

## **Personal, Background, and Future Goals Statement**

I like understanding things that scare me by dissecting them and parsing their inner workings. Software engineering is really scary to me. Software is all-encompassing, and software engineers are the ones tasked with keeping the wheels turning on our digital lives. I find the challenge of understanding and improving the software engineering process fascinating because it is the most complex and dynamic process I know of. From one angle, it's just writing code; then from another, it's creating and maintaining large-scale code structures; from another, it's debugging and building on undocumented legacy code; from another, it's the social coordination of complex technical tasks. Software engineering research sparked my interest because it's so many different things simultaneously, like a kaleidoscope of challenges. I have explored various research methods, and I enjoy empirical methods the most, first during high school doing clinical research in medicine, then in Computer Science. Empirical research makes me feel safe like I'm laying bricks down on top of previous work, slowly expanding the reach of human knowledge.

I want to do empirical software engineering research studying developers and the challenges they face. I have had multiple opportunities to work within this area, and I find the work very interesting and satisfying. Within this general area, I have many broad interests that I have explored within my undergraduate work. My research has explored topics including the disengagement of open source software (OSS) contributors, the affects of working from home (WFH) on developer team productivity, and the use of natural language processing (NLP) to identify patterns in the issue comments of OSS contributors. I see the backbone of my interests as identifying friction points in the developer experience, like a mechanic checking hinges for spots that need WD-40.

OSS is commonly considered the backbone of modern software. OSS allows for rapid innovation by reducing the overhead need to build new projects. Within OSS projects, there are often established contributors at the center of the team who are responsible for the majority of the work and coordination efforts. When these established contributors disengage, knowledge is lost and the project is less likely to survive.

The critical yet fragile nature of established OSS contributors motivated my first project to identify and explore factors that could predict their disengagement which I did during my first Summer at the REUSE program in 2018. I identified established contributors who had disengaged and surveyed them about why they left the OSS community. I then used their responses and previous research to inspire factors of disengagement that I operationalized using GitHub trace data and public CV data. Then I designed a survival model to determine which of these factors could successfully predict disengagement. Experiencing a job transition was the strongest predictor of disengagement. We also found that different kinds of core contributors (volunteers vs. paid) tend to disengage for different reasons. Other factors including the popularity of the projects contributors work on, when they work, and how much work they do can also be used as predictors disengagement. This work was published at the 15th International Conference on Open Source Systems (OSS '19) [1]. Having the opportunity to present this work was a great experience that taught me how much I enjoy discussing interesting questions and collaborating with my peers.

Beyond the survey and model findings, this project served as a proof of concept for operationalizing developer-related factors using publicly available trace data to study disengagement. It also inspired questions about what else could be used to help understand contributor disengagement. In my honors thesis project, I'm currently investigating the impact of issue comments. For this project, I am expanding on an ICSE '19 paper which built a classification tool for GitHub issue comments [2]. Based on the classification problem at hand, I believe a slightly different model is called for than is used in the original paper. In addition to my interest in empirical software engineering research, I am also interested in data

science and machine learning (ML). I believe finding the right model for a project, whether it be logistic regression, or a recursive neural network is akin to a fine art. I like to call the process of identifying, building, and evaluating the models farm-to-table ML. I am using PyTorch to build a deep learning model that includes a Long Short Term Memory (LSTM) layer to create the comments' sentence-level embeddings, followed by a regression layer for classification. The LSTM layer will learn to develop useful sentence-level embeddings using the errors from the regression layer via backpropagation. This updated model will create improved sentence embeddings, which I predict will improve the performance of the model.

I will identify OSS contributors who have disengaged, pull their issue comments from GitHub, and classify them using the improved model. The classified issue comments will be used to determine if any interesting patterns tend to signal disengagement. These patterns will be compared to those of non-disengaged contributors to see if there are differences in the patterns.

Branching out from researching disengagement, in the Summer of 2020, I studied developer productivity during my internship at Microsoft Research with Tom Zimmermann. One of the biggest challenges faced by developer teams isn't writing code, but rather coordination and communication. Team culture factors like quality of communication and social connectedness are factors that affect team productivity [3]. These team culture factors are often affected by the team's geographic distribution. For example, at a large software company, many teams are used to interacting face to face, knocking on doors when there are questions, and participating in low effort social events like eating lunch together. However, when Coronavirus struck, many of these teams suddenly had to go remote. The way they work and interact had to rapidly be refactored. This led to the question of how working from home affects team productivity?

Using my survey design experience, I created a survey asking developers from a large software company who were working from home about how various team culture factors had changed, and their perceived change in team productivity. From this survey, we learned important insights about the day-to-day challenges that are affecting team productivity, like the ability to speak up in team meetings, which used to be taken for granted. I then designed a model to determine which factors were most effective at predicting change in perceived team productivity. This work resulted in a paper submission that is currently under review.

Looking forward, I want to continue working on questions within the broad area of empirical software engineering research focused on developers. I am particularly passionate about studying OSS ecosystems, how people collaborate, and the sustainability of communities and teams. I also enjoy exploring the day-to-day challenges developers face, developer productivity, and other technical topics.

I have a passion for teaching. When teaching students, I always learn something new, as I experienced as a TA. I also find pedagogy to be a captivating topic since it is so subjective yet essential to the learning process. I also think diversity is very important to the Computer Science community. I personally know how frustrating a negative initial experience can be, and I want to make sure others have a better initial experience, especially students who do not have preexisting experience, or a different background than the average Computer Science student.

Thank you for your consideration.

## References

[1] C. Miller, D. G. Widder, C. Kästner, and B. Vasilescu, "Why do people give up flossing? a study of contributor disengagement in open source," in IFIP International Conference on Open Source Systems. Springer, 2019, pp. 116–129.

[2] D. Arya, W. Wang, J. L. Guo, and J. Cheng, “Analysis and detection of information types of open source software issue discussions,” in 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). IEEE, 2019, pp. 454–464.

[3] S. Wagner and E. Murphy-Hill, “Factors that influence productivity: A checklist,” in Rethinking Productivity in Software Engineering. Springer, 2019, pp. 69–84.