





Using a Low Code Development Environment to Teach the Agile Methodology

Mary Lebens^(✉)  and Roger Finnegan 

Metropolitan State University, Saint Paul, MN 55106, USA
{mary.lebens, roger.finnegan}@metrostate.edu

Abstract. The Agile development methodology is soaring in popularity in the business world. Companies are turning to Agile to develop products quickly and to achieve digital transformation of their organization. Because of this push, companies need employees who understand Agile. Therefore, higher education is obligated to provide an understanding of Agile to students as they enter the workplace. Providing Agile experience to students who are new to programming is difficult because they are so worried about the coding aspects of the assignment, they cannot take time to think about the methodology they are using. The coding crowds out the time needed to get an understanding of how Agile actually works. One remedy for this is to use a low or no-code development platform. With this type of platform students spend less time learning to create apps, freeing them to experience the rituals and roles of Agile. This study examines using the Agile methodology along with the Microsoft Power Apps platform to provide an Agile experience to students. Two course sections were surveyed to learn if students perceived that they acquired a better understanding of Agile and to learn their perceptions of a no-code platform experience. The students completed surveys to ascertain their comfort with the Agile methodology and whether the no-code environment increased their comfort level. The results showed students perceived the no-code platform increased their comfort with using the Agile methodology. The implication is that no-code platforms can be used broadly to help students to gain experience with Agile.

Keywords: Agile methodology · No-code · Experiential learning

1 Introduction

Although the Agile software development methodology is now ubiquitous in the business world, it is not common in beginning software development courses due to the difficulty in teaching Agile. Agile is best learned in the context of a software development project. However, students who are new to programming often need the entire semester to learn the programming structures and syntax, leaving no time for completing an Agile project. In addition, the normal challenges of a group project are still present in Agile student projects, such as getting students to work effectively together and avoiding free riders who do not complete their fair share of the work.

© The Author(s) 2021

P. Gregory et al. (Eds.): XP 2021, LNBP 419, pp. 191–199, 2021.

https://doi.org/10.1007/978-3-030-78098-2_12

One approach to freeing up class time to teach Agile is using a low- or no-code development environment in place of teaching a traditional programming language. Low- and no-code environments allow students to develop applications through a Graphical User Interface (GUI), reducing time spent on coding. This paper examines using the Agile software development methodology in conjunction with a no-code environment in a beginning application development course. Two course sections of students were surveyed to learn if they perceived that they acquired a deeper understanding of Agile, as well as to learn their perceptions of the no-code platform. Both course sections were taught in an online, asynchronous format.

2 Related Work

Although the *Agile Manifesto* [1] was written over twenty years ago, few University courses cover more than a mention of Agile and its general characteristics [2]. Part of the difficulty in teaching Agile is that it is best taught through student projects, but the complexity of projects and the lack of sufficient time during the academic calendar make this difficult [3]. In addition, textbooks provide only cursory coverage of the Agile methodology, exacerbating the time crunch by requiring professors to spend more class time discussing Agile in order to make up this deficit [4].

Despite the difficulty of teaching Agile, there are proven benefits to using Agile practices with students [5]. Agile projects featuring feedback after each iteration increase students' software development skills more than traditional group projects [6]. Students find Agile practices such as pair programming beneficial [7]. College students in asynchronous online courses reported that using Agile led to a more effective learning experience and allowed them to produce deliverables of a higher quality [8]. Additionally, using small teams and timeboxed iterations overcomes some of the hurdles of traditional group projects, like free-riding students who do not complete their fair share of the work [9].

Due to the difficulty in teaching students coding along with Agile practices in a single course, some college faculty have tried alternative no-code approaches, such as assigning students to build a wiki, solve an IT business case, or construct a building using Legos [10–12]. In addition to these no-code approaches in college courses, the no-code environment Scratch has been used to successfully teach middle school students Agile practices [13]. No-code software development environments are not only becoming more popular for teaching, but for use by business professionals as well, with Microsoft PowerApps as one of the leading platforms [14]. Combining a no-code environment with Agile practices provides benefits students and teaches skills that employers desire.

2.1 Microsoft PowerApps

The low-code development platform Microsoft PowerApps was used during this research. PowerApps was chosen in part because it is freely available to students. PowerApps is part of the Office 365 suite used by Metropolitan State University. The University supplies students with Office 365 free of charge. PowerApps is a subscription-based platform to create applications. It is used to create browser-based apps that connect to a data

source. Students can create screens to view and edit data for business processes [15]. Gartner has recognized PowerApps as a leader in enterprise low-code platforms [16].

2.2 Course Background

The course targeted for this study is MIS 328-Applications Development I, an introductory software development course that is part of the MIS major at Metropolitan State University. The course curriculum assumes that the students do not have any previous programming experience. The learning outcomes include students experiencing the process, tools and methodologies used to create and revise computer applications. In addition, the students experience the existing and future paradigms of application development with the Agile framework currently being the most important paradigm. The course is at the junior level and is required for all MIS majors. Due to its historic reliance on teaching a programming language, such as C#, Visual Basic, or even Cobol in an early iteration, it carried the reputation of being the hardest course in the program. Many students avoided taking the course until their last year or even their last semester.

The course was redesigned to give students the experience of being on an Agile development team while reducing the stress of learning a development language. That is, the development language was deemphasized in order to allow the students the time to better learn the Agile methodology. It was surmised that the gain that the students achieved with the Agile experience would more than offset the reduction in the experience with a programming language. This gain in Agile methodology knowledge is expected to help students in their job search.

The course, MIS 328, is Application Development I. As the name implies it was developed to teach students how applications are created in organizations. In years past the waterfall methodology would have been emphasized, but since the Agile framework is becoming more and more important in organizations [17] it is now being taught. The course includes the opportunity to write user stories, create prototypes, develop a product, participate in sprint planning meetings, participate in standup meetings, and interact with a development team.

2.3 Hypothesis

As a result of reviewing the literature and related work, the following hypothesis was developed.

- H_0 : Students do not perceive they have a better understanding of the Agile methodology after using a low-code development environment to experience participating in an Agile team.
- H_1 : Students perceive they have a better understanding of the Agile methodology after using a low-code development environment to experience participating in an Agile team.

3 Research Methods

3.1 Participants

To gauge the students' reactions to the no-code platform teaching method for Agile we conducted a survey across two sections in the fall semester of 2020. The students involved were undergraduates majoring in Management Information Systems (MIS). Fifty-seven students participated in the survey across the two sections.

3.2 Materials

The goal of the survey was to determine how the students felt about their familiarity and comfort with the Agile methodology after using it as part of a no-code development team. The specific things that the authors wanted to learn from the survey were the students' comfort level with the Agile framework and if our reliance on a no-code platform was the correct decision. The authors realized that students would not have any firsthand experience with the way in which the course had been previously taught with its use of a programming language but were confident that students would be able to gauge if they would be overwhelmed as they needed to learn both the Agile framework and a new programming language at the same time. The survey instrument used the following questions:

1. I feel that I have an understanding of the Agile software development methodology.
2. I feel that I have more comfort with the Agile methodology after experiencing it in MIS 328.
3. Going through the development process with Power Apps helped me better understand the Agile methodology.
4. It would have been harder to learn the Agile methodology if I had also been required to learn how to write computer code at the same time.
5. I feel that I will be a more effective employee for companies with my new knowledge of the Agile methodology.

The possible responses were recorded on a Likert scale:

- Strongly agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree

3.3 Procedure

The survey was administered at the end of the course. At this point in the course the students had completed the second sprint in their development team final project. It was felt that they would have sufficient experience with the Agile methodology at this point in the course to express an opinion. The students were given extra credit points as

an incentive to complete the survey. It was administered anonymously through Survey Monkey. Survey Monkey was chosen to get more candid answers than might have been possible using the learning management system.

4 Results

The results of the survey strongly demonstrated that the students felt that learning Agile by using a no-code platform increased their understanding of the methodology. One hundred percent of students strongly agreed or agreed that they understood the Agile methodology. This affirmed hypothesis H_1 that students will perceive they have a better understanding of the Agile methodology after using a low-code development environment to experience participating in an Agile team. Agile has been described as a culture [18]. Giving the students the ability to experience that culture as part of their learning was an important part of the planned experience.

As shown in Fig. 1, ninety-eight percent of students strongly agreed or agreed that they were comfortable with using the Agile methodology. This helped with the goal of increasing their comfort with Agile in order to better prepare the students for careers with companies that are making strong use of the method. This comfort level is expected to be very important to student's career success [19].

Ninety-six percent of students strongly agreed or agreed that using the Power Apps no-code platform helped them get a better understanding of the development platform, as shown in Fig. 1. Power Apps was selected as the platform since it is already part of the Office 365 suite. The University supplies the students access to Office 365 so there was no additional cost to the students to access the platform. The Mendix platform had been used in previous semesters but the migration to Power Apps was seen as advantageous to the students' careers since it is more popular in industry. Power Apps is also seen as becoming more important by Gartner [20].

Eighty-four percent of the students strongly agreed or agreed that learning the Agile methodology would have been more difficult if they also had been required to learn to write computer code, as shown in Fig. 1. This corroborated the authors assumption that having to learn a programming language and the Agile methodology at that same time would be difficult and stressful for students. The course was already known for its difficulties and adding the additional requirements for Agile could have increased the apprehension regarding it. The need to teach Agile [21] outweighed the need to teach coding.

Figure 1 shows ninety-six percent of students strongly agreed or agreed that they will be better employees with their knowledge of the Agile methodology. Providing the students the skills that they need to be productive employees and competitive in the job market was the most important factor in the decision to emphasize the Agile method over programming skills. Companies that use the Agile framework have a 30 percent higher project success rate than companies using the traditional waterfall methodology [22].

The results of the survey gave the authors the sense as well as the metrics to show that their decision to use the no-code platform and emphasize the Agile methodology increased students' confidence with the methodology. Additionally, the survey results

showed students felt confident in their ability to use the knowledge to become valuable employees and be successful in the job market.

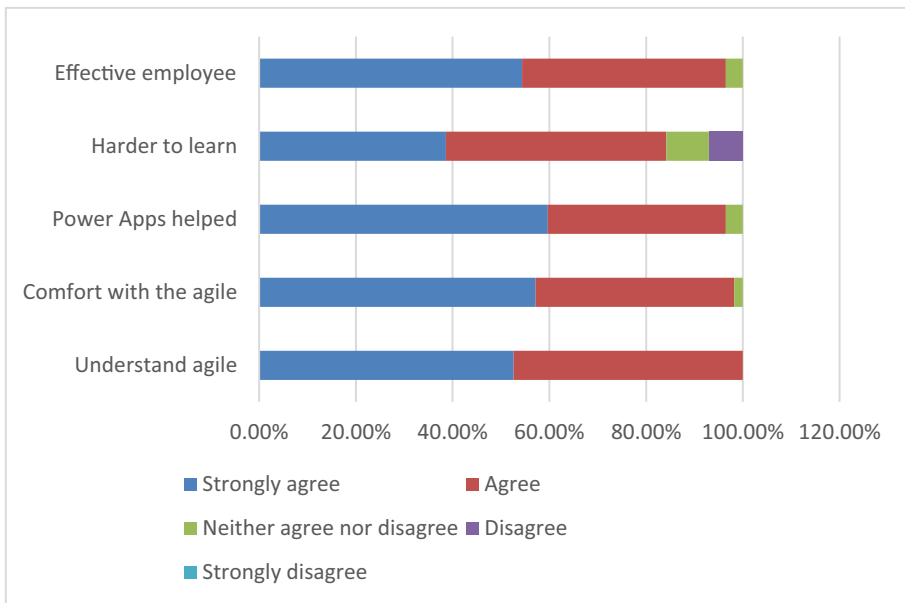


Fig. 1. Survey results for the five questions posed to the students.

5 Discussion

This study expands on previous research which found other types of no-code approaches were useful for teaching Agile, such as building a wiki, solving an IT business case, or constructing a building using Legos [10–12]. A key contribution of this study is that this course design not only teaches students Agile, but also teaches them software development using a popular and common no-code platform [14]. The combination of no-code with Agile allowed students in this study to gain valuable skills as a part of an Agile development team while simultaneously gaining experience with a leading application development platform, Microsoft PowerApps [14].

Another key contribution of this study is that it demonstrates a no-code environment allows students to learn Agile in a short timeframe. Using a no-code environment to teach the Agile methodology allowed students to grow their Agile skills and successfully develop fully functioning mobile apps in a single semester. This short timeframe is a boon for college professors who are seeking to teach students career-boosting Agile skills under the constraints of the academic calendar.

The implication of these findings for the Information Technology (IT) and Information Systems (IS) disciplines is that no-code platforms can be used more broadly to

allow students to gain experience with Agile, without the added stress and time requirements of learning a traditional programming language. This approach to teaching Agile methodology will be helpful for college programs which only require a single software development course, such as IT, IS, or networking, since the course time devoted to programming is extremely limited in that case. Additionally, this approach may be valuable for business programs, since this approach could also help non-technical students learn the Agile methodology and later apply it to projects in their careers.

5.1 Limitations and Further Research

A key limitation of this study is the sample was limited to two course sections, since the University only offers two sections of MIS 328. Further research on additional sections is needed to determine if the results hold true over a larger sample. In addition, further research is needed to determine if this approach works in programs outside of IS, particularly in more narrowly focused IT programs such as networking and in non-technical programs such as business. The two course sections used for this study were both held in an asynchronous online format, so additional research into the effectiveness of teaching a no-code or low-code environment in different types of formats is warranted. The asynchronous format does not include course meetings, so further research on face-to-face, hybrid, and synchronous online courses would be useful to determine if the students' perceptions remain the same across formats.

6 Conclusion

In this study, the authors examined using the Agile methodology in conjunction with a no-code environment to provide students with the experience of participating in an Agile development team. The results overwhelmingly showed students perceived the no-code platform increased their comfort with using the Agile methodology. The stressors and time constraints of teaching both application development and Agile in a single semester was overcome using the no-code development environment. The implication is that no-code platforms can be used broadly to allow students to gain experience with Agile across the IT and IS disciplines.

References

1. Agile Alliance: Agile manifesto and principles (2001). <http://agilemanifesto.org/principles.html>
2. Devedžić, V., Milenković, S.R.: Teaching agile software development: a case study. *IEEE Trans. Educ.* **54**(2), 273–278 (2011). <https://doi.org/10.1109/TE.2010.2052104>
3. Fitsilis, P., Lekatos, A.: Teaching software project management using agile paradigm. In: Proceedings of 21st Pan-Hellenic Conference on Informatics, Larissa, Greece, (2017). <https://doi.org/10.1145/3139367.3139413>
4. May, J.: Play ball: Bringing scrum into the classroom. *J. Inf. Syst. Educ.* **27**(2), 87–92 (2016)
5. Missiroli, M., Russo, D., Ciancarini, P.: Learning agile software development in high school: An investigation. In: Proceedings - International Conference on Software Engineering, pp. 293–302, ACM, Austin, Texas (2016). <https://doi.org/10.1145/2889160.2889180>

6. Yang, J., Zhang, X.L., Su, P.: Deep-learning-based agile teaching framework of software development courses in computer science education. *Procedia Comput. Sci.* **154**, 137–145 (2018). <https://doi.org/10.1016/j.procs.2019.06.021>
7. Monett, D.: Agile project-based teaching and learning. In: *Proceedings of the 11th International Conference on Software Engineering Research and Practice*, pp. 377–383. (2013)
8. Hulshult, A.R., Krehbiel, T.C.: Using eight agile practices in an online course to improve student learning and team project quality. *J. High. Educ. Theor. Pract.* **19**(3), 55–68 (2019). <https://doi.org/10.33423/jhetp.v19i3.2116>
9. Stapel, K., Lübke, D., Knauss, E.: Best practices in extreme programming course design. In: *Proceedings - International Conference on Software Engineering*, pp. 769–775 (2008). <https://doi.org/10.1145/1368088.1368197>
10. Cubric, M.: Agile learning & teaching with wikis: building a pattern. In: *Proceedings of the 2007 International Symposium on Wikis*, pp. 11–24 (2008). <https://doi.org/10.1145/1296951.1296953>
11. Rush, D.E., Connolly, A.J.: An agile framework for teaching with scrum in the IT project management classroom. *J. Inf. Syst.* **31**(3), 196–207 (2020)
12. JSteghöfer, J.P., Knauss, E., Alégroth, E., Hammouda, I., Burden, H., Ericsson, M.: teaching agile: addressing the conflict between project delivery and application of agile methods. In: *2016 IEEE/ACM 38th IEEE International Conference on Software Engineering Companion*, pp. 303–312 (2016). <https://doi.org/10.1145/2889160.2889181>
13. Fronza, I., El Ioini, N., Corral, L.: Teaching computational thinking using agile software engineering methods: a framework for middle schools. *ACM Trans. Comput. Educ.* **17**, 4 (2017). <https://doi.org/10.1145/3055258>
14. Rymer, J.R., Koplowitz, R.: *The Forrester Wave™: Low-code development Platforms for AD&D professionals*. Technical report, Forrester (2019)
15. Leung, T.: *Beginning PowerApps*. Apress, Berkeley (2017)
16. Vincent, P., et al.: *Magic Quadrant for Enterprise Low-Code Application Platforms* (2020). <https://www.gartner.com>
17. Brower, T.: Why agile is the mindset to get us through the Covid Crisis: 4 lessons from agile for today and the new normal. *Forbes* (2020). <https://www.forbes.com/sites/tracybrower/2020/04/12/why-agile-is-the-mindset-to-get-us-through-the-covid-crisis-4-lessons-from-agile-for-today-and-the-new-normal/?sh=4f350a3631d3>
18. Prieto, R.: The agile classroom: Embracing an agile mindset in education. *Medium* (2016). <https://medium.com/laboratoria/the-agile-classroom-embracing-an-agile-mindset-in-education-ae0f19e801f3>
19. Nikolova, Z.: 3 things that you should know about career development in an agile context. *Leanify* (2016). <http://leanify.com/3-things-about-agile-career/>
20. Cunningham, R.: Gartner magic quadrant names Microsoft Power Apps a leader for low code application platforms. *Microsoft* (2020). <https://powerapps.microsoft.com/en-us/blog/gartner-magic-quadrant-names-microsoft-power-apps-a-leader-for-low-code-application-platforms/>
21. Zenagile: Agile training: What it is and why you should adopt it. (2020). <https://www.zenagile.com/agile-training/>
22. Keita, B.: Top 5 benefits of agile certification. *Invensis Learning* (2020). <https://www.invensislearning.com/blog/benefits-of-agile-certifications/>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

