James Byrne

October 10, 2018

CMSI 401

ValuJet Article Summary

ValueJet 592 was a flight that was the victim of an amalgamation of systematic errors that led to its demise. The chain of blame is so convoluted that no one person can be sourced as the origin of the problem. The flight was spotted by a observant fisherman in Everglades Park, Florida. The commercial airliner was seen diving into the marsh ablaze and going dangerously fast. All of the passengers and crew died in the crash. Investigators uncovered clues from the wreckage including a black box from the cockpit. This had a recording of audio from the cockpit that gave insight into the cause of the crash. People were recorded reporting electrical malfunctions in the cockpit, then total electronic failure. Smoke and toxic fumes started to seep into the cockpit from below. As the fumes leaked into the main cabin, panic ensued. The cause of this fire was an accidental ignition of retired oxygen mask systems in the cargo bay directly below the cockpit. What led to this cause is a chain of problems too large to fully solve.

The chain went like this. The air traffic controller might have directed the plane incorrectly. The baggage worker and pilot should have checked the cargo for hazardous waste before they let it on the plane. The people putting the air canisters should have gotten caps to put on them and marked them as hazardous waste. The cargo managers should have never signed off that the cargo was safe. Finally, the FFA should have had better inspections to enforce their regulations. The game of pointing fingers and trying to place blame on any one party quickly gets complicated. Systematic failures are often this complicated, especially when they have such a devastating result such that of ValueJet flight 592.

In the software development realm, system failures are just as possible as in the airline industry. The main variable in system failures are human behavior, whether that be laziness, impatience, or incompetence. Human error is a big part of the software industry. There are two general categories where errors can occur in software. The first are bugs in the code, and the second is organizational/communication errors. The bugs in the code relate the most directly to the workers who physically put the oxygen mask devices into the boxes without proper caps or checking their status. Bugs in your code directly affect the experience of the end user when using your software. They are the causes that affect the outcome most directly. The organizational/communication errors in software development are the ones that are the hardest to catch. Programmers have been fixing bugs in their code since they learned how to program. They are trained to continuously check for new bugs after big changes, and to write unit tests to make it easy to test their code. However, most are not formally trained in communication and organizational skills.

Communication skills often manifest in the form of requirements not met. The requirements could be given by the client you are making the software for, and lost throughout the chain of managers like a game of telephone. Managers who might not know the technical purpose of features prioritize them differently than a programmer would or scrap them entirely. On the programmer level, communication can affect how software developers work with each other. If there is not almost constant communication between all the parties developing software, it is really hard for all the parts to fit together at the end and for developers to not overlap in the work they do. Organizational can greatly affect software development as well. Unclear or unrealistic deadlines can cause pressure about finishing on time or not exceeding the budget and adding all the features that the client wants you to. This can cause people in the chain to cut corners to make up for time or money as ValueJet did.

One problem that I can see from the ValueJet article in software development is "paper pushing". In the case of the article, this was when workers signed off on inspection forms when not all of the required items on the checklist were completed. This led to hazardous waste being allowed to board the cargo bay of ValueJet flight 592. Paper pushing in software development is done when people who are supposed to thoroughly review code just sign off on it without checking it for errors or incorrect implementation. One example of this causing a system failure in computer science is when NASA lost mars orbiter costing them 125 million dollars. The issue was that while NASA was using metric units in their code, Lockhead Martin working as a contractor with them was using english units of measurement. This was a very obvious error that could have been easily solved by more strict code review when the two companies used their programs in conjunction.

Organizational strategies for software development are steps that are already being used by the industry to try and reduce systematic errors. The two main strategies in affect are the waterfall and the agile techniques. The waterfall method goes through each step once from conception, to development, then testing. The agile strategy is different because it goes through many iterations of all the steps that the waterfall method goes through. The waterfall method is very structured but can be hard to stick to a timeline with. The agile method is good at making a quick prototype which allows for time to critique before the next iteration where improvements are made. The downsides to the agile strategy is that if you are running out of time you only have a stripped down version of the final product. For preventing systematic errors, I think the agile method is better than waterfall because of the many opportunities for errors to be caught. For each iteration that you go through of the software product, more new eyes are checking your work for bugs, requirements not met, or things that can be improved. Imagine if ValueJet's oxygen canisters had gone through not one, but multiple times unloading and loading the material. There would be a much higher probability that the errors in the cargo would be spotted and acted upon.

Once we realize the systematic errors occur wherever there is a system set up, we realize the importance in analyzing these errors and trying to find solutions to them. When ValueJet's problem was compared to the likes of NASA's loss of one of the Mars orbiters, the first glance thought was "how can a oxygen system that started a fire be a similar problem to a spacecraft getting lost in space?". The systems that humans create as the backbone of every industry is extremely similar throughout. There is a chain of command in every operations with humans there introducing the possibility of human error that leads unknowingly to system errors. I think the depth of the complexity of systematic errors means that they will never be completely eliminated. I think the best we can do is learn from our lessons and try to force these system errors to become less frequent and occur in instances where they won't cause substantial damage to human life.