

Thesis Project Portfolio

Maintenance vs Growth: Two Sides of the Same Coin

(Technical Report)

The Exploitation of the Open-Source Philosophy

(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

Eric Knocklein

Spring, 2023

Department of Computer Science

Table of Contents

Sociotechnical Synthesis

Maintenance vs Growth: Two Sides of the Same Coin

The Exploitation of the Open-Source Philosophy

Prospectus

Sociotechnical Synthesis

Introduction

This portfolio is divided into two parts. The first, my technical report, is a discussion of my time as an intern with a Virginia-based software company, and how this experience relates to the debate between software growth and software maintenance. The second, my STS research topic, is a detailed analysis and discussion of the ethics of how proprietary software companies interact with the Open Source Software community.

Technical Topic: Maintenance vs Growth

In the development of software, it is crucial to continually improve. A software that receives no updates is likely poorly secured and not marketable. This is especially true for web-based applications, such as the one that I worked on in my internship. In systems such as this, maintenance and growth are both viable and important methods of improving the product. In the paper, I discuss how these concepts apply to the software I worked with.

I start by discussing relevant research within these fields. Next, I discuss the various systems of the company and how they relate to the two main concepts. Within these systems, I identify some shortcomings in guidelines or the enforcing of the guidelines that were in place. These shortcomings hindered the company's ability to both grow and maintain the software. The first system I analyze is the ticketing system, the system through which tasks are assigned. Next, I analyze the quality assurance system, which relates directly to the maintenance of the software.

The structure of a software system's code is crucial for both its maintenance and growth, and it is this structure that I analyze next. I identify various improvements that could be made in the code's modularity and maintainability. After this, I discuss how the work I did fits into these topics as well as the expected outcomes of the work.

I conclude with some ideas for future work and how the University of Virginia prepared me for this work, noting some discrepancies between the structure of the workplace and the structure of school-work that may hinder a smooth transition into the work environment.

STS Topic: The Exploitation of the Open-Source Philosophy

In this section of my portfolio, I analyze the open source philosophy and its interplay with proprietary software vendors. I start by defining the open source philosophy and how multiple different open source organizations define its tenets. Within a review of the literature surrounding this topic, I identify key stakeholders of the systems I am analyzing. I also identify their desires and, crucially, how these desires interact between the stakeholders.

There are two other perspectives that are also made clear in the literature review. The first is a view of the larger system in which these interactions take place. The second is a view of the security concerns that are common in these interactions. To better understand how to analyze the system in question – the open source software (OSS) ecosystem – I review literature that provides a framework known as the Motivation-Practice Framework, which I use throughout the paper.

After establishing some of the crucial facts about the topic in question and a lens through which to interpret them, I analyze the system. I draw the motivations to their logical conclusions,

identify a power imbalance within the interactions between OSS communities and proprietary software communities, and discuss how these things impact the end-users of the software.

Most of the problems I identify arise from the power imbalance between the OSS community and the proprietary software community and the difference in motivation between the two communities. These problems include the unjust extraction of OSS innovations from the community, the “brain drain” of OSS developers into proprietary software, the disregard for licensing laws in the usage of OSS in proprietary software, a decrease in software security due to improper use of OSS, and a decrease in the maintainability of software due to the improper use of OSS.

Lastly, I discuss some methods of rectifying the issues that are made apparent in my analysis. These solutions are both legal and cultural. First, I propose a change in copyright laws that would make it more difficult for OSS to be unfairly extracted from the OSS community. Secondly, I discuss how the use of OSS experience in the hiring processes of proprietary software companies is more complicated than expected. Lastly, I discuss a concept that was explored in one of the readings within the literature review: minimum wage for creative work.