Jackson Watkins

Professor Johnson

Computer Science 401

10 October 2018

ValuJet Article

William Langewiesche argues that the reason behind the ValuJet 592's fall was the compilation of failures that came with the maintenance of the plane. The myriad of failures that Langewiesche calls as part of the "system accident" that were attributed to the crash of the ValuJet 592 include the multiple issues surrounding the miscommunication involved surrounding the incident of old oxygen generators to exist below the cockpit and the normalization of deviance apparent in the attempt at pencil-whipping and SabreTech employee non-adherence to work order rules and standards.

The SabreTech workers' miscommunication in preparing oxygen cannisters for the ValuJet 592 was apparent in their misinterpretation of the labeling associated with the cannisters and the "engineerspeak" associated with the notes left for other workers to interpret from.

Langewiesche also points out that SabreTech's normalization of deviance to standard work procedure plays a huge role in the incident. The SabreTech workers failed to follow standard work order procedure. This is apparent in their failure to report the empty oxygen cannisters to supervisors and their failed attempt at fixing the problem of oxygen cannister leakage by putting the cannisters in a cardboard box to prevent the leakage of the poisonous gas into the workplace air. The workers also failed to place the required plastic safety caps over the firing pins of the oxygen cannisters which is one of the instructions in the work order. Langwiesche points out that this attitude that is present in the workers at SabreTech is due to the inherent nature of the company to meet the required deadlines for delivering the oxygen cannisters to ValuJet. Langwiesche calls this action “pencil whipping” because the term implies signing off on completed work at a fast pace.

There are many other issues that Langewiesche points out that may have contributed to this chain of events which all contributed toward the ValuJet 592 crash either directly or indirectly but the point is that if any events in the pipeline go awry, issues and mistakes are likely to happen. A big reason that a lot of these mistakes happened is due to failure of communication between multiple systems of systems in the pipeline of preparing a passenger plane. If communication was well maintained in the process of preparing a plane, then workers that contribute in any way toward the plane development/maintenance pipeline would be better informed on the state that that system was left off so as to continue properly.

This problem can be solved with physical ledgers which explain the current system of systems but these physical ledgers would be a lot to handle especially when these physical copies may be lost in the pipeline when passed from one system to the other. A software solution would be the most practical with regard to documenting the updates that occur to the different items that go into the creation and maintenance of specific planes.

One software solution that I can come up with that would address the issue of communication failure in the system of systems is the creation of a database that documents the parts, maintenance, and other plane details associated with all of the planes that are owned by ValuJet. Since the ValuJet airliner essentially owns this plane, they would be in charge of maintaining the database to make it easily viewable by any third party that the company works with including SabreTech. In this fashion, the planes that ValuJet owns would have up to date information regarding the upkeep and parts associated with them. Even more, the third parties that Valujet works with would be given access to read and modify the specific data within the database. The third parties would be able to read and modify this database in an attempt to continue off of where the last modification left off. Since one of the database properties as part of the ACID properties include consistency, the database would have the most up to date information associated with it. Essentially this way of tracking the system of systems would ensure that the parties involved in the construction/maintenance of the plane would know exactly what to work on next.

Since there is a database at the ValuJet level of the maintenance of each plane that is owned by ValuJet, there should also be a record keeping of the work orders of a particular project at SabreTech. Since large mistakes that affected the whole pipeline of the maintenance of planes were made at the third party level, the third party should ultimately find a solution to track the actions made from each worker. If SabreTech, were to enact the implementation of a work order tracking system so as to track the progress of each work order it would be beneficial for other workers to pick off from where the last worker left off from. This work order information could be tracked in a database or any other accessible means so that other employees are aware of the actions made on a particular project and can continue off where last left off without any confusion.

Another software-based solution to the issue that is apparent as part of a failure of “system of systems” that Langewiesche likes to call it is to implement a software blockchain structure for the maintenance done for each plane. Blockchains mimic a back-linked list software data structure that is prevalent in banking and are great for tracking progress made on completing a project. Each block has access to the block that was created before it so every party can be cognizant of the most recent change to the pipeline of a project. This could help minimize the issues that would come with not understanding the most recent progress that was done for maintenance the plane.

Besides the software solutions to the issues surrounding the failure of the system of systems, there are many times where a failure of communication similar to the miscommunication occurred at SabreTech would happen in software engineering. A situation which mimics the type of issues that were apparent in the ValuJet crash but on the software side is if code that is not understandable from an outside perspective is not well documented with comments and can cause bugs. This program is set up for failure if the next person to review that code does not catch the error made by the most recent change. Forms of software development version control were created to help solve the problem of miscommunication on the software level. With regard to git version control, contributors to a git repository can see the most recent changes and fix bugs and ambiguous code on the fly.