Andy Shin

Software Engineering for UAVs Spring 2019

UAV application is relevant to the 'Therac paper' because of the physical component of UAV development. Drones are objects which can cause harm, and they require similar care in coding as the Terac-25 required. The article specifically mentions that sophisticated modeling and analysis tools are required to analyze data. It makes sense that in this class we are learning to write software for these UAVs in a simulator before attempting to put this code on physical drones.

One of the biggest things seems to be having less confidence in things such as risk assessments and the safety of software. The article also mentions that the Navy operates on a different mindset where the design is made in the worst-case scenario, opposed to the average case. This will likely help us to install safer code with safe-checks and other mechanisms to catch the worst-case scenario. The Therac-25 software itself also didn't contain any self-checks or error-detection which led to the fatal radiations.

Although UAV development is exciting in the growing field of cyber-physical systems, the 'Therac paper' helps remind us of the dangers in overconfidence of safety stemming from the reliability of software. As computer scientists, we need to be better educated and aware of safety issues to make more fail-proof software for UAVs.

I chose to take this course because it seemed like an elective that stood apart from the other computer science electives. It seemed interesting and was also a course that would promote project-based learning. I believe if I can find myself engaged and challenged with a project that I find myself proud of to share in both interviews and friends and family would make this course a success for me.