Cristalina Nguyen

Dr. BJ Johnson

401 Software Engineering Laboratory

October 1, 2019

ValuJet Article Summary

In May of 1996, ValuJet Flight 592 crashed into the Florida Everglades and killed all passengers on board. The crash was caused by, what author William Langewiesche called the "most elusive kind of disaster", an onboard fire triggered by illegally stowed chemical oxygen generators that were not safely capped. In this article Langewiesche states that there are 3 types of airplane accidents: procedural, which "result from single obvious mistakes, that can immediately be understood in simple terms, and that have simple resolutions", engineered, which "consists of those surprising materials failures that should have been predicted by designers or discovered by test pilots but were not…defy understanding, but ultimately they yield to examination and result in tangible solutions", and finally system accidents, which "lie beyond the reach of conventional solution" and is how he explains Flight 592. Langewiesche then goes into detail about the characteristics of the crew, the airline, and more specifically the flight itself as he hints at the combination of factors that led up to this unfortunate event. He also timelines the conversation between the co-pilot and radar controller leading up to the crash itself before explaining his experience being at the crash site during the investigation period, as well as the specifics behind the cause of the crash.

In short, ValuJet had hired a company called SabreTech to perform some maintenance work, who actually hired contract mechanics from other companies to fill three-fourths of the positions on the project. SabreTech handled the extraction of the oxygen generators and replaced them into the aircraft without caps which is what caused the fire, but the investigation, understandably so, could never pin the entire accident on a single person or persons. A large question posed by Langewiesche was why the FAA's administrator, an airline boss named David Hinson, did not shut down ValuJet before the accident. ValuJet was eventually grounded indefinitely 5 long weeks after the accident and proved that while ValuJet had some blame in the accident, the FAA's neglect towards their duties were much more crucial.

Engineerspeak is a topic that Langewiesche touches on, being that many of the instructions were written in a language that the workers did not speak or were too lengthy and complicated for them to try and figure out. This is very common in software development and applies to situations where communicating with stakeholders, investors, designers, or product managers requires different language than communicating with engineers would. The ValuJet crash was caused by many factors, one of them

being a lack of communication. While many situations do not end as tragically as ValuJet Flight 592 did, bad communication can be one of the top problems that can arise in the midst of a software engineering project and could contribute to this kind of disaster. It can oftentimes delay, impede, or completely derail a software engineering project that could have otherwise gone very well. Langewiesche also mentions that "mechanics who are too careful will never get the job done. The airline system as it stands today requires people, in flight or on the ground, to compromise, to make choices, and sometimes even to gamble." This is an interesting statement because it applies to software engineering so well. In an engineering project there are deadlines, guidelines, rules, stakeholders, and many other limiting factors that can contribute to the behavior that Langewiesche described. The airline system, while it has more dependencies than most software projects, very much assembles the engineering system. It requires compromise, choices, and sometimes gambling as well and these, like communication, are factors that could contribute to disaster.

Although it was not the case with Flight 592, software can contribute or even directly lead to this kind of disaster as well. Airline systems and airplane cockpits have always largely depended on software whether it be the communication system from plane to control center, or the software system that tells the plane where it is and where to go. Malfunctions and errors in this type of software can wreak havoc and even cost lives if proper care is not taken to make sure that these kinds of software systems are always running smoothly.

Ensuring that design is always user centric and takes care of edge cases can also help prevent this type of disaster. As it was shown in the Flight 592 accident, bad design can lead to misuse of products or systems and cause organizations to fail. All systems, especially airlines, should be designed so that in situations of emergency and those edge case situations, the design is capable of communicating to the user what should be done. In the case of this specific accident, if the instructions for the mechanics who were performing maintenance on the airplane were given clearer instructions or a better designed system, smaller errors may have been made.

The "Lessons of ValuJet" article by the *Atlantic* can apply to software discipline as well as other ethical dilemmas and engineering faults in that failure almost always results in a poorly thought out and badly designed system. This includes communication, product and procedure design, lack of informed workers or members, and more. Preparation and investing lots of forethought into a project is always going to better the result, ensuring that thought is put into each system within the large system. This idea of the failure of the "systems of systems" translates to me as a combination of failures that led to the big one.

Each small mistake or sacrifice that was made along the way of Flight 592 led to the disaster but this experience can be used 23 years later as a learning experience for people from airline system

executives to small university group project members like us. Going forward as a leader of my 401 team and a member of any team going forward, I will make sure to remind myself daily of the lessons learned through this article and how I can apply them to my software development processes. While it was tragic that so many individuals lost their lives for lessons like this to be learned, we can be hopeful that the lessons learned and the precautions taken from now on save many lives.

Source: www.theatlantic.com/magazine/archive/1998/03/the-lessons-of-valujet-592/306534/