Complexity Beyond Imagination: The Boeing 737 Max

If only we could go back in time…

If it were possible to wait for highly improbable, highly impactful, fatal accidents to occur and go back to the exact moment in any of a series of mistakes, we would of course do so. If some human in 2014 could go back to GM's ignition switch decision in 2002, she would. If in 2019 she could go back a year and ensure that a coordinate measuring machine was set to "ABS" and not "REL", or go back a few days and ensure that the angle-of-attack sensor was calibrated as it was installed on the Boeing 737 Max operated by Lion Air, or encourage the first crew to fly that aircraft to return to the airport as soon as the stick-shaker event occurred (as they were "required" to do), the crash of LNI610 would be avoided and hundreds of lives would be saved.

But we cannot go back in time and are limited to imagining the future, rather than correcting the past. And even if a time-shifter suddenly appeared who could perfectly describe future events, would anyone believe her? She would need to overcome organizational inertia and an unwillingness to believe in the possibility of such unlikely scenarios. Since we are limited to imagining the future, our best option is through coordinating many perspectives, developing systemic empathy, and challenging status quo decisions. For this, we need collective learning.

MOBILITY KILLS

Being mobile is inherently dangerous. Traveling long distances in short times requires high velocities and a need to decelerate gracefully. In an aircraft the pilot decelerates when landing by simultaneously losing speed and altitude and bumping gently (in most cases) onto a runway, where remaining speed is lost as the aircraft travels for miles. In an automobile the driver decelerates by pressing the brake to stop in a few seconds and a few dozen feet (on dry pavement).

Idea in Brief

Modern products are increasingly intelligent, and their development increasingly complex. Such complexity is managed through documented requirements, but these are decomposed and assigned to subgroups, eventually leading to a lack of product clarity and organizational dysfunction.

The Systems Engineering methods used to manage complexity are not up to the challenge, and we need Systems Thinking. Product Lifecycles are more complex than we can imagine, and we need to reduce, rather than manage, complexity.

This paper investigates how complexity and dysfunction led to two crashes and the eventual grounding of the Boeing 737 Max aircraft and notes similarities in the case of the GM Ignition Switch Recall.

Aircraft operate in a structured human environment, built on laws, customs, and practices. The operator's (pilot's) actions are coordinated with a central authority (air-traffic control), and in-flight safety is focused on maintaining maneuverability in the air and avoiding other aircraft. While it is highly unlikely for one airborne aircraft to hit another, they remain in a constant struggle against gravity. Aircraft offer little in the way of physical protection in the event of a crash, and from 2010 to 2018 the US averaged about 440 airline deaths per year.

Automobiles operate in a much less structured environment, where each driver makes moment-by-moment decisions uncoordinated with any central authority. We assume that cars will hit other objects, and passenger safety is found through physical protection; anti-lock brakes to handle ice, defined crush zones to absorb impact, reinforced passenger compartments to protect occupants, plus seat belts, headrests, and airbags. In the period 2010 to 2018 the US averaged about 34,000 fatalities per year, or 90 times the number of aircraft fatalities (USDOT, 2021).

In the case of GM airbag non-deployments, the fatality scenario was that in the few seconds prior to impacting a tree a vehicle would encounter enough bumps to knock the key from "Run" into the "Accessory" position, telling an onboard computer to disable the airbags. This small piece of autonomy was designed to increase safety by not deploying airbags when the car is parked, but in over 120 cases the autonomy failed to inflate the airbags when needed. This was a sin of *omission*; the autonomy caused the airbag to do nothing, thus allowing humans to come to harm.

In 346 fatalities involving two Boeing 737 Max crashes in 2019 and 2020, an Angle of Attack indicator was mis-calibrated, causing on-board software to think that the plane's nose was pitched too high, in what in-fact were level aircraft. The plane's autonomy continually forced the airplane's nose down, despite the flight crew's best efforts. This was a sin of *commission;* the robotic pilot harmed humans when it should have done nothing (KNKT, 2018).

In his "I, Robot" series, Isaac Asimov developed *"Three Laws of Robotics"*, the first being that:

> *"a robot may not injure a human being,*
> *nor through inaction, allow a human being to come to harm."*

By causing a properly functioning aircraft to crash, the plane's autonomy (which was being fed misinformation from one of its sensors) took control of two aircraft away from their human flight crews and forced the planes to crash. In these cases, the robot pilots killed 346 humans, violating the first clause of Asimov's First Law.

As a control sample, let's briefly explore another option – high speed trains. In the US train fatalities are about twice that of aircraft fatalities, but a better example may be the German train system, Deutsche

Bahn. It is famous for its punctuality and is a very safe means of mobility; from 2010 to 2019 there were only two incidences totaling twelve fatalities,[*] or one fatality per year. (Wikipedia: List of Rail Accidents)

The point being that if we think about "safer mobility" rather than either "safer automobiles" or "safer aircraft" it would lead us to different modes of travel. It is hard to imagine an automobile which is 34,000 times safer than what is available today, although that is what a DB Bahn train might offer. They are also comparatively fast and can deliver passengers directly to and from dense population areas. DB Bahn carries about two billion passengers annually (Wikipedia: Deutsche Bahn).

But most of article is about aircraft, specifically the Boeing 737 Max, and the challenges in attempting to develop and manage increasing product complexity, in particular as our products become more intelligent.

Developing Aircraft Requirements

In autonomous products human decision-making is supplanted by that of the machine, but the machines in-turn are developed by human organizations. As it is not possible to travel to the past to correct problems, product developers must imagine the future.

To plan against unintended consequence, aircraft manufacturers develop the lists (and lists, and lists… and did I mention the many, many lists?) of requirements which must be fulfilled to achieve a high-level goal. These are developed to suit overlapping and at times conflicting needs, from basic physics (e.g., the amount of wingspan needed to generate the necessary lift), to business, safety, and governmental regulations. To manage these requirements the aerospace industry follows objectives described by the Society of Automotive Engineers (SAE, 2010).

> "Section 5.3: Requirements Capture:
>
> "The top-level process in the aircraft development cycle includes the identification of aircraft functions and the requirements associated with these functions. The aircraft functions, including functional interfaces and corresponding safety requirements, form the basis for establishing the system architecture. At each phase of the requirements identification and allocation process (i.e., system and item) both additional detail for existing requirements and new derived requirements are identified. Choices made and problems encountered during implementation are a primary source for derived requirements and may lead to identification of new system safety requirements. Detailed design activities will invariably introduce new requirements or modify existing requirements.

---

[*] Eleven of the twelve are due to a single incident caused by a rail controller who was distracted by a video game.

> "Section 5.4 Requirements Validation
>
> Validation of requirements is the process of ensuring that the specified requirements are sufficiently correct and complete so that the product will meet the needs of customers, users, suppliers, maintainers and certification authorities.
>
> "Section 5.5 Implementation Verification
>
> "The verification process ensures that the system satisfies the validated requirements. Verifications consists of inspections, reviews, analyses, tests, and service experience…"

If this all seems familiar, it is essentially a text description of the Vee model (INCOSE, 2021) discussed in my post *"For Better Products We Need Better Cultures"* (Hillberg, 2021). In the Vee, high-level requirements are defined in the top-left, validated and implemented towards the bottom, and then verified as the process moves towards the top-right.

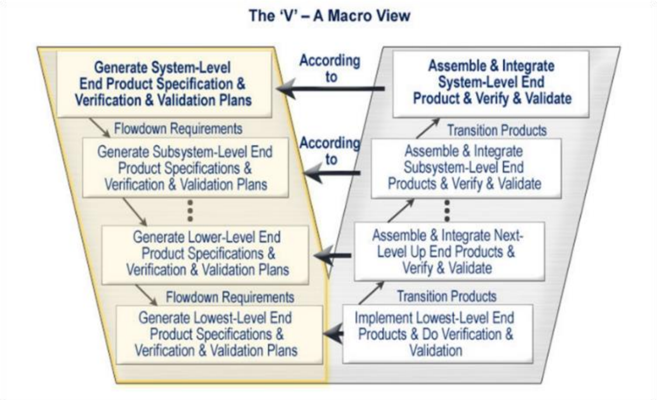


Figure 1: System Vee Model
© International Council of Systems Engineers

But note the latter portion of Section 5.2: *"At each phase … new derived requirements are identified. Choices made and problems encountered during implementation are a primary source for derived requirements…"* and thus it accepts the inevitability that not all requirements will be identified up-front, and that new ones will develop over time. The practice accepts that it is not possible to fully imagine the product's complexity.

Derived requirements will grow as the implementation moves down the left-hand side of the Vee, as work is decomposed into smaller and smaller groups. Subgroups will then derive sub-requirements, and in while theory conflicts will be found as the implementation moves up the right-hand side, this is a late and expensive time to find problems. As seen in the GM Ignition Switch post, it is organizationally difficult to admit the existence of a problem which delays the product launch, and by extension, delays revenue generation.

THE HIDDEN ROLE OF ASSUMPTIONS IN THE BOEING 737 MAX

The requirements process is based upon a foundation of assumptions, whether stated or not, and it is the assumptions which lead to scandalous failures. With the Boeing 737 Max, it was assumed that the introduction of an automated MCAS system did not necessitate a new failure analysis on the previously designed angle-of-attack sensor. It was further assumed that all pilot and maintenance activities would be performed properly, and pilots would behave as the training prescribed. There is no evidence of malicious intent in the 737 Max; there was no serial killer bent on the death of hundreds of people, but the fatalities happened all the same.

Boeing's Assumptions in the 737 Max

The following is reported by the New York Times in discussing factors which led up to the first Boeing 737 Max crash (Gelles, 2019):

> "Boeing faced an unthinkable defection in the spring of 2011. American Airlines, an exclusive Boeing customer for more than a decade, was ready to place an order for hundreds of new, fuel-efficient jets from the world's other major aircraft manufacturer, Airbus.
>
> "To win over American, Boeing ditched the idea of developing a new passenger plane, which would take a decade. Instead, it decided to update its workhorse 737, promising the plane would be done in six years…
>
> "The 737 Max was born roughly three months later…
>
> "And losing the American account would have been gutting, costing the manufacturer billions in lost sales and potentially thousands of jobs.
>
> "A former senior Boeing official said the company opted to build the Max because it would be far quicker, easier and cheaper than starting from scratch, and would provide almost as much fuel savings for airlines."

Thus, an assumption was baked into the development of the 737 Max, that the most effective means for Boeing to develop a new line of fuel-efficient airplanes would be to modify the existing 737 NG model.

However, in researching the first accident, the Joint Authorities Technical Review (JATR, 2019) found that:

> "…some elements of the design and certification remain rooted in the original 1967 certification of the B737-100. … and while (some later advancements and design concepts) have been incorporated into the B737 MAX, other advancements have been determined to be impractical."

As is often the case when developing new products, there is a balance to be made. An aircraft based on a 50-year-old design may be limited in its advancements but has 50 years of collected knowledge, experience, and infrastructure. By 2011, over ten thousand 737 aircraft had been manufactured, carrying 12 billion passengers more than 74 billion miles, and the model accounts for 25% of the world-wide airplane fleet (Wikipedia, 2021). It is a popular and highly stable model of aircraft, and it is a not-unreasonable decision to leverage its history and infrastructure. There are airports around the world with terminals designed to accommodate the Boeing 737.

Cascading Strategies Lead to MCAS

Boeing customers demanded fuel savings and the manufacturer saw two alternatives: design an entirely new aircraft line or modify the existing 737 NG model. The new line would require a longer process for FAA certification (ten years vs. six) plus an additional expense of retraining pilots. This training is not trivial and would require the construction of new aircraft simulators and would keep pilots out of the air (thus not ferrying passengers) while in training. For the airlines, different models create logistical hassles, as a pilot qualified to ferry passengers on one model may not be certified to ferry them on another.[†] Per the New York Times (Boeing Built Deadly Assumptions Into 737 Max, Blind to a Late Design Change, 2019):

> *"Boeing wanted to limit changes to the Max, from previous versions of the 737. Anything major could have required airlines to spend millions of dollars on additional training. Boeing, facing competitive pressure from Airbus, tried to avoid that."*

Relating this back to the Vee-model in Figure 1, two top-level requirements were created within Boeing: Improve fuel economy of the 737 NG and do so without the need to retrain pilots. For the former, Boeing chose to use the CFM LEAP-1B engine, which had a long track record in the industry. But this engine has a larger diameter the previous engines, creating derived requirements to move the wings forward, changing the flight characteristics in such a way that pilot-retraining might be needed. To avoid the retraining, Boeing engineers and test pilots chose to include an autonomous system, known as the Maneuvering Characteristics Augmentation System, or MCAS. (Sider & Tangel, 2019)

Verifying and Validating Designs

The US FAA's Joint Authorities Technical Review (JATR, 2019) noted that *"Boeing's integration of the design and safety analysis is heavily reliant on the chief test pilot…"* which is in the Validate and Verify steps at the top-right of the Vee. This is also home to the greatest pressure and risk as the product is moving closer to its release date.

Again from the New York Times article:

> In 2012, during the early development of the 737 Max, chief test pilot Ray Craig was testing high-speed situations on a flight simulator, like maneuvers to avoid an obstacle or to escape a vortex from another plane. Such moves might never be necessary in traditional passenger travel, but the F.A.A. requires that a jet handle these situations. Ray felt that the plane was not flying smoothly due to the Max's larger engines.
>
> At some point a suggestion appeared to use the MCAS system on the 737 Max, as it had solved a similar problem on the KC-46A fueling tanker. The 737 Max designs adopted the MCAS to address the problem, intending it to work in the background, so pilots would not need to know it was there. Mr. Craig preferred to avoid systems that took control from pilots and would have preferred an aerodynamic fix, but the engineers who tested the

---

[†] As the market for air-travel increases, there is the possibility that the size of the flying public will exceed the availability of trained pilots.

> Max design in a wind tunnel were not convinced aerodynamic changes would work, and Mr. Craig relented. Such high-speed situations were so rare that he expected that the software would never actually kick in. Further, engineers designed the MCAS to trigger only when the plane exceeded two separate thresholds: both angle of attack, and acceleration force.

This is a situation of a derived requirement, where people with different backgrounds and expertise attempt to imagine what may happen in rare but possible situations. Pilot and engineers have a reasonable difference of opinion, and develop a compromise based on Mr. Craig's acceptance that this is a highly unlikely event, and the engineering team's commitment to require both an angle and acceleration trigger, making MCAS activation unlikely in a pilot's career.

The article continues (now in January 2016, four years later):

> Ed Wilson is the chief test pilot for the Max, having taken over the role from Mr. Craig the previous year. Mr. Wilson feels that the Max is not handling well when nearing stalls at low speeds and tells engineers that the issue would need to be fixed; he and his co-pilot propose expanding the role of MCAS to low-speed maneuvers. The change does not elicit much debate in the group as it was considered "a run-of-the-mill adjustment". One person interviewed said "I don't recall ever having any real debates over whether it was a good idea or not".
>
> However, expanding the use of MCAS to lower-speed situations required removing the acceleration criteria, meaning that it would now function based only on a single angle-of-attack sensor. Although the 737 has two sensors, the new version of MCAS took data from only one.
>
> Test flights were uneventful, looking at a high-speed maneuver in which the system does not trigger, and a low-speed stall when it activates but then freezes. In both cases, the test pilots were able to easily fly the jet. But in those flights, they did not test what would happen if the MCAS activated as a result of a faulty angle-of-attack sensor — which was a problem in the two crashes.

The Conversation That ***Didn't*** Happen

To recap, four years after an engineering decision to activate the MCAS in highly unlikely situations (a decision which still needed to overcome the initial test pilot's reticence), a second test pilot advocates to expand the MCAS scope, and it doesn't even raise significant conversation. This is now only one year prior to the planned FAA certification, the Boeing team is further up the right-hand side of the Vee, and after the expanded MCAS scope is implemented, it is flight-tested only against the failures that they have recognized.

But no one on the team tested against systemic failures that they *hadn't* recognized.

Assumptions and Rationalizations

The FAA defines four classifications of failures (or hazards): Minor, Major, Hazardous, and Catastrophic. Boeing classified two hazards associated with "uncommanded MCAS" (meaning it was invoked by the automation, not by the pilot) as "Major: Reduced control capability, reduced system redundancy, and increased crew workload". According to the (KNKT, 2018) report:

> "This assessment of 'Major' failure effect did not require Boeing to more rigorously analyze the failure condition in the safety analysis, using Failure Modes and Effects Analysis (FMEA) and Fault Tree Analysis (FTA), as these are only required for Hazardous or Catastrophic failure conditions."[‡]

I have no insight into Boeing's decision-making process in choosing the Major classification, but let's imagine the situation within the Boeing culture at this decision point. If the possible hazard had been deemed 'Catastrophic' (and considering the two accidents, it certainly *was* catastrophic) it would have required additional pilot training, which contradicted the business strategy. Continuing the New York Times story:

> Mark Forkner was the chief technical pilot for the 737 Max and was the liaison to the FAA regarding the needs for pilot training. In March 2016, Mr. Forkner lobbied the FAA to remove mention of the MCAS from the pilot's manual. The FAA was aware of the MCAS in its initial form (limited to high-speed maneuvers per then chief pilot Ray Craig) and approved the removal from the training, but neither Mr. Forkner nor Boeing informed the FAA that the MCAS was in the midst of an overhaul.

However, by November 2016, Mr. Forkner is privately having concerns. In a text exchange with fellow pilot Patrik Gustavsson: (Forkner, 2019):

> **Mark Forkner:**
> Oh shocker alerT!
> MCAS is now active down to (Mach) 0.2
> It's running rampant in the (simulator) on me
> at least that's what Vince thinks is happening
> so I basically lied to the regulators (unknowingly)
>
> **Gustavsson, Patrik:** it wasnt a lie, no one told us that was the case
>
> **Mark Forkner**: I'm levelling off at like 4000 ft, 230 knots and the plane is trimming itself like (crazy). I'm like, WHAT?

Note that Mr. Forkner is surprised to learn about MCAS changes that had been made without his knowledge, and the implication that he unknowingly provided inaccurate information to the FAA. This is

---

[‡] For reference on FMEA and FTA, see
[Understanding the Cause of Faults in the Lean Factory | Engineering.com]

now just five months prior to the planned FAA Certification, which is the milestone which allows Boeing to deliver planes to the airlines (and generate revenue). The development of the 737 Max is well up on the right-hand side of the Vee, and the pressure on those involved is building.

A later criminal probe into Boeing's actions focused on Forkner and Gustavsson. Boeing settled a criminal suit brought by the US Justice Department, reported in (Boeing Reaches $2.5 Billion Settlement of U.S. Probe Into 737 MAX Crashes, 2021).

Assumptions Define What Is (and Is Not) Tested

Pilot assessments of MCAS hazards were conducted in flight simulators, but the tests were limited to activating MCAS without first simulating the failure modes which led up to these events. In the actual accident flights, the pilots were overwhelmed by alerts, stick shakers, an aircraft which seemed to be fighting them, and substantial concern for their own and their passengers' safety. The accident flight crews needed to weed through extraneous and urgent information to determine the root cause, which was that an erroneous angle-of-attack sensor was triggering autonomous actions by an MCAS system of which they were unaware. In the two accident flights, and the pre-accident flight (discussed shortly), the workload substantially increased as the pilots tried to debug what was going wrong. Per their report (NTSB, 2019), "while Boeing considered the possibility of uncommanded MCAS operation as part of its hazard assessment, it did not evaluate all the potential alerts and indications that could accompany a failure that also *resulted* in uncommanded MCAS operation".

The 4-second Assumption

Further, the JATR observed that while Boeing test pilots follow FAA guidance to delay their actions to account for the time it takes a pilot to recognize (1 second) and respond to (3 seconds) a malfunction, Boeing and other aircraft makers incorporate this as "a design assumption that the pilot will be able to respond correctly within 4 seconds". But the report found that "no studies were found which substantiate this guidance", and that there are studies "demonstrating that general aviation pilots may take many seconds, and in some cases many minutes, to recognize and respond to malfunctions". Finally, "there is substantial difference between the situation of a test pilot who is testing a particular malfunction with a precise foreknowledge, … and of a line pilot on a routine revenue flight who is not expecting any malfunction." (JATR, 2019, p. 8.1) Thus, across the industry there is a design assumption about pilot actions which has no support and a good deal of contradiction.

The (NTSB, 2019) report observes that "because the FAA allows for such assumptions to be made…without consideration of multiple flight deck alerts in evaluating pilot (workload)", and "because the FAA routinely harmonizes with… other international regulators", similar assumptions may have been used in aircraft manufactured world-wide. Thus, the crew on any flight deck in any aircraft world-wide may be susceptible to excessive workload in hazardous situations.

THE LION AIR ACCIDENT

On October 29th, 2018, Lion Air Flight LNI610 crashed minutes after takeoff from Jakarta, killing all 189 people on board.

Lion Air Flight LNI610 from Jakarta 29-October-2018

First Officer; in control at the time of the accident

According to the (KNKT, p. 179) report, at the time of the accident the flight was under the command of the First Officer whose "training records showed… difficulty in air handling," and an inability in the performance of "memory items" that pilots must know. In the end, the flight crew was unable to work through the "Non-Normal Checklist" (NNC) which would have provided guidance in resolving the flight's problems, and the First Officer's training records show an inability to identify the NNC from memory. Under the First Officer's control, the plane plunged into the sea, killing 189 people.

Captain; a few minutes prior to the accident

A few minutes prior, the flight's Captain had transferred control to the First Officer so that he could better think through the problems which they had been unable to mitigate since the flight left the ground. Per the KNKT report:

> "The activation of stick shaker indicated that the aircraft was about to stall while the cockpit instrument indicated the pitch was relatively level and the speed relatively high. The cockpit instrument did not indicate that the aircraft was close to stall condition which contradicted to the stick shaker activation. The aircraft was not equipped with AOA (angle of attack) indicator and the AOA DISAGREE message was inhibited, so there was no information provided to the flight crew of which AOA sensor triggered activation of stick shaker. The flight crew was not aware of the real aircraft condition." (KNKT, p. 181)

The crew was not trained in, nor even aware of, the MCAS system as it had been a Boeing design criterion that training would not be required. A mis-calibrated angle-of-attack sensor signaled the MCAS that the plane was approaching stall, even though it was not. The AOA DISAGREE message may have given the flight crew information which would help them identify the miscalibration, but through a manufacturing mistake it was not enabled. The MCAS triggered the stick-shaker signal, indicating that the plane was in a stall (though it was not) and continually forced the nose down, as it was programmed to do.

Per Asimov's rule, the MCAS robot was attempting to keep humans from harm, but instead plunged them to their deaths.

> "In the accident flight, MCAS repeatedly moved the horizontal stabilizer based on a combination of the erroneous AOA inputs and flight crew manual electric inputs. The Captain managed to control the aircraft with the pitch trim. The DFDR data showed that control was maintained by keeping pitch trim above 5 units to counter the repetitive MCAS activations. After transfer of control to the FO (First Officer), the FO did not apply sufficient manual electric trim to counter repetitive MCAS activations resulting in compounding mis-trim which required significant control column force."

> "Movement of the stabilizer due to uncommanded MCAS activation during normal flight would be easier to identify if there were no other distractions in the cockpit. However, during the accident flight erroneous inputs…resulted in several fault messages … that affected the flight crew's understanding and awareness of the situation. The stick shaker activated continuously after lift-off and the noise could have interfered with the flight crew hearing the sound of the stabilizer trim wheel spinning during MCAS operations.

Lion Air Flight LNI043 From Denpasar to Jakarta
Day prior to accident flight

On 28-October, the same aircraft was flown successfully from Denpasar to Jakarta as flight LNI043. This crew faced similar challenges in controlling the aircraft but were able to land successfully. When the stick shaker and numerous caution lights activated during lift-off, the Captain of LNI043 transferred control to the First Officer and accurately performed the Non-Normal Checklist. By chance, LNI043 had a third pilot in the cockpit, who was transporting between airports and could help with the mental workload. In this case, the Captain cut the stabilizer trim which brought the aircraft under control and allowed the aircraft to fly normally under manual trim. Again, per the KNKT report:

> "In the LNI043 flight, the crew required 3 minutes and 40 seconds… to recognize and understand the problem, during which repetitive uncommanded MCAS activations occurred. During the accident flight (LNI610), recognition of the uncommanded stabilizer movement as a runaway stabilizer condition did not occur, thereby the execution of the non-normal procedure did not occur".

At the end of flight LNI043 the Captain made entries in the Aircraft Flight Maintenance Log (AFML) about three warning lights but failed to record the activation of the stick-shaker, erroneously assuming that the shaker was a redundant signal to the warning lights.

> "The Captain's incomplete report in the AFML was based on his incomplete understanding of the interrelationship between the effects experienced during the flight and the system failures that caused those effects… Further, the requirement to report all known and suspected defects is very critical… to maintain the airworthiness of the aircraft." (KNKT, p. 176)

Aircraft Maintenance in Denpasar
27-October-2018, two days prior to the accident

The evening prior to the nerve-shattering yet successful flight of LNI043, the left-side AOA (angle-of-attack) sensor was replaced in order to deal with recurring issues on previous flights. An AOA test fixture was not available in Denpasar, but an alternative test is prescribed in which the vane is moved through its range of motion and the output values observed and recorded. This action was to be performed by an engineer employed by Batam Aero Technic (BAT), which was contracted by Lion Air to perform such maintenance. While BAT's procedures required the recording of the results, no such recording was done.

> "The engineer in Denpasar stated that the test result was satisfactory, … (then) provided to the investigation some photos of the SMYD unit during an installation test as evidence

> of a satisfactory installation test result. The investigation confirmed that the SMYD photos were not of the accident aircraft and considered that the photos were not valid evidence."

Which implies unethical behavior[§] by the maintenance engineer in Denpasar in falsifying the photos. In flights prior to this AOA replacement, Captains were reporting erroneous Speed (SPD) and Altimeter (ALT) flags. Replacing the AOA removed these flags, but the new AOA sensor was mis-calibrated by 21 degrees.

Angle of Attack (AOA) Refurbishing in Miramar, FL
October-November 2017, one-year prior to accident flight

A year prior to the accident, BAT removed an angle-of-attack sensor from a different aircraft and sent it to the company Xtra Aerospace of Miramar Florida for repair. After the accident, representatives of the NTSB, FAA, Boeing, and Collins Aerospace reviewed the test equipment at Xtra, and found that Xtra was unable to produce written instructions on how to operate the AOA calibration tool. But of greater impact was the following observation:

> "During calibration… selection of the REL/ABS switch in the REL position may have led to improper calibration because there were no written instructions for correct use… in accordance with requirements. … With the switch in the REL position… improper calibration would not be detected."

Thus, an AOA sensor was improperly calibrated in 2017, and though there were numerous opportunities to recognize and correct this problem, none occurred, and the plane crashed in 2019, killing all on board.

COMPLEXITY BEYOND OUR IMAGINATION

The "Swiss Cheese" Safety Model

Imagine that your safety on an aircraft is protected by layered slices of swiss cheese. The defenses against a failure are a series of barriers (the cheese slices) each with possible failures (the "holes") hence, the Swiss Cheese model. This is an accident causation model developed by James Reason at the University of Manchester in the late 1980's. But each of the barriers is an expense, and it is difficult to recognize all the holes. If we trace the events leading up to the crash in chronological order, we see the following:

- Developing a new fuel-efficient aircraft model would require significant additional time and expense, due to a new FAA certification, additional pilot training, and logistical challenges for both airlines and airports.
- The new engines used in the 737 Max changed the aerodynamics compared to its predecessor; this would necessitate pilot training to account for rare circumstances. Automation was introduced into the Max design in 2012, such that software would account for these rare

---

[§] And likely a prevention focus, which is discussed in the GM Ignition Switch article.

instances. The test pilot initially objected but was swayed by the expectation that the automation would only rarely be used.

- A different set of test pilots, a few years later, worked with engineers and decided to increase the scope of the MCAS automation to remove a condition that would restrict its use. Test pilots and engineers did not think about the ramifications of increased scope, and characterized the failure mode as Major, but not Catastrophic (thus avoiding the need for pilot training). Testing did not invoke the mental workload faced by a normal pilot in the midst of a crisis.
- A third test pilot was then tasked with persuading the FAA that introduction of the MCAS did not require additional pilot training, in an environment in which a strong corporate goal was to eliminate the need for new training.
- Once the 737 Max product was launched, proper instructions were not provided to an operator in Florida when calibrating a device after a repair of an angle-of-attack (AOA) sensor.
- A year later, this AOA sensor was installed on an aircraft and again left uncalibrated.
- The following day, pilots on the pre-accident flight encountered a stick-shaker event (created by the mis-calibrated AOA sensor) but did not immediately return to the departure airport and did not properly record the event in the flight log for additional maintenance.
- And on the craft's final flight, a pilot who was unaware of the MCAS, and a co-pilot who flew despite a documented inability to memorize the Non-Normal Checklist were overwhelmed by an autonomous system which was attempting to avoid a stall, but instead drove the plane into the ground, killing all aboard.

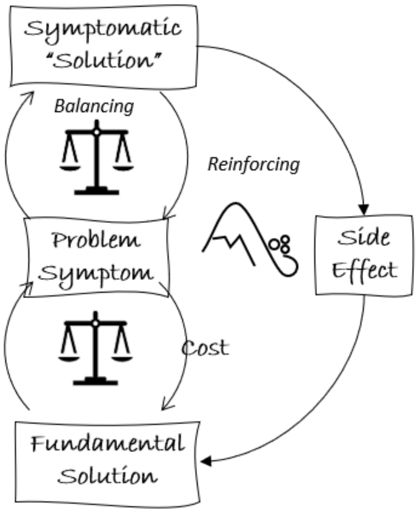
And four months later, the MCAS system drove a similar 737 Max into the ground in Ethiopia.

Shifting the Burden

(Senge, 2006) describes another Systems Thinking model called Shifting the Burden:

In it, you can see that there is some problematic symptom, and two possible solutions; one addresses the obvious symptoms, and has a lower cost, immediate payback, and seemingly positive results. A fundamental solution exists (and after a crisis is adopted) but it is costly and time-consuming compared to the symptomatic solution which seems to work so well.

Eventually, a side effect appears, leading to a systemic collapse. The collapse itself is reinforcing; as one infrastructure element collapses, two which depended on it also collapse, followed by four, and by eight, etc.



Within Boeing, a burden was placed on the product development team to modify the 737 NG aircraft with higher fuel efficiency, yet not require retraining of existing 737 NG pilots. This burden was shifted onto engineers and test pilots, who shifted the burden further onto the MCAS system. As the MCAS was behaving strangely for Mark Forkner, he felt the burden of meeting a schedule for certification and "lied (unknowingly)" to the FAA.

Once certified, the MCAS shifted the burden of onto a single Angle of Attack indicator, and the burden of calibrating these indicators fell onto both the onsite maintenance technician (who may have felt burdened by the need to repair the plane quickly) and the remote technician responsible for setting a switch in the proper "REL" vs. "ABS" position. This technician placed a burden on an operations manual which wasn't properly placed at the machine. The layers of "swiss cheese" included a small set of aligned "holes" which allowed fatal errors to pass.

Decomposition Leads to Dysfunction

In *The Fifth Discipline*, (Senge, 2006) posits that organizations have "learning disabilities". He says:

> "We learn best from experience, but we never experience the consequences of many of our most important decisions…
>
> "Traditionally, organizations break themselves into components (and) institute functional hierarchies that are easier for people to 'get their hands around'. But functional divisions grow into fiefdoms, and what was once a convenient division of labor grows into stovepipes… the complex issues that cross functional lines become a perilous or non-existent exercise".

Complex products are decomposed arbitrarily into manageable pieces, and organizations build fiefdoms around the decompositions. But as products become more autonomous, components interact in unimagined ways.

In June 2019, a few months after the second 737 Max crash, then-Boeing CEO Dennis Muilenburg stated that a shortage of pilots represented "one of the biggest challenges" to the airline industry, and that 800,000 new pilots would be needed over the next 20 years (Meredith, 2019). Automating flight through the MCAS was not only critical to 737 Max sales, but to all Boeing sales going forward. There was clearly a cultural bias within Boeing to reduce the level of pilot training needed, and in this environment engineers and test pilots made highly consequential decisions with little ability to experience the consequences.

Air Travel in a Pandemic World

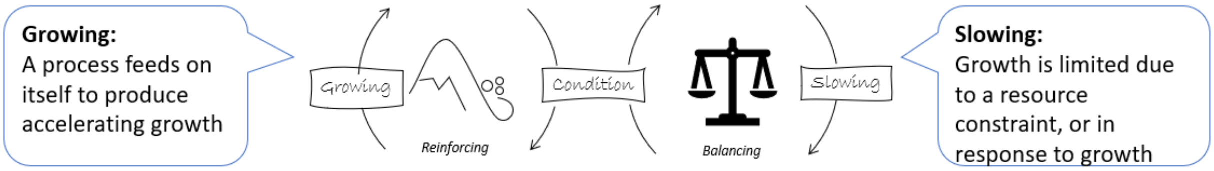
But Muilenburg's comments were made a few months prior to the COVID-19 pandemic, and by the following year thousands of pilots would be furloughed (Chokshi, 2020). The original strain of the virus moved from China to various locations around the world in the matter of a few weeks, and new strains originating in Britain, South Africa, Brazil, India and future countries continue to travel the world quickly… via airplane.

The worldwide system of air travel is far more complex than the autonomous features of individual planes as airlines deliver the diseases which result in travel barriers and the industry's own downfall. This may be the greatest unexpected consequence of the industry's most important decisions.

REDUCING COMPLEXITY

Systems Thinking

(Senge, 2006) describes *Systems Thinking*, one tenet of which includes two feedback loops, *reinforcing* and *balancing*. In a reinforcing loop growth encourages growth, but the loop eventually reaches some resource limit such that continued growth is not possible. At this point the system enters a balancing loop, in which any new growth is counteracted by opposing forces. Viral growth is reinforcing while hosts without immunity continue to infect each other, but eventually growth slows as immunity develops from past infections. The virus' early success eventually creates a 'herd immunity' against its growth, and the system finds a balance. It follows a Limits to Growth model:



Growing: A process feeds on itself to produce accelerating growth

Slowing: Growth is limited due to a resource constraint, or in response to growth

In similar fashion, for the past century product functionality has followed reinforcing growth, as products accomplish more and more and become nearly "intelligent". But in *The Innovator's Dilemma* (Christensen, 1997) warns of an 'Oversupply of Performance' which creates room for something low-priced and simple to disrupt the complex and overly functional. Think of the "Settings" screen on your cell phone… do you understand what every option does? Will you after the next update? Or do you simply use a subset of the device's functionality, meaning that you are paying for (and confused by) functions that you do not use? At some point customers to choose to pay less for a device which is less functional and less complex, and the market is disrupted.

Mobility devices (cars, planes, trains, etc.) involve large masses moving at high velocities, and thus include inherent dangers. The complexity of the MCAS system, intended to keep passengers safer without pilot need for retraining, was eventually too complex for pilots to solve when it misfunctioned. The nature of danger in autonomous devices does not originate within some nefarious artificial intelligence (as sci-fi movies would have us believe) it instead originates within the complexity of the *human* organization which develops the product. Organizational dysfunction leads to unresolved product complexity creating unexpected dangers and limiting growth. After two crashes the fleet of 737 Max's were grounded for two years, and as this is written in the spring of 2021 are just now resuming flights. Boeing now agrees that pilots of the 737 Max need special training, but this increases the cost of ownership and the complexity of airline scheduling. The systemic burden shifts from Boeing to the airlines, though Boeing is compensating the airlines for additional training needed.

We need new thinking to develop better Systems.

Systems Engineering

I will admit to being frustrated with "Systems Engineering" in spite of having two graduate degrees and a faculty position with those words in their titles. I am not 100% sure what the term means, as Engineering is a *subsystem* within product lifecycles. "Systems Engineering" seems self-contradictory. Only a small percentage of the people who contribute to the lifecycle of a product consider themselves "engineers" and if all activities are engineering activities, then the definition of 'engineering' has no boundaries. Most notably, Systems Engineering is *not* Systems Thinking, as it doesn't account for cultural behavior and feedback loops.

The International Council of Systems Engineers (INCOSE, 2021) is a professional group "designed to connect systems engineering professionals with educational, networking, and career-advancement opportunities in the interest of developing the global community of systems engineers and systems approaches to problems". It boasts of over 18,000 members, though even this group seems to struggle for a definition, and posit three as follows:

> *"A **system** is an arrangement of parts or elements that together exhibit behavior or meaning that the individual constituents do not."*

Per this definition, the components acting within a system exhibit behavior which is different than if they behaved independently. Birds within a flock behave in concordance with the flock's behavior; the flock is a loosely coupled 'system'.

> *"An **engineered system** is a system designed or adapted to interact with an anticipated operational environment to achieve one or more intended purposes while complying with applicable constraints."*

But note the implication that the design of engineered systems has an *intended purpose* and *applicable constraints*, then think of the assumptions, rationalizations, and conversations which did not occur in either the Boeing 737 Max or the GM Ignition Switch, in which true purpose and constraints were not understood throughout the organization.

> *"**Systems Engineering** is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods."*

But this definition is circular, in that "Systems Engineering" is assumed to use "systems principles" without describing what those principles are. Further, to limit Systems Engineering to *"scientific, technological,* *and management methods"* fails to account for the important complexities which do not obviously fall into those responsibilities. And what about the interfaces between these three fields, and the fields unmentioned, where no one feel responsible?

In October 2018, I attended an aerospace conference where a Boeing economist provided rosy projections for the future of aircraft manufacture. She stated that more and more of the public wanted to fly, the airlines were doing so profitably, and Boeing had released a new airplane (the 737 Max) which lowered fuel costs and raised margins. But later that same month the Lion Air flight crashed, followed a few months later by an Ethiopian Air flight. The organization's desire to meet economic projections created cultural norms and led to decisions which eventually resulted in the loss of 346 lives, and tens of billions of dollars (Isidore, 2021). It is *because* the organization tried to live up to its own economic projections that it failed to meet those projections.

How then, can an unimaginably complex product, developed by an even more complex organization, whose members are driven by conflicting goals, live up to Asimov's First Law?

"A robot may not injure a human being,
nor through inaction, allow a human being to come to harm."

The Need to Fix Systems Engineering

That "Systems Engineering" requires fixing was delivered in a 2010 speech (titled "How to Fix Systems Engineering") by Dr. Mike Griffin, the former director of NASA. (Griffin, 2010)

> "[S]ystem engineers have some explaining to do. How is it that we continue to encounter failure of important and complex systems where everything thought to be necessary in the way of process control was done, and yet despite these efforts the system failed? Each time this occurs, we as an engineering community vow to redouble our efforts to control the engineering process, and yet such events continue to occur. The answer cannot lie in continuing to do more of the same thing while expecting a different outcome. We need to rise above process, to examine the technical, cultural, and political mix that is "system engineering", and to examine the education and training we are providing to those who would practice this discipline."

Thus, Griffin claims that to increase societal value, "Systems Engineering" must also include culture and politics (which are not included in the INCOSE definitions). He goes on to say:

> "failures continue to occur, often of the most glaring and consequential nature, commonly at the boundaries or interfaces between elements, often due to uncontrolled, unanticipated and unwanted interactions between elements, in many cases between elements thought to be entirely separate."

Which I rephrase as "Decomposition leads to Dysfunction". Henry Ford's Model-T was a system of maybe a thousand parts, but the assembly line which could build hundreds per day was an even more complex system of assembly lines, machine tools, and the cultural and political actions of the humans involved. Ford decomposed work on the assembly line and in the engineering offices such that each person was highly experienced at their own narrow set of tasks, but now a century later we have learned that the whole is *less* than the sum of its parts. In the GM Ignition Switch case, the engineer responsible for spring tension switch did not recognize that the airbag system depended on his decisions, and by failing to connect the dots hundreds of people died and billions were lost. At Boeing, three different test pilots and associated development teams made decisions about the 737 Max, but they unknowingly relied upon service technicians around the world to perform their tasks as the developers expected, without understanding the cultural and political environments in which those technicians worked. The pilots, the engineers, and the maintenance technicians did not understand the full consequences of their decisions within the overall system.

Returning to Griffin's speech:

> "Failures of system engineering processes have in the past typically resulted in the addition of more, and more detailed, processes. In the world of 1950, … this would likely have been the right answer in any given case. But in the world of 2010, it is this author's view that the addition of more or new system engineering processes is likely not the right answer in response to any failure. It puts one in mind of the jocular definition of insanity: continuing to do the same things over and over, while expecting a different outcome."

In other words, attempting to manage complexity by adding more detail to existing methods has reached its limit to growth. It is insanity to try.

Completing the "Job to be Done" with Less-Complex Systems

The (Christensen Institute, 2021) speaks of a "Job to be Done" by a product or service: we purchase products to perform some "job". Aircraft manufactures and airlines see their job-to-do as flying people and cargo through the air; auto makers see theirs as transport via the roadways.

But what the provider sees as the "job-to-do" is not necessarily the same as the consumer's desire for the "job-to-be-done". As air and auto travel become more complex, simpler solutions will become available and disrupt the current players. Rather than developing ever more extensive means of managing ever-increasing complexity, we should instead focus on complexity's reduction.

For example, the jobs done by personal automobiles include "driving to work or school", "going shopping", "visiting friends, family, and attractions", and in the US using an automobile largely requires owning one outright. But a vehicle driven 1,000 miles per month[**] sits idle 97% of the time, and some of that "idle" time is being stuck in traffic, unable to move and unable to do anything else. What job is being done while you sit frustrated in traffic?

Is it possible to accomplish these jobs with less driving, less complexity, and more environmental sustainability? There were signs of this in 2020, as people and businesses adapted to the COVID-19 pandemic. Online shopping and delivery boomed, even in perishable items like groceries. Offices operated

---

[**] At 45 mph

with reduced staff as people worked from their homes, eliminating commute times. Imagine a future two-income couple, with two cars and kids in the local schools. If in-person office attendance switches to twice per week, rather than Monday through Friday, family driving will be reduced by 60%. Will they still purchase two cars? Do they still need a two-car garage? What is the impact on childcare if the couple alternates days at home?

As online shopping and delivery becomes more prevalent, it reduces trips to the store and even reduces the number of stores. (One can imagine shopping malls becoming Amazon distribution centers.) Think of the acres of unused parking space; what is the impact on local wetlands and water tables if reduced in-person shopping also reduces the amount of asphalt? A mall parking lot designed to handle Black Friday traffic sits empty every day from 10 pm to 10 am, shuttling rains into retention basins which exit to the local waterways, rather than being absorbed and filling the local water tables. The systemic complexity of 1.4 billion automobiles goes far beyond the individual complexity of a single autonomous vehicle. It goes far beyond the scope of the auto manufacturers.

The Elegance of Rail Travel

In "How to Fix Systems Engineering", (Griffin, 2010) calls for "design elegance", with four attributes:

- Does it work as intended? Will the system produce the anticipated behavior over the expected range of conditions?
- Is it robust? The system should not produce radical departures from expected behavior when faced with small changes from the original intent?
- Is it efficient? Does it produce the intended result for fewer resources than we might expect?
- Does it minimize unintended actions, side effects, and consequences?

Based on these criteria, neither the GM Ignition Switch nor the Boeing 737 Max achieved "elegance". Most notably, the products' intent was murky, they were not robust, and they led to fatal unintended consequences. As automobiles and aircraft look to add autonomy, with its inherent complexity, product solutions become less and less "elegant".

We should refocus on an old means of travel, which in many cases is more capable of accomplishing the job to be done. In their report *The Future of Rail* (IEA, 2019) the International Energy Agency makes the following conclusions:

> Urban and high-speed rail hold major promise to unlock substantial benefits throughout the world. In an era of rapid urbanisation, urban rail systems can provide a reliable, affordable, attractive, and fast alternative to travel by road: metro and light rail can reduce congestion, increase throughput on the most heavily trafficked corridors and reduce local pollutant and greenhouse gas emissions. With coordinated planning, urban rail systems increase the attractiveness of high-density districts and boost their overall economic output, equality, safety, resilience, and vitality of metropolises. High-speed rail can provide a high-quality substitute for short-distance intracontinental flights. As incomes rise, demand for passenger aviation, a mode of transport that is extremely difficult and expensive to decarbonise, will continue to grow rapidly. If designed with

> comfort and reliability as key performance criteria, highspeed rail can provide an attractive, low-emissions substitute to flying.
>
> …
>
> Yet while rail is among the most energy efficient modes of transport for freight and passengers, it is often neglected in public debate.

Trains (and electric buses powered via overhead wires) do not need to carry their sources of power with them; there is no need for an on-board battery or fossil fuel. Motion is of course limited to their tracks, but this creates an elegance in that they easy to monitor and control. A stuck train ahead is easily sensed, and trains behind are brought gracefully to a stop. Per Griffin's definition of systemic elegance, the intent of a train is clear, it is efficient, and by limiting its motion to the track which powers it, unintended actions are minimized. Further a *system* of trains is elegant; I have worked, read, slept, and watched TV while traveling at hundreds of kilometers per hour through Germany, and as thousands of people are simultaneously transported on dozens of trains (which are all constrained to their tracks), high-level controls keep them running on time.

There is much talk of a future in which autonomous vehicles perform the jobs already accomplished by trains, but this seems unlikely. Will there be a sufficient supply of sustainably available lithium to provide for billions of independently driven vehicles? Wind and solar power are intermittent, with possibly too much during sunny days, and too little at night, in the winter, or when it rains. Batteries will be needed, but how will people choose to use a scarce supply of lithium? To drive their EV during the day, or to watch TV in the evening?

And is there a sufficient supply of integrated circuits? Electric vehicles require ICs to manage their electric motors, but a world-wide shortage in 2021 is forcing automakers to reduce production of their petroleum vehicles (Jeong & Strumpf, 2021), and the need for ICs will be much, much higher for autonomous electric vehicles. IC manufacturing requires large quantities of water and the nation of Taiwan, which manufactures two-thirds of the world's semiconductors, is facing a drought. According to (Yang, 2021):

> Taiwan officials and scholars have warned that water scarcity could become a more persistent problem in the years to come because of climate change, a worrying possibility for the global semiconductor industry given the concentration of chip production in Taiwan.

Automakers seek to lower emissions by replacing petroleum with electricity, but this shifts the burden onto lithium mining and IC manufacture. At some point, electric vehicle production will compete for people's need for clean water, food, and nighttime power. Prices will rise, and consumers must choose: EVs or TVs? General Motors will find themselves in a supply chain competition with tech companies like Facebook, Apple, Amazon, Netflix, and Google.

CONCLUSION

This paper covers a long arc, beginning with the Boeing 737 Max, and ending with trains and autonomous cars[††]. It discusses the failure of requirements planning in the 737 Max, a recognition that this and many systems may be too complex to plan, and that we need less-complex systems. Requirements decomposed and assigned to groups with differing cultures, norms, and goals eventually create the sorts of dysfunction which disables airbags when needed and overrides pilot commands to plunge aircraft into the Earth.

But – a long arc *is* the point if we are to think systemically. An autonomous car needs to operate in an environment in which the need for integrated circuits competes for the water needed to grow food. Aircraft need to fly in such a way that they do not turn local diseases into worldwide pandemics.

"Managing complexity" implies shifting burdens onto others. We must strive for less complex systems.

BIBLIOGRAPHY

Chokshi, N. (2020, 7 6). *Pilots Were Once in Short Supply. Now They're Losing Their Jobs.* Retrieved from nytimes.com: https://www.nytimes.com/2020/07/06/business/pilots-jobs-shortage-airlines.html?referringSource=articleShare

Christensen, C. (1997). *The Innovator's Dilemma.* Cambridge: Harvard Business Review Press; 1st edition.

Christensen, C. (2021, 5 8). *Jobs to be Done*. Retrieved from Christensen Institute: https://www.christenseninstitute.org/jobs-to-be-done/

Forkner, M. (2019, 10 18). Read the Boeing Pilot Messages From 2016: 'The Plane Is Trimming Itself Like Craxy'. *New York TImes*. Retrieved from https://www.nytimes.com/2019/10/18/business/boeing-pilot-messages.html

Gates, B. (2001). *How to Avoid a Climate Disaster.* New York: Doubleday.

Gelles, D. (2019, 3 23). Boeing was 'Go, Go, Go to beat Airbus with the 737 Max',. *New York Times*. Retrieved from https://www.nytimes.com/2019/03/23/business/boeing-737-max-crash.html

Griffin, M. D. (2010). HOW DO WE FIX SYSTEM ENGINEERING? *61st International Astronautical Congress.* Prague, Czech Republic.

Hillberg, P. (2021, 05 01). Retrieved from PatrickHillberg.com: https://www.patrickhillberg.com/my-writing

---

[††] Cue the Steve Martin and John Candy movie *"Planes, Trains, and Automobiles".*

IEA. (2019). *The Future of Rail, IEA, Paris.* Paris: IEA.

INCOSE. (2021, 5 18). *About Us*. Retrieved from incose.org: https://www.incose.org/about-incose

Isidore, C. (2021, 3 10). *The 737 Max crisis costs continues to climb two years after the second fatal crash*. Retrieved from cnn.com: https://www.cnn.com/2021/03/10/investing/boeing-costs-737-max-crisis/index.html

JATR. (2019). *Joint Authorities Technical Review - Boeing 737 MAX Flight Control System.* Washington, DC: US Federal Aviation Administration.

Jeong, E.-Y., & Strumpf, D. (2021, 04 19). Why the Chip Shortage Is So Hard to Overcome. *The Wall Street Journal*, pp. https://www.wsj.com/articles/why-the-chip-shortage-is-so-hard-to-overcome-11618844905?mod=article_inline.

KNKT. (2018). *Komite Nasional Keselamatan Transportasi Aircraft Accident Investigation Report PT. Lion Mentari Airlines Boeing 737-8 (MAX), 29-October-2018.* Retrieved from http://knkt.dephub.go.id/knkt/ntsc_aviation/baru/2018%20-%20035%20-%20PK-LQP%20Final%20Report.pdf

Meredith, S. (2019, 6 17). *Boeing CEO says a global pilot shortage is 'one of the biggest challenges' facing the airline industry.* Retrieved from CNBC.com: https://www.cnbc.com/2019/06/17/boeing-ceo-says-global-pilot-shortage-is-one-of-the-biggest-challenges.html?referringSource=articleShare

Michaels, D., Tangel, A., & Pasztor, A. (2021, 1 7). Boeing Reaches $2.5 Billion Settlement of U.S. Probe Into 737 MAX Crashes. *Wall Street Journal*. Retrieved from https://www.wsj.com/articles/boeing-reaches-2-5-billion-settlement-of-u-s-probe-into-737-max-crashes-11610054729

Nicas, J., Kitreoff, N., Gelles, D., & Glanz, J. (2019, 6 1). Boeing Built Deadly Assumptions Into 737 Max, Blind to a Late Design Change. *New York Times*, pp. https://www.nytimes.com/2019/06/01/business/boeing-737-max-crash.html.

Nicas, J., Kitroeff, N., Gelles, D., & Glanz, J. (2019, 6 1). Boeing Built Deadly Assumptions Into 737 Max, Blind to a Late Design Change. *New York Times*. Retrieved from https://www.nytimes.com/2019/06/01/business/boeing-737-max-crash.html

NTSB. (2019). *Safety Recommendation Report: Assumptions Used in the Safety Assessment Process and the Effects of Multiple Alerts and Indications on Pilot Performance.* Retrieved from https://www.ntsb.gov/investigations/AccidentReports/Pages/ASR1901.aspx

Penn, I., & Lipton, E. (2021, 05 06). The Lithium Gold Rush: Inside the Race to Power Electric Vehicles. *The New York Times*.

SAE. (2010). *Guidelines for Development of Civil Aircraft and Systems ARP4754A.* Society of Automotive Engineers. Retrieved from https://www.sae.org/standards/content/arp4754a/

Senge, P. M. (2006). *The Fifth Discipline.* New York: Currency.

Sider, A., & Tangel, A. (2019, 9 29). Before 737 MAX, Boeing's Flight-Control System Included Key Safeguards. *Wall Street Journal*. Retrieved from https://www.wsj.com/articles/before-737-max-boeings-flight-control-system-included-key-safeguards-11569754800

TSA checkpoint travel numbers. (2021, 5 28). Retrieved from tsa.gov: https://www.tsa.gov/coronavirus/passenger-throughput

USDOT. (2021, 5 20). *National Transportation Statistics*. Retrieved from Bureau of Transportation Statistics: https://www.bts.gov/content/transportation-fatalities-mode

Wikipedia. (2021, 5 18). *Boeing 737*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Boeing_737

Wikipedia. (n.d.). *List of Rail Accidents (2010–present)* . Retrieved from https://en.wikipedia.org/wiki/List_of_rail_accidents_(2010–present)

Wikipedia: Deutsche Bahn. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Deutsche_Bahn

Wikipedia: List of Rail Accidents. (n.d.). *List of Rail Accidents (2010–present)*. Retrieved from https://en.wikipedia.org/wiki/List_of_rail_accidents_(2010–present)

Yang, S. (2021, 04 16). The Chip Shortage Is Bad. Taiwan's Drought Threatens to Make It Worse. *The Wall Street Journal*.