12-21-2021

Adopting Agility in Academia through Pilot Projects

Omar Ochoa
Embry-Riddle Aeronautical University, ochoao@erau.edu

Massood Towhidnejad
Embry-Riddle Aeronautical University, towhid@erau.edu

Scholarly Commons Citation

Ochoa, O., & Towhidnejad, M. (2021). Adopting Agility in Academia through Pilot Projects. , ().
https://doi.org/10.1109/FIE49875.2021.9637458

Adopting Agility in Academia through Pilot Projects

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

Timothy Wilson
Department of Electrical Engineering and Computer Science
Embry-Riddle Aeronautical University
Daytona Beach, FL 32114, USA
wilsonti@erau.edu

James Pembridge
Department of Engineering Fundamentals
Embry-Riddle Aeronautical University
Daytona Beach, FL 32114, USA
pembridj@erau.edu

Erin Bowen
Department of Psychology
University of Texas at Arlington
Arlington, TX 76019, USA
erin.bowen@uta.edu

Carlos Castro
NEAR Lab
Embry-Riddle Aeronautical University
Daytona Beach, FL 32114, USA
castroc@erau.edu

2021 IEEE Frontiers in Education Conference (FIE) | 978-1-6654-3851-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/FIE49875.2021.9637458

***Abstract*—This innovate practice work in progress paper presents our initial findings about the application of agile process in the day to day operation of an academic department. The benefits of agile methods, which are processes that are lightweight and people-oriented, are well-known. Agile methods provide the ability to respond faster to the changing needs of stakeholders, improved product quality, team over individual goals, and better transparency. Therefore, one would expect adaption of the agile process within academic environment would be welcomed, however, unique issues exist within faculty that impact adoption; faculty have different goals that change through their tenure, and faculty have their own priorities that drive their time investment. For the last 18 months, the Electrical Engineering and Computer Science department at the Embry-Riddle Aeronautical University has been involved in adopting the use of agile process in its day-to-day operations, namely Scrum, which is one type of Agile method. To assess the feasibility of departmental adaptation of agile process, we established two pilot projects to identify advantages and disadvantages such adaptation. In this paper we report on the outcome of the projects in addition to the observations, challenges, and opportunities regarding the adaptation of the agile processes in the day-to-day operation of an academic department.**

***Keywords*— Scrum, Academia**

I. Introduction

The benefits of agile methods are well-known, which include the ability to respond faster to the changing needs of stakeholders, improved quality, team over individual goals, and better transparency. However, compared to industry, adopting agility in academia is difficult because changing faculty culture is a very hard, and typically academic processes are slow. Unique issues exist within faculty that impact adoption; faculty have different goals that change through their tenure, and faculty have their own priorities that drive their time investment. This is mainly due to the university's promotion and tenure recognitions are heavily weighted based on the individual faculty accomplishment. Therefore, the culture of the individualism is built into the earlier stages of the faculty career, and changing this behavior requires a complete rethinking both on the faculty, and university administration.

The Electrical Engineering and Computer Science (EECS) department at the Embry Riddle Aeronautical University (ERAU) has been involved in adopting the use of agile process in its day-to-day operations. The EECS Department conducted a survey of the faculty, and based on the results of this survey, two initiatives were identified that the department should focus on, one dealing with modifications to the curriculum, and the second dealing with graduate recruitment. This paper reports on the outcome of the projects in addition to the observations, challenges, and opportunities regarding the adoption the agile process in the day-to-day operation of an academic department.

II. Background

Traditional approaches to software development involve the use of process-oriented methods that are document-driven and require following a disciplined plan [1]. These disciplined strategies have well-defined life cycle phases and templates for how to complete the required artifacts [2]. These defined 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.

A. Agile Software Development

The agile approaches to software development are 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 the 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

This work is supported by the National Science Foundation Grant 1920780

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 number of events and artifacts. Figure 1 represents an overview of the scrum framework [7]. The Scrum framework consist 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 Product Backlog represents what the customer needs, Sprint Backlog represents what the team plan to deliver during the sprint, and Burndown Chart represents the team's accomplishments during the sprint in real-time.

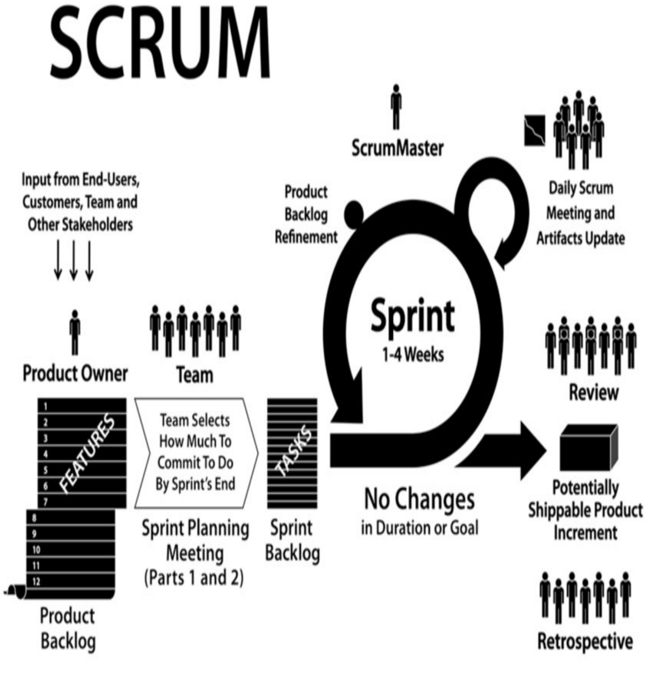
SCRUM
ScrumMaster
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Sprint
1-4 Weeks
Product Owner
Team
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes
in Duration or Goal
Potentially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

Figure 1: Scrum Framework

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.

III. Related Work

This section describes some approaches to using Agile approaches in both non-software projects and in academia. However, it is not the intention of this work to review the use of agile approaches in the classroom.

The benefits of using an Agile approach can be applied to non-software projects. Agile approaches have successfully been used in other domains e.g., manufacturing, product design, library management, education, pharmaceuticals, and healthcare [8]. These projects report benefits of using agile such as better teamwork, customer interaction, productivity, and flexibility. However, using agile approaches in these domains requires customization of the approach to better fulfill their unique needs [9].

Furthermore, studies have shown that the use of Agile in higher education can provide benefits, which include increased student engagement and responsibility for their learning, enhancing the quality of collaborations and providing higher quality deliverables [10].

IV. Approach

Through a faculty survey of the EECS Department, each faculty member was asked to identify their top three initiatives that they believe the department should focus its efforts on. Based on the results of this survey, two initiatives were identified that the department should focus on, one dealt with the modification to the curriculum, and the second dealt with graduate recruitment. To assess the feasibility of departmental adaptation of an agile process, two pilot project areas were established, each addressing one of these initiatives. The faculty that identified one of these initiatives as their top two initiatives were selected to serve on the corresponding pilot project team. To control the workload, no faculty could participate in more than one pilot project. Before the project initiation, faculty participating in the pilot projects were trained on the agile Scrum process by an external entity. Although we had number of qualified candidates to conduct this training, we believed involving a third party for would eliminate any perceived biases regarding what is covered and not covered in this training.

A. Curriculum Project

The Curriculum project includes three sub projects, each with specific goals and teams. The Curriculum Response to ABET visit (CR-ABET) goal was to respond to the ABET 30-day response letter, that was the result of the ABET visit in the fall 2019. The Software Engineering Graduate Curriculum (SEGC) project had the goal of modifying the existing software engineering graduate curriculum to better match the industry needs and the research goals of the department. Finally, the Undergraduate Curriculum (UC) team had the goal of revising and enhancing the department's four undergraduate degrees (computer engineering, computer science, electrical engineering, software engineering) to better serve stakeholders (i.e., students and the industries that hire them).

The CR-ABET team comprised of six members which includes the department chair, an ABET consultant (who is member of the department with ABET PEV and Commissioner experience), and four members of the department assessment committee. Every team member except one had previous scrum training. The team was charged to address all the issues that were raised as a result of the latest visit for the department's four undergraduate programs. In order to address the sighted shortcomings, there was a need to implement specific changes to three of four undergraduate programs. Typically, a curriculum change takes between two to four months. However, using the Scrum process, the team was able to propose the curriculum modifications, in two one-week sprints. During this time, the department chair, who was serving the product owner role, interacted with the college, and university decision makers, having all the stakeholders participating in the process. All the proposals were approved during the third sprints, and we completed our 30-day response within three one-week sprints, which resulted resolving all the shortcomings.

The SEGC team comprised of three members, all experts with Scrum, and have previously used Scrum in their teaching. The three members represents all the stakeholders in graduate software engineering program. The software engineering program coordinator served as both the product owner and the scrum master. The team identified the industry needs, surveyed competing graduate software engineering programs, and reviewed the department research and teaching capabilities. The proposal was generated in two sprints (two-week sprints) and was passed on to the remainder of the university curriculum change process on June 5th, 2020. The proposal was finally approved (completed the university curriculum change process) during the late March 2021. It is interesting to note that once the agile process interfaces with the old traditional processes that have been in practice by the rest of the university community, the advantages of using the agile process will be eliminated, and the old practices will dominate the overall process.

The UC team is comprised of four members, one serving as both Scrum Master and Product Owner, with each member of the team representing one of the undergraduate degrees. As well as proposed curriculum revisions, work completed includes examining curricula of similar programs in other universities, reviewing ABET accreditation guidelines, and consulting bodies of knowledge for each domain. Following the Scrum framework, Sprints are defined to be two weeks and include Sprint Planning (one hour), Sprint Review (demonstrating the progress to the department chair, the Chief Product Owner) and Sprint Retrospective (one hour combined with Sprint Review) and includes twice weekly standup meetings. The UC team has completed 13 sprints and has generated several major artifacts including:

- New proposed curriculum changes to the four B.S. degree plans (CS, CE, EE, SE). Including two new classes to be added to the department's offerings.
- Planned a new B.S. degree offering in Cybersecurity, which will be offered first as a track within the CS degree.
- Potential new curriculum directions for the team for the next year, including revisions to two minor degrees.

B. Recruitment Project

The EECS department offers five different Master of Science degrees, and over the last four years the number of incoming students to these programs has been significantly reduced. The major reason behind such a reduction is the significant reduction in the admission of international students due to the policies implemented by the U.S. government between 2016 and 2020. This not only affected the departments in delivering courses, but also significantly affected the faculty research production due to lack of high-quality graduate students to work on their projects. As a result, the department faculty identified graduate recruitment from U.S. undergraduate program as one of the highest priorities during the 20-21 academic year. A team of four faculty, and two undergraduate students formed the recruiting team, who were all trained in Scrum. The team followed the scrum framework having two-week sprints, with two daily scrums per week. At the completion of the sprints the team holds review, retrospectives, and planning meeting. In addition to the team, the recruiting team also kept a line of communication with the admission department and invited a member of the of the admission department to participate in number of sprint review and planning meetings to make sure they are aware of our activities and provide us with the resources that the team may need. The recruiting team has been in place for 10 sprints, and during this time, it has generated number of major artifacts which include.

- Department Graduate Community website, which allows students and the department to share information, such as available GRA, GTA positions, course offering, research opportunities, etc.
- Developed a virtual recruiting campaign, including identification of the potential undergraduate school feeders, development of the advertising materials, delivery multiple virtual information sessions.
- Identification of over 100 scholarship opportunities for current and future graduate students.
- Development of talking points material for all the graduate programs to be used by the admission department.

V. Analysis

The analysis of these pilot projects yields some interesting insights, which we represent here as challenges, and opportunities.

A. Challenges

Team members in an industry agile team 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 one of their top five the highest priority. In an industry setting, once an organization decides to adopt agile process for an organization and/or a project, then almost all employees will follow the decision or they may face precautions such as bad evaluation, or other disciplinary actions. However, in an academic setting, faculty are more independent and less constrained by direct supervision, and they have more freedom in how they perform their tasks. As long as they deliver the products that they are responsible for (course delivery, conducting research, publishing, advising etc.) they usually receive favorable evaluations. Therefore, there are limited negative implications for a faculty member to not fully participate in the agile implementation.

There are other differences between the industry and academy that require a number of customizations to adopt traditional agile processes in academia. For example, one of the key aspects of the scrum process is its daily stand ups, where the team members meet every day for 15–20 minutes capturing what has been accomplished in the previous day, and what they plan to work on during the day. We ended up scheduling the standup meeting twice a week instead of daily since faculty have wide array of varying time commitments, e.g., office hours, classes, research meetings, etc., while working on multiple obligations.

Another major difference between industry and academia is associated with sprint planning. In an industry setting, since team members are 100% or almost 100% dedicated to the project, once the project starts, and the team establishes its sprint velocity (how much work they can accomplish), there is typically limited variation in that velocity. At the beginning of each sprint, team members use velocity and team member availability to estimate sprint capacity (what they can accomplish during this specific sprint). However, sprint estimation can be challenging in the academic setting. As discussed, 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. As a result, sprint capacity is not as accurate as on an industry project. In addition, as faculty directly deal with at least one their customers i.e., students, they do not have the luxury of ignoring their customer or being shielded from them by the Scrum Master. Therefore, once the student requests a portion of the faculty member's time, the faculty member must dedicate that time to them. This results into a volatility in their time availability.

B. *Opportunities*

Since these two projects were pilot projects, and the participants had volunteered for the project, we achieved a high level of commitment of participants and a bigger sense of ownership about the products generated. The shared sense of ownership by every member of the team, and the inclusion of participants with the diverse set of goals, priorities, and views resulted in generation of high-quality products that may not have been generated following traditional process. For example, the recruiting project has representatives from most department graduate programs as well as student representatives. The team recognized the product must satisfy the needs of all the participants including students, and at a project postmortem faculty recognized students' contribution to the overall product quality.

On the curriculum team, the need to identify commonalities while highlighting program distinctions required compromises to achieve team goals. Having to make these compromises seems to have resulted in good team chemistry, with the team reporting satisfaction with the process and team operations. In addition, we also observed that there was a faster turnaround on projects and delivery of products compared to the traditional academic committees. Although this might be a byproduct of the increased attention to the projects, it is still a valuable outcome that might require further research. Finally, teams report that frequent interactions help product delivery with a more efficient operation. Additionally, it was observed that the right product must match with the right team members, the more aligned this is the more efficient the team can work.

VI. Summary and Future Work

This project is at the early stage of implementing an agile department. To date we have identified two major pilot projects, recruiting and curriculum development, which are using an agile process, e.g., Scrum, to deliver products. Through these early pilot experiences we have identified number of observations. These observations can be grouped into challenges and opportunities. Observed challenges for the adoption of Scrum in academia include that unlike in industry, faculty members cannot devote 100% of their time to a single project due to other obligations. This was exemplified by the need to switch from the Daily Standup into a twice-a-week daily meeting to accommodate faculty schedules. Additionally, a challenge that was observed is that it can be difficult to correctly plan a sprint due to the volatility in the faculty's available time for a sprint. However, certain observations can be seen as positive opportunities for the adoption of Scrum in academia. We observed that 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 [9]. Future work includes identifying additional products that can adopt an agile process to further create an agile department.

Acknowledgment

The authors would like to acknowledge the many Scrum team members, including Dr. Akbas, Prof. Behi, Dr. Garfield, Dr. Jafer, Dr. Liu, Dr. Pang, Dr. Rojas, Dr. Stansbury, and the following undergraduate students, Mr. Abeshi and Mr. Van De Brook. Finally we would like to thank National Science Foundation (grant # 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: 26- Apr- 2021].

[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] P. Deemer, G. Benefield, C. Larman, and B. Vodde, “The Scrum Primer’, goodagile.com, 2010. [Online]. Available http://www.goodagile.com/scrumprimer/scrumprimer.pdf. [Accessed: 26-Apr-2021]

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

[8] 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.

[9] L. Gerdes, R. Phaal, and V. Lynch. “Agile New Product Development in Not-Purely Software Projects.” https://doi.org/10.17863/CAM.66232.

[10] 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.