Aidan Esposito, Matthew Savitt, Parker Jazayeri, Alex Gordon

Professor Johnson

CMSI 4071 Senior Project I

30 September, 2024

ValueJet Article Summary

The ValueJet crash in 1996 is one example of an event that, as tragic as it is, can be used to learn lessons about the failure of "systems of systems" and inspire ideas to help prevent such catastrophes in the future. The largest point of failure was simply a lack of communication, but the disaster also demonstrates the necessity of failsafes and accounting for even the seemingly smallest of flaws which could have a larger effect.

The article begins by giving us the perspective of Walton Little, a computer engineer and pilot, who witnessed the crash one May afternoon in the Everglades Holiday Park. By the time Walton was hit by the shock wave of the plane, 110 lives were lost and led to a poison that would be felt in generations to come. With this, the Federal Aviation Administration led an investigation into the cause, and what they found is shocking, changing the airline industry's rules and standards forever.

The accident is described to be a system accident, which can be summarized as a combination of external factors behind the incident. For example, a lack of safety precautions caused a rapid fire in the cockpit that swarmed it with smoke. This not only caused movement issues with the plane, but eventually spread the fire around the area. The plane, taking damage, then pushed into a nosedive but miraculously recovered due to possible autopilot interference.

The airplane eventually reached 500 mph and, on a right turn, crash-landed into the swamp where Walton watched in horror.

The recovery effort soon began, having news stations and technician camps being set up near the crash site. With the recovery, one of the main issues to blame the airline for became present, the plane lacked complete fire detection and extinguishing systems opting instead to rely on outdated oxygen tanks and new MD-80 engines for security and travel, starting the original fire. Even the oxygen tanks were badly mismanaged, with no one knowing what to do with them before sitting in cargo during the event. With this lack of oxygen precautions and the rapid fire, the smoke quickly rose to a poisonous level, preventing the passengers and cabin to do anything because of their lack of air. In this event, the final outcome was determined: the passengers and crew died an agonizing death with oxygen deprivation and intense burning, and the airline was to blame. The groups ending up taking scrutiny from the incident included ValueJet, the FAA, and the airline industry as a whole, losing the trust of passengers everywhere.

There are many lessons that can be learned based on this incident that we would like to mention. One of the most prevalent ideas we saw is the importance of communication between developers. As mentioned previously, the fault on the oxygen take location can be blamed on multiple different parties not knowing what to do. ValueJet gave the order for replacing the caps and just wanted results. SabreTech, the group handling installation, ignored a major safety precaution, that being the caps that kept generators from firing, which led to the main problem of the incident. The shipping clerks had no idea where to send the oxygen, leading to a mislabeling of the boxes and the final location of the cargo being chosen. The maintenance is also at fault here because, at the point of arrival, the cargo was considered too hazardous for the plane because of its long wait time yet, was still placed on the plane anyway leading to the eventual

fire. All the issues caused in the plane could have been prevented by a single party having communication and feedback to the other to keep the oxygen off the plane. ValueJet, in our eyes, is ultimately in charge because they were in the management position, but that is not to say that no one else here is at fault. It would have taken one order to stop the event from happening, and highlights the importance of communication for all projects.

Another major lesson we learned was the importance of precautions and back up plans during projects. In this event, when the fire started, there was not a single tool on the plane to prevent the fire and extinguish it when it spread. The only supplies given to the crew and passengers were badly made oxygen masks that failed to stop the smoke from spreading into human lungs. If you are not prepared for a project, then it will ultimately lead to failure. Precautions in all industries are necessary to prevent possible scenarios. This can be seen from industries such as airlines all the way to technical industries by using tests and writing error catches for certain bits of code. Having error checking to prevent even any small possible error can fix code and, in the case of this event, even save lives.

The final main idea we gathered from this incident is the process of handling complexity properly. With this event, there were so many external factors around the situation that could not be controlled. These can go from the structure of the aircraft itself, the way the airline built the plane systems, the cargo and emergency supplies on the aircraft, the supplies and training given to both crews and passengers, and the structure and modifications ValueJet put into their product. If a project becomes too complex to handle, it leads to supposedly small errors in the system. The lack of caps on the oxygen can be seen as a small error at the time, but in the larger picture, it led to a massive fire that led to the deaths of over one hundred people. If any error is ignored in a project and the complexity continues to grow with it, it will become harder and harder to scope

out the original error and fix it by the time of product delivery. In the coding space, we can see this as an example of ignoring a seemingly small bug in a program but, as time goes on, the bug increases in size affecting more parts of the program and system as it becomes more and more complex. The lesson we have taken from this part of the issue is not letting seemingly small problems in a complex system get unaddressed as, if left untouched, could become a major headache later into development.

Overall, the ValueJet crash in May 1996 can be seen as a lesson for not only installing precautions, having good communication, and managing complexity in the airline industry, but also as a lesson for projects in the software industry for the near future and beyond.