Thesis Project Portfolio

Enhancing Test Management and Quality

(Technical Report)

Organizational Structures and Vulnerabilities in the Software Industry

(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

August Diamond

Spring, 2024

Department of Computer Science

Table of Contents

Sociotechnical Synthesis

For my capstone project, I wrote about my experience working as an intern with the quality assurance team of a financial technology company. The company wanted to improve the efficiency of their test suite, and I helped them do this by revising and automating software tests.

Computer software has become a staple of modern life, used widely in everything from entertainment to education to healthcare. Given this increasing reliance, software failures have the potential to be devastating. When such failures occur, we as engineers tend to focus on the immediate technical problems, the questions of "what line of code was the fault on?" or "how can we automate tests for this error going forward?". These are the questions that existing software failure prevention practices, like the software testing I performed in my capstone project, are designed to handle. Comparatively little research has been done on the social, or more broadly, non-technical components of software failures, such as how the developers responsible for the code were managed or which stakeholders' goals were prioritized during development. While the objectivity of the technical questions is compelling, it would be careless to neglect the social side of software failures, as we cannot divorce the quality of software from the inherently social process of its development.

To work towards a deeper understanding of the nontechnical components of software quality, I gathered and analyzed 30,891 reports on vulnerabilities found in specific software products, then looked for how the vulnerability severity values assigned in these reports related to the social structures of the responsible organizations. I found that software developed by open-source groups tended to have less-severe vulnerabilities than products produced by corporations, and that the balance of power between stakeholders in corporate organizations, as reflected by compensation and job satisfaction ratings, did not significantly influence vulnerability severity. The social construction of technology (SCOT) theory and its notion of interpretive flexibility, whereby interactions between stakeholders guide the technological development process, can be used to provide a social explanation for these findings. SCOT suggests that the difference in management between open-source and proprietary groups may be related to the differences in quality between these projects.

Applying the results of my STS research to the software industry may yield new non-technical practices that can supplement software testing in the development of high-quality software, e.g. adjusting the balance of power between different stakeholder groups involved in the project.