2023

A Blueprint for Adopting Agility in Teaching, Research and Service in an Engineering Department

Omar Ochoa
ochoao@erau.edu

Massood Towhidnejad
towhid@erau.edu

James J. Pembridge
pembridj@erau.edu

Radu Babiceanu
babicear@erau.edu

A Blueprint for Adopting Agility in Teaching, Research and Service in an Engineering Department

Omar Ochoa
Department of Electrical Engineering and Computer Science
Embry-Riddle Aeronautical University
Daytona Beach, FL 32114, USA
ochoao@erau.edu

Massood Towhidnejad
Department of Electrical Engineering and Computer Science
Embry-Riddle Aeronautical University
Daytona Beach, FL 32114, USA
towhid@erau.edu

James J. Pembridge
Department of Electrical Engineering and Computer Science
Embry-Riddle Aeronautical University
Daytona Beach, FL 32114, USA
pembridj@erau.edu

Radu F. Babiceanu
Department of Electrical Engineering and Computer Science
Embry-Riddle Aeronautical University
Daytona Beach, FL 32114, USA
babicear@erau.edu

2023 IEEE Frontiers in Education Conference (FIE) | 979-8-3503-3642-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/FIE58773.2023.10343318

Abstract— **This innovate-practice, work in progress paper presents a blueprint on the adoption of agile process in an academic engineering department. The blueprint offered in this paper can be used by engineering departments to introduce an agile approach in their teaching, research, and service activities. This paper provides a framework for using Scrum as it has been implemented in the EECS department at Embry-Riddle Aeronautical University. For each of the common facets of academia, i.e., teaching, research, and service, we report on our experiences, lessons learned, and proposed approaches in adopting Scrum. Furthermore, we highlight the value added by integrating this Scrum approach, such as the increased marketability of students with agile experience, a greater accountability from students and faculty, and/or building a much stronger communities between faculty, and between faculty and students. We also discuss possible obstacles and required mitigations to facilitate adoption. Results of adopting agility in all three facets of the academic department will be reported, including how it benefits computing and engineering education and the academic department.**

Keywords—Scrum in Academia, Agile Research, Agile Service, Agile Teaching

I. Introduction & Motivation

With the fast-paced changes in technology and industry requirements, it is essential for academia to keep up and adapt accordingly. As such, within industry, agile methods offer numerous benefits, such as the ability to quickly adapt to changing stakeholder needs, enhance quality, prioritize teamwork over individual goals, and improve transparency. There is an opportunity in integrating the proven benefits of agility into academia by adapting the approach to suit academic processes. The unique approach presented in this paper focuses on bringing the many proven benefits from agility in industry and integrating the approach into academia, which requires certain modifications. This paper presents an approach on how to integrate agility in academia, i.e., a blueprint. This blueprint provides a set of guidelines, recommendations, and lessons learned from an exploratory implementation of the approach being used as experiment within our department. The goal is that engineering departments could follow this blueprint to introduce an agile approach in their teaching, research, and service activities. This paper provides a framework for using Scrum, a widely popular agile method, as it has been implemented in the Electrical Engineering and Computer Science (EECS) department at Embry-Riddle Aeronautical University.

Scrum can be used in each of the three components of an active faculty member's responsibilities in an academic department, i.e., teaching, research, and service. For teaching, this paper reports on lesson plans, class objectives and assignments that can be used for any project-based class using Scrum, followed by a discussion on incorporating feedback stemming from students that have completed agile-based courses, and how to modify those courses to continue to improve the Scrum approach. Within research, this paper describes how to structure a research group that follows a Scrum-like approach, including responsibilities and research processes that involve researchers from all experience levels. For service, this paper reports on how to adopt Scrum in committees, and offers possible rewards and incentives approaches that can be utilized by a department for faculty. For each of these areas, the value added by integrating this approach is presented, such as the increased marketability of students with agile experience, a greater accountability from students and faculty, and/or building a much stronger communities between faculty, and between faculty and students. In addition, this paper discusses possible obstacles and required mitigations to facilitate adoption.

II. Background

Non-agile approaches to software life cycles involve the use of process-oriented methods that are document-driven and require adhering to a disciplined plan [1]. These disciplined strategies have well-defined strict phases and templates for how to complete the required artifacts [2]. These phases and templates are perceived to be heavyweight, i.e., they are very slow to respond to changes in requirements. This slowness led to the development of more lightweight, agile, approaches. The following describes agile approaches, in particular Scrum.

A. *Agile Software Development*

Agile approaches to software development can be characterized as lightweight methods that are people-oriented, adaptable to change, and characterized by short incremental iterations, with meaningful deliverable at the completion of each iteration. To better develop software, the agile manifesto was created by a group of motivated and experienced individuals that value customer interactions over following a plan [3]. All agile methods follow the twelve principles backing the agile manifesto, which involves continuous delivery of working software, a high level of customer involvement, flexibility to change, face-to-face communication, and improvements to the process [4]. The benefits of these methods are better software quality, improved productivity, frequent delivery, and customer satisfaction [5].

B. *Scrum*

Scrum is an agile software development method that manages software development in iterations, called sprints [5]. The agile process focuses less on the heavyweight and formalized processes [6] and more on daily progress and process improvement through retrospective meetings after each sprint and planning before the next sprint. The Scrum framework consists of a number of events and artifacts. Figure 1 represents an overview of the scrum framework [7]. The Scrum framework consists of number of events which include a Planning Meeting that is conducted at the beginning of each sprint, where the team identify what would be delivered during the upcoming sprint. Major activities during the planning meeting include prioritization of the deliverable, estimation of the effort, and identification of the team's availability during the upcoming sprint. A sprint represents a fixed time (typically 2–4 weeks) that the team will participate in development of the product, the Daily Scrum is a 15-minute time slot, at the beginning of each day where team members discuss their accomplishments during the previous day, and what they plan to work on during the current day, and a Sprint Review at the completion of each sprint where the product developed during the sprint is reviewed by the team, and stakeholders, and Sprint Retrospective at the completion of the sprint, which the team review the process used during the sprint, and look for opportunities to improve the process

Scrum artifacts include the Product Backlog representing what the customer needs, Sprint Backlog representing what the team plans to deliver during the sprint, and Burndown Chart representing the team’s accomplishments during the sprint in real-time.

There are three roles in the Scrum framework, these are the Product Owner who serves as the main interface between the customer and the team and responsible for the product backlog, and the prioritization of its content, the Development Team which is responsible for the delivery of the product during the sprint, and the Scrum Master who is part of the Development team, but also has the responsibility of the mentoring the team throughout the process, and also support the development team by removing impediments the team would face

Scrum utilizes the roles of Product owner and Scrum Master to streamline the development process. The Product Owner is the sole source and validation of requirements. While the Scrum Master is the main point of communication between the team and the outside organization, allowing the Scrum Master to protect the rest of the team from unnecessary distractions.

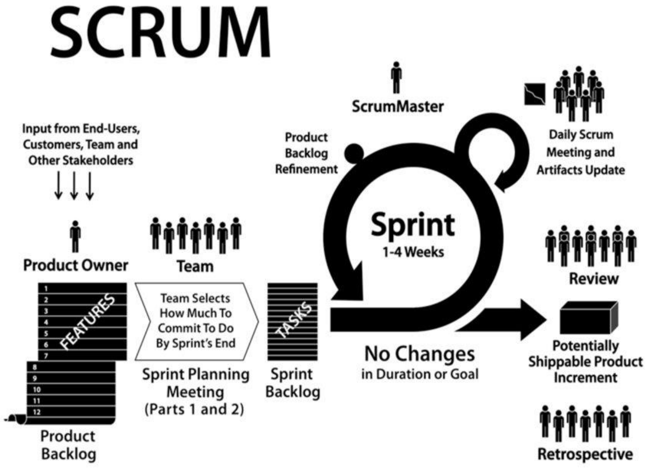
SCRUM
ScrumMaster
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Sprint
1-4 Weeks
Product Owner
Team
Review
1
2
3
4
5
6
7
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
8
9
10
11
12
No Changes
in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting (Parts 1 and 2)
Sprint Backlog
Product Backlog
Retrospective

Figure 1: Scrum Framework

III. Academic Adoption

Integrating Scrum into the engineering department can be done through the traditional three areas that support faculty endeavors. The following sections discuss adopting Scrum within each.

A. *Teaching*

Over the last decade and a half, a multitude of computing disciplines—including software engineering and computer science—have incorporated agile methodologies into their capstone design courses. Despite this, other engineering fields have been slow to follow suit. In line with many other institutions, our school first implemented Scrum in our two-semester capstone design course for Software Engineering and Computer Science programs in 2007. Shifting gears in 2011, we amalgamated our Electrical Engineering and Computer Engineering capstone design course with our established Software Engineering and Computer Science capstone course [10]. This necessitated students to engage in a multidisciplinary project. This multidisciplinary capstone design course extends over both the fall and spring semesters, wherein teams of four or five students undertake their project through twelve to thirteen two-week sprints. When the project is externally defined, a customer, faculty member, or teaching assistant can act as the Product Owner. However, for student-defined projects, the individual who proposed the project assumes the role of the Product Owner.

The Scrum Master plays a pivotal role in determining the success or failure of a project, making it ideal to assign this role to someone with prior agile experience. However, it's not always feasible to do so. Consequently, it becomes crucial to carefully supervise the performance of the Scrum Master. If their performance is found lacking, we first offer additional mentoring. Should this not rectify the situation, we then reassign the Scrum Master role.

Since the introduction of Scrum in the capstone sequence, we've observed an appreciable improvement in the quality of end products. This led us to incorporate Scrum in our earlier team project class, Introduction to Software Engineering, which is taken during the first semester of sophomore year.

The inclusion of Scrum yielded two significant outcomes. Firstly, the quality of products created within the course saw marked improvement. Secondly, these students progress to undertake the capstone design course roughly two years later, arriving equipped with a firm grasp of the Scrum process. This proficiency results in a far smoother process execution within the capstone course.

In our continuous effort to enhance student understanding of Scrum, we are now introducing components of Scrum in our introductory programming class, offered in the first semester of freshman year. While the projects in this class are individual efforts, students gain familiarity with crucial Scrum concepts such as backlog generation, planning a sprint, and project tracking.

B. Research

Scrum can be applied to research by structuring a research group that follows a Scrum-like approach [11]. This includes clearly defined responsibilities, research processes involving researchers of all experience levels, and regular progress updates. Using Scrum in research requires some modifications to Scrum. For example, the sprints are defined to be one week. This amount of time provides enough days for each member of the research group to gain insights into the research area and enough time to develop some research artifacts. The roles also need to be adapted. First, a Product Owner is the main driver for the research and could be a faculty member or a motivated student, responsibilities are to manage the research that is being conducted and ensuring that it aligns the research group's projects and is of high quality. Secondly, a Scrum Master role facilitates the research work (i.e., removing impediments) by providing guidance and technical advice, mentorship, and training. This role must be fulfilled by someone with expertise in research, i.e., faculty member or senior graduate student. Finally, the remaining members of the group, i.e., other students, act as researchers and work on one or more of the projects simultaneously depending on their interests and schedules, akin to the Development Team in traditional Scrum.

After each sprint, a research meeting is held where the Product Owner of the team gives an update on the progress of the research, gives any demos, if necessary, discuss any hurdles, and explain goals for the upcoming sprint. Once a presentation is complete, the Product Owner answers any questions posed by other Researchers and considers their inputs as related to the research presented.

Since this project is in its early stages, it was difficult to gather data as the current sample size is small. Because of this, it is important to note that trends could be misleading, and the given results must be treated critically. Despite this, the consistency of results gathered from the survey lends credibility to our conclusion and encourages us to further investigate the approach.

Preliminary use of Scrum in research has shown increased motivation in the research group when compared to the traditional control group [8]. Additionally, it seems that the Scrum research group has enhanced scalability, and with that is more productive with their research tasks. The evidence also supports that, with a properly structured Scrum research group, knowledge and skill transfer happens at a faster rate than with traditional research groups.

C. Service

Agile approaches can be adopted in committees and departmental services by redefining traditional responsibilities [12]. Committees are converted into Scrum teams, where the responsibilities are adapted to meet the needs of the committee. For example, instead of having a committee chair, a Scrum team will have a Product Owner. The Product Owner is responsible for ensuring that the work done by the committee aligns with the needs and desires of the department. Likewise, the Scrum Master focuses on removing impediments, which often requires the Scrum Master to talk to the department chair or other external people. The Scrum Master ensures the process is being followed, by organizing meetings and guiding the faculty to adhere to the process, e.g., only spending 15 minutes on standups. The remaining faculty act as team members and work on completing the team's deliverable.

Under Scrum, teams must meet regularly, as they work on their sprints, which should be short, i.e., 2 weeks. This is so because faculty have a wide array of varying time commitments, e.g., office hours, classes, research meetings, etc., while working on multiple obligations. Each sprint consists of a planning for the sprint, "daily standups" (at least twice a week), time for a sprint review and retrospection. The Product Owner manages the product backlog, using it to guide the team during the spring planning meeting. The team creates a sprint backlog, which is the work that they will be working on during a sprint. At the end of the sprint, the Product Owner reviews the work created by the team and offers feedback, followed by the retrospective, where the team reviews their performance during the sprint and aims to identify things that can improve the process.

The use of Scrum for service in the engineering department has been observed to have positive results [12]. Participants had a higher level of commitment and a bigger sense of ownership, with a faster turnaround in project delivery and higher efficiency, which align with other research results [13].

IV. Challenges & Mitigations

Adopting agility in an academic engineering department can face various challenges, such as resistance to change, lack of understanding of agile principles, and insufficient support and understanding from upper administration.

A. Teaching

As emphasized earlier, the success of a project largely hinges on an effective Scrum Master. Therefore, it becomes essential for a teaching assistant or faculty member to act as a mentor to the Scrum Master, particularly during the initial sprints. This mentorship ensures that the Scrum process is implemented and followed accurately, laying a solid foundation for a successful project.

B. Research

There exist potential drawbacks and pitfalls when using Scrum for research. Agile research groups depend on student researchers to take a more active role in research, which can be challenging with more novice students. To alleviate this pitfall, the student researcher must receive encouragement and mentoring from their peers or speak to the Scrum Master to receive guidance. The goal is to slowly give the student researcher confidence about learning the skills to conduct research. Additionally, this type of research approach requires more hands-on involvement from faculty, as it requires more mentoring and guidance on how to conduct research until the team has been trained. This can be mitigated by the faculty creating an environment where the teams help each other adopt the skills necessary to conduct research, thus offloading some of the responsibility to the group.

C. Service

Team members in an agile team working in industry are typically fully dedicated to their project. In an ideal industry setting, the team members are 100% assigned to a single project, and even in a nonideal case, the team members are usually not assigned to more than couple of the projects at the time, and in most cases both projects have relatively high priority for the organization. However, unlike industry, in an academic setting, team members (faculty) are involved in multiple projects with different priorities (either teaching, service, or research obligations), depending on their academic rank and career paths, and thus these projects may not be the highest priority. One mitigation that can be applied is to reward and incentivize Scrum usage in promotion and tenure. This way faculty can raise the priority of Scrum-based service and get recognized for it.

Another major difference between industry and academia is associated with planning a sprint. Sprint estimation can be challenging in the academic setting. Faculty work on multiple projects concurrently, and in most cases, they do not have any historical data associated with how much work it takes to accomplish a specific task. Not knowing this information causes the team members to erroneously estimate their availability during the upcoming sprint. A mitigation can be to identify the volatility ahead of time, i.e., if a proposal or paper is due in a sprint to adjust their availability accordingly. Although it's not perfect, and certain things can pop up of higher priority, the teams should be resilient enough to make up for the time lost.

V. Future Work & Conclusions

To sum up, this paper presents a comprehensive framework for integrating agility into pedagogy, research, and service functions within an engineering department. This preliminary blueprint for implementing agile methods, specifically the Scrum approach, outlines the steps to incorporate Scrum into academic practices. The adoption of agile methodologies can prepare students and faculty more effectively for the fast-paced professional environment, elevate research standards, and cultivate a culture of collaboration and ongoing improvement. Future research could explore the influence of agility adoption on student outcomes, research productivity, and departmental efficiency. These investigations could further refine this proposed blueprint and encourage its broader implementation within the realm of engineering education.

Acknowledgment

The authors would like to acknowledge the many Scrum team members involved in this research from the EECS department. Finally, we would like to thank the National Science Foundation (NSF Award #1920780) for their support.

References

[1] V. Mahnic, "Teaching Scrum through Team-Project Work: Students' Perceptions and Teacher's Observations", International Journal of Engineering Education, vol. 26, no. 1, pp. 96-110, 2010.

[2] M. Marques, S. Ochoa, M. Bastarrica and F. Gutierrez, "Enhancing the Student Learning Experience in Software Engineering Project Courses", IEEE Transactions on Education, vol. 61, no. 1, pp. 63-73, 2018.

[3] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. Martin, S. Mellor, K. Schwaber, J. Sutherland and D. Thomas, "Manifesto for Agile Software Development", Agilemanifesto.org, 2001. [Online]. Available: http://agilemanifesto.org [Accessed: 22-Apr- 2023].

[4] G. Kumar and P. Kumar Bhatia, "Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies", 2014 Fourth International Conference on Advanced Computing & Communication Technologies, pp. 189-196, 2014.

[5] G. Matharu, A. Mishra, H. Singh and P. Upadhyay, "Empirical Study of Agile Software Development Methodologies", *ACM SIGSOFT Software Engineering Notes*, vol. 40, no. 1, pp. 1-6, 2015.

[6] N. Ruparelia, "Software development lifecycle models", *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 3, p. 8, 2010.

[7] T. Gustavsson, "Benefits of Agile Project Management in a Non-Software Development Context : A Literature Review", in Project Management Development – Practice and Perspectives : Fifth International Scientific Conference on Project Management in the Baltic Countries, Conference Proceedings, 2016, pp. 114–124.

[8] V. Mahnic, "A Capstone Course in Agile Software Development Using Scrum", in IEEE Transaction on Education, 55(1) pp 99-106.

[9] J. Kuhl, "Incorporation of Agile Development Methodology into a Capstone Software Engineering Project" in Proceeding of the ASEE Midwest Section Conference, 2014.

[10] R. Stansbury, M. Towhidnejad, J. Clifford, M. Dop, "Agile Methodologies for Hardware/Software Teams for a Capstone Design Course: Lessons Learned", in Proceedings of the 118th Annual Conference for the American Society for Engineering Education (ASEE 2011), 2011.

[11] O. Ochoa and S. A. Reynolds, "Introducing Agility into Research Teams," 2022 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2022, pp. 1-8, doi: 10.1109/FIE56618.2022.9962665.

[12] O. Ochoa, M. Towhidneiad, T. Wilson, J. Pembridze, E. Bowen and C. Castro, "Adopting Agility in Academia through Pilot Projects," 2021 IEEE Frontiers in Education Conference (FIE), Lincoln, NE, USA, 2021, pp. 1-5, doi: 10.1109/FIE49875.2021.9637458.

[13] T.C. Krehbiel, P. A. Salzarulo, M. L. Cosmah, J. Forren, G. Gannod, D. Havelk, A. R. Hulshult, and J. Merhout. "Agile Manifesto for Teaching and Learning", in Journal of Effective Teaching, v17 n2 p90-111 2017.