**Therac-25 and the Boeing 737-Max**

Anthony Wong

Georgia Institute of Technology

CS3001: Computing & Society

Professor Melody Moore

October 3, 2022

The two Boeing 737-Max crashes of Lion Air Flight 610 and Ethiopian Airlines Flight 302 were preventable tragedies caused by complacency, overconfidence in software, and lack of defensive design. In describing these phenomena as it relates to this type of aircraft, we first need to delve into the technical causes that downed the two planes in 2018 and 2019 respectively.

Cost savings are very important for nearly any company, and Boeing is no different. In pursuit of lower and lower costs for flying, airplane engines have gotten larger and larger, since this concept increases efficiency as described by the principle of Carnot efficiency (Travis, 2021). As these engines get increasingly large, they risk colliding with the floor. The 737-Max combated this with architectural changes that fixed this issue but also caused the aircraft to have a propensity to pitch up, causing aerodynamic stall when power was applied to the engine, among other scenarios. Boeing designed a software solution called the Maneuvering Characteristics Augmentation System (MCAS) to remedy these issues by pushing the nose of the airplane downward when it thinks it is going into a stall. Malfunctions in what are called angle-of-attack sensors that determine this, along with the inability for a pilot to override this sensor reading are what caused the two aforementioned flights to crash in a nosedive (Travis, 2021).

With events such as this, often times we can look towards the people “at the top of the food chain” for root causes. This case is no different, as it seems complacency within Boeing was an overarching reason why many of the issues down the line occurred in the first place. For all the companies that exist in the aircraft industry, Boeing may perhaps be the most recognizable or successful. The establishment of this credibility is no doubt part of the reason why the Federal Aviation Administration both rushed their safety reviews and deferred much of the work back to Boeing itself. This general sentiment can be seen from Former Boeing CEO Dennis Muilenburg who said that they followed the same usual steps they always had which had previously consistently produced safe and reliable airplanes (Gates, 2022). Along of course with malintent, you can see why this attitude to aircraft design would lead to major oversights, inadequate validation testing, and rushed/incomplete safety assessments. This kind of complacency was also present in the case of the Therac-25 where the AECL became complacent about the safety of their devices given that, as a medical physicist wrote, “the use of medical accelerators has been remarkably free of serious radiation accidents until now” (Leveson, 1993).

Another causal factor of the failure of the Boeing 737-Max is the clear overconfidence in software within the design of the aircraft’s systems. Software was not the only way that Boeing could have fixed the aerodynamic issues that originated from their design changes. There was a more expensive solution that involved “extensively modifying the airframe to accommodate larger engines” (Travis, 2021). This includes things like longer landing gear, more wing dihedral, etc. Because that type of change is more costly and takes a significant amount of time to develop, Boeing went with MCAS instead. The idea that this single software could be a substitute for such substantial structural changes shows this overconfidence in software. This same idea was present in the case of Therac-25, where designers made the software an integral part of the whole device. Similarly, this kind of overconfidence and subsequent overreliance on software meant that malfunctions resulted in catastrophic damage (Leveson, 1993).

Finally, another very important causal factor was the lack of defensive design when it came to the Boeing 737-Max. As we’ve previously stated, the insistence on using software alone is proof of this. A proper design would at least supplement the software with some kind of hardware or structural help such that we ensure that if software breaks, we have something to fall back on. Additionally, is the lack of defensive design within the software itself. A big issue was that the MCAS system only relied on the sensors on one side of the plane, meaning that decisions would be made from one sensors’ readings (Travis, 2021). Given the environment these sensors are in, faulty readings are not uncommon. If there was defensive design, they could’ve utilized other sensors that could provide the same information to ensure decisions don’t hinge on a single sensor. Finally, the near literal nail in the coffin is how MCAS denied the pilot’s “sovereignty”, meaning that it denied pilot intervention. The system was “blind” to any other evidence that it was wrong (Travis, 2021). With defensive design in this case, the pilot is the one physically in the cockpit and should have the option to step in. If that were the case these flights may not have been flown into the ground. A lack of defense design was also present in the case of Therac-25, where the lack of any form of error-handling and error-detection allowed overdoses of radiation to occur (Leveson, 1993).

After discussing the technical causes of the failure of the Boeing 737-Max, it’s clear to see that a combination of complacency, overconfidence in software, and lack of defensive design were all present, as they were in the case of the Therac-25, and resulted in the tragic crashes of Lion Air Flight 610 and Ethiopian Airlines Flight 302.

**Bibliography**

Gates, D. (2020, November 22). Q&A: What led to Boeing's 737 max crisis. The Seattle Times. Retrieved September 30, 2022, from https://www.seattletimes.com/business/boeing-aerospace/what-led-to-boeings-737-max-crisis-a-qa/

Leveson, Nancy. (1993, July). Medical Devices: The Therac-25. *University of Washington*.

<http://sunnyday.mit.edu/papers/therac.pdf>

Travis, G. (2021, July 29). How the Boeing 737 MAX disaster looks to a software developer. IEEE Spectrum. Retrieved September 30, 2022, from https://spectrum.ieee.org/how-the-boeing-737-max-disaster-looks-to-a-software-developer