2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
ASEE
Paper ID #36703

Board 423: Using Agile Principles for Cohort Building in a Graduate Software Engineering Program

Dr. Stan Kurkovsky, Central Connecticut State University

Stan Kurkovsky is a professor at the Department of Computer Science at Central Connecticut State University. Stan earned his PhD from the Center for Advanced Computer Studies of the University of Louisiana in 1999. Results of his doctoral research have been applied to network planning and industrial simulation. Stan's current research interests are in software engineering, mobile, and pervasive computing. He published over 90 papers in refereed proceedings of national and international conferences, scientific journals, and books. Stan serves as a reviewer and a member of program committees for a number of national and international conferences. During his academic career, Stan received over seven million dollars in funding from private and federal sources.

Using Agile Principles for Cohort Building in a Graduate Software Engineering Program

Abstract

This report describes an approach to building a cohort of students in a graduate software engineering program supported by the Scholarships in Science, Technology, Engineering, and Mathematics (S-STEM) Program of the National Science Foundation. We used many agile principles for building and sustaining the cohort, which is scaffolded around the students' academic studies and their simultaneous work on an externally sourced software development project. We discuss how the agile principles were applied in practice in this S-STEM project, how they helped build a cohesive student cohort, and how they helped bring the software development project to a successful completion. This report describes the work in progress, which is limited in scope by the software project duration and the number of participants.

Introduction

The Computer Science Department at Central Connecticut State University (CCSU) offers a Master of Science program in Software Engineering to better address the needs of its graduates and the industry's demands. As a regional university, CCSU serves the needs of central Connecticut and the greater Hartford area. Since many large insurance companies (Aetna, Cigna, The Hartford, United Healthcare, and The Travelers) either have a very substantial presence in the area or are headquartered in Hartford, it has long been known as "the insurance capital" of the US. Today, these companies employ a very significant cadre of software engineers who develop and support various digital platforms, portals, and services that drive all aspects of the companies' business. These and many other companies have been extremely active in recruiting and hiring our graduates due to their high level of competencies and because 85% of CCSU graduates stay in Connecticut upon graduation. This benchmark is much higher for CCSU compared to graduates of other universities in the state [1].

Since 2021, our Master's in Software Engineering program has been supported by an NSF S-STEM award "Advancing Graduates in Learning Experiences (AGILE)" to attract and retain academically talented financially disadvantaged first generation and underrepresented minority students. The author of this report serves as the PI on this NSF award. AGILE scholarship program builds on the strengths and experiences of three prior S-STEM programs supporting Computer Science undergraduates at CCSU. This program implements and offers industry-centric curricular and co-curricular activities proven to improve student persistence. Another aim of this program is to study how student success in impacted by these activities and other factors enabled by the scholarship program. AGILE scholarship program extends the pathways for students with interdisciplinary backgrounds to earn a graduate degree in software engineering, a discipline with a severe deficit of highly skilled workers.

Program activities are aligned with three strategies for sustaining a diverse engineering program: recruitment, mentoring and support, and workforce development. Prospective students are recruited via a broad assortment of orchestrated efforts to attract a diverse range of students with Computer Science and other STEM backgrounds. Scholarship recipients are selected based on

their financial need, professionalism, and academic potential to succeed in the graduate software engineering program. Once in the program, they form a student cohort that plays an essential role in supporting their academic and professional success. Scholars are also assigned faculty, near-peer, and industry-based mentors and receive a wide range of support including academic and career planning, counseling, and internship opportunities. AGILE program leverages our existing synergies with industry to infuse industry-proven practices, activities, and connections into many aspects of the program. This includes its co-curricular and curricular components as well as the program's logistics.

In this paper, we discuss how agile principles were used to build and sustain a cohort of graduate students in our AGILE scholarship program over a period of one year.

The Context

The program was fully rolled out in the Fall 2021 semester when it offered support to five graduate students. Consistent with the program goals, all of them were first-generation college students and represented a broad range of ethnic and racial backgrounds. Their educational background was also diverse and included students with undergraduate degrees in Computer Science, Secondary Education in Mathematics, as well as minors in Biology and Music.

All five students were actively involved as software developers working as a part of a research project funded by the National Institutes of Health [2]. In this project, students developed G-Date, an iPad-based dating application specifically designed to research risk factors for sexual violence among men who have sex with men (MSM). This research project was led by a team of psychologists from Rowan University, Widener University, and CCSU. The research team developed a functional specification for G-Date and worked closely with the software architect (the author of this report) and the development team, consisting of the five graduate students, throughout the entire project. G-Date is intended to be used by MSM in a controlled lab environment where they would be led to believe that this is a fully functional application running on a mobile device. During the experiments, MSM would be told that they are interacting with other MSM, while, in fact, they are interacting with a scripted chatbot. Here, we describe only the software engineering aspects of this project that involved CCSU students and how this project was used for building and sustaining a student cohort.

Using Agile Methods

One of the AGILE Scholarship Program goals is to support the students by using high-quality, evidence-based practices and professional/workforce development activities. In addition to offering a strong industry-focused curriculum in software engineering, we implemented a metacognitive approach to introduce scholars to agile methods [3, 4] by using these methods to manage various aspects of the program. Over the last decade, agile methods have become an industry standard for running most small and medium scale projects [5, 6]. Scrum, the most common agile framework, allows self-organizing teams to achieve their goals by working with clearly established priorities. Scrum teams can easily track their progress and (re-)plan their goals, if necessary, during regular timeboxed meetings.

The twelve principles of agile software projects [3] were originally formulated in the context of software engineering with the purpose of creating and supporting a customer-focused work environment that aligns with business objectives. Agile principles provide developers with the guidance about responding and adapting quickly to inevitable changes in user needs and market forces. Today, agile principles are applied to manage projects in many other industries including finance, healthcare, and manufacturing [7, 8, 9]. In software engineering, agile principles form the foundation for several software project frameworks, such as Scrum.

To provide the necessary scaffolding facilitating the teamwork logistics and to compensate students for their effort with academic credits, all five students were enrolled in a special topics course whose main objective was to work on the G-Date project in both semesters of the 2021-22 Academic Year. In our context, we used agile principles and many Scrum elements to run this software development project while simultaneously applying them beyond the project and to manage many other aspects of students' progress towards reaching their academic goals.

The cohort met once a week during the regular meetings of the above-mentioned special topic class. The cohort's *goals* for each semester were established during the first week of class. These goals included those specific to the G-Date project, as well as broader goals, such as learning about a particular industrial technique, establishing stronger connections with near-peers in the industry, or achieving a specific technical credential.

The author of this report served as the instructor on record for the class that provided time and place for all project-related and cohort-related meetings. It was natural for the author to also play the role of the *Scrum Master* who facilitated all student team meetings, supported the team and individual team members in their development (and academic) tasks, and helped them avoid or resolve any potential conflicts.

The research team worked in close cooperation with the student team throughout the semester. The research project lead acted as a *product owner* serving as the primary point of contact with the team. The product owner worked with the team to populate the *product backlog* with a set of specific technical requirements and establish specific *priorities* at every step of the project. Additionally, the product owner was readily available to the team to provide detailed answers to any day-to-day questions concerning the G-Date functionality.

For the purposes of completing the G-Date project, the team's workflow was organized into six *2-week sprints* every semester, which allowed for a two-week inception phase at the beginning of the semester, a one-week transition phase at the end of the semester, as well as an accommodation for the semester breaks (Thanksgiving and Spring break). Working closely with the product owner, the team identified a set of *sprint goals* for every sprint, created a *sprint backlog* by extracting the corresponding set of requirements from the product backlog, and identified specific development tasks to be completed by the end of the sprint to meet the sprint goals.

A stand-up *Scrum meeting* took place during the first 15 minutes of every class, when each student briefly described their accomplishments, challenges, and their progress in two contexts. First, each student would discuss their progress towards the G-Date sprint goals, and then they

would discuss their progress towards the cohort goal in the scope of the entire semester. In each context, the students would also discuss what changes may need to be made to ensure that the goals are met.

At the end of every sprint, the team would hold an online *sprint review* with the entire research team. Since G-Date is a mobile application specifically designed for the iPadOS platform, the team would make their bi-weekly releases available to the researchers via the TestFlight app, so that they could test it on their devices prior to the sprint review. During the sprint review, the student team would offer a live demo of the current functionality of the app and answer any questions the researchers might ask. The product owner collected the feedback from the researchers and, if needed, made changes to the product backlog to reflect the refinements to the application functionality identified by the research team. A *sprint retrospective* conducted only by the development team typically followed, where the team discussed the lessons learned during that sprint.

Throughout the project, we used a number of technologies and tools to support the cohort's progress. Slack was used for all aspects of team communication. Trello was used for high-level project planning and interacting with the product owner, while GitHub Projects was used for tracking software development progress.

The last meeting of this class concluded each semester with a project retrospective, during which the cohort reflected on what went well, what didn't, and how students can become more productive both in the context of software engineering and in the context of reaching their academic goals.

Building and Sustaining the Cohort with Agile Methods

The S-STEM program emphasizes a broad use of cohorts to support the attainment of students' academic goals. In particular, NSF S-STEM program solicitation states that "cohorts should be formed in a way to enable scholars to support each other academically and socially" [10]. Based on our experience gathered from continuously running an NSF S-STEM sponsored scholarship program that has been supporting undergraduate computer science students at CCSU since 2006, we believed that working on the software project described above would be a good foundation for forming and sustaining a strong student cohort. While Scrum was a natural fit for managing the project itself, we wanted to see if working on this project would help students bond and form a cohort and whether the agile principles underpinning Scrum could be applied to a broader scope of student academic work.

Having students taking at least one class together is a proven practice in forming student cohorts. However, due to their varying academic backgrounds, it was impossible to expect all five students to simultaneously enroll in any of the existing courses fitting MS Software Engineering curriculum. This issue was successfully resolved by enrolling all five students in a special topics class that met once a week, which was created specifically for the purpose of working on the G-Date project and supporting the cohort.

Scrum provides an excellent framework to scaffold an existing software process and measure tangible progress. In particular, Scrum events, such as daily (in our case, weekly) scrums and sprint retrospectives, provided us with an opportunity to apply a formative assessment lens to measure student progress both in the software development project and in the broader academic scope.

From the outset, we knew that weekly scrums taking place at the beginning of the weekly class meetings would serve as a valuable tool to get students to talk about their project progress, any possible blockers that they may have encountered, and how they were able to resolve them. For aspiring software developers who are also new to a graduate academic program, it is vital to establish the sense of belonging [11]. This allows the students to see that they experience the same or a very similar set of issues as their teammates [12]. Additionally, this sense of belonging may help resolve any symptoms of the impostor syndrome that many of them are likely to experience [13]. Giving all students an opportunity to talk about their daily work and struggles allows the rest of the team to hear about these experiences, which most of them find very relatable. During these weekly scrums, we also asked students to reflect on their current progress in other classes that they are taking, if they felt comfortable discussing it with their teammates. We believe that asking students to reflect about their academic work was a successful strategy to bring out another dimension along which students were able to relate to each other and strengthen their cohort. Even though there were clusters of students who did take some classes together during each semester of the 2021-22 academic year, they did not form study groups until they discovered the similarity of their experiences during the weekly scrums. However, once they heard each other saying that they are experiencing very similar issues in a certain other class during a weekly scrum, they immediately moved to form a study group.

The project retrospective conducted at the end of each semester allowed us to apply the lens of summative assessment and review the effect of using agile methods throughout the semester on forming and sustaining the student cohort. It also served as an opportunity to conduct an end-of-semester review of how students were able to benefit from the cohort-building activities supported by Scrum. Each semester, project retrospective meetings consisted of two distinct parts. The first part of the meeting focused on the G-Date project and reflected only the software engineering aspects of the cohort work. It was facilitated by the author of this report. The second part focused on the cohort-building activities and the overall student experiences and satisfaction with the scholarship program. It was conducted by the external project evaluator with the author of this report absent from this part of the meeting. Below is a summary of student feedback:

- Students were generally very comfortable discussing their “trials and tribulations” with their teammates. The only topic they did not always feel comfortable discussing in that environment were their grades. However, this had no bearing on helping them relate to each other and form study groups outside of the G-Date project. Furthermore, since the author of this report served as their academic advisor, each student was able to discuss their grades, as they saw fit, during regular one-on-one meetings.
- Study groups were vital for most students to succeed in many of their graduate classes. Some students noted that they probably would not have formed these study groups if it were not for the weekly scrums where they learned about other students in the same class and the difficulties they experienced.

- Regular and structured interactions scaffolded by the Scrum framework in the software development project paved the way to more frequent communication among students outside of classes or projects. This helped them further develop a strong sense of the cohort sharing similar academic goals that helped them support each other both in the software engineering project and in their regular coursework.
- G-Date project served as a valuable experience where students practiced many technical and professional skills valued by the industry and expected of the graduates. Technical skill included several mobile application development frameworks, unit and integration testing, as well as iOS development workflow. Professional skills included strong teamwork and time management skills, as well as the ability to communicate with highly non-technical project stakeholders.

Summary

This report described how agile principles were applied to the process of building and sustaining a small cohort of graduate students in a Master of Science in Software Engineering program over the course of one academic year. We used Scrum to run a year-long software development project that served as a backbone for many of the cohort's activities. We believe that the agile project methodology was successful in reaching beyond the software development project. Based on the student feedback, using agile principles helped them build a strong cohort that was instrumental in helping them achieve academic success. Following this experience, we will continue applying agile principles to manage and sustain this evolving cohort as some of the students graduate and new students enter the scholarship program. We will also experiment with other supporting activities for the cohort that may not include a software project involving all students. It will be valuable to evaluate the applicability and benefits of agile methods to manage a group of students outside of the traditional software engineering project context.

References

[1] Megan, K. Almost Three-Quarters of CSCU Grads Get Jobs in Connecticut. Hartford Courant, July 31, 2016. https://www.courant.com/education/hc-cscu-students-stay-0728-20160731-story.html.

[2] Angelone, D.J. The development and validation of a novel paradigm for assessing sexual assault risk perception in MSM. https://reporter.nih.gov/search/ZvDy4NX6fkurBbzy9tT69w/project-details/10046308.

[3] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for agile software development. https://agilemanifesto.org/

[4] Fowler, M., & Highsmith, J. (2001). The agile manifesto. Software Development, 9(8), 28-35.

[5] Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum (Vol. 1). Upper Saddle River: Prentice Hall.

[6] Sharma, S., & Hasteer, N. (2016). A comprehensive study on state of Scrum development. In 2016 International Conference on Computing, Communication and Automation (ICCCA) (pp. 867-872). IEEE.

[7] Begel, A., & Nagappan, N. (2007, September). Usage and perceptions of agile software development in an industrial context: An exploratory study. In *First International*

Symposium on Empirical Software Engineering and Measurement (ESEM 2007) (pp. 255-264). IEEE.

[8] Conforto, E. C., Salum, F., Amaral, D. C., Da Silva, S. L., & De Almeida, L. F. M. (2014). Can agile project management be adopted by industries other than software development?. *Project Management Journal*, *45*(3), 21-34.

[9] Denning, S. (2013). Why Agile can be a game changer for managing continuous innovation in many industries. *Strategy & Leadership*, *41*(2), 5-11.

[10] National Science Foundation. NSF Scholarships in Science, Technology, Engineering, and Mathematics (S-STEM), Program Solicitation, NSF 23-527, December 6, 2022, https://www.nsf.gov/pubs/2023/nsf23527/nsf23527.htm.

[11] Xu, C., & Lastrapes, R. E. (2022). Impact of STEM sense of belonging on career interest: The role of STEM attitudes. *Journal of Career Development*, *49*(6), 1215-1229.

[12] Bronner, C. E., & Wakefield, A., & VanderGheynst, J. S., & Moloney, K. (2019, June), *Student-centered Strategies for Promoting Inclusive, Supportive, Diverse Environments in Graduate STEM Education* Paper presented at 2019 ASEE Annual Conference & Exposition , Tampa, Florida. 10.18260/1-2—33305.

[13] Tao, K. W., & Gloria, A. M. (2019). Should I stay or should I go? The role of impostorism in STEM persistence. *Psychology of Women Quarterly*, *43*(2), 151-164.