

Thesis Project Portfolio

Leveraging Communication: Learning and Achieving in an Agile Workplace

(Technical Report)

“Crowd” Computing – Volunteer Computing, Cryptojacking and their Policy Needs

(STS Research Paper)

An Undergraduate Thesis

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Ryan Pope

Spring, 2022

Department of Computer Science

Table of Contents

Sociotechnical Synthesis

Leveraging Communication: Learning and Achieving in an Agile Workplace

“Crowd” Computing – Volunteer Computing, Cryptojacking and their Policy Needs

Prospectus

Sociotechnical Synthesis

In the modern software-driven world, the responsibilities of every software engineer are not insignificant. Even software written by entry-level engineers can potentially affect large groups of people – a fact which is often lost in the noise. As an intern at Yext in the summer of 2021, I was frequently pushing code to production with minimal code-review, not only impacting internal systems at my company but also the publicly-facing pages of many of my company's clients. While the accountability I maintained throughout my internship was expected, it was not necessarily required or explicitly mentioned by my superiors. The downfalls of this unspoken expectation of accountability can be seen in projects like Coinhive, in which software engineers took advantage of the power of web browsers to mine cryptocurrency without the users' knowledge. This revealed a need for an external driving force of accountability in software engineering, one example of which I discuss thoroughly in my research topic.

In my technical paper, I detail the progression of my summer-long project for improving Yext's internal tools. Through a long process of collaboration, communication, and expansion of my own skills, I was able to develop and deliver a project which had a significant impact on the efficiency of one of Yext's internal processes. Though this project's development process was not without a fair number of obstacles, having a number of senior engineers helping me throughout the summer not only helped me overcome them, but also helped me to develop my soft skills to better communicate with my fellow engineers in the future.

In my research paper, I examine several case studies in volunteer computing, a field which has been developing rapidly over the past couple decades. Volunteer computing is a computing model which capitalizes on the massive quantity of computing hardware which exists in the modern day, allowing people to volunteer their computing resources to contribute to some

larger computational cause. One of these case studies, Folding@Home, reveals the potential for major developments to be made in protein simulations – by 2020, the project was one of the world’s fastest computing systems. On the other hand, the computing potential of volunteer systems also opens the door for bad actors to take advantage. The other case study, Coinhive, involved non-consensual use of web users’ computing resources, signaling a need for more distinct legislation in the field of volunteer computing. I go on to draw inspiration from adjacent fields and propose a new legislative model that can better protect users while simultaneously maximizing the potential for volunteer computing to continue solving some of society’s largest computational challenges.

I feel that my work discussed in both my technical and research topics has given me a greater perspective into accountability in software engineering. While I am certain there is still work to be done in developing appropriate legislation to enforce this expectation of accountability, I feel my research paper provides a solid starting point for future researchers to continue investigating volunteer computing policy.