

Overview of Agile frameworks in Computer Science education

Pavlo V. Zahorodko^[0000–0002–8254–7899]

Kryvyi Rih State Pedagogical University,
54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine
`zahorodko.pavlo@kdpu.edu.ua`

Abstract. This paper explores the application of Agile frameworks in education to enhance the learning process. The lack of a generalized approach to Agile in education prompted an analysis and comparison of various frameworks, including Scrum, eduScrum, Agile Learning Loops (ALL), and other Scrum-based approaches. The Agile Manifesto was also reinterpreted for educational settings. The review highlights the differences and similarities among these frameworks in terms of their suitability for education, support for classic increments, project-based approach, documentation, technological stack, and role distribution. The analysis concludes that while Agile is increasingly used in education, there is no standardized methodology. The reviewed frameworks offer valuable insights, but their implementation requires experimentation and adaptation to individual educational contexts.

Keywords: Agile · Agile frameworks · Scrum · Agile Learning Loops · Scrum in education · eduScrum · Computer Science education · project-based learning · teaching methodologies

1 Introduction

Nowadays, Agile is one of the most popular approaches in Computer Science. Agile is also popular in such fields as education, even though it is widely accepted to consider Agile only for Software Engineering. Many teachers and educators try to adopt Agile to enhance the learning process. However, there is no generalized approach to applying Agile in education. This paper aims to resolve this issue by comparing various Agile frameworks and their applicability in education.

Agile contains many frameworks, such as Scrum, Kanban, and Lean. However, these frameworks were originally designed for application in Computer Science and Software Engineering, which can complicate their usage in an educational context. To simplify the understanding of Agile in education, the first attempts to apply Agile involved reinterpreting the Agile Manifesto [2]. The Agile Manifesto was initially reinterpreted by Steve Peha [8], who replaced technical terms with educational ones.

Research on the application of Agile in education is relatively limited, with most use cases focused on high school education and few on higher education.

Consequently, there is limited knowledge about the methods and approaches for implementing Agile in educational settings. Furthermore, due to the lack of a standardized methodology, each research study provides an independent solution, often not fully compatible with other frameworks or interpretations.

This paper is organized as follows: First, a literature review is conducted, focusing on the Agile Manifesto and the various Agile frameworks used in education. Next, a comparison of these frameworks is presented, highlighting their differences and similarities. Finally, a conclusion is drawn based on the findings and recommendations for future research in this area.

2 Literature review

2.1 Agile Manifesto

As was mentioned before, the first attempt to bring Agile to the educational paradigm was made by Peha [8]. For greater clarity, let us compare Steve Peha's Agile Manifesto with the original one. The comparison is shown in table 1.

Table 1. Comparison of Steve Peha's Agile Manifesto and the original one.

Steve Peha's Manifesto	Original Manifesto
Individuals and interactions over processes and tools	Individuals and interactions over processes and tools
Meaningful learning over the measurement of learning	Working software over comprehensive documentation
Stakeholder collaboration over constant negotiation	Customer collaboration over contract negotiation
Responding to change over following a plan	Responding to change over following a plan

As shown in table 1, only two points were changed. In the second one, 'Working software' replaced 'Meaningful learning'. It can be treated as a synonymical replacement since there is no working software in school, but working software is meaningful learning in terms of education. Regarding the third point, the term "customers" was replaced with "stakeholders," which is a more appropriate term from an educational perspective.

The Agile Manifesto should not be considered a final and ready-to-use solution. In terms of Agile, such Manifesto creates the basis for further development. In this case, Steve Peha's interpretation allows us to create a generic and stable learning environment using only these principles.

A more substantial interpretation belongs to Krehbiel et al. [7] in the paper "Agile Manifesto for Teaching and Learning" [7]. Their Manifesto for Teaching and Learning stated the following [7, p. 7]:

- Adaptability over prescriptive teaching methods.

- Collaboration over individual accomplishment.
- Achievement of learning outcomes over student testing and assessment.
- Student-driven inquiry over classroom lecturing.
- Demonstration and application over-accumulation of information.
- Continuous improvement over the maintenance of current practices.

This Manifesto is completely different from the previously reviewed by its size and values. The first two points are described similarly to Steve Peha's approach but with some changes. "Adaptability over prescriptive teaching methods" aligns with the Agile principle of "Responding to changes over following the plan". This point tries to overcome what Steve Peha tried to describe. So that schools would be more adaptable to new approaches and more agile. At the same time, "Collaboration over individual accomplishment" is similar to "Customer collaboration over contract negotiation" but from a different perspective. Both of those points can be fairly merged, but for clarity, they are separated. Additionally, the second point follows the original Agile because collaboration between different team members is among the most important points in software development. This is clearly stated in the fourth Agile principle [3]: "Business people and developers must work together daily throughout the project." In educational interpretation, it would mean that students and teachers should collaborate and work together. Moreover, this principle interferes with popular Individualized Learning, but not in a bad way; on the contrary, it enhances the individual's knowledge through teamwork and constant learning.

The third point of the Agile Manifesto for Teaching and Learning is extensively valued but also controversial. Agile is more concerned about visible value, which, from an educational perspective, would be an achievement of learning outcomes. However, it is near impossible to remove or omit student testing and assessment in real cases. Any discipline or course is inclined to student testing as the only way of knowledge estimation. Therefore, it is hard to implement this point fully, but it is still worth considering.

The fourth and fifth points are related to motivating students and helping them to demonstrate their knowledge. Student-driven inquiry is tightly connected to the following principle of Agile: "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done." From an educational perspective, students need to be interested and highly motivated to have the best results. Therefore, it is essential to start a new discipline or course by motivating students. Worth to notice that student-driven inquiry does not replace classroom lecturing. Both of them are connected and cannot be separated. Though, in this case, it means that lectures should not be the main source of knowledge but rather be a part of the whole learning process. Demonstration and application are also important and related to the Minimum Viable Product (MVP). MVP is an approach to building new projects fast with minimum investments [10]. Demonstrating the applicability of students' knowledge is essential because it helps them better understand how to apply their knowledge practically. Additionally, this point is connected to the

third point, whereas the latter appeals to achieving learning outcomes. Such connecting allows us to implement Agile in a project-based way.

The final point is about continuous improvement. This one is one of the most important in Agile and education. Improvements should be noticed and applied in the process, significantly improving knowledge and framework. Not all of the frameworks support this approach. For instance, Kanban is not an iterative framework. It uses continuous development [1]. At the same time, implementation of real Scrum with the standard of a one or two-week sprint will not be possible for educational institutions. Therefore, following a fully compliant way of Scrum continuous development would be hard. However, this point is only limited to improvement in new practices, which means that teaching techniques should not be fixed but rather agile.

2.2 Agile frameworks

Most of the modern Agile frameworks are generic and universal. As for this research, the only framework for a review will be Scrum. The reason is the lack of research papers regarding other frameworks.

Scrum is an iterative framework without any particular organizational rules. Scrum is a set of recommendations and instructions with a guide created by Schwaber and Sutherland [9]. This framework is built upon five principles or core elements [9, p. 4]:

- Commitment.
- Focus.
- Openness.
- Respect.
- Courage.

This freedom allows to create a framework upon Scrum easier. One of the most prominent examples is eduScrum, created by eduScrum team [5]. eduScrum is a framework which reinterprets Scrum from an educational perspective. It does not change core principles but rather changes roles and instructions for them to be more suitable for teachers.

Traditionally, Scrum has a few roles [9, p. 5]:

- *Scrum Master*. This role controls the process for it to be completely Agile.
- *Product Owner*. This role is responsible for producing requirements for a team.
- *Developers*. This role is responsible for actual implementations of the Product Owner's requirements.

Migrating these roles to educational institutes is straightforward. According to eduScrum [5, p. 9-10], a teacher implements two first roles. Thus, the teacher represents a role that controls the process and creates requirements (assignments) for students. However, in eduScrum's scope, Scrum Master is called eduScrum Master or team captain. Developers are replaced with students as they

are implementing assignments. eduScrum provides a new ambiguous role called 'Team Captain'. Team Captain is a mediator between a team and a teacher (Product Owner). At the same time, the teacher supports the team as a team captain whenever the team is stuck or cannot move toward a goal.

As mentioned before, Scrum is an iterative framework, which implies that the whole working process should be done in iterations or time intervals. Such intervals are usually called Sprints. In Scrum, each sprint should have a fixed amount of time up to a monthly interval [9, p. 7]. The issue with applying Sprints in education is that they are designed to have a 40 hours working week. However, it is not possible for students to spend such an amount of time on one particular course. eduScrum solves this issue by making the sprint duration approximately one-semester [5, p. 15]. This makes eduScrum a non-iterative framework because iterations are too long to have continuous improvements.

Similarly to Scrum eduScrum has Celebration Criteria (CC). CC is a criterion of what needs to be done, how, and with which deadline. In Agile, these criteria are called the definition of done; thus, when a conclusion about the task being finished can be made.

Another similar approach to combining Scrum is Agile Learning Loops (ALL). Learning loops are similar to Scrum by iterations. Böhm and Unnold [4] developed the concept of ALL [4]. This paper focuses on Loop Learning Methods, which are divided into three:

- *Single Loop Learning*. Focused on solving specific problems.
- *Double Loop Learning*. Focused on deeper analysis and making assumptions.
- *Triple Loop Learning*. Focused on studying the underlying context of the problem.

According to Böhm and Unnold [4], Triple Loop Learning is the most suitable for higher education [4, p. 2]. It allows to teach students to formulate their hypothesis independently of application scope.

ALL adopt the same strategy as Scrum for organizing work. They have 2-3 weeks Sprints. The only difference is the Students Learners Diary. Since students cannot meet daily about one subject, they reflect on their work daily with a diary. Reflections are also more often in comparison with eduScrum. Traditionally, the same as for Scrum, after each sprint team has a sprint review, each member, including teachers, can give feedback about the sprint.

A role distribution in the ALL methodology is slightly different. In their case, a Scrum Master is only a teacher who helps to coordinate the process. Meanwhile, a Product Owner is a role shared by teachers and students, similar to eduScrum's Team Captain. The development team is represented only by one person, which can be interpreted as a team captain, who helps to coordinate local processes within the team.

In the ALL methodology, Learning Loops and Scrum are merged. As each sprint is distributed along the semester, learning loops, starting from the single, make a smooth transition from one to another. In this way, at the end semester, students will reach Triple Loop Learning by working on a proposed project.

A more practical example of pure Scrum was done by the Jo et al. [6]. Their examples include the technology stack they used and how they organised it. For them, the appropriate sprint duration is 3-8 weeks, which is reasonable in terms of 10 working weeks for students. As for work organization, they prefer to use Notion instead of Jira. The Notion is simpler in comparison with Jira and can combine a lot of features. Additionally, their framework adopts a project-based approach to organizing work. Thus, students should present a ready project at the end of the course.

From the technical side, they have used classical tools for Version Control, which is GitHub. An interesting approach was taken for organizing parallel work between students. Using branches within GitHub, each team was working in parallel and then, at some point, merged into one general branch at the end of the project. As for boards, the authors suggest a combination of Scrum and Kanban boards. From Scrum, they are using Product Backlog, which displays all available tasks, and a Task board with particular tasks with assigned responsible person and priority. As for the Kanban, there is a general board where the whole process and all the tasks are shown to track progress.

The organization of the process was done in the following way [6, p. 6]:

- Week 1. Sprint and project planning.
- Week 2. Creation of Backlog with tasks.
- From week 3 to week 8. Every two weeks are divided into sprints.
- Week 9. Presentation of finished projects.

3 Comparison of approaches

Each of the reviewed approaches differs from another by a set of features. We will use a set of distinguishable features for each of the selected frameworks to compare them. A comparison of the frameworks is presented in table 2.

Table 2. Comparison of Scrum frameworks in education.

Framework	Scrum	eduScrum	ALL	Scrum by Jo et al. [6]
Supports classic increments	+	-	+	+
Project-based approach	+	+	+	+
Can be used in higher education	-	-	+	+
Has documentation	+	+	-	-
Has description of necessary technological stack	-	-	-	+
Supports different roles	+	+	+	-

The frameworks differ in their support for classic increments, project-based approach, suitability for higher education, documentation, description of the necessary technological stack, and role distribution.

Scrum supports classic increments, making it suitable for iterative development. However, eduScrum and ALL deviate from classic increments due to the longer duration of their sprints or learning loops. The approach described by Jo et al. [6] also supports classic increments.

All the frameworks embrace a project-based approach, emphasizing the completion of projects or assignments. This aligns with the Agile principle of delivering working software and encourages students to apply their knowledge in practical contexts.

While Scrum and eduScrum are primarily designed for school education, ALL and the approach described by Jo et al. [6] can be used in higher education settings. This flexibility allows for the adaptation of Agile principles to different educational contexts.

Scrum and eduScrum have documentation that provides guidelines and instructions for implementation. However, ALL does not have detailed documentation, making its implementation less structured. The approach described by Jo et al. [6] provides practical examples but lacks comprehensive documentation.

The description of the necessary technological stack is absent in Scrum and eduScrum. However, the approach described by Jo et al. [6] includes a description of the technological tools used, such as Notion for project management and GitHub for version control.

The frameworks differ in their approach to role distribution. Scrum and eduScrum support different roles, including Scrum Master, Product Owner, and Developers (students). ALL also has role distribution, with a teacher acting as the Scrum Master and students and teachers sharing the Product Owner role. In contrast, the approach described by Jo et al. [6] utilizes a simplified role distribution with a teacher as the Scrum Master and Product Owner and only one student representing the development team.

4 Conclusion

The application of Agile principles in education is gaining popularity, but there is a lack of standardized methodologies. This paper reviewed and compared various Agile frameworks used in education, including Scrum, eduScrum, ALL, and other Scrum-based approaches. The Agile Manifesto was reinterpreted for educational settings to provide a foundation for the frameworks.

The comparison revealed differences and similarities among the frameworks, highlighting their suitability for education, support for classic increments, project-based approach, documentation, technological stack, and role distribution. Scrum and eduScrum are well-documented frameworks that support classic increments and a project-based approach, but they are primarily designed for school education. ALL and the approach described by Jo et al. [6] can be used in higher education and provide practical examples but have limited documentation.

In conclusion, while Agile is increasingly being used in education, there is no standardized methodology. The reviewed frameworks offer valuable insights into the application of Agile principles in education, but their implementation

requires experimentation and adaptation to individual educational contexts. Further research is needed to develop comprehensive and well-documented methodologies for implementing Agile in education.

5 Future work

Based on the findings of this study, the following recommendations for future research in the application of Agile in education are suggested:

1. Conduct more empirical studies and case studies in higher education to explore the effectiveness of Agile frameworks in different educational contexts.
2. Develop comprehensive and well-documented methodologies that provide guidelines and instructions for implementing Agile in education.
3. Explore the integration of Agile frameworks with existing educational practices and curricula to identify areas of synergy and potential challenges.
4. Investigate the role of technology in supporting Agile practices in education, including the use of project management tools, version control systems, and collaboration platforms.
5. Explore the scalability of Agile frameworks in education, considering larger class sizes and diverse learning environments.

By addressing these research recommendations, educators and researchers can further advance the understanding and implementation of Agile principles in education, ultimately enhancing the learning experience for Computer Science students.

References

- [1] Atlassian: Kanban - A brief introduction (2023), URL <https://www.atlassian.com/agile/kanban>
- [2] 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), URL <https://agilemanifesto.org/>
- [3] 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.: Principles behind the Agile Manifesto (2001), URL <https://agilemanifesto.org/principles.html>
- [4] Böhm, K., Unnold, Y.: Agile Learning Loops Combining Agile Approaches in Higher Education Programs. In: Nunes, M.B., Isafas, P., Powell, P. (eds.) IADIS International Conference Information Systems - IS IADIS International Conference Information Systems 2021, pp. 177–184 (2021), URL <https://www.iadisportal.org/digital-library/agile-learning-loops-combining-agile-approaches-in-higher-education-programs>

- [5] eduScrum team: The eduScrum Guide: “The rules of the Game” (Jan 2020), URL https://eduscrum.com.ru/wp-content/uploads/2020/01/The_eduScrum-guide-English_2.0_update_21-12-2019.pdf
- [6] Jo, S., Park, S., Kwon, G.: Student-Friendly Scrum Framework for Undergraduate Education. *The Journal of Korean Institute of Information Technology* **20**, 199–206 (11 2022), <https://doi.org/10.14801/jkiit.2022.20.11.199>
- [7] 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 (Apr 2017), URL <https://eric.ed.gov/?id=EJ1157450>
- [8] Peha, S.: Agile Schools: How Technology Saves Education (Just Not the Way We Thought it Would) (Jun 2011), URL <https://www.infoq.com/articles/agile-schools-education/>
- [9] Schwaber, K., Sutherland, J.: The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game (Nov 2020), URL <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- [10] ScienceSoft USA Corporation: Minimum Viable Product (MVP) Development: Step-by-Step Guide (2023), URL <https://www.scnsoft.com/software-development/mvp>