

**Agile Frameworks: How Appropriate Problem Definition Can  
Streamline Software Development**

A Technical Report Submitted to the Department of Engineering

Presented to the Faculty of the School of Engineering and Applied Science  
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree  
Bachelor of Science, School of Engineering

**Samuel Veliveli**

Fall 2024

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

Rosanne Vrugtman, Department of Computer Science

Briana Morrison, Department of Computer Science

# Agile Frameworks: How Appropriate Problem Definition Can Streamline Software Development

CS4991 Capstone Report, 2024

Samuel Veliveli  
Computer Science  
The University of Virginia  
School of Engineering and Applied Science  
Charlottesville, Virginia USA  
rrs4bw@virginia.edu

## ABSTRACT

Software development in corporate America is often slowed down by red tape, unclear requirements, and conflicting desires for applications. To combat this problem, I learned about and utilized agile frameworks in software development during my summer internship. More specifically, I focused on clear requirements gathering, effective standups, and communicating with stakeholders in an efficient manner. I found that these techniques helped me focus on the bigger pictures of my deliverables without getting bogged down by technical requirements. As I continue to use these techniques, I want to integrate agile frameworks to eliminate miscommunication in software development teams to prevent unnecessary delays in production.

## 1. INTRODUCTION

Internships are coveted opportunities that students often see as the pinnacle of their academic career and the beginning of their professional career. One of the reasons internships are so valuable is that they expose students to the real-life applications of classroom learning. Additionally, students can often see shortcomings, obstacles, and failures in their real-world working environments that can inspire them to create innovative solutions and begin a cycle of positive change. After all, a chief reason why companies hire new grads

and interns is to get a fresh perspective and bring new ideas to the workplace.

During my summer internship at FreddieMac—one of the leaders in the Secondary Mortgage Market in the U.S., I had the opportunity to work on software development in a corporate environment. I saw firsthand many applications from my classroom, and lessons from my lectures come to life. However, I also saw room for improvement, particularly in the transition from business requirements in problem definition to software development. The transition I saw was slowed down by red tape, poor planning, and unclear requirements.

The lessons I learned in the UVA CS department regarding agile frameworks helped me focus on clear requirements gathering, effective standups, and efficiently communicating with stakeholders. Doing so allowed me to minimize miscommunication in my software team and prevent unnecessary production delays.

## 2. RELATED WORKS

In his paper about scaling agile frameworks in a global software development environment, Beecham (2021) discussed three main areas where agile frameworks failed to factor in necessary risk in global software development. All of these unaccounted for risks were related to environmental concerns outside of the

companies' control. However, there was support for agile frameworks allowing internal decisions to be faster, and iterative development to work better inside corporations. Beecham's research supports what I experienced this summer in utilizing agile methodologies to streamline my software development work and minimize miscommunication with my team.

A study by Mishra (2023) compared projects that used agile frameworks to ones that did not. The study did a deep dive into improving agile methodologies by combining other frameworks, explaining that agile frameworks have a significantly higher project completion rate and lower failure rate. I advocate for agile frameworks in the corporate environment, and having studies showing the effectiveness of these frameworks provides additional evidence in support of my premise. Since I am primarily discussing my personal experience, a plethora of sources citing the quantitative evidence—success rates, productivity boosts, etc.—supports the usefulness of agile frameworks

### **3. PROPOSAL DESIGN**

This section introduces a proposed methodology to test the effectiveness of agile frameworks through CS students.

#### **3.1. Design Introduction**

For my proposed project design, I plan to demonstrate the effectiveness of agile frameworks particularly in corporate America, through a mock experience. More specifically, I plan to design an experiment to test my premise that agile frameworks truly boost productivity and minimize miscommunication in the corporate environment.

#### **3.2. Design Implementation**

To do this, I plan to have two groups of Computer Science students with a heavy technical background assigned the task of

completing a sizeable coding deliverable such as a real-time stock monitoring application or dashboard displaying weather data. One group of students will be required to implement agile practices such as requirements gathering, daily standups, having a scrum master, and keeping track of tasks through stories. The other group will simply be free to organize themselves in the best way they see fit to accomplish the deliverable.

#### **3.3. Design Evaluation**

To analyze which group performed better, three judges will be selected to evaluate the groups' deliverables: an industry professional, a CS professor, and a CS student at an accredited university. The categories for judging will be quality of deliverable, readability of code, and quality of presentation. The last category is in reference to a required presentation that the groups will give regarding their deliverable and how it meets/exceeds the requirements. The judges will choose a winning group based on their evaluations.

In addition to judges, the group members themselves will analyze their experience. Group members will be required to fill out a survey with questions related to their experience on the team based on the following categories: team dynamic, overall experience in the group, and group improvements. Team members will also give evaluations to other team members regarding each member's productivity and attitude toward the group.

#### **3.4. Design Conclusion**

My goal is that this experiment design will give a clear picture demonstrating agile frameworks facilitate a flexible structure that allows students to streamline their work process. Especially in technical fields, it is extremely important not to lose sight of non-technical requirements and soft skills that are imperative to have in the corporate

environment. The judges' feedback as well as that of the individual group members will be invaluable as I seek to integrate and advocate for agile frameworks.

#### **4. ANTICIPATED RESULTS**

My hope for the results is that they show agile frameworks allow technical teams to produce better quality deliverables and foster better team dynamics. I anticipate that the group of students who use agile frameworks will produce a better quality deliverable, have readable and understandable code, and experience a positive team dynamic. Furthermore, I anticipate that the judges will see a stark difference in the quality of deliverables, and the internal group evaluations will show a better team dynamic for the group that used agile frameworks.

In terms of specific metrics, I plan to have both the judges and the teams use a Likert-type numerical scale from 1 to 5 (1 being the worst and 5 being the best) for their evaluations. Doing this will allow me to compare the average score, standard deviation, and other basic statistics to get a basic quantitative measure of each team's performance. Numbers are important, so I will investigate other measures to translate subjective fields into quantitative factors.

#### **5. CONCLUSION**

In our modern age, where software controls almost every aspect of our lives, the development of software needs to be streamlined, efficient and effective. Unfortunately, the reality in corporate America is that software development is often slow, inefficient and ineffective.

My internship experience on a professional software development team at Freddie Mac this past summer gave me a potential framework using agile methodologies to help improve the process of software development

in corporate environments. My hope is that by adopting these methodologies, more companies will be able to streamline their software development and, thereby, produce better value for their customers.

#### **6. FUTURE WORK**

Some next steps for my project would be to accomplish the proposed experiment I designed. The Computer Science Department at UVA is very open to student research, and I have formed some great connections with staff and faculty who could help me further refine my design, as well as implementation. My hope is that conducting this experiment with UVA students will produce results showing the effectiveness of agile frameworks in accomplishing software projects. Furthermore, this will hopefully lead to agile frameworks and methodology becoming even more integrated into the UVA curriculums.

#### **7. ACKNOWLEDGMENTS**

I would like to thank Freddie Mac for providing me with a great internship opportunity, and I would also like to extend a special thank you to my managers Francine Heisler and Bassel Sibbay for incorporating agile frameworks into our software team. I would also like to acknowledge the CS department at UVA for their incorporation of agile methodologies into their curriculum.

#### **REFERENCES**

- Beecham, S., Clear, T., Lal, R., & Noll, J. (2021). Do scaling agile frameworks address global software development risks? An empirical study. *Journal of Systems and Software*, 171, 110823. <https://doi.org/10.1016/j.jss.2020.110823>
- Mishra, A. and Alzoubi, Y.I. (2023). Structured software development versus agile software development: a comparative analysis. *Int J Syst Assur Eng Manag* 14, 1504–1522 <https://doi.org/10.1007/s13198-023-01958-5>