Brett Derham

10/2/19

Doctor Johnson

Software Engineering Laboratory

ValuJet Article Summary

The ValuJet Article describes a story about system failure and how it led to the loss of many lives. Walton Little was fishing in a lake in Everglades Park, Florida, when he saw an airplane crash land near him. Walton quickly analyzed the situation and called 911 to alert them of the crash. Walton, a computer engineer and private pilot, quickly described the plane in detail to the operator and they were able to identify it was a twin-engine DC-9 with ValuJet colors painted on its exterior. Walton had witnessed the crash of flight 592, a flight expected to deliver passengers from Miami to Atlanta. All 105 passengers, three flight attendants, and two pilots lost their lives in the crash due to failure.

The article continues to describe the three types of failure. The first type of failure

mentioned is Procedural failure, which are accidents that result from single obvious mistakes, that can immediately be understood in simple terms, and that have simple resolutions. An example of this type of failure would be a pilot driving into a thunderstorm and the plane getting struck by debris or lightening. The second kind of accident is Engineered failure. It describes failure due to material failures that should have been predicted by designers or should have been discovered by testers. At first these failures might have no explanation, but upon further examination the cause of the failure can be found materially. The final form of failure is described as Normal or Systematic failure. This kind of failure occurs through many small decisions that cause one final problem due to unpredictability and interactively complex situations. This brings up the concept of Murphy’s Law which says, “What can go wrong usually goes right.”

The ValuJet incident is an example of a Systematic failure because many small decisions were made that led to the unfortunate crash of the plane. ValuJet hired a company called SabreTech to work on their new planes that they had purchased. When SabreTech needed more mechanics, they hired outside workers to complete their promise made to ValuJet. The article claims that about 75% of workers were not a part of the SabreTech company. The article also claims that many mechanics were overworked to keep the ValuJet prices low for customers. The beginning of this systematic failure started when some mechanics chose to ignore the ValuJet work order. The work order clearly instructed the mechanics to place safety caps on expended oxygen tanks. These mechanics chose to ignore this part of the work order and signed off that they had done it anyway. This was the first dominoes in the systematic failure. These oxygen tanks were then shipped taken across the airport and were prepped to be loaded onto planes and flown back to headquarters in Atlanta. Both the SabreTech manager and the Ramp attendant didn’t think to clear the packages and make sure they were not hazardous. The packages were loaded onto flight 592 and caused the plane to go up in flames six minutes after takeoff. The proper people were punished and SabreTech soon lost their business and were forced to shutdown.

After reading this article, I believe that this is the perfect example to explain systematic or normal failure without faulting any one person with too much blame. Although many people were clearly wrong in this situation it fits the exact description Perrow gives for systematic failure. The unpredictability of the of the oxygen tanks catching fire and ending up on a commercial flight and the many faulty decisions describe the interactively complex situation and show why this is a perfect example. I believe the first person/company to cause the chain reaction that resulted in flight 592 going down was ValuJet. Their need to stay competitive and profitable by keeping their flight prices so low caused the first mistake. Their workers were working overtime and possibly disgruntled. This led to an attitude that resulted in safety caps not being installed and started the process of failure. This can directly relate to Sofware design because we will need to make the right decision at every step of the way to avoid bigger problems in the future. For example, thinking about how our website is going to function and choosing a method of operation may limit some of our decisions in the future.

Another huge problem that led to the crash of the plane was the complex terminology and linguistics of the communication between the mechanics and the ValuJet and SabreTech policies. The article talks about the word expendable and how the two sides took it to mean two different things. The mechanics thought is meant the oxygen tanks were completely exhausted, whereas the companies meant that safety caps should be placed on all tanks that were not completely full. Miscommunications like this are extremely important when it comes to Software design. We must avoid “engineer speak” and make sure that every piece of documentation that we put out is very clear and concise. It is very important that a user gets the exact experience that is described in any documentation. If not many lawsuits or complaints may follow due to the lack of proper language. For example, in terms of my application BallparkBookie, it would not be appropriate for us to claim that any bets can be made through our website. Not only to we not have the legal power to do that, but many users may be unhappy when they discover our website is strictly advice based and they cannot bet. That is why it is very important that we are clear when we write our documentation and properly describe exactly what our application does.

In this situation both pilots had plenty of flight experience and were unable to avoid the circumstances of the systematic failure. As a beginner software developer, this article left a big impact on the way I will be handling our project going forward. While the consequences of our app failing and a plane failing mid-flight are very different it still provides a good example of the dangers of systematic failure.