University Honors Theses   University Honors College

6-16-2021

Case Study of Scrum Methodology as Used by a Capstone Team

Lilly I. Yeaton
Portland State University

Case Study of Scrum Methodology as Used by a Capstone Team

by

Lilly Yeaton

An undergraduate honors thesis submitted in partial fulfillment of the

requirements for the degree of

Bachelor of Science

in

University Honors

and

Computer Science

Thesis Adviser

Warren Harrison

Portland State University

2021

Abstract

Scrum is widely used in the software industry to manage all kinds of projects. This case study examines the way in which a capstone team used the methodology and models the specific project management processes they used over the course of their project. These models and the process modifications therein are then compared to the team's velocity at different points in the project. The results of this analysis suggest a correlation between asynchronous daily meetings and sprint reviews and improved velocity.

Introduction

Parts of the process which we now call scrum, and indeed agile methodologies in general, have been in use in some form since the 1950s [1, 2]. However, it was in the early 2000s, in part due to the publication of the Agile Manifesto, that agile methods became more widely used in the software development industry. Ken Schwaber and Jeff Sutherland, two of the contributors to the Agile Manifesto, formalized the framework and published the Scrum Guide, first in 2010 with the most recent version being released in 2017 [3]. The version of Scrum described in this guide will be considered the canonical, baseline form of Scrum for the purposes of this research.

A scrum team performs certain key events and produces and maintains some key artifacts. The key artifacts consist of two backlogs, the sprint and product backlogs. These are essentially prioritized lists of tasks that need to be done in the current sprint and the entire project lifetime, respectively. The basic loop of a scrum process begins with sprint planning, where the team decides what work can be done during the next sprint (timeboxed to some time between one and four weeks). During the sprint, the team meets daily to assess each member's state and progress, as well as any blockers. At the end of each sprint, a sprint review is held, where the stakeholders and scrum team review increment (work that got done) together and adapt the product backlog if needed. The scrum team also meets privately for a sprint retrospective, where the team inspects itself and discusses improvements to be made to their process for the next sprint. The goal of all of this is to adapt to the various unpredictable changes that often occur throughout a project and to be able to create a working product despite them.

Research has been done in the past on first-time Scrum users and the specifics of the process that work for them. Mauricio Cristal et al. [4], for example, examined two teams within global companies using Scrum for the first time. This study concluded that new Scrum teams, particularly those that are separated physically, should consider things like keeping a global task board, integrating testers and developers, keeping detailed meeting minutes, and other practices not specifically outlined in the Scrum guide.

Capstone teams using Scrum have also been studied in the past. Viljan Mahnic's 2012 case study of the topic looks at a specific undergraduate capstone class at the University of Ljubljana and reports on how well the teams were able to estimate story points, track velocity, and plan their projects. While the main goals of this study were to assess the course and provide guidance for future educators, there are useful insights about how undergraduate teams

tend to interact with Scrum. For example, Mahnic [5] notes that students' planning abilities usually improved over the course of the project, just three sprints. In sprint three, the class on average was able to complete 91% of the story points they planned to. Students in the study also tended to report that their daily Scrum meetings were less helpful when reporting progress only to the Scrum master. Mahnic, therefore, recommends a focus on keeping team members up to date on other team members' progress during daily Scrum.

Finally, Scrum is an evolving methodology, and Jeff Sutherland [6], one of the inventors of Scrum, has published work in the past speculating on the future of Scrum and the changes that might be made to it as more teams use it. In a 2005 paper, Sutherland discusses how advanced agile teams might go beyond the basic definition of Scrum and utilize an advanced Scrum framework, involving overlapping sprints. He also discusses MetaScrum, wherein all parts of a company organize using Scrum, not just software development, allowing those managing multiple project teams using Scrum to do so by using Scrum themselves. In this way, we can see that the methodology is not set in stone, has changed over time, and according to Sutherland is likely to change more in the future. It is therefore important to examine the effectiveness of certain tweaks to the Scrum framework, especially in the context of a team that is not used to Scrum as it is currently defined.

Methodology

Capstone Project Overview

This paper will consider a capstone project created by a Portland State University (PSU) computer science capstone team (Team A). The PSU computer science undergraduate degree requirements include the completion of an undergraduate capstone project. This is a group project where students, under the direction of a professor and a peer team lead, create some kind of software project for a sponsor. Sponsors are usually a representative from a local business or other organization. The timeline for the project is two academic terms, which is just over six months in total. However, the majority of the first term is concerned with forming teams, matching teams to sponsors, and pre-sprint tasks such as requirements gathering. This paper will therefore mostly concern the second term, in this case, March 2020 - May 2020, where almost all actual development occurred.

Given this timeline, it is necessary to mention the effect of the COVID-19 pandemic on this project. For the last two weeks of the first academic term and the entirety of the second, quarantine was in effect and, because of this, all meetings were necessarily remote and communication was entirely digital. While many professional Scrum teams work entirely remotely due to geographical distance, in Team A's case it was an unexpected shift. Certain changes in the Scrum process that the team decided to implement, such as the move to partially asynchronous sprint reviews, were likely influenced by this situation.

Development was tracked using a Trello task board. This board was populated with user stories, or "tickets", each representing an incremental task or feature to be implemented. Tickets are considered the basic unit of work for this study. They are intended to be of similar complexity, and Team A separated work into equally sized tickets to the best of their ability. The Trello board was divided into six lists representing the various states of development: Backlog,

Groomed, Sprint Backlog, In Progress, In Review, and Complete. Fig. 1. shows the relationship between these states.

Backlog is where tickets were placed upon initial creation. Tickets in this state didn't necessarily contain all their necessary details. For example, Backlog tickets often lacked specific acceptance criteria, which were the conditions a feature needed to satisfy in order to pass review and be completed. In team meetings, the highest priority Backlog tickets would be "groomed" - i.e. the team would decide on the details of the task that the ticket describes, and what it should look like when completed. Essentially, the Groomed list contained tickets ready to be assigned. These tickets could then, during sprint planning meetings, be moved to the Sprint Backlog, which was the list of tickets that the team planned to complete in the coming sprint. When a developer began working on a ticket, they would move it to In Progress. When the developer(s) working on a ticket thought the acceptance criteria were met, the ticket would be moved to In Review. The review process, in this case, consisted of approval by two team members as well as the team lead. If the team felt more work needed to be done or changes needed to be made, the ticket would be moved back to In Progress until it was ready for review again. If the team felt that the feature was implemented adequately and the acceptance criteria were fulfilled, the ticket was moved to Complete.
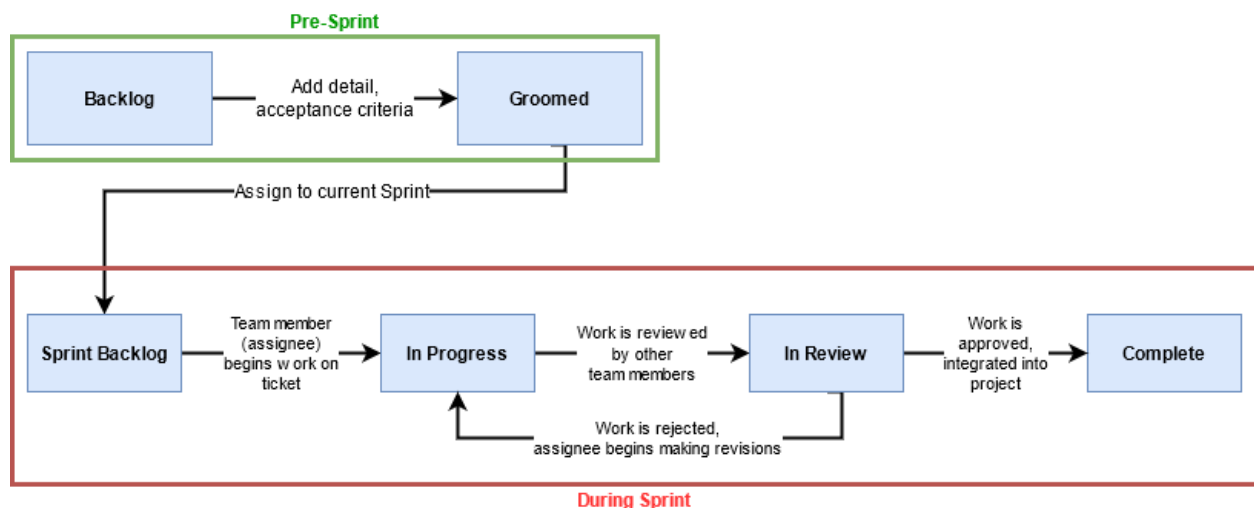

Fig. 1.   Diagram of development states that tickets would move through during the project. Blue boxes represent states, arrows denote the transitions between states and are labeled with the reasons for those transitions.

In the next section, the dates on which tickets were moved from state to state will be examined in order to form a retrospective timeline for the project's development. The time tickets spent in the states In Progress, In Review, and Complete will be most relevant here because those are the states in which the ticket was in active development or review and thus provide the most insight into the efficacy of Team A's project management methodology.

Process Modeling

In order to graphically display the variations on Scrum to be examined, the metamodeling system presented by Damiani et al. [7] will be used. In this model, a development process is defined as a set of phases, which are comprised of activities. Note that Damiani et al.

do present their own model of Scrum in [7], it is different from the one presented in the next section, as their model is based on a version of Scrum presented in 1995, not based on the Schwaber and Sutherland [3] text used here.

Data

Models

Team A began the first term of their capstone project, in January 2020, with a baseline understanding of the Scrum framework as shown in Fig. 2. During the initial stages of the project, they negotiated a slightly modified version of this framework, Process 1, shown in Fig. 3. One change was the replacement of the traditional daily standup with a bi-weekly one. Baseline Scrum assumes a team that works full time on a project and ideally meets in the office every day. Even if a team isn't physically in the same space, much of the Scrum framework is written for a team that would get value out of daily, synchronous (which is to say, "in person", either physically or in a video conference) meetings.
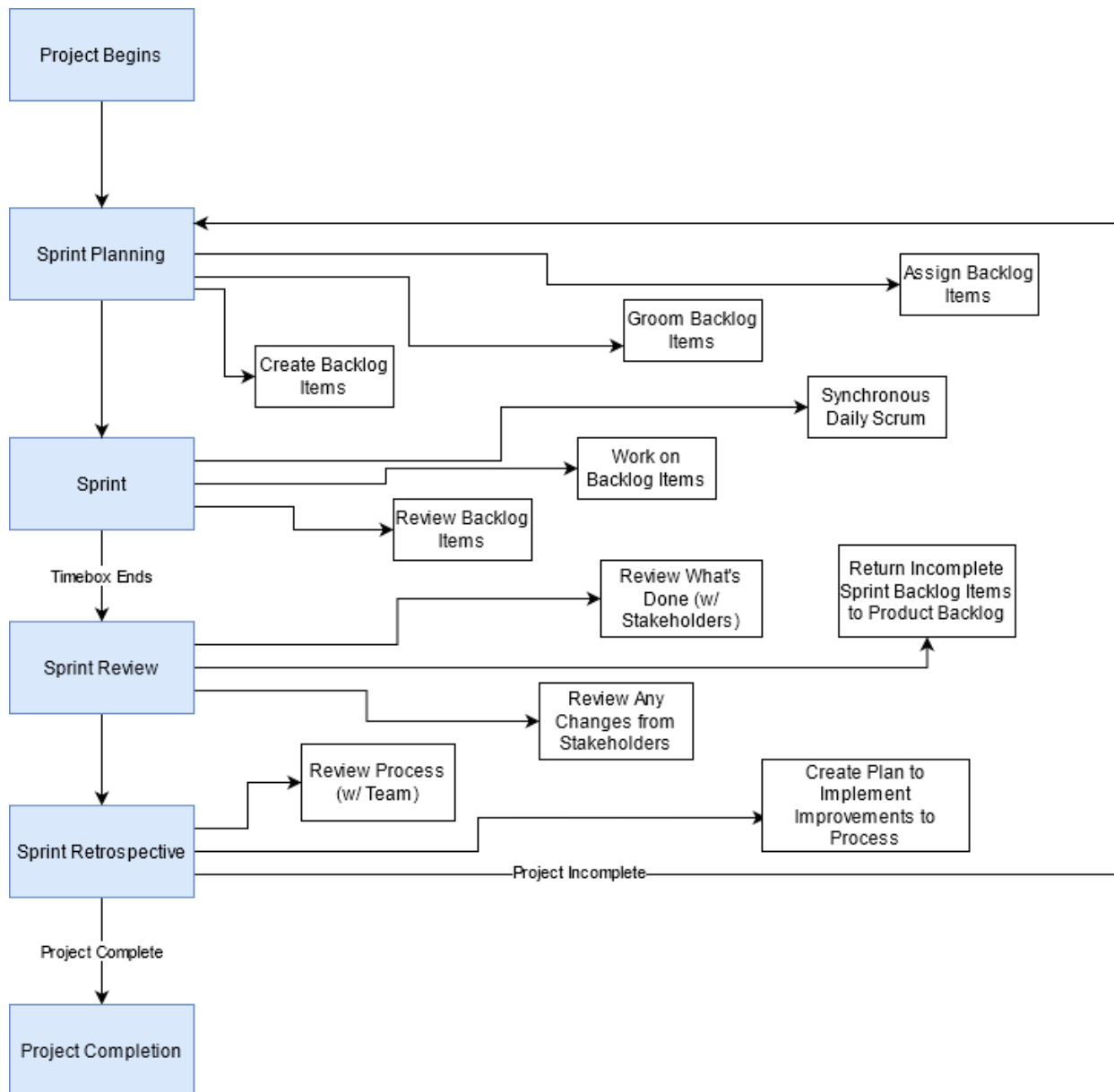
Fig. 2.   Model of "baseline" Scrum, i.e. Scrum as described in *The Scrum Guide (2017)* [3]

This was not the situation for Team A, however. As college students, most team members were taking other classes in addition to the capstone class. Unlike a typical team of developers employed at the same company full time, Team A's schedules differed significantly from one another. Scheduled class meetings for the capstone class turned out to be the only time it was possible for the entire team to meet synchronously. Additionally, the team assumed that because they would be working on the project part-time, there would not be significant progress to report on a daily basis. Due to this and in the interest of keeping standup meetings synchronous, Team A first implemented biweekly status update meetings.

Team A also implemented weekly meetings with their sponsor, who acted as a stakeholder for the project. Weekly stakeholder meetings are not typical in Scrum teams. This was thought to be helpful due to the team's inexperience, however, and was something the

sponsor and team were both interested in. Along with this, demos of the team's current progress were planned for every sprint review. This was to ensure the project was progressing in the way the sponsor expected. These demos were also something that the sponsor and team explicitly expressed interest in prior to the beginning of development.
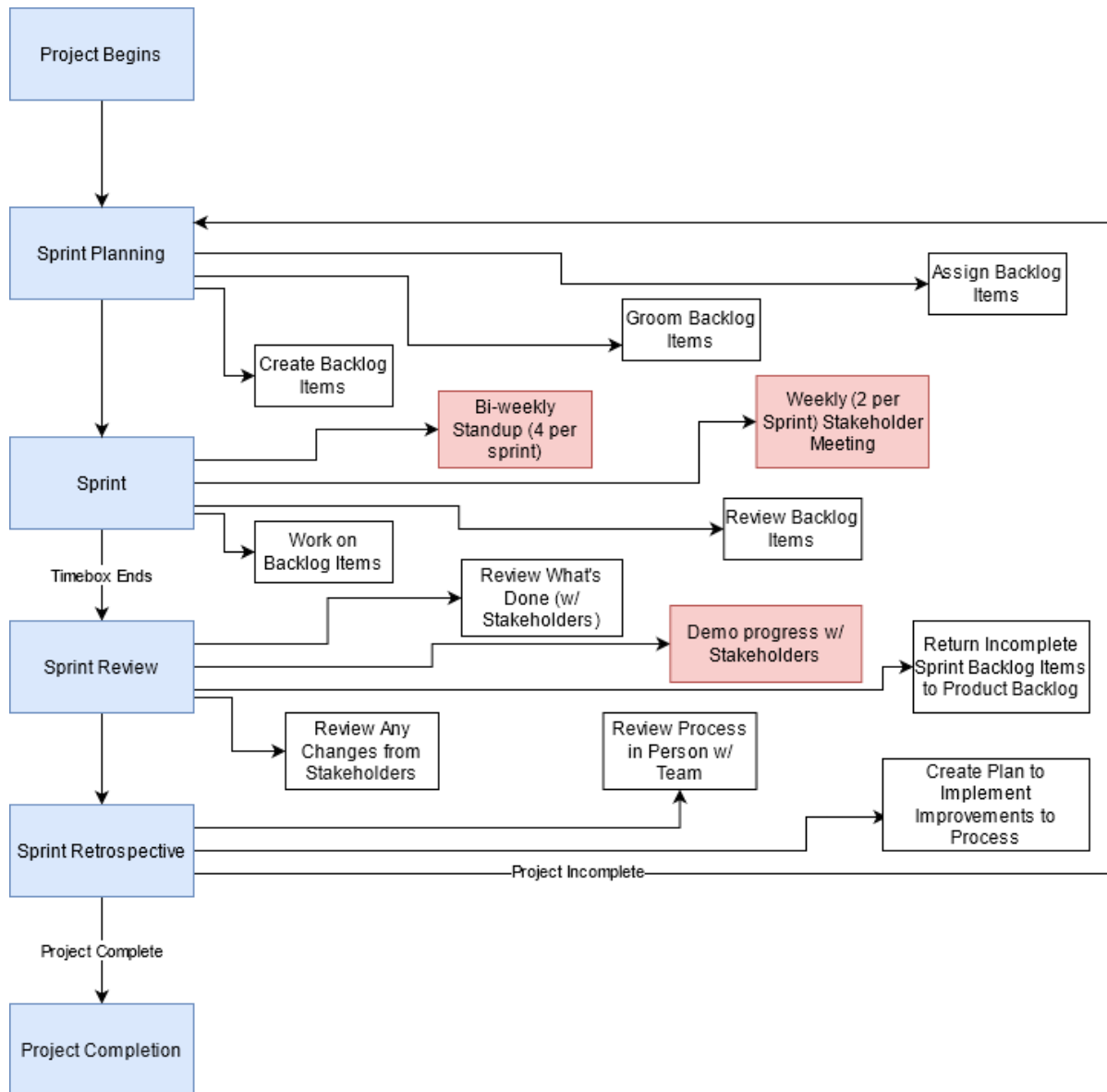

Fig. 3. First version of Scrum used by Team A, Process 1 (from approx. Jan 2020 - March 2020). Changes from Fig. 2. highlighted in red.

On April 14th, 2020, directly after the end of Sprint 1, Team A had a retrospective meeting about the previous sprint and discussed possible changes to their process. The final process, Process 2, that came out of this meeting can be seen in Fig. 4. The team noted a slow code review process and a lack of communication, both with each other and the sponsor, during the first sprint as areas to improve on.

To improve team communication and aid in review speed, Team A members decided to implement daily asynchronous standups over Slack instead of biweekly synchronous meetings.

This was in order to foster a greater awareness of other team members' progress, as well as encourage people to reach out sooner for help from other team members or guidance from the team lead. With regards to review speed, team members were to mention in their daily standup message when they had code that needed to be reviewed, ensuring awareness of that fact. Making these status updates asynchronous, i.e. everyone was free to post their update at any time during the weekday, avoided the scheduling burden of meeting daily for standups.

Another meeting that was made asynchronous during this time was the sprint review. During the first sprint, a synchronous review meeting was held and this was found to have a few issues. Firstly, the sprint review was sharing the same timeslot as that week's stakeholder meeting and demo. This meant the two ceremonies were competing for time, making both less effective. To alleviate this, Team A decided to have asynchronous sprint reviews where team members could reply to a slack thread with their feedback about the previous sprint at their leisure during the day when the synchronous meeting would have been held.
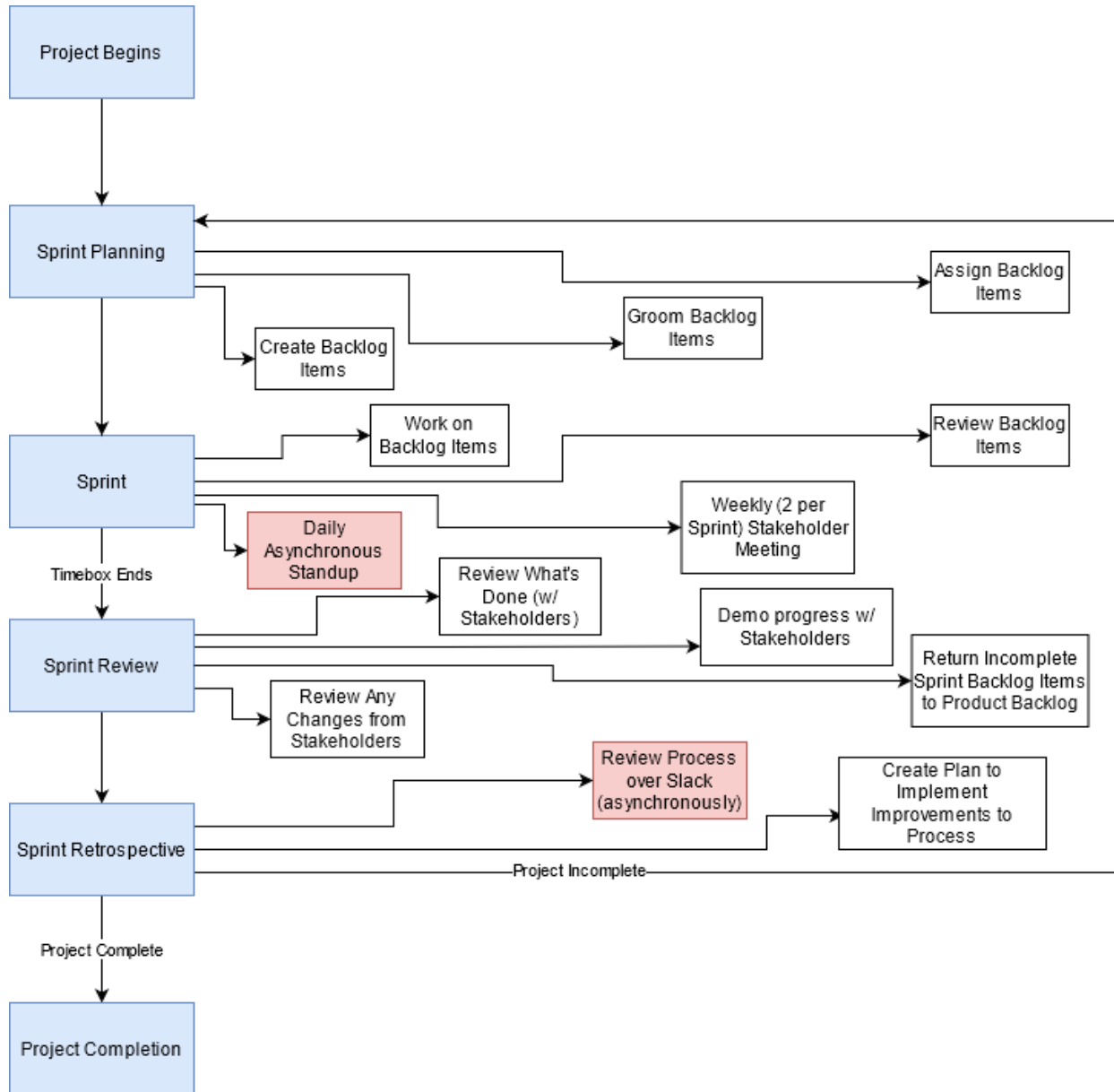
Fig. 4. Final project management methodology, Process 2, used by Team A (from approx. Apr 2020 - Jun 2020). Changes from Fig. 2. highlighted in red.

Ticket Statistics

Fig. 5 shows the average time a ticket spent in the In Review state based on the sprint it first entered review. This is an important statistic as long review times were something the team specifically cited as a reason for making process changes.
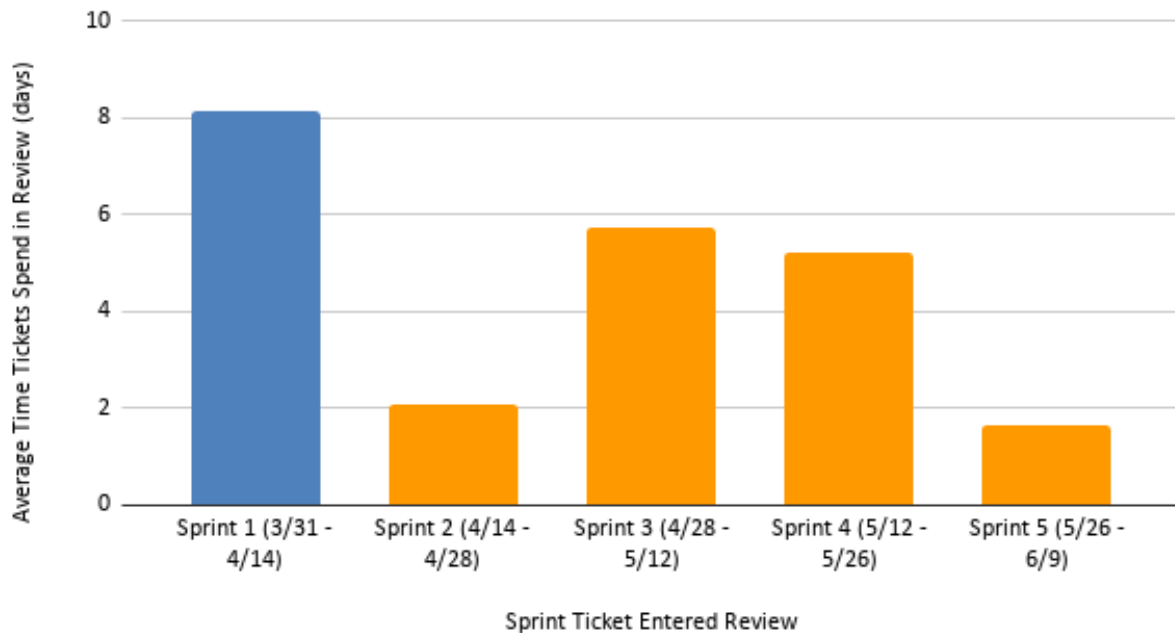
Fig. 5. Average ticket review time by sprint that the ticket was put into review for the final time before being approved. Blue (Sprint 1) denotes Team A using Process 1. Orange (Sprints 2 - 5) denotes Team A using Process 2.

Fig. 6. shows the number of tickets completed, which here means moved to the Complete state, in each sprint. All sprints to 2 weeks. It is also notable that the final presentation for the capstone class fell on June 4th, 2020, within the second week of Sprint 5.

The average tickets completed per sprint (velocities) achieved using each Process are shown in Table I. Due to the special nature of Sprint 5, a few different considerations are made for it.
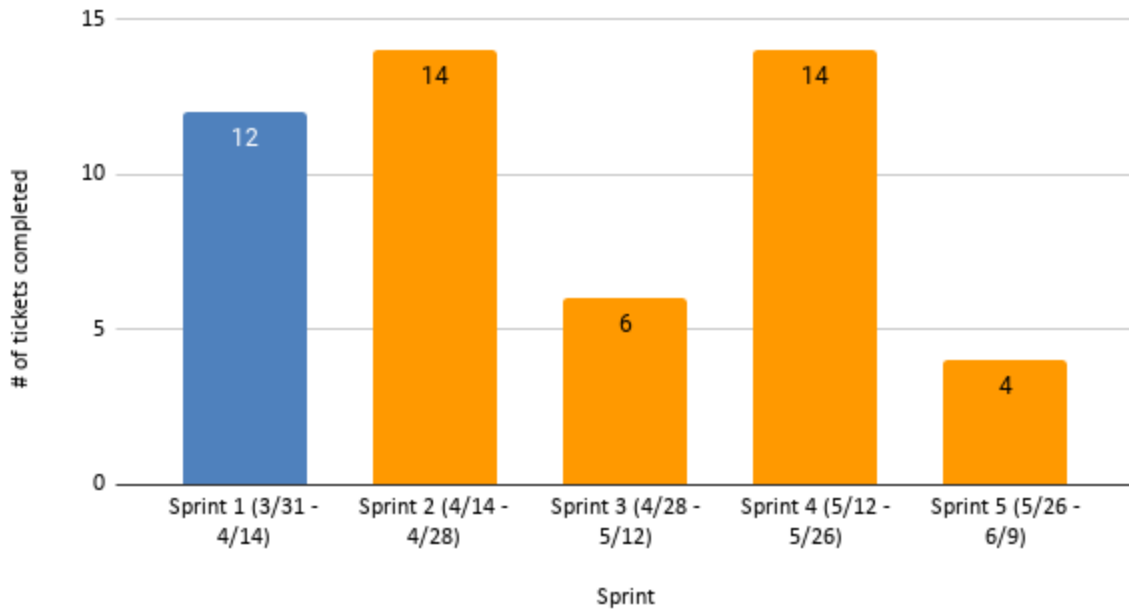
Number of Tickets Completed by Sprint

Fig. 6. The number of tickets completed in each sprint. Blue (sprint 1) denotes Team A using the process shown in Fig. 3. Orange (sprints 2, 3, 4, and 5) denotes Team A using the process shown in Fig. 4.

Table I
Velocity by Process

Process	Velocity		
Process 1	12		
Process 2	Including Sprint 5 as a whole Sprint	9.50	
	Excluding Sprint 5	11.33	
	Including Sprint 5 as one week (half a sprint)	10.86	

Discussion

Taken as an average, the data for Process 2 shows similar and slightly decreased productivity compared to Process 1. This would tend to suggest the changes made from Process 1 were detrimental to Team A's productivity and efficiency overall. There are some anomalous things to consider, however, such as how different Sprint 3's velocity is from the surrounding sprints. Sprint 3 had a velocity of only 6 compared to Sprints 2 and 4's 14. Unlike Sprint 5, which had a similarly low velocity, Sprint 3 was not cut short by any outside deadlines. This sprint did cover the typical spring midterm exam period, April 27th - May 8th. However, comments collected from Team A members after the project indicate that most team members were not taking other courses with significant workloads around this time. A more likely cause of the low velocity is a critical and unexpected bug in the project the team was attempting to fix around this time. Work on this bug took place from approximately May 5th - May 21st, which roughly coincides with Sprint 3. Many team members were devoting their time to this one issue and this may have impacted the total amount of tickets that could be completed. This is supported by the fact that most of the tickets that were completed in Sprint 3 were assigned to team members not assigned to the bug ticket.

Sprints 2 and 4 might therefore provide a better picture of the effects of Process 2, as they weren't impeded by another variable. The fact that the velocity of these two sprints was slightly higher than Sprint 1's velocity suggests a benefit to changes made in Process 2. The fact that Sprints 2 and 4 had identical velocities is also promising in that case, suggesting that Process 2 facilitates consistent efficiency.

Another theory is that the velocity of Sprint 3 being so drastically affected by the appearance of an unexpectedly time-consuming ticket suggests an inflexibility in some part of Process 2. This would be detrimental to any potential agile process as one of the founding philosophies of agile, and indeed the reason many teams choose it, is how it facilitates adapting to changing conditions. Ultimately, there isn't enough data to make any generalizations.

Additionally, average review time trends downwards as the project went on. This means that Process 2 correlates with decreased review time. Again, more frequent communication in the form of asynchronous daily standups may have lead to increased awareness of what tickets were in review and encouraged more prompt reviews by team members.

The changes from Process 1 to Process 2 and indeed between Process 2 and baseline scrum have to do with asynchronicity. Any correlations that exist between Process 2 and increased velocity or decreased review time, therefore, may point towards the benefits of asynchronous work practices, at least in capstone teams or inexperienced scrum teams. It would follow, for example, that teams like Team A who are unable to meet physically or even synchronously on a daily basis might benefit from daily communication regardless. Team A members were not all working on their capstone project every day of the week, but checking in daily may have improved productivity regardless.

In addition, if asynchronous sprint reviews improved productivity this could be due to a variety of things, and likely a combination of factors. As mentioned previously, the move to asynchronous reviews prioritized stakeholder meetings and allowed more time for real-time discussion with sponsors. This may have contributed to both Team A's work (feedback from sponsors allowing work efforts to be better directed) and morale (positive reinforcement from

sponsors improving excitement about the project). There is also the fact that an asynchronous sprint review allows team members more potential time to reflect on the previous sprint and may have led to more thorough or insightful feedback because of this.

Finally, it must be noted that all observations about the efficacy of one process over another must be considered in addition to the rest of the project context. Team A was a team of students using agile for the first time and, in the case of some team members, weren't used to working on a project of this scale in a team. At least some of the differences between sprints could be due to the team simply becoming more comfortable with the work and each other over time. Because of the relatively short duration of the project, it's hard to separate the impact of process changes from the impact of practice.

Conclusion

In conclusion, the data collected from Team A suggests a possible correlation between the frequency of meetings and overall productivity (i.e. higher velocity and lower review times). The data also seems to support previous research that suggests capstone teams improve in terms of productivity over time.

The scope of this study is relatively limited and there is a lot of future research that could be done on the topic of project management in capstone teams. For example, this study examined only one team. Future studies could compare productivity between teams using the same project management processes but developing different projects to examine the efficacy of that process across different workloads. It could also be interesting to see how projects with a longer timeline are affected by mid-project process changes, or if there is any difference between physical, in person meetings and virtual ones in terms of velocity. Overall, there is a lot more to learn from capstone teams about how inexperienced scrum users interact with the system for the first time, as well as how well scrum works for teams that don't have a standard full-time schedule.

References

[1] Lei, Howard, et al. "A Statistical Analysis of the Effects of Scrum and Kanban on Software Development Projects." *Robotics and Computer-Integrated Manufacturing*, vol. 43, Feb. 2017, pp. 59–67. *ScienceDirect*, doi:10.1016/j.rcim.2015.12.001.

[2] Larman, Craig; Basili, Victor R. (June 2003). "Iterative and Incremental Development: A Brief History". *IEEE Computer*. 36 (3): 47–56. doi:10.1109/MC.2003.1204375. S2CID 9240477

[3] Schwaber, Ken, and Jeff Sutherland. The Scrum Guide. Scrum Guides, scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf.

[4] Cristal, Maurício, et al. "Usage of SCRUM Practices within a Global Company." *2008 IEEE International Conference on Global Software Engineering*, 2008, pp. 222–26. *IEEE Xplore*, doi:10.1109/ICGSE.2008.34.

[5] Mahnic, Viljan. "A Capstone Course on Agile Software Development Using Scrum." *IEEE Transactions on Education*, vol. 55, no. 1, Feb. 2012, pp. 99–106. *IEEE Xplore*, doi:10.1109/TE.2011.2142311.

[6] Sutherland, J. "Future of Scrum: Parallel Pipelining of Sprints in Complex Projects." *Agile Development Conference (ADC'05)*, 2005, pp. 90–99. *IEEE Xplore*, doi:10.1109/ADC.2005.28.

[7] Damiani, Ernesto, et al. A Metamodel for Modeling and Measuring Scrum Development Process. 2007, pp. 74–83. ResearchGate, doi:10.1007/978-3-540-73101-6_11.

Alegría, Julio Ariel Hurtado, et al. "Analyzing the Scrum Process Model with AVISPA." *2010 XXIX International Conference of the Chilean Computer Science Society*, 2010, pp. 60–65. *IEEE Xplore*, doi:10.1109/SCCC.2010.18.

Rising, L., and N. S. Janoff. "The Scrum Software Development Process for Small Teams." *IEEE Software*, vol. 17, no. 4, July 2000, pp. 26–32. *IEEE Xplore*, doi:10.1109/52.854065.