Blake Collins

Data Structures

October 26, 2019

CSCI 315

Grade: 100

Dangers of Technology

Between the years of 1985 and 1987, there was a terrible string of incidents where patients were inflicted with heavy doses of radiation due to heavy programming errors because of little testing of the product as a whole unit. The machine in question is the Therac-25, a radiation therapy machine, that caused several patients severe injury and even death in some instances, due to its lack of testing. (Wikipedia) This incident raised awareness of the dangers of irresponsible programming, especially when it concerns a system that manages hazardous materials, including but not limited to: radiation, chemicals, electricity, heat, etc.. As such, several laws were made to prevent further incidents, and the programmer was held accountable for the incident. However, not all things are so clear cut, and no programmer or company can test for literally every problem, especially when those problems may be caused by time itself. Companies must hold themselves accountable, while maintaining reasonable standards that allow for them to continue to make a profit.

It is unrealistic to ask any product to be absolutely perfect, and to fulfill every promise that has been assigned to it, but it is certainly more realistic to believe that every product be tested as best as it possibly could be, and very reasonable on the part of the customer to believe that the product or service that they purchased will not cause them harm, permanent or otherwise. The chances of such instances can be reduced or eliminated by having company rules state that there is a clear and firm testing period for any product set to be released to the public, where the time presented varies depending on the company and product. Of course, there can be exceptions to that rule under extraneous circumstances. If the technology was designed to combat some plague or outbreak, then it would be best to administer it as soon as possible, especially if there is no easy way to test it. People survived the terrible epidemics thanks to pockets of people who thought they knew what they were doing after the first dozen times, and technology could be used the same way in those circumstances. I would try it if the chance of death was weighed against just death or another critical affliction.

In most circumstances, however, there would be no need to consider the question of using such a dangerous piece of equipment, from both a customer perspective and a company perspective. The customer, as stated above, could potentially suffer critical or fatal injury from such a faulty device, and the company would be held accountable for any harm that befalls the consumer of its product. The ACM Code of Ethics in this instance, highlights that "harm" can be any negative consequence and being in violation of that could put a company in serious trouble. (ACM, section 1.2) While this might be softened in dire circumstances, those circumstances hardly ever come up in an ideal society.

Engineers hold themselves to a certain standard and have certifications that verify their individual ability to maintain this standard. However, software engineers and programmers do not. I believe that having a mandatory safety certification would instill confidence in companies about the validity and skill of their programmers as well as incentivize those companies to hire new programmers who have proven themselves with those certification tests. This should also reduce potential problems in the future that could bring someone harm. "When you build a new house, make a parapet around your roof so that you may not bring the guilt of bloodshed on your house if someone falls from the roof." (Deuteronomy 22:8)

Works Cited

"The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society." *Code of Ethics*, https://www.acm.org/code-of-ethics

King James Bible. Holman Bible Publishers, 1973.

"Therac-25." *Wikipedia*, Wikimedia Foundation, 2 Oct. 2019, https://en.wikipedia.org/wiki/Therac-25.