John Scott

Software Engineering Laboratory

October 10th, 2018

ValuJet Article Write Up

The article "The Lessons of ValuJet 592" written by William Langewiesche, an experienced pilot, captures the reader's attention by beginning with a chilling call to an emergency dispatcher after, as the shocked caller explains, "a large jet aircraft has just crashed." When Walton Little, the caller and "a computer engineer and a private pilot," felt the immense shock wave of the airplane colliding with Earth, 110 lives were lost, not to mention the emotional impact on the loved ones of the departed. The airplane was a twin-engine DC-9 and belonged to "ValuJet, an aggressive young discount airline," whose disregard for proper safety procedures and excessive hiring of "temporary employees and independent contractors" in pursuit of cheap air travel led to a catastrophic "system accident."

Langewiesche lists and explains the three kinds of airplane accidents: "procedural," "engineered," and "system accidents." Procedural accidents "result from single obvious mistakes" and can generally be avoided by following safety rules/procedures. Engineered accidents are caused by "surprising materials failures" that should've been prevented by testing parts and equipment or predicting how materials will, or won't, function under certain circumstances. However the third and "most elusive" disaster, a system accident, is born from the confusion and complexity of these massive organizations that we rely on to manage and ensure the safety of the "dangerous technologies" that we access.

Naturally, the public demanded to know how specifically this tragedy occurred and who are to blame. An insultingly brief explanation for how this horrific event took place is low-paid employees and contract mechanics hired on "an as-needed basis" from SabreTech, an outside company hired by ValuJet to handle maintenance, overlooked standard safety procedures and signed off on work being completed without much care in an effort to get things done by an unsafe deadline. Blame was unsurprising placed on ValuJet but, inexplicably, the Federal Aviation Administration's administrator asserted "that ValuJet was a safe airline" immediately after the tragedy occurred, linking the FAA's reputation to ValuJet and therefore taking a brunt of the blame too.

One of the first things I noticed while reading the article was the frightening number of parallels between the complexity and scale of the procedures of the organizations involved in the situation that caused the ValuJet catastrophe and the complexity and scale of the procedures required in substantial software engineering projects. For example, "SabreTech inspectors and supervisors signed off on work" certifying that proper safety steps were taken without giving much thought as to whether it was actually done correctly, or even done at all. At the most absolutely basic level of analysis, this was a direct cause of the tragedy. However, the heart of the problem definitely lies in the breeding ground for system accidents created by the confusion and inherent complexity of the several different organizations and individuals tasked with managing the "dangerous technology" associated with air travel or software development. This particular system accident I believe can be attributed to three main reasons: a general apathy for getting the job done in the most complete way possible, the impending feeling of a deadline approaching, and the deception that can form from the tediousness of paperwork.

Apathy can be a very dangerous symptom in projects of all types of disciplines if the parties involved don't feel invested in, or will benefit from the success of, them. In Langewiesche's article he states that a majority of the people hired by SabreTech to handle maintenance for ValuJet were temporary, "three fourths of the people on the project were just such temporary outsiders." These temporary employees more likely than not did not feel much passion for the work they were doing and possibly saw the job as just a simple paycheck. Relating to software development, a software engineer might not feel motivated to go above and beyond with his/her implementation or testing of a certain assignment if they are hired temporarily or won't receive recognition for the extra hard work. Apathy for getting a project or large number of tasks done can easily be caused if an individual is only hired temporarily and doesn't feel they will personally benefit from the absolute completion of the project.

Deadlines are necessary to manage massive projects with several moving parts, but can often cause individuals to cut corners if the deadline is fast approaching and can't be moved. As SabreTech hired employees nearing the end of their work period signed off on tasks being completed that were not, "In the rush to complete batches of paperwork … two mechanics routinely 'pencil whipped' the problem by signing off on the safety-cap line … certifying that the work had been done. With a deadline closing in, SabreTech mechanics, inspectors, and supervisors all signed off on work without completing it or ensuring it had been done; an extremely reckless action to take when dealing with safety precautions. Similarly, making sure no obvious security breaches could occur in a client's software by following all procedures could be an example of something that would be dangerous to not implement if a deadline was close.

Deadlines can lead to people feeling pressured to *appear* to get everything completed and result in cutting corners which can be especially dangerous with safety and/or security procedures.

Paperwork is no doubt a necessity when dealing with the substantial systems and organizations that exist in society today but can result in harmful deception. Near Langewiesche's closing statements he explains how the problem with paperwork is "not just the burden that it places on practical operations but also the deception it breeds" by stating how the pencil-whipping committed by the two mechanics, supervisors, and inspectors is part of "an entire pretend reality" created by paperwork in which "system accidents are born." Regarding the ValuJet accident, various employees signed off on work being done without ensuring it had been possibly because it was seen as just another checkbox on a list of tasks to complete. Following through on everything listed in a requirements specification document and not just checking it off the list is probably a relatable scenario in software development. The falsification of completed procedures or tasks that can result from an exorbitant amount of paperwork is extremely dangerous in our systems.

I believe the tragic "system accident" that was ValuJet 592 can be attributed to, but not limited to, three main issues that are "natural" results of these massive systems and organizations that manage our dangerous technologies. The three main issues are apathy for getting a job done in the most complete way possible when not personally invested, the impending feeling of a deadline approaching and the corner cutting that may result from it, and the deception that can form from the "checkbox mentality" of paperwork. In a similar fashion, apathy, deadlines, and paperwork can all exist in a software development environment. It's important to learn from the lessons of ValuJet 592 to prevent "system accidents" in both hardware and software systems.