Issues in Information Systems
Volume 26, Issue 3, pp. 340-349, 2025

DOI: https://doi.org/10.48009/3_iis_2025_2025_127

Applying agile principles to pedagogical activities and strategies in a learning-centric environment

John C. Stewart, *Robert Morris University, stewartj@rmu.edu*
G. Alan Davis, *Robert Morris University, davis@rmu.edu*

Abstract

In this research, we present an update to our prior work on the initial mapping of Agile principles to pedagogical strategies. We have found that Agile teaching methods have been primarily confined to project-based and computer information systems-related courses. However, these methods can have a much broader impact on all teaching disciplines. In this update to our prior research, we review the literature over the last 16 years on the application of Agile methods to pedagogical approaches. Further, we attempt to expand upon our initial framework to reflect a more comprehensive implementation, rather than simply a strategy for technology-based courses. In the current research, we reiterate our contention that the congruence of principles of the Agile Manifesto and pedagogical strategies (including Active and Cooperative learning approaches) can improve teaching effectiveness, facilitate learning, create a learning-centric environment, and contribute to student success and retention.

Keywords: agile methods, active learning, cooperative learning, agile pedagogy, higher education curriculum

Introduction: Agile Methods in the Classroom

In 2001, the Agile Alliance recognized the need for an alternative to rigid procedural documentation-driven, heavy-weight software development processes. The Agile Alliance was a group of representatives from the disparate, but similar, philosophies on the implementation of software development: Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming et al. The Agile Alliance created the Agile Manifesto (Beck et al., 2001) to outline a standard of Agile methods in software development. These principles of software development have migrated into the academic arena, where there have been several studies on the application of Agile methods to software development in college-level engineering and computer science courses (Alfonso & Botia, 2005; Bergen, 2005; Boehm et al., 2002; Hislop, 2002).

In 2009 we proposed that Agile principles could be applied to teaching methods and pedagogy (see Table 1) to facilitate effective and improved learning outcomes (Stewart et al., 2009). In that research, we showed how Agile principles outlined in the Agile Manifesto could be directly mapped to pedagogical strategies. The research demonstrated that such pedagogical strategies could then, ultimately, facilitate a learning-centric environment in the classroom.

The similarities between software development methodologies and educational methodologies are easily observed. For example, both teaching and software development require detailed planning and scheduling. Each discipline requires management, ongoing assessment, and feedback to and from all involved stakeholders. In addition, the challenge of developing and delivering a higher-education course effectively and on time presents similar difficulties to those encountered in software development projects.

In the years since our original and formative paper, a number of studies have used our initial application of Agile methods to learning strategies and outcomes to further research in this area. Some have used our initial mapping and proposed application of agile principles to pedagogical strategies and attempted to expand upon them (Niculescu,et. al., 2021; Dewi & Muniandy, 2014; Salza, et. al., 2019). However, most research still focuses on the specific application of Agile methods to computer or information systems-related courses (Alfonso & Botia, 2005; Fox and Patterson, 2017; Guercio & Sharif, 2012; Layman et al., 2006; Razmov & Anderson, 2014; Reed, 2008).

In recent years, a few studies have surfaced on the use of Agile methods in general education or teaching environments other than in areas of computer science or information systems, (Krehbiel et al., 2017). The limited number of more recent studies, however, generally focus on describing Agile principles but lack the specific strategies and methods to implement those principles in the classroom (Fitsilis, et al., 2023). Other research studies simply evaluate the awareness of the Agile approach (Janos, et al., 2024).

One recent study proposed the benefits of the synergistic effects of Agile teaching strategies, along with a customized learning focus within AI courses that uses collaboration, adaptability, flexibility, and continuous improvement. However, the researchers made no mention of specific methods nor strategies to accomplish the actual implementation of Agile in the classroom (Adenubi, et al., 2004).

Statistical or Quantitative studies that actually measure the effectiveness of Agile teaching are almost non-existent, despite efforts to do so (Pócsová, et al.,2020). One inherent difficulty in developing a quantitative comparison of the outcomes of an Agile course with those of a non-Agile course would be the question of which aspects of the agile principles would be denied to the control group? For example, what level of ongoing feedback to students would be implemented or limited in the control group?

In summary, the use of Agile methods in general education and college curricula has only been observed to a very limited extent. The current research urges the expansion of Agile principles across disciplines and academic curricula.

Active and Cooperative Learning

Within the context of Agile pedagogical principles, we have focused on the implementation of Active and Cooperative learning methods. We see these learning strategies as natural methods, and avenues for applying and benefiting from Agile practices. Traditional lecture-based content delivery does not easily lend itself to the application of Agile principles.

It has been proposed that, to improve learning outcomes (i.e., for learning to take place), students must actively participate in the knowledge-transfer process (Bonwell & Eison, 1991). In other words, the traditional lecture-based format, where students sit quietly and passively take notes, is far outdated.

The passive focus of the traditional college lecture has been replaced by more Active Learning techniques, such as discussions, group problem-solving, reading, writing, evaluating, comparing, discussing, and debating (Chickering & Gamson, 1987). Active Learning and active involvement in the learning process means that students are participating in higher-level thinking through analysis, evaluation, and synthesis (Bonwell & Eison, 1991).

When students are actively engaged in a course, student learning and information retention are more likely to be the result (Tinto, 1993). One overview of several studies points to Active Learning techniques as having a major impact on student learning. In this overview study, it is suggested that student achievement in course activities that incorporate Active Learning is comparable to lectures, in terms of mastery of content. However, the same study found that Active Learning is superior to the lecture format when it comes to promoting student thinking skills (Bonwell & Eison, 1991).

In a seminal work, researchers determined that specific indicators of student engagement influenced knowledge transfer, the capability of students to apply the material, and the overall experience of the student. Several of these factors are determined by the instructor (Chickering & Gamson, 1987):

- Encouraging cooperation among students
- Encouraging Active Learning
- Communicating high expectations
- Encouraging contact between students and faculty
- Using active learning techniques

Cooperative Learning is a subset of Active Learning where students actively participate in tasks as groups of three or more students (Paulson & Faust, 1998). Students can also be paired for certain classroom learning activities. This pairing of students is similar to the way that programmers are paired for problem solving in Agile software development. Past studies have suggested that teams succeed at problem solving more than individuals. This increased success at problem solving reinforces the idea that Cooperative Learning is more effective than individual learning (Sharan & Shlomo, 1990).

In order to incorporate cooperative learning methods into instruction, instructors must look at their role as more than simple transmitters of information and instead view themselves as guides and facilitators (Sharan & Sharan, 1994). In the Agile model, and in Active and Cooperative learning, feedback is an important component. Activities that include group participation has the added value of immediate evaluation by the instructor and instantaneous feedback to the student. In addition, research has shown a strong association between improved student learning and faculty-student interaction and contact (Umbach & Wawrzynski, 2005). According to such research, the greater the level of interaction between the student and the instructor, the greater the likelihood of student engagement and success.

Mapping Agile Principles to Pedagogical Strategies and Activities

The Agile Manifesto identifies the following values:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan (Agile Manifesto, 2001)

In our 2009 research paper, we proposed a set of values for Agile Pedagogy, which applied the Agile Manifesto as a template for teaching strategies (Stewart et al., 2009). Since that time, academia has progressively learned to value the interaction between students and instructors over a specific approach or method of teaching and learning. Academia has also valued working knowledge over rote memorization, and communication over negotiation. Finally, the trends in academia increasingly favor being receptive and responsive to changes, rather than adhering to a specific schedule.

In the following sections, we offer an update to our prior research, and we modify the principles and values that we initially proposed:

> **Learning-centric processes over traditional processes and tools.** Many institutions of higher learning advertise and stress a student-centric learning environment. We believe that learning environments should be neither instructor-centric nor student-centric. Traditional lecture-driven environments are instructor focused, which lack any interaction between the instructor and the student. The student-centric approach implies that students are in charge. In a pure student-centric environment, students can submit late assignments, expect adaptation of requirements to their schedule, and expect certain grading considerations, etc. Learning-centric environments focus on just one thing: learning. In a learning-centric environment, the instructor uses Active and Cooperative Learning techniques. When such techniques are used, students are motivated and attentive to the course requirements and deadlines. In addition, course requirements and deadlines are flexible within a framework focused solely on learning. Exploration of learning content is encouraged, as students take an active role in the learning process, both individually and as members of a group. Finally, flexibility, on the part of both the instructor and the students, helps to address the uncertainty associated with a dynamic learning environment.
>
> **Working projects over comprehensive documentation.** In project-based courses (or courses with semester long deliverables), projects are due near the end of the semester and are often presented during the last class meeting. Without an ongoing evaluation of progress, students will typically wait until the "last minute" to begin the project. In a learning-centric environment, the use of an iterative and progressive deliverable structure (combined with ongoing feedback from the instructor) allows for a greater level of immersion and depth to a course component. Ultimately, this approach leads to increased student learning and better learning outcomes.
>
> **Student and instructor collaboration over rigid course syllabi.** In a learning-centric environment the focus is on what the student is doing and what pedagogical methods are working to facilitate learning. In most courses, the syllabus is outlined as the contract between the student and the instructor. The syllabus lists expectations and specifies the requirements for deliverables. Instructors often attempt to create an "iron-clad" syllabus to prevent all manner of student "workarounds." This approach to syllabus development is often used instead of trusting that most students are motivated learners. A more flexible relationship, with greater access to the instructor, can lead to a more collaborative relationship between the instructor and the student. For example, in a machine-learning course, access to the instructor in the physical classroom (or via online class platform and/or via e-mail) is an Agile approach to Active Learning. In such an example, the instructor uses the increased access to offer suggestions to the student, and to ask probing questions in an iterative way about the student's approach. Similarly, in a writing course, iterative updates and revisions to a paper or written work allow the student to address their shortcomings, and helps the student to learn from incremental improvements in their work.

> **Responding to feedback rather than following a plan.** Often instructors outline the course to the students with specific topics to cover on specific days with fixed due dates. In addition, instructors have a standard method of delivering the material and are often unwilling or incapable of adapting the delivery methods to a specific class dynamic. Each time a course is delivered, the class dynamic is different, much to the surprise of new instructors. Keeping to a rigid timetable can leave students behind and require them to rush to catch up. With such a rigid schedule, learning outcomes can be diminished. It can be argued that it is far better to progress at a pace where students are able to manage the material and the deliverables, as well as comprehend and apply the course concepts. Such a flexible pace can be more conducive to learning than trying to adhere to the planned syllabus outline. Therefore, each iteration of the course may require the adjustment of delivery methods and timing of assignments to facilitate learning outcomes.

Agility is about adapting, within reason, to diverse learning styles and to the class dynamic. To be Agile, the instructor must be willing to "change course" if the current methods are not producing the desired results. In such an environment, interaction and feedback are paramount. Students need to know if they are on the right trajectory, and if they are submitting the desired deliverables. By contrast, instructors need to be cognizant of the effectiveness of their pedagogical approach.

Table 1: Mapping Agile Principles to the Classroom Environment (Adapted from Stewart et al., 2009)

Principles of the Agile Manifesto	Corollary to the Pedagogical Environment
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	Our highest priority is to prepare the student to make relevant and applicable contributions through continuous delivery of course components that engage and create competence. Students' highest priority is to make continuous and progressive proof of competence through ongoing delivery and improvement of course component deliverables.
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	The instructor and students welcome and adapt to changes to course requirements and deliverables even late in the semester. Agile pedagogical methods leverage problems and change as an opportunity to facilitate learning and better develop problem solving and other marketable skills in the students.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	Requiring working deliverables from the students over short periods of time, allowing for frequent feedback. Provide assignments conducive to guided problem solving, and guided experimentation.
Businesspeople and developers must work together daily throughout the project.	There is iterative interaction between the instructor and students (or student groups) during each iteration of course components.

Principles of the Agile Manifesto	Corollary to the Pedagogical Environment
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	Trust that most students are motivated. Give them the environment and support necessary for them to be successful.
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	To the extent possible, allow for direct face-to-face interaction with students or student groups. If class time and structure allow, offer individual attention to each student or group.
Working software is the primary measure of progress.	Working deliverables (i.e., models, software, project deliverables, presentations, research papers, etc.) are the primary measure of student progress (not necessarily midterm & final exams that require rote learning and memorization).
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	Guided problem solving and guided experimentation, where students actively seek guidance and tools to solve problems, and learn to work independently (or in groups) at a constant rate is the basis for teaching the skills needed for life-long learning.
Continuous attention to technical excellence and good design enhances agility.	Continuous attention to technical excellence and good design enhances learning and competence.
Simplicity--the art of maximizing the amount of work not done is essential.	While in education there is some value in exploring subjects in depth because there is student interest, understanding the problem, and solving it as simply and clearly as possible enhances student confidence and capability.
The best architectures, requirements, and designs emerge from self-organizing teams.	Student groups and teams should self-organize, but all should participate equally in the effort with the premise and objective that collaboration and cooperation among group members is synergistic: the group will accomplish more as a whole than group members individually.
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	At regular intervals, the students, student groups, and instructor reflect and offer feedback on how to be more effective. All stakeholders then adjust accordingly with the goal of being more effective.

Conclusion and Discussion

Enormous opportunities exist for the application of Agile software development principles to pedagogical methods in higher education. Despite the general focus of applying Agile teaching methods to college-level courses, many such initiatives have focused on project-oriented or software development courses. In the current research, we argue that these same Agile principles can be applied much more broadly within the pedagogical environment.

Despite the absence of statistical support for Agile, vast opportunities exist to apply Agile software development principles to teaching methodologies, particularly in the areas of Active and Cooperative Learning. In any case, it can be seen that methodologies that value the interaction between students and teachers, as opposed to more rigid approaches, are more favorable to learning outcomes. In this current work, we argue the need for a broader view of Agile capabilities and an adapting perspective in all areas of higher education pedagogy. Confining Agile teaching methods to project and software development courses continues to stifle the full capacity of Agile methodologies and limits its potential for markedly improved learning outcomes. Regardless of the assignment or the deliverable (i.e., papers, projects, programs, assignments, in-class activities, etc.), students can benefit from an Agile approach.

Agile teaching is about encouraging and addressing student questions, guided problem solving and guided experimentation. With Agile methods, the curriculum is project or assignment-based (both large and small), and the emphasis is on student learning experiences, while the student can be working collaboratively. Agile teaching is about working closely with students, and a willingness to be flexible and respond to the students' needs. With Agile teaching, there is direct involvement of the students in the learning process. Delivering the course material "agilely" requires being goal-driven, as opposed to plan-driven. As we have outlined in this research, plan-driven delivery is typically the standard with more conventional course design.

There are clear parallels between the utilization of Agile methods in teaching (i.e., pedagogical methods) and Active/Cooperative Learning techniques commonly found in the education literature. The current study proposes that each approach complements the other, and that using Agile delivery methods (combined with active learning activities) leads to improved learning outcomes and higher levels of student engagement and satisfaction.

Further research is needed, particularly in the area of quantifying the effectiveness of Agile teaching methods that have been outlined in Table 1. Despite the difficulties in doing so, measuring the learning outcomes from Agile teaching practices (compared to traditional methods) will need to be quantitatively explored in order to ensure wide acceptance of these teaching principles.

Finally, these principles, and their application to pedagogical considerations, is not confined to project-based or software development courses. It is our contention that instructors (as well as students) in all academic disciplines can benefit by applying Agile methods more broadly across all curricula. Agile teaching methods have broad application in all areas of academia. Ultimately, the application of such methods is only limited by the imagination of the instructor.

References

Adenubi, A. O., Samuel, S., Oyenuga, A. O., & Adewale, K. A. (2024). Artificial intelligence with agile teaching strategies for personalized learning. *Nigerian Online Journal of Educational Sciences and Technology (NOJEST)*, *6*(1), 364–379.

Alfonso, A., & Botía, J. A. (2005). An iterative and agile process model for teaching software engineering. In Proceedings of the 18th Conference on Software Engineering Education & Training (CSEET'05) (pp. 70–77). IEEE. https://doi.org/10.1109/CSEET.2005.37

Beck, K. et al. (2001). Manifesto for Agile Software Development. Agile Manifesto. http://agilemanifesto.org/

Bergin, J., Kussmaul, C., Reichlmayr, T., Caristi, J., & Pollice, G. (2005). Agile development in computer science education: Practices and prognosis. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '05)* (pp. 273–274). ACM. https://doi.org/10.1145/1047344.1047446

Boehm, B., Port, D., & Brown, A. W. (2002). Balancing plan-driven and agile methods in software engineering project courses. Computer Science Education, 12(3), 187–195. https://doi.org/10.1076/csed.12.3.187.8611

Bonwell, C. C., & Eison, J. A. (1991). *Active learning: Creating excitement in the classroom* (ERIC Digest No. ED336049). ERIC Clearinghouse on Higher Education, The George Washington University.

Chickering, A. W., & Gamson, Z. F. (1987). Seven principles for good practice in undergraduate education. *AAHE Bulletin, 39*(7), 3–7. ERIC Document No. ED282491.

Chun, D. (2004). The agile teaching/learning methodology and its e-learning platform. In W. Liu, Y. Shi, & Q. Li (Eds.), *Advances in web-based learning* (Lecture Notes in Computer Science, Vol. 3243, pp. 11–18). Springer. https://doi.org/10.1007/978-3-540-30195-9_3

Clarke, J. (1994). Pieces of the puzzle: The jigsaw method. In S. Sharan (Ed.), *Handbook of cooperative learning* (pp. 34–50). Greenwood Press.

Dewi, D. A., & Muniandy, M. (2014, September). The agility of agile methodology for teaching and learning activities. *In 2014 8th. Malaysian Software Engineering Conference (MySEC)* (pp. 255-259). IEEE.

Fitsilis, P., Damasiotis, V., & Boti, E. (2023). *Agile learning: An innovative curriculum for educators*. Qeios. https://doi.org/10.32388/RQX9T9

Fox, A., & Patterson, D. (2013). Is the new software engineering curriculum agile? *IEEE Software, 30*(5), 85–87. https://doi.org/10.1109/MS.2013.105

Guercio, A., & Sharif, B. (2012). Being agile in computer science classrooms. *AURCO Journal, 17*(1), 41–62.

Hislop, G., Ardis, M., Bagert, D., Ekstrom, J. J., Jorgensen, H., Lutz, M., McCauley, R., Naveda, J., & Rains, D. (2002). Integrating agile practices into software engineering courses. Computer Science Education, 12(3), 169–185. https://doi.org/10.1076/csed.12.3.169.8610

Kagan, S., & Kagan, M. (1994). The structural approach: Six keys to cooperative learning. In S. Sharan (Ed.), *Handbook of cooperative learning* (pp. 115–136). Greenwood Press.

Krehbiel, T. C., Salzarulo, P. A., Cosmah, M. L., Forren, J. P., Gannod, G. C., Havelka, D., Hulshult, A., & Merhout, J. W. (2017). Agile manifesto for teaching and learning. *The Journal of Effective Teaching, 17*(2), 90–111. https://files.eric.ed.gov/fulltext/EJ1157450.pdf

Layman, L., Cornwell, T., & Williams, L. (2006). Personality types, learning styles, and an agile approach to software engineering education. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '06)* (pp. 428–432). ACM. https://doi.org/10.1145/1121341.1121478

Niculescu, V., Suciu, D., and Bufnea, D. (2021). Agile principles applied in learning contexts. *In: Proceedings of the 3rd International Workshop on Education through Advanced Software Engineering and Artificial Intelligence (EASEAI 2021). Association for Computing Machinery*, New York, NY, USA, 31–38. https://doi.org/10.1145/3472673.3473963

Paulson, D. R., & Faust, J. L. (1998). Active learning for the college classroom. *Journal on Excellence in College Teaching, 9*(2), 3–24.

Pócsová, J., Bednárová, D., Bogdanovská, G., & Mojžišová, A. (2020). Implementation of agile methodologies in an engineering course. *Education Sciences, 10*(11), 333. https://doi.org/10.3390/educsci10110333

Razmov, V., & Anderson, R. (2006). Experiences with agile teaching in project-based courses. In *Proceedings of the 2006 ASEE Annual Conference and Exposition*. American Society for Engineering Education.

Reed, D. (2008). An agile classroom experience. In *Proceedings of the Agile 2008 Conference* (pp. 206–211). IEEE Computer Society. https://doi.org/10.1109/AGILE.2008.37

Salza, P., Musmarra, P., Ferrucci, F. (2019). Agile Methodologies in Education: A Review. *In: Parsons, D., MacCallum, K. (eds) Agile and Lean Concepts for Teaching and Learning*. Springer, Singapore. https://doi.org/10.1007/978-981-13-2751-3_2

Sharan, S., & Sharan, Y. (1994). Group investigation in the cooperative learning classroom. In S. Sharan (Ed.), *Handbook of cooperative learning methods* (pp. 97–114). Greenwood Press.

Sharan, S., & Shlomo, S. (1990). *Cooperative learning: Theory and research*. Praeger Publishers.

Simon, B., & Hanks, B. (2008). First year students' impressions of pair programming in CS1. *ACM Journal on Educational Resources in Computing (JERIC), 7*(4), Article 5. https://doi.org/10.1145/1348750.1348755

Stewart, J., DeCusatis, C., Kidder, K., Massi, J., & Anne, K. (2009). *Evaluating agile principles in active and cooperative learning*. Pace University. Retrieved from http://csis.pace.edu/~ctappert/srd2009/b3.pdf

Tinto, V. (1993). *Rethinking the causes and cures of student attrition* (2nd ed.). University of Chicago Press.

Umbach, P. D., & Wawrzynski, M. R. (2005). Faculty do matter: The role of college faculty in student learning and engagement. *Research in Higher Education, 46*(2), 153–184. https://doi.org/10.1007/s11162-004-1598-1

Varga, J., Csiszárik-Kocsir, Á., & Garai-Fodor, M. (2025). The development of modern-day competences in education, in the context of an agile approach. Frontiers in Education, 9. https://doi.org/10.3389/feduc.2024.1485273