Sociotechnical Synthesis

(Executive Summary)

Software Verification and Regulation in Safety Critical Systems

A bug in Netflix might end a movie session, but a bug in flight software might end a life. Generally speaking, there is a tradeoff between the correctness of the software, and the time and effort spent looking for imperfections. In some systems, like aviation, healthcare, and autonomous vehicles, the weight of failure is so high, that it encourages great lengths to be taken in order to prevent failure. Being inspired about software analysis in an undergraduate compilers class, I decided to focus on and research software safety for my capstone. Throughout my sociotechnical research, I examined technical capabilities that allow for greater guarantees of software correctness in the form of automated provers. For my STS research I examined the sociotechnical landscape of software regulations in safety critical systems, with a focus on autonomous vehicles.

My technical research looks into the viability of automated software provers as they relate to medium and large scale software projects. Automated provers allow software developers to write specifications for their source code in a first order logic system, and then the provers attempt to verify if the code matches the conditions laid out in the specification. This ensures that the software "does what it is supposed to do" in the sense that the code is verified to correctly carry out certain tasks (this technology says little about whether or not those tasks are defined correctly). The specific verification tools I looked at in my research were Frama-C and the ANSI-C Specification Language (ASCL). At the time of starting my research, many of the examples and projects that used these technologies were mainly for educational or proof of

concept purposes, often using somewhat contrived examples to show the power of the automated provers. The goal of the research was to conduct a case study, wherein I applied the ASCL to an existing C program, that had not been written with verification in mind, to gain insights as to how this technology can be adopted in more real world settings. Throughout this process, I was able to generate a list of suggestions for how to improve such tools, such as the introduction of predicate libraries, macros, and language tooling to allow for easier writing of function specifications, which is by far the most costly part of the verification process.

While the verification technology is very powerful, and the technical research showed promising results, the areas where it would be relevant, safety critical systems, rely on regulations. Software regulations are similar to a dual sided blade when it comes to improving the quality of software. On one hand, they are essential for enforcing practices that promote safety that companies might be enticed to overlook. On the other hand, software regulations can be slow moving, and might prevent newer, and safer, technologies from being used. For the STS portion of my research, I looked into the regulatory landscapes of both the Aviation and Autonomous Vehicle industries to understand how the autonomous vehicle industry, which is very behind in terms of regulations, can move forward. The most interesting results that I found were how the Autonomous Vehicle industry is viewed less as a safety critical system when compared to the aviation industry, but rather as a means of harm reduction, where driving is already dangerous, and any improvement is a good thing. This way of thinking about AVs has led to a lack of robust guidelines and regulations, which may become detrimental in the future as the industry grows.

Oftentimes as engineers, we get pigeonholed into thinking about technical capabilities that we are producing. This poses multiple problems, in that there is a lack of strong

understanding of the problem you are attempting to solve, the process of implementation and adoption remains unclear, and there is a failure to examine broader ethical impacts of the capability being created. Engaging in STS research helps to resolve all of these problems. By engaging in STS research, I gained a better understanding of the problem. That it is less so the capabilities of verification that are falling short, but rather the societal pressure to regulate the Autonomous Vehicle industry. Similarly, I gained insights into industry norms change, and how industry norms get adopted into federal regulations. Finally, taking a step back to focus on the cultural and organizational dimensions allowed me to better understand how complex the regulatory system is, as well as the various weaknesses that it currently has. Similar to the case study on Hurricane Katrina, the system is far too complex for any one person to understand, but the clear result is that the existing system is not sufficient.