Merci Magallanes

CMSI 401

10 October 2018

Assignment #1: “The Lessons of ValuJet 592” Response

In William Langewiesche’s article, “The Lessons of ValuJet 592”, Langewiesche details the series of events that led up to the tragic crash of the commercial airliner, ValuJet 592, in 1996. In the article, Langewiesche discusses how “engineerspeak” led to miscommunication which, when paired with “incompetence and cronyism”, culminated in the loss of one-hundred and ten lives aboard flight 592. Similar to the development of software engineering projects**,** when “engineerspeak” is lost in translation and combined with the lackadaisical treatment of paperwork, the result is often malfunctioning software that have the potential to cause bigger problems.

Langewiesche opens the article with the accident from the perspective of Walton Little, a witness who dialed 911 and reported the crash. According to Langewiesche, Little “was a born accident observer—a computer engineer and a private pilot with pride in his technical competence and a passion for detail”. Little saw the plane descend into the swamp and commented how he knew something was wrong with the plane by the lack of reflection from “the lower (outboard) portion of the right wing”. Being a pilot himself, Langewiesche describes three different types of airplane accidents: procedural, engineered, and system. Flight 592’s accident was a “system accident”. Langewiesche describes system accidents as “science’s illegitimate children, bastards born of the confusion that lies within the complex organizations with which we manage our dangerous technologies.”

Although it was initially thought that an electrical fire was the cause of the crash, in actuality, “flight 592 had been loaded with a potentially dangerous cargo of chemical oxygen generators—more than a hundred little firebombs that could have caused this accident, and that indeed did.” How the oxygen generators got onboard the flight was in part due to a long chain of events such as apathetic workers mishandling paperwork, “engineerspeak” being lost in translation, and valuing money over passenger safety, which culminated into the system accident that Langewiesche discusses. Ultimately, the accident shed light on “the sort of collective relaxation of technical standards that the Boston College sociologist Diane Vaughan has called ‘the normalization of deviance’”.

“The creation of an entire pretend reality [within the ValuJet company] that includes unworkable chains of command, unlearnable training programs, unreadable manuals, and the fiction of regulation, checks, and controls” embodies the “normalization of deviance” that Vaughan describes. On more than one occasion this can also occur in the realm of software development. For example, in the early stages of development, the fixing of a small bug could be postponed continuously and eventually forgotten about. Or the bug could be approved by a supervisor who wished to push the code to meet an upcoming deadline and the bug just moves further down the line until it’s hidden amongst thousands of lines of code and causes problems that would only be found after it’s too late.

Due to the specialization of the field of aeronautics, technical jargon, or “engineerspeak” as Langewiesche characterizes it, is often used to talk shop. However, certain terms’ significance can often be misconstrued, which is exactly how the oxygen generators managed to get onboard flight 592. “When [the shipping clerk] prepared the boxes for shipping, he noticed the green ‘repairable’ tags mistakenly placed on the canisters by the mechanics, and misunderstood them to signify ‘unserviceable’ or ‘out of service,’…[and] drew the unpredictable conclusion that the canisters were therefore empty”. Due to this miscommunication and the previous mistakes made earlier, it was only a matter of time before the domino effect of the airplane crash occurred.

The field of computer science is no different in terms of misunderstanding “engineerspeak”. For example, during the creation of a software requirements documentation, the definition of the terms “will”, “shall”, and “should” might become blurred, which can lead to complicated legal problems, especially since some of the terms are more legally binding than others. Therefore, if the software requirements document states that an application “shall” be available on iOS and Android, the software developers are legally bound for the application to be available on both iOS *and* Android.

Although the tragic, the crash and following investigation “served as a necessary reminder [for all airlines] of the possible consequences of cost-cutting and complacency” by shedding light onto the “normalization of deviance” and the resulting outcome, that when profit is placed above the safety of the passengers, the passengers are the most likely to suffer for these mistakes. Similar to terms of software development, when profit is valued more than the consumer, it’s only a matter of time before a corner is cut too sharply and millions of customers’ personal and private data are accidentally released. However, in the case of ValuJet 592, the NTSB shook up the FAA, reminding “the agency of its mandate to oversee the safety of the airlines…[and] prodding the FAA into a renewed commitment to inspections and a resolution to hold airlines responsible for their actions and for the performance of outside shops.” Although lives were lost due to malpractice, because of the lessons garnered from the accident, more lives were saved due to stricter regulations and a more vigilant FAA.

From my Ethics for Engineers class, the professor brought up the idea that we, as engineers and scientists, are professionals. We’re specialists in our field because we know more than the general population about *our* field. And because we have this certain knowledge, it is our responsibility to act for the benefit of others and not for ourselves. According to the article, “The Lessons of ValuJet 592”, it seems that this is one of the implicit lessons that needed to be learnt. That those involved in the crash of flight 592 forgot that they have a responsibility to act for the benefit of others, not just for themselves to keep their average day moving along.

Overall, the compilation of carelessness, miscommunication, apathy, and sheer bad luck culminated in the ill-fated accident of ValuJet 592 in 1996. Langewiesche cites Charles Perrow who pointed out that the events, such as those that led up to the accident, “will occasionally combine in unforeseeable ways, and if they induce further failures in an operating environment of tightly interrelated processes, the failures will spin out of control, defeating all interventions.” Ultimately, the crash could have been prevented if ValuJet had been grounded at the very beginning, but once those oxygen generators were taken off, it led to a string of failures that spun out of control and became unsalvageable for the flight. But for future flights and mechanics and pilots and the FAA and the NTSB and others who learned to prevent those events from ever occurring again, it guaranteed their passengers’ safety.

Work Cited

Langewiesche, William. “The Lessons of Valujet 592.” *The Atlantic*, Atlantic Media Company, 1 Mar. 1998.