Software Development Internship Experiential Learning Takeaways and Observations

Investigating the Effect of Social Pressures Found in the Scrum Methodology

A Thesis Prospectus
In STS 4500
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By
Nathan Barnette

Fall 2023

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Nathan Barnette

ADVISORS

Joshua Earle STS 4500 advisor, Department of Engineering and Society

Introduction

This past summer I gained valuable experience leading a group of software development interns. I learned many things during this time, but the most valuable part of the experience was the lens it offered into agile development and the Scrum process - something I felt I had been missing in my current curriculum. I detail these insights in my technical project, and further investigate them in my STS project, in which I discuss how to refine the inefficiencies in current agile development processes. My STS project hopes to answer the research questions: *What challenges do teams and team members encounter when determining success metrics within the Scrum process? How does team performance vary under different degrees of pressure?* I chose this as a question because while interning I felt that the pointing system we used to determine how much effort each task would take might have been flawed. In this system, each developer estimates a numerical amount of effort that they expect it will take to accomplish a task, allowing the eventual estimation of how many tasks to commit to each sprint. During my time voting for how much effort a task would take, I often found myself wanting to inflate numbers to put myself under less stress. Essentially wanting to be credited for doing something difficult that wouldn't actually take much effort to accomplish. Any improvements in the Scrum system would help teams work more efficiently and help management better understand how effective each individual developer can be, leading to quicker innovation and better business practices. This prospectus continues by providing the abstract of my exploration into the experiential learning opportunities internships have provided me, then gives details about my STS project and sources I will use in it.

Technical Topic

I plan to detail the significance that my internship experience had in my technical report. I have interned as a software developer a total of four times at a tech company that specializes in sending marketing communications for enterprise businesses to their customer base, mainly in the form of emails and push notifications. While I have learned many fundamental skills majoring in Computer Science here at UVA, my internship experience has definitely been a place of continual outgrowth. I have been exposed to many technical, interpersonal, organizational, and even managerial skills during my time working at MessageGears that I never would have had I just been in class.

This past summer I helped plan the internship program for the company I have worked for. I coordinated with management to discuss an onboarding process for the group and personally onboarded half of the group for the first week of work this summer. Afterwards, I had three direct intern reports while working on converting our front-end webapp from a more out of date framework, Angular, to the more modernized React framework. To do this conversion I had to learn the new React framework, utilize the technical skills I learned from Angular, and share my knowledge of how both connected to the backend to make the transition seamless. Learning first how to get the other interns up to speed, then later how to best communicate tasking strategy with them had its own challenges and rewards. Ultimately, I feel like I did a good job in this regard - three out of the four interns asked back in totality were the three under me. Feedback I received during my exit interview even said that myself and two interns I took under my wing had started to perform on par with or better than some of the more junior developers. I'm hoping to broaden my horizons elsewhere next fall, but it is nice knowing that I have made enough of an impact on a company to have a standing job offer should I need it.

In summary, this report will detail everything I have learned in my time as a software development intern, highlighting things I learned that I would not have otherwise been exposed to in the UVA curriculum. I plan to go into much further detail about the significance my work had to the company I worked for as well as myself.

STS Project

Agile development is an iterative and flexible approach to software development that emphasizes collaboration, adaptability, and customer satisfaction. Scrum is one of the most popular frameworks within the agile methodology. The Scrum process is designed to help teams deliver high-quality software by breaking down complex projects into smaller, more manageable tasks. All of these tasks are compiled into a product backlog, which a product manager combs through to determine an individual sprint's work. A sprint is a time-boxed iteration during which a specific set of tasks or work items are supposed to be completed. They typically last between two to four weeks. Each sprint has its own planning period, daily standups, and a retrospective review period (Sutherland, 2014).

I have described above how the creator of Scrum, Jeff Sutherland, advocates for his methodology to be used. His intentionally loose-fitting guide to Scrum has had little updating since the 2000s, serving as an easily adaptable template for any development team to mold into to best fit their needs. This format has been widely adopted since its inception, but it is by no means perfect. Successful sprint planning and execution relies on developers getting a feel for how much work can be accomplished over the course of a sprint. However, there is no empirical evidence that suggests an improvement in the predictive abilities of developers over any time frame (L. Cao, 2022). Speaking anecdotally, this lack of improvement is puzzling. After years of experience at the same company I have been able to estimate with much more confidence how long it will take for me to accomplish a task if I have worked in or around it before. To be clear, I still am often wrong when estimating - just much less so than when I first started working. I want to explore qualitative factors that might be hindering the improvement of the pointing process, from lack of internal team communication to outside stressors.

While the challenges of stress and tight deadlines on development teams are widely acknowledged, they remain underexplored in the STS field. I want to take a further look at the Scrum process, highlighting social and external pressures found within development teams and examining how they affect the development process. I hope to answer the following research questions: *What challenges do Scrum teams and team members encounter when determining success metrics within the Scrum process? How does team performance vary under different degrees of pressure?* I want to learn how the ideal Scrum team would operate under "sterile" conditions. How it should be managed. How it should be structured. How big it should be. How its members communicate with one another. How they ask questions. Any improvements in the software development process will have a significant impact on the amount of time it takes for product roadmaps to be completed. Innovation spurs innovation, and minimizing the time it takes for consumers to see improvement in products can have a significant impact on consumers of technology as a whole.

Software developers and product managers are the social groups these questions apply to most. These are the two groups that would immediately benefit from and use any Scrum optimizations in their day to day lives. In fact, anyone working in engineering - meaning anyone who's work impacts or is impacted by a sprint cycle - could be seen as relevant in this scenario. All of these employees depend on methodical sprint planning and execution to get their jobs done correctly. Due to the nature of software development, many more social groups could come to see the benefits provided by more efficient development. Software users as a whole could gain by seeing less bugs make it in production code or gaining access to products released sooner than otherwise possible. However, I would be hesitant to include those impacted by such a ripple effect as a relevant social group to not point the research in the wrong direction. To keep the scope of this research manageable, relevant social groups will be focused around those using Scrum practices day to day.

To explore the agile development process I plan to employ a variety of different STS methodologies and frameworks. My research will be collected in a variety of different ways, mainly focusing on tracing connections of relation within the Scrum process. Scrum teams are by definition their own network, and I hope to apply the actor network theory to their team members and technologies used to see how power flows through them, what challenges teams face, and how these challenges affect them. I also want to examine external, namely stakeholder and managerial, pressures Scrum teams might face using the actor network theory as well. In addition to analyzing the flow of communication within a Scrum team, I am using ethnography as an additional method of research. Several of my managers at my former place of employment studied the Scrum process in person with Jeff Sutherland and have years of collective experience in managing sprint cycles. I plan to talk to them to discover their frustrations with their current team's dynamic and hope to learn any insights they might have to offer into forming effective agile development teams.

The timeline for this research is relatively straightforward, with the only thing under any sort of time pressure being the ethnographic portion. To avoid any last minute issues with this course of action I plan to reach out to my former managers over winter break.

Key Texts

The first key text I am using is *Reassembling the Social: An Introduction to Actor-Network-Theory* by Bruno Latour. *Reassembling the Social* serves as a foundational text for understanding actor network theory (ANT). It provides theoretical tools for analyzing the complex interactions and entanglements in a social network by emphasizing the agency of both human and non-human actors in such a network. Latour also notes the importance of translation and alignment within ANT, which involves the alignment of interests, goals, and meanings among actors. In Scrum teams, understanding how team members and various elements align or translate their intentions can shed light on the negotiation processes within the team. ANT can help us understand the dynamics of the team as a whole (Latour, 2005).

The second key text I am using is *Scrum: The Art of Doing Twice the Work in Half the Time* by Jeff Sutherland. This book, written by one of the co-creators of Scrum, provides insights into the Scrum framework and its principles. It offers a foundational understanding of Scrum from the perspective of someone intimately involved in its development. It gives specific case studies, talks about how to scale scrum at large companies, and gives general practical advice to practice Scrum (Sutherland, 2014).

The third key text I am using is *Laboratory Life: The Social Construction of Scientific Facts* by Bruno Latour and Steve Woolgar. This is a book about the ethnographic nature of a laboratory environment. Latour argues that scientific facts are not discovered but rather constructed through a complex social process involving negotiations, debates, and the use of various scientific instruments. The book emphasizes the importance of social interactions, power dynamics, and the role of non-human actors (such as scientific instruments) in shaping scientific knowledge. Social interactions in the laboratory are not unlike those found on a Scrum team and the insights into team dynamics found in this book can be applied to my thesis (Latour & Woolgar, 1979).

The fourth key text I am using is *An Introduction to Science and Technology* by Sergio Sismondo. Sismondo touches on two crucial topics for my thesis in this piece: actor network theory and the social influence on construction of scientific realities. Sismondo lays out a clear explanation of what ANT is and how it is widely used. He then goes on to list several problems the ANT faces and how it can be limited in scope. Having a clear understanding of the limitations of ANT will help put my research of its role in the Scrum process into perspective. Further investigating the social construction of technical realities can provide insight into how social dynamics affect a team (Sismondo, 2004).

References

Broza, G. (2012). *The Human Side of Agile: How to Help Your Team Deliver*. 3P Vantage Media.

Cao, L. (2022). *Estimating Efforts for Various Activities in Agile Software Development: An Empirical Study*. IEEE Access, 10, 83311-83321. doi: 10.1109/ACCESS.2022.3196923

Chirra, S. M. R., & Reza, H. (2019). A Survey on Software Cost Estimation Techniques. *Journal of Software Engineering and Applications, 12(6)*, 226-248. doi: 10.4236/jsea.2019.126014

Cohn, M. (2005). *Agile Estimating and Planning*. Pearson.

Latour, B. (2005). *Reassembling the Social: An Introduction to Actor-Network-Theory*. OUP Oxford.

Latour, B., & Woolgar, S. (1979). *Laboratory Life*. Sage Publications.

Sage, D., Dainty, A., & Brookes, N. (2011). How actor‐network theories can help in understanding project complexities. *International Journal of Managing Projects in Business, 4(2)*, 274-293. doi: 10.1108/17538371111120243

Schwaber, K. (2010, June 15). *About scrum.org*. Scrum.org. https://www.scrum.org/about

Sismondo, S. (2004). *An Introduction to Science and Technology Studies*. Blackwell Publishing.

Sutherland, J. (2014). *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Currency.