). https://doi.org/10.1145/1656255.1656259
21. Missiroli, M., Russo, D., Ciancarini, P.: Agile for Millennials: A Comparative Study. In: 2017 IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM). pp. 47–53. IEEE, Buenos Aires, Argentina (2017). https://doi.org/10.1109/SECM.2017.7
22. Missiroli, M., Russo, D., Ciancarini, P.: Teaching Test-First Programming: Assessment and Solutions. In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC). pp. 420–425. IEEE, Turin (2017). https://doi.org/10.1109/COMPSAC.2017.229
23. OECD: Embedding Values and Attitudes in Curriculum: Shaping a Better Future. OECD (2021). https://doi.org/10.1787/aee2adcd-en
24. Papert, S., Harel, I.: Constructionism. Ablex Publishing, Norwood, NJ (1991)
25. Resnick, M.: All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In: Creativity and Cognition 2007, CC2007 - Seeding Creativity: Tools, Media, and Environments. pp. 1–6 (2007). https://doi.org/10.1145/1254960.1254961
26. 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. pp. 48–57. WiPSCE '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2481449.2481461
27. Salza, P., Musmarra, P., Ferrucci, F.: Agile Methodologies in Education: A Review: Bringing Methodologies from Industry to the Classroom. In: Agile and Lean Concepts for Teaching and Learning: Bringing Methodologies from Industry to the Classroom, pp. 25–45. Springer Singapore (2019). https://doi.org/10.1007/978-981-13-2751-3_2
28. Schifferle, T.M., Kollegger, N.: Enabling agile rapid product development in K12 classrooms by enhancing an educational exoskeleton. In: FabLearn Europe / MakeEd 2021. pp. 1–3. ACM, St. Gallen Switzerland (2021). https://doi.org/10.1145/3466725.3466768
29. Schwartz, S.: Universals in the Content and Structure of Values: Theoretical Advances and Empirical Tests in 20 Countries. In: Advances in Experimental Social Psychology, vol. 25, pp. 1–65. Elsevier Science & Technology (1992). https://doi.org/10.1016/S0065-2601(08)60281-6
30. Sharp, J., Lang, G.: Agile in teaching and learning: Conceptual framework and research agenda. Journal of Information Systems Education **29**, 45–52 (2018)
31. Stettina, C.J., van Els, V., Croonenberg, J., Visser, J.: The Impact of Agile Transformations on Organizational Performance: A Survey of Teams, Programs and Portfolios. In: Agile Processes in Software Engineering and Extreme Programming. pp. 86–102. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-78098-2_6
32. Voštinár, P.: Teaching programming using eduScrum methodology. PeerJ Computer Science **10**, e1822 (2024). https://doi.org/10.7717/peerj-cs.1822
33. Wijnands, W., Stolze, A.: Transforming Education with eduScrum. In: Agile and Lean Concepts for Teaching and Learning: Bringing Methodologies from Industry to the Classroom, pp. 95–114. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-2751-3_5