ISSN: 2473-3857
v2 n4019

Agile Learning: Sprinting Through the Semester

Guido Lang
guido.lang@quinnipiac.edu
Quinnipiac University
Hamden, CT 06518

Abstract

This paper introduces agile learning, a novel pedagogical approach that applies the processes and principles of agile software development to the context of learning. Agile learning is characterized by short project cycles, called sprints, in which a usable deliverable is fully planned, designed, built, tested, reviewed, and launched. An undergraduate elective Computer Information Systems course on web development was redesigned to implement a semester-long agile learning experience. Results of a student survey conducted at the end of the semester reveal that agile learning combines learning and application of learning, while allowing students to fail more and fail faster. At the same time, agile learning takes longer than traditional project-based learning and makes it easier for students to fall behind. Nevertheless, students indicated a strong preference for agile learning over traditional project-based learning. Importantly, students' preference for and performance in agile learning was not influenced by their learning style. However, agile learning requires significant amount of planning, balancing the need to provide instructions with the need to provide explanations, as well as significant amount of one-on-one student support.

Keywords: agile learning, pedagogy, learning style

1. INTRODUCTION

The proliferation of massive open online courses (MOOCs) with the goal of "learn how to code" has spurred the development of innovative pedagogical approaches that have the potential to disrupt Information Systems (IS) education, as well as higher education in general (Drachsler & Kalz, 2016; Fox, 2016). For example, popular MOOCs offered by Code Academy (https://www.codecademy.com/), Treehouse (https://teamtreehouse.com/), and One Month (https://onemonth.com/) teach various aspects of coding by guiding students through the iterative development of multiple increasingly sophisticated software applications.

I term this pedagogical approach "agile learning." Agile learning applies the processes and principles of agile software development to the context of learning. It is characterized by short project cycles, called "sprints," in which a usable deliverable is fully planned, designed, built, tested, reviewed, and launched. Through several sprints, students iteratively expand and improve the deliverables. Agile learning stands in contrast to traditional project-based learning, which is often characterized by a linear process through which students develop deliverables (Lee, Huh, & Reigeluth, 2015; Melles et al., 2015).

The present work reports the results of a first implementation of agile learning in the context of undergraduate Computer Information Systems (CIS) education. In particular, this work addresses the following research questions:

(1) What are the advantages and disadvantages of agile learning, as perceived by students?

(2) Do students prefer agile learning to traditional project-based learning?

(3) Does learning style affect students' preference for and performance in agile learning?

ISSN: 2473-3857
v2 n4019

(4) What are the challenges of designing and implementing an agile learning experience?

The following sections describe agile learning, the methodology, the results, as well as the contributions and limitations of this research.

2. AGILE LEARNING

Traditional project-based learning is often implemented in a linear process that begins with theoretical lectures before asking students to plan, design, build, test, review, and ultimately launch a useable deliverable (Lee, Huh, & Reigeluth, 2015; Melles et al., 2015). Similar activities are part of nearly all student projects – such as an English paper, a financial report, or a marketing presentation. Interestingly, traditional project-based learning was popularized in the early 2000s, a time when the traditional "waterfall" systems development methodology was prevailing (Condliffe et al., 2015; Matkovic & Tumbas, 2010). Just like project-based learning, traditional systems development involves executing the aforementioned activities in a linear fashion. Figure 1 depicts the traditional project-based learning process.

Figure 1: Traditional Project-Based Learning Process

I propose the term agile learning to refer to the application of the processes and principles of agile software development to the context of learning. Agile software development is characterized by short development cycles, called sprints, in which a working software application is fully planned, designed, built, tested, reviewed, and launched (Anand & Dinakaran, 2016; Matharu et al., 2015). Through several sprints, developers iteratively expand and improve the software application. In the context of learning, development cycles and working software applications are replaced by project cycles and useable deliverables, respectively. In other words, an agile learning experience consists of multiple short project cycles, called sprints, in which a useable deliverable is fully planned, designed, built, tested, reviewed, and launched. One of the defining features of agile software development – and by extension agile learning – is the fact that each sprint ends with a useable deliverable that is increasingly being expanded and improved upon. Although originally introduced in the early 2000s, agile software development only became widely adopted in the last few years (Anand & Dinakaran, 2016). Figure 2 depicts the agile learning process.

Figure 2: Agile Learning Process

In addition to the above-mentioned processes, agile learning also applies the four principles of agile software development to the context of learning (Jørgensen et al., 2015; Bustard & Keenan, 2009). The four principles of agile software development were first stated in the "Manifesto for Agile Software Development" (Beck et al., 2001), which was drafted by 17 leading software development experts that recognized the need for an alternative to documentation-driven, heavyweight software development processes.

The first agile principle is "individuals and interactions over processes and tools." Applied to the context of learning, it suggests for the instructor to focus on working with students one-on-one and to be flexible in adjusting the processes and tools used in the classroom. The second agile principle is "working software over comprehensive documentation," which suggests shifting the focus from students writing reports to students producing something that can be used in a professional environment. The third agile principle is "customer collaboration over contract negotiation." Applied to learning, it suggests for the instructor to collaborate with students instead of strictly enforcing assignments and associated rules. Lastly, the fourth agile principle is "responding to change over following a plan," which further emphasizes the need for the instructor to be willing to depart from the traditional semester-long course schedule and instead to adjust the schedule in response to students' needs as they arise. The goal of the agile learning principles is to improve the instructor's ability to facilitate learning in an agile learning experience.

3. METHODOLOGY

Course and Content Development

An undergraduate elective Computer Information Systems (CIS) course on Web Development (CIS 381) at Quinnipiac University was completely redesigned to implement a semester-long agile learning experience. CIS

ISSN: 2473-3857
v2 n4019

381 guided students through the process of building web applications from idea to deployment, placing an equal emphasis on front and back end aspects of web development. Student learning objectives of the course included:

- Evaluate and justify choices in design patterns and technologies used in web development
- Explain and configure the fundamental structure of a web application
- Implement responsive design in a web application frontend using Bootstrap
- Develop a secure web application backend using Django/Python
- Understand and implement the basic principles of web services from the perspective of both the client and service provider

Students developed web applications that adhere to industry best practices and leverage professional tools, such as Django (web application framework), Github (version control system), Bootstrap (front end library), Nitrous (cloud-based IDE), and Heroku (deployment platform). The author of this work was the instructor for this course in Fall 2015 (N = 37).

The semester was divided into four sprints of increasing length (i.e. 1 week for sprint 1, 2 weeks for sprint 2, 3 weeks for sprint 3, and 4 weeks for sprint 4). Each sprint consisted of a web development project that required students to repeat and add to the work conducted in the previous sprint. For example, in sprint 1, students developed a simple splash page. In sprint 2, students developed a more advanced single-page website. In order to complete sprint 2, students had to repeat most of the steps from sprint 1 before being introduced to new content. The instructions in sprint 2 asked students to identify and repeat the necessary steps from sprint 1 on their own, before giving them step-by-step instructions for the new aspects of sprint 2. This continued until sprint 4, when students were asked to develop a complex web application with few instructions, thus requiring them to apply their learning from the previous sprints.

With each sprint, students were given increasing creative freedom over the actual content of their web development project. For example, while the first sprint required all students to implement the same project, the last sprint specified only functional requirements and gave students full control over the content domain. Requirements of sprint 4 included "the web app shall include user account management," "the web app shall allow users to view, add, edit, and delete objects and related objects," and "the web app shall include a search function." At the end of sprint 4, students had developed different web apps featuring e.g. restaurant reviews, college sports forums, and travel logs. Holding the functional requirements constant across all students reduced the complexity of potential technical problems and thus allowed the instructor to assist each student throughout the sprints (without the help of a teaching assistant). Table 1 provides an overview of the four sprints.

Sprint	Duration	Project
1	1 week	Splash page
2	2 weeks	Landing page
3	3 weeks	Web app
4	4 weeks	Final project

Table 1: Overview of the Sprints

The course also applied the agile learning principles. For example, students were provided instructions through video tutorials (that were recorded by the instructor), which gave the instructor time to respond to students' questions and work with them one-on-one. Students developed working websites of increasing sophistication, using professional tools, and industry best practices, as needed. The above-mentioned agile learning process was used as a guideline and not as a strict process – thus allowing the instructor to adjust the pace and deliverables to students' needs.

Data Collection and Analysis

At the end of the semester, students completed a survey, which measured perceived advantages and disadvantages of agile learning, preference for agile learning over traditional project-based learning, and learning style. The survey included definitions of agile learning and traditional project-based learning, thus allowing students to draw on their personal experience when comparing the two pedagogical approaches. A total of N_{Final} = 34 students completed the survey for extra credit (worth approximately 5% of their final grade), for a response rate of 92%. The open-ended questions were analyzed using qualitative content analysis (Mayring, 2000). Learning style was measured using the Learning Style Inventory (LSI; Kolb & Kolb, 2005), which asks participants to rank the endings of 12 sentences according to how well they think each

ISSN: 2473-3857
v2 n4019

one fits with how they would go about learning something. Detailed instructions on the LSI, including how to calculate the learning style dimensions, can be found in Kolb and Kolb (2005). The survey was not anonymous, thus allowing me to correlate students' responses with their performance in the course. The full survey instrument can be found in Appendix A.

The challenges regarding the design and implementation of the agile learning experience are the outcome of reflection-on-action performed by the instructor (Schön, 1983).

4. RESULTS

Advantages and Disadvantages

Four themes – two advantages and two disadvantages – emerged from the qualitative content analysis of students' responses to the open-ended questions. The first major advantage of agile learning, as perceived by the students, is that it combines learning and application of learning. By introducing new concepts, as they are needed, and immediately applying these concepts in practice, students are able to decrease the time lag between learning and the application of learning. As stated by one student: "I'm able to implement what I'm learning right away instead of waiting until I learn other material and then having to do everything at once."

The second major advantage of agile learning is that it allows students to fail more and fail faster. By going through multiple iterative projects, or sprints, students are able to recognize the shortcomings of their understanding more often and faster than in traditional project-based learning. One student observed: "I know exactly where my weak points are and can easily fix them because I know what portion or part I'm having trouble with." Likewise, another student stated "you can see your mistakes and areas that you can improve on while working on different projects."

The first major disadvantage is that agile learning takes longer than traditional project-based learning. As agile learning involves iteration and repetition, it is likely that traditional project-based learning conveys the same amount of learning material in a shorter amount of time. In line with this concern, one student remarked that "maybe it takes longer but that did not seem to be a problem here because each project led up to the big final project."

The second major disadvantage of agile learning is that it is easier for students to fall behind than in traditional project-based learning. Since students are working, hands-on, on projects in every single class, they are required to stay up-to-date – especially when they miss class. As one student put it, "[it] is very necessary to be on top of the work, it was very important to go to class and to follow along with the lessons and be able to ask questions." Similarly, another student noted that "if a student did not understand one concept taught early, they could fall behind. All of the concepts are built off of each other and if you miss one section you could end up very lost."

Preference for Agile Learning

The three items measuring students' preference for agile learning (i.e. "I prefer agile learning to project-based learning", "I believe agile learning helps me achieve my learning better than project-based learning", "I wish more classes would use agile learning") are highly correlated (all *r*s > .60, *p*s < .001), as is also evident in the aggregate responses shown in Figure 3.

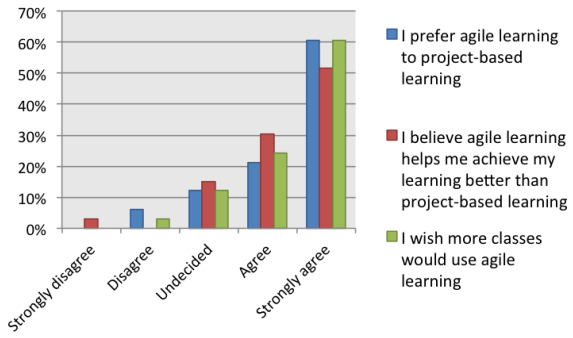

Figure 3: Preference for Agile Learning

Eighty-two percent of the participants agree or strongly agree with the statements "I prefer agile learning to project-based learning" (*M* = 4.30, *SD* = 1.10) and "I believe agile learning helps me achieve my learning better than project-based learning" (*M* = 4.27, *SD* = .94). Moreover, 85% of the participants agree or strongly agree with the statement "I wish more classes would use agile learning" (*M* = 4.42, *SD* = .83). Taken together, these responses indicate a strong preference for agile learning over project-based learning.

Influence of Learning Style

The students exhibited a diverging learning style, which is characterized by an emphasis of Concrete Experience (CE; *M* = 36.15, *SD* =

ISSN: 2473-3857
v2 n4019

4.85) over Abstract Conceptualization (AC; *M* = 28.15, *SD* = 5.44) and Reflective Observation (RO; *M* = 30.73, *SD* = 5.60) over Active Experimentation (AE; *M* = 24.97, *SD* = 5.03). The participants' aggregate learning style profile is shown in Figure 4.

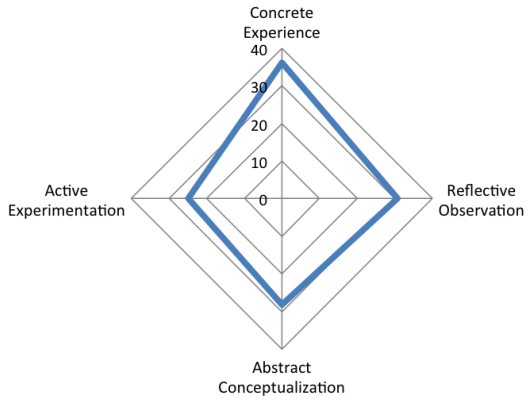

Figure 4: Learning Style Profile

Individuals with a diverging learning style are best at viewing concrete situations from many different points of view. They tend to perform better in situations that call for generation of ideas, such as brainstorming sessions (Kolb & Kolb, 2005). It is noteworthy that the diverging learning style is highly atypical of students in CIS/IS. As previous research has shown, the prevalent learning style among CIS/IS students is assimilating (Kolb & Kolb, 2005). Individuals with an assimilating learning style are best at understanding a wide range of information and putting it into concise, logical form. To better understand if and to what extent learning style might influence preference for and performance in agile learning, two multivariate regression analyses were performed.

The first multivariate regression analysis was used to test if the learning style dimensions (i.e. CE, AC, RO, AE) predict the preferences for agile learning (i.e. "I prefer agile learning to project-based learning", "I believe agile learning helps me achieve my learning better than project-based learning", "I wish more classes would use agile learning"). Results suggest that learning style does not affect preference for agile learning ($F(3, 29) = .66$, $p > .05$).

The second multivariate regression analysis was used to test if the learning style dimensions predict students' performance in the course (i.e. assignment grades, midterm grade, final grade). Results suggest that learning style does not affect performance in agile learning ($F(3, 29) = 2.15$, $p > .05$). Moreover, the student performance in this course (as measured by the assignment grades) suggests that the agile learning approach allowed students to achieve the stated learning objectives (*M* = 86.13%, *SD* = 21.43%). Taken together, these findings suggest that learning style, as measured by the LSI, does not influence preference for and performance in agile learning.

Challenges

Three challenges for the design and implementation of agile learning became apparent from the instructor's reflection-on-action: First, agile learning requires a significant amount of planning by the instructor. As each sprint repeats and builds upon the previous sprint, it is crucial that the projects are chosen and developed in a way that introduces increasingly complex concepts over time.

Second, agile learning requires balancing the need to provide students with step-by-step instructions on how to do something with the need to provide students with explanations on why to do something. As students are in the midst of a sprint, it is often easier to just give instructions on what to do next than to step back and explain why something needs to be done.

Third, agile learning requires significant amount of one-on-one student support from the instructor. Given that students work hands-on for almost the entire semester, many problems and questions arise that need to be addressed one-on-one with the instructor. Since this implementation of agile learning made extensive use of online videos, the instructor was able to address most of the problems and questions in class.

5. CONTRIBUTIONS AND LIMITATIONS

The present work contributes to IS education in two ways: First, it introduces the concept of agile learning, which has hitherto not been explored in the IS and education literatures. This has the potential to improve our understanding of teaching and learning and lays the groundwork for future research in this area. Second, it implemented and evaluated agile learning in an undergraduate CIS course. This, in turn, has the potential to improve the practice of teaching and learning in IS and beyond.

However, the present work is not without limitations. First, the design and implementation

ISSN: 2473-3857
v2 n4019

of the agile learning experience did not follow previously established guidelines. As such, it is possible that one could have designed and implemented a purer agile learning experience and thus conducted a better test of the viability of agile learning in IS education. Second, the quantitative and qualitative results must be seen in light of the relatively small sample size and students exhibiting a learning style that is unusual of CIS/IS students. Future research is clearly needed to replicate and deepen the insights derived from this work.

6. CONCLUSION

The present work introduces agile learning. Agile learning is a novel pedagogical approach that applies the processes and principles of agile software development to the context of learning. Agile learning was implemented and subsequently evaluated in an undergraduate CIS course. Results of a student survey suggest that agile learning combines learning and application of learning, while allowing students to fail more and fail faster. At the same time, agile learning takes longer than traditional project-based learning and makes it easier for students to fall behind. Nevertheless, students indicated a strong preference for agile learning over traditional project-based learning. Importantly, students' preference for and performance in agile learning was not influenced by their learning style, as measured by the LSI. From the instructor's point of view, agile learning requires significant amount of planning, balancing the need to provide instructions with the need to provide explanations, as well as significant amount of one-on-one student support. This work opens avenues for future research on the potential of agile learning in IS education and beyond.

7. REFERENCES

Anand, R., & Dinakaran, M. (2016). Popular Agile Methods in Software Development: Review and Analysis. *International Journal of Applied Engineering Research*, 11(5), 3433-3437.

Beck, K. et al. (2001). Agile Manifesto for Software Development. Retrieved 5/18/2016 from http://agilemanifesto.org

Bustard, D., & Keenan, F. (2009). Soft Systems Methodology: An Aid to Agile Development. In Barry, C., Conboy, K., Lang, M., Wojtkowski, G., Wojtkowski, W. (Eds.), *Information Systems Development: Challenges in Practice, Theory, and Education*, New York: Springer, 25-38.

Condliffe, B., Visher, M. G., Bangser, M. R., Drohojowska, S. & Saco, L. (2015). Project-Based Learning: A Literature Review. MDRC. Retrieved 5/18/2016 from https://s3-us-west-1.amazonaws.com/ler/MDRC+PBL+Literature+Review.pdf

Drachsler, H., & Kalz, M. (2016). The MOOC and Learning Analytics Innovation Cycle (MOLAC): A Reflective Summary of Ongoing Research and Its Challenges. *Journal of Computer Assisted Learning*, 32, 281-290.

Fox, R. (2016). MOOC Impact Beyond Innovation. In: Ng, Chi-hung Clarence, Fox, Robert, Nakano, Michiko (Eds.), *Reforming Learning and Teaching in Asia-Pacific Universities* (Education in the Asia-Pacific Region: Issues, Concerns and Prospects, Volume 33), Berlin: Springer, 159-172.

Jørgensen, M. T. N., Hovmøller, H., Nielsen, J. R., & Tambo, T. (2015). Improving offshoring of Low-Budget Agile Software Development Using the Dual-Shore Approach: An Autoethnographic Study. *Proceedings of 36th Information Systems Research in Scandinavia Seminar*, 2-17.

Kolb, A. Y., & Kolb, D. A. (2005). The Kolb Learning Style Inventory—Version 3.1 2005 Technical Specifications. Department of Organizational Behavior, Weatherhead School of Management, Case Western Reserve University, Cleveland, OH,

Lee, D., Huh, Y. & Regeluth, C. (2015). Collaboration, Intragroup Conflict, and Social Skills in Project-Based Learning. *Instructional Science*, 43(5), 561-590.

Matharu, G., Mishra, A., Singh, H., Upadhyay, P. (2015). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *ACM SIGSOFT Software Engineering Notes*, 40(1), 1-6.

Matkovic, P., & Tumbas, P. (2010). A Comparative Overview of the Evolution of Software Development Models. *Journal of Industrial Engineering and Management*, 1(4), 163-172.

Mayring, P. (2000). Qualitative Content Analysis. *Qualitative Social Research*, 1(2), Art. 20, Retrieved 5/18/2016 from http://nbn-resolving.de/urn:nbn:de:0114-fqs0002204.

ISSN: 2473-3857
v2 n4019

Melles, G., Anderson, N., Barrett, T., Thompson-Whiteside, S. (2015). Problem Finding through Design Thinking in Education. In Patrick Blessinger, John M. Carfora (Eds.), *Inquiry-Based Learning for Multidisciplinary Programs: A Conceptual and Practical Resource for Educators* (Innovations in Higher Education Teaching and Learning, Volume 3), Bingley, UK: Emerald, 191-209.

Schön, D. (1983) *The Reflective Practitioner. How Professionals Think in Action*. London: Temple Smith.

ISSN: 2473-3857
v2 n4019

Appendix A: Survey Instrument

At the beginning of the survey, students were provided the following introduction to agile learning:

"This course used a novel pedagogical approach called agile learning. Agile learning contrasts traditional project-based learning, which is often implemented in a linear process that begins with theoretical lectures before asking students to plan, design, build, test, review, and ultimately launch a useable deliverable. An agile learning experience consists of multiple short project cycles, called sprints, in which a useable deliverable is fully planned, designed, built, tested, reviewed, and launched. Over the course of the semester, you completed four sprints: the splash page, the landing page, the web app, and the final project."

Advantages and Disadvantages:
The following were open-ended questions.

What would you say are the advantages of agile learning (compared to project-based learning)?

What would you say are the disadvantages of agile learning (compared to project-based learning)?

Preference for Agile Learning:
The following items were answered using a 5-point Likert scale, labeled 1 – Strongly disagree; 2 – Disagree; 3 – Undecided; 4 – Agree; 5 – Strongly agree.

I prefer agile learning to project-based learning.

I believe agile learning helps me achieve my learning better than project-based learning.

I wish more classes would use agile learning.

Kolb Learning Style Inventory:

(See Smith & Kolb, 1985)