Interstices in the Certification of Safety Critical Avionics Software: Boeing 737-MAX MCAS Case Study

Aiman Gannous
University of Benghazi
Benghazi, Libya
aiman.gannous@uob.edu.ly

***Abstract*—Two Boeing planes of the same model were crashed in October 2018 and in March 2019. All the passengers and crew onboard were killed. Investigations showed that an external failure in the newly installed critical software Maneuvering Characteristics Augmentation System (MCAS) was responsible of these two fatal accidents. In this paper we reviewed the reasons behind the failure of MCAS that installed in Boeing most selling airplane and investigated engineering and certifying such software systems. The case study revealed an urgent need to clarify the certification elements in a more practical way and to be linked to advancements in software engineering.**

***Keywords*—Boeing 737-Max, MCAS, Fail-safe Testing, Software Safety Certification, DO-178 C.**

I. Introduction

In a fatal two crashed Boeing 737-Max consequently in October 2018 and March 2019 in a similar circumstances minutes after takeoff, a total of 346 passengers and crows lost their life's. As a trivial result, to avoid more life lose, the new airplane has been grounded globally and a series of investigations started to determine what went wrong and caused the unfortunate accidents [1], [2].

Investigations showed that for almost the first time and at the best of our knowledge, a software was mainly the cause of these two crashes called Maneuvering Characteristics Augmentation System (MCAS).

It turns out that all of this started on the ground of competition. Boeing's main competitor Airbus had announced and started marketing their new A320 neo airplane, the updated version of the famous A320 series that has a long history of safety. This new A320 neo has a larger, fuel-efficient engines that adheres to the pressure of environment regulators to reduce Co2 emission that harms our planet. As a result, for Boeing to stay in the market, the company decided to go with updating their famous Boeing 737, which also has a long and safe history record too. They also decided to upgrade the 737-NG by installing new engine that is fuel efficient as well. However, it turns out that installing the new engine (LEAP-1B) on the 737-NG body will pose an engineering challenge because the 737-NG clearness from the ground is already low and the new engine will worsen the ground clearance problem even more. The solution that Boeing engineers came with was to mount the engines higher and more forward on the wings as shown in Figures 1 and 2 [3].

Fig. 1. New Leap Engine size on 737 MAX-8 (Right) vs. old engine on the 737 NG (Left) [4].

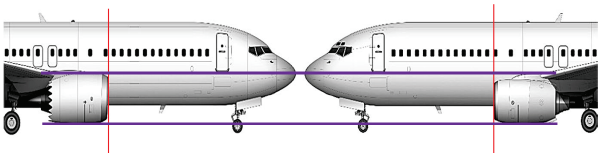

Fig. 2. New Leap Engine mount up and forward on 737 MAX-8 (Right) vs. 737 NG (Left) [3].

This solution however caused a new problem as it changes the aerodynamics of the plane. Moving the engines up and forward on the same body caused an extra lift generated at high Angle of Attack (AOA) and low speed, specifically at takeoff, hence, the handling characteristics will be different than the previous 737 models under certain flight conditions. Here comes the role of the MCAS system, a software that provides a solution to these problems designed to take control of the airplane to avoid stalling at low speed. MCAS rely on an important external component called the AOA sensors. AOA sensors send AOA data to MCAS and the system main function is to bring the nose down if AOA reading was high autonomously without the intervention of the pilots [3].

Now, what actually killed those 346 people is still debatable. There are several technical and certification factors. The most critical technical factor that MCAS was actually relying on

979-8-3503-2759-5/23/$31.00 ©2023 IEEE
DOI 10.1109/CSCE60160.2023.00425

one AOA sensor and does not have a mitigation action against external failures of the AOA sensor. On the certification side, defining the level of safety for the MCAS was the major issue.

In this paper, we present a qualitative and interpretive investigation to understand why Boeing ignored the external failure factors and if the use of proper safety engineering practice could prevent this failure. The analysis is limited to the MCAS system installed on the new Boeing 737 MAX airplane. The paper is structured as follows: in section II, we present a background about safety certification and testing safety critical systems. In section III, we describe the MCAS system. Analysis and results of the technical and certification flaws are presented in section IV. In section V, we discuss our findings. Conclusion and future work are presented in section VI.

II. Background

A. Fail-Safe Testing

In the development process of safety-critical systems (SCSs), testing is more rigorous than regular systems. Testing SCSs target is to verify that safety requirements were implemented correctly. Possible hazards shall be identified using safety analysis techniques such as Fault Tree Analysis (FTA) and Preliminary Hazard List Analysis (PHLA) [5] [6]. In addition, mitigation action should be proven to be working as intended against the identified hazards. Therefore, testing SCSs should include testing for proper mitigation of expected failures from internal and external system components.

B. Safety Certification

Software certification defined as the process of verifying that software products and processes comply with the corresponding domain defacto standards [7]. Domain standards certification have a set of requirements that guides the development process towards a successful certification that proves the safety of the system. DO-178C [8] is an example of such a defacto standard for embedded software development and certification in the avionics domain. DO-178C illustrates the processes of the software life cycle through activities and objectives. Requirement-based testing is adopted in DO-178C. DO-178C emphasizes the development of normal test cases to verify system functions and robustness test cases to verify system safety [9].

III. Maneuvering Characteristics Augmentation System (MCAS)

The description is covering MCAS when the crashes occurred and before the updates Boeing made to resume 737-MAX operations. MCAS is an automated system that augment pitch flight control law without pilots interference. MCAS is suppose to use two Angle of Attack (AOA) sensors to move the plane's horizontal stabilizer at 0.27 degree rate per second. MCAS is automatically activated when the AOA reading exceeds the allowed threshold during takeoff or crossing at low speed [10].

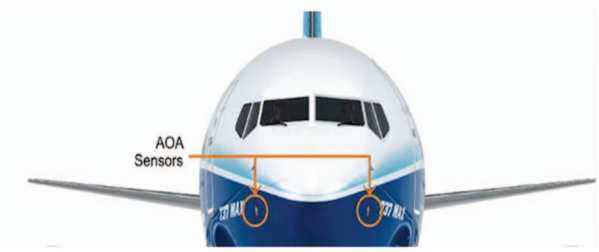

Fig. 3. AOA sensors location on the plane [1]

Even Though MCAS was not developed for stalling prevention in high speed, it's main function was to handle an undesired pitch-up action that the new engine lift causes, therefore, prevent the pilot from pulling the yoke unwittingly harder than usual to avoid an accidental stall [11].

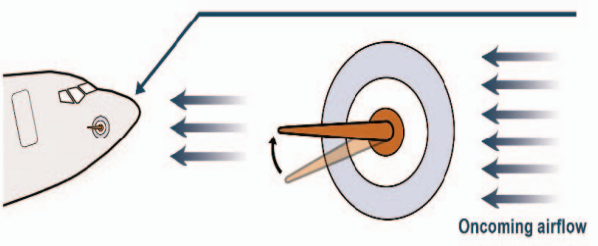

Fig. 4. AOA changes as the airplane nose goes up/down [4]

With data coming from airspeed, AOA, and altitude sensors, MCAS can detect if the AOA is higher than the safe threshold, therefore, it will control the plane trim to lower the nose, modify the rear stabilizer and push the yoke down [12].

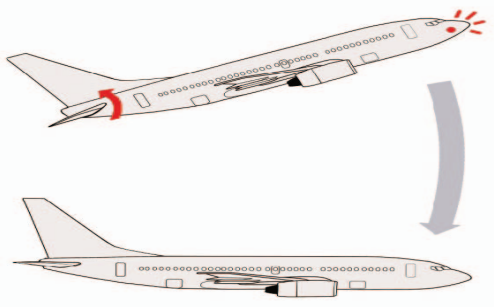

Fig. 5. MCAS controls the stabilizers to bring the aircraft nose down [4]

Pilots can override the MCAS using trim controls, however, if the the trim switcher released and AOA is still over the threshold, MCAS will be reactivated again after 5 seconds. To fully disengage MCAS pilots must use a CUTOUT switch [12].

IV. Analysis

In this analysis, we will investigate elements of the practice in developing and certifying MCAS with respect to two

major highly correlated aspects: certification and software engineering.

1. Certification

The beginning was with the false identification of the MCAS safety class. Design Assurance Level (DAL) as defined in DO-178C is determined from the safety assessment process by examining the effects of a failure condition in the system. The failure conditions are categorized as follows [13]:

Catastrophic: Failure may cause deaths, usually with loss of the airplane.
Hazardous: Failure has a large negative impact on safety or performance, or causes serious or fatal injuries among the crew and passengers.
Major: Failure significantly reduces the safety margin or significantly increases crew workload. May result in passenger discomfort or minor injuries.
Minor: Failure slightly reduces the safety margin or slightly increases crew workload. Examples might include causing passenger inconvenience.
No Effect: Failure has no impact on safety, aircraft operation, or crew workload. Examples might include passenger entertainment system or WiFi connections.

The certification authorities usually require that the correct DAL should be established using comprehensive analyses methods and and to be justified as well. Any software that commands, controls, and monitors safety-critical functions should receive the highest DAL - Level A which is the Catastrophic class [8].

Second, it is the independence problem. The software safety level also determines the number of objectives to be satisfied for successful safety certification. Independence which refers to a separation of responsibilities where the objectivity of the verification and validation processes is ensured by virtue of their ”independence” from the software development team. For objectives that must be satisfied with independence, the person verifying the item (such as a requirement or source code) may not be the person who authored the item and this separation must be clearly documented [8].

2. Software Engineering

Regarding software engineering, the MCAS design and implementation should be escalated to obtain safety safety requirements as the MCAS system correct classification is a safety critical component in the flight control system according to the DO-178C certification. Hence, at least three critical elements should be respected in the software development process: Redundancy, Fail-safe testing, Mitigation Testing and Traceability. It is obvious that in the implementation of MCAS redundancy was ignored as a critical safety requirement. External failures are highly expected when the system relies on external components such as sensors. AOA has a long history of faulty reading, therefore MACS should rely on at least two or three sensors as recommended in the design of such systems. AOA reading should be received based on agreement between multiple independent AOA sensors to avoid failures. Unfortunately, there is no evidence that Fail-safe testing nor Mitigation testing were executed on the MCAS. Traceability was also ignored. In the certification of safety-critical systems, safety requirements should be verified and evidence must be presented by linking testing results to each and every safety requirement.

V. Discussion

In this research, we were able to determine the main reasons behind the failure of MCAS on the two crashed Boeing 737-MAX airplanes. The certification process was not executed as expected starting from miss-classifying the MCAS system level of safety up to not proving independent verification. With respect to software engineering practice, the development process of the MCAS system and deployment was missing important aspects such as ensuring that the system is safe and/or show that system failure are mitigated. An important fact worth mention and could be surprising, is that MCAS was not actually new. In other words, the 737-MAX aircraft was not the first plane to use the MCAS system. MCAS was actually implemented first on the Boeing KC-46 Pegasus military aerial refueling tanker. It was also used to stabilize the tanker but not for the same design issues of the 737-MAX. In KC-46 Pegasus, MCAS used to stabilize the aircraft because of the weight and balance shifts when the tanker dispense fuel in the air. Suspicions raised about the possibility that Boeing engineers or/and the FAA assumed that since the system was used before and certified on the KC-46 Pegasus, there is no need to re-certify it again. This could be another research problem that could be tackled by researchers to contribute by studying the need of re-certifying such systems when used for the same purposes but in different circumstances. At the best of my knowledge, there is nothing in the literature discussing what testing methodology had been used by MCAS developers to test their product. This raises a hard question actually as we cannot determine if the industry is working closely with software engineering research institutions to adopt the latest advancement in the field of verification and validation of safety-critical systems. Unfortunately, there are some questions we couldn’t find an adequate answers for them: 1. Where is DO-178C in all this? 2. What kind of safety evidence that Boeing provided to show that MCAS is safe? 3. Since MCAS has been used in another plane, did Boeing reused a previous certification? Finally, we cannot ignore the fact that safety certification documents still contain some ambiguities regarding the guidance in obtaining certification and prove that system is safe. Therefore, arguably, the industry of safety-critical avionics systems could use this as an escape ticket from a complete responsibility of these failures.

VI. Conclusion and Future Work

In the two 737-MAX crashes, investigations showed that a software was actually charged of killing 346 people. As an avionic safety-critical system, MCAS was designed to overcome an engineering problem caused by installing new

bigger engine on the same body of the Boeing 737-NG. The new engine is a cutting edge technology in fuel efficiency, however, it changes the aerodynamics of the airplane, for this reason MCAS was developed to overcome issues in controlling the airplane and keep the passengers and crew safe. However, this took a hard U turn into two catastrophic accidents. Questions raised about who is actually responsible? How MCAS software actually designed? and What is the benefit of safety certification if such systems can cause death? In this paper, we tried to analyze the failing elements in the certification and software engineering of MCAS. Contradictions were found as the correct DAL was not established correctly at the beginning. In addition, other elements in certification such as "independence" was not clearly reported in the validation process. Regarding software engineering, the most surprising part is that external failures were ignored since MCAS relies on only one AOA sensor at a time. In future work, we will further investigate MCAS system failures by applying different fail-safe testing techniques and formal verification methods to discover if such failures could be detected and how they could be linked to certification objectives to be presented as safety evidence.

REFERENCES

[1] H. Matt. (2020) Killer software: 4 lessons from the deadly 737 max crashes. [Online]. Available: https://www.fierceelectronics.com/electronics/killer-software-4-lessons-from-deadly-737-max-crashes

[2] G. Dominic. (2019) Flawed analysis, failed oversight: How boeing, faa certified the suspect 737 max flight control system. [Online]. Available: https://www.seattletimes.com/business/boeing-aerospace/failed-certification-faa-missed-safety-issues-in-the-737-max-system-implicated-in-the-lion-air-crash/

[3] J. Herkert, J. Borenstein, and K. Miller, "The boeing 737 max: Lessons for engineering ethics," *Science and engineering ethics*, vol. 26, no. 6, pp. 2957–2974, 2020.

[4] US Department of Transportation. (2021, February) Weaknesses in faa's certification and delegation processes hindered its oversight of the 737 max 8.

[5] E. Clifton, *Hazard Analysis Techniques for System Safety*, 1st ed., 2005.

[6] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2015.

[7] R. T. V. Braga, O. Trindade, Jr., K. R. L. J. C. Branco, and J. Lee, "Incorporating certification in feature modelling of an unmanned aerial vehicle product line," in *Proceedings of the 16th International Software Product Line Conference - Volume 1*, ser. SPLC '12. New York, NY, USA: ACM, 2012, pp. 249–258. [Online]. Available: http://doi.acm.org/10.1145/2362536.2362570

[8] RTCA Inc. (2013) Software Considerations in Airborne Systems and Equipment Certification. [Online]. Available: https://www.rtca.org

[9] L. Rierson, *Developing safety-critical software: a practical guide for aviation software and DO-178C compliance*, 1st ed., 2013.

[10] Federal Aviation Administration. (2020, November) Summary of the faa's review of the boeing 737 max.

[11] S. Makó, M. Pilat, P. Šváb, J. Kozuba, and M. Čičváková, "Evaluation of mcas system," *Acta Avionica Journal*, pp. 21–28, 07 2020.

[12] P. Johnston and R. Harris, "The boeing 737 max saga: lessons for software organizations," *Software Quality Professional*, vol. 21, no. 3, pp. 4–12, 2019.

[13] S. M. H. Yelisetty, J. Marques, and P. M. Tasinaffo, "A set of metrics to assess and monitor compliance with rtca do-178c," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, Sep. 2015, pp. 8D2–1–8D2–6.