The Societal Impact of Untested Software: A Boeing Case Study

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Jassiel Mendoza

Spring, 2024

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor

Bryn E. Seabrook, Department of Engineering and Society

STS Research Paper

Introduction

Steve Jobs's introduction of the iPhone in 2007 marks a critical turning point in modern technology, as it then became possible to wield the power of a computer in the palm of a hand. Since then, technology has rapidly evolved. Breakthroughs in chip manufacturing processes have led to increases in computing power, enabling technology to power more and more things, so much so that technology has become ubiquitous in daily life. The evolution of technology goes hand in hand with that of the software that drives it, becoming increasingly complex as well. But unlike physical hardware, software is much harder to comprehensively test. And yet, software has infiltrated every aspect of daily life. The old way of doing things is no longer viable as "there is an app for that". For the most part, software works and does exactly what is needed from it, but what happens when it does not? No piece of software will ever be perfect, given the right conditions, it can fail. At what point, then, is a piece of software tested enough? Who makes this decision? Who is responsible if things go wrong? These are important considerations software engineers face as dependency on software continues to increase. Through the analysis of a case study about the software failure on Boeing's 737 Max, this paper answers the question, what are the societal ramifications of releasing software to the public that will never be thoroughly tested?

Methods and Research Question

Through a deep dive investigation that aggregates insights, technical understandings, and detailed descriptions from various sources gathered from articles, expert reports, aviation industry news outlets, and testimonies, the complex case study of Boeing's 737 MAX failures in 2019 is examined. The involved actors are first identified, and the ethics behind their decisions and how these decisions affect other actors and the overall network are analyzed. The points made for the Boeing case study are then extrapolated and generalized to reiterate the sentiment

that companies must prioritize ethics over monetary gain when developing and testing software. Then, the limitations of the research are addressed, acknowledging that although software will never be fully tested, this should not inhibit change from occurring to minimize risk. Ultimately highlighting the importance of proper software testing for applications released to the public, and answering the question: what are the societal ramifications of releasing sensitive software to the public that will never be thoroughly tested?

Background Information

To better understand these circumstances, it must be understood how software testing is currently done within the software engineering industry and the standards in place, if any, used to ensure the quality of software products. Software testing is a continuous process that ideally starts from the beginning of the software development cycle, the design phase, with the main objective being to verify and validate an application meets the user requirements elicited by the client. There are various methods and types of software testing that each have specific objectives and strategies. In general, though, software testing is split into three broad categories. Functional testing, which focuses on ensuring the software functions as intended by comparing actual to expected output. Non-functional testing focuses on assessing properties of software that contribute to the user experience, such as stress testing. And maintenance, which ensures that changes or updates to the software post-release do not break previous functionality. All of these categories are pivotal in delivering a quality software product. However, it is important to note that no piece of software will ever be "perfect". Software will always be prone to unforeseen bugs as it would be impossible to test for every edge case imaginable, apart from it being impractical. Additionally, after software is released it "ages", meaning that new bugs arise over time due to bugs in libraries it is dependent on or being vulnerable to new attack vectors

employed by malicious actors. Even software with minimal errors is still prone to the bugs and vulnerabilities of the underlying operating system, as well as the inconsistencies of the hardware it is running on (Stack Exchange Inc., n.d.).

In practice, various software testing standards have been developed and outlined by reputable organizations, such as IEEE 29119, which cover testing processes, documentation, techniques, and management (LinkedIn, n.d.). However, since software spans multiple industries, all of which have different requirements, these standards are not mandatory unless there is a legal requirement. Often this is seen in software used for expensive or dangerous applications, such as the software running on an airplane computer where a bug or failure can lead to a loss of life. In such cases, there are strict standards software engineers must adhere to for software to be accepted and released to the public; the FAA would be such an organization that enforces a strict set of standards.

Boeing's 737 MAX was grounded in 2019 over safety concerns after 346 people died in two similar crashes. This is a complex story whose roots are traced back to the 1960s when the first iteration of the 737 aircraft was designed. Boeing had purposefully built the aircraft low to the ground for it to be compatible with smaller airports as a way to increase its market share with airlines since only portable staircases would be needed to board and unboard passengers (Travis, 2019). Since then, engine manufacturers have made breakthroughs in engine efficiency, with the tradeoff of engine sizes increasing compared to the small counterparts of the 1960s. Since the initial 737 was built low to the ground, however, these bigger engines caused problems as they were so big that they would not fit onto the wings in the original design of the 737. But, to maintain its popularity with airlines, Boeing wanted to keep the same low-to-the-ground profile that made the original 737 flexible for various airports. Another reason Boeing wanted to

maintain the design is so that they would be able to classify this new aircraft as belonging to the 737 platform that airlines and pilots were already accustomed to and familiar with, eliminating the need for expensive pilot training. Boeing was able to fit the new engines by moving the engines further forward and raising them to achieve enough clearance. However, the engine nacelle, which is the casing that houses the engine itself, had an aerodynamic quirk that caused the nacelle to generate additional lift at a high angle of attack (Travis, 2019). Boeing's solution to this was to install a new system, the maneuvering characteristics augmentation system (MCAS), which used sensors to detect when the angle of attack was too high and would consequently pitch the nose of the plane down. However, this would be considered a new system by the FAA which would require Boeing to notify and train pilots, which is what Boeing wanted to avoid. Consequently, Boeing requested that the FAA remove references of MCAS from the flight manual, which the FAA approved (Committee on Transportation and Infrastructure, 2020). Therefore, when the angle of attack sensors failed, causing MCAS to pitch the plane into an uncontrollable nosedive, the pilots did not know how to react accordingly as they were fighting a system that they had not been made aware of.

Actor-Network Theory:

Actor-Network Theory (ANT) is an STS framework which's creation is frequently attributed to Michel Callon, Madeleine Akrich, Bruno Latour, and John Law. The theory is predicated on the idea that social and non-social networks are a result of ever-evolving relationships between human and non-human actors. ANT tries to explain the internal processes of these networks by "tracing the complex relationships that exist between governments, technologies, knowledge, texts, money and people" (Cressman, 2009). ANT refers to actors both human and non-human as having equal agency in a network and reinforces the view that

"technology emerged from social interests and that it thus has the potential to shape social interactions" (Cresswell et al., 2010).

ANT is a difficult concept to grasp and as a consequence is often criticized for being too abstract. Critics argue that ANT fails to "come up with any detailed suggestions of how actors should be seen, and their actions analysed and interpreted" and therefore should not be considered a "theory" (Cresswell et al., 2010). ANT critics also argue that the theory fails to "take into account human intentions, interests between different groups, morals, learning, backgrounds, routines, culture and previous experiences of human actors," all of which shape an actor's role in a network (Cresswell et al., 2010). However, ANT is still regarded as an innovative approach used to "reveal the complexities of our sociotechnical world" (Cressman, 2009). Especially when paired with other theories, it is still a widely used framework in the field of STS.

In the context of this paper, ANT language is used to serve as a guiding framework for the analysis of the Boeing case study. The principal actors identified in the case study are Boeing executives, Boeing engineers, the Federal Aviation Administration, and the general public. This network of actors represents the different parties involved or affected by the events leading to the Boeing 737 MAX crashes in 2019. Each actor group is evaluated to determine how their actions affect the network and others or how the actions of others impacted them. Through the dissection of the case study using the ANT framework, the moral obligations between these groups and their interactions are analyzed to examine if ethical duties were upheld.

Case Study

Releasing untested, buggy software to the public can have devasting repercussions on society as a whole, as well as the reputation of the company. Therefore, should it not be in a

company's best interest to advocate for the proper testing of their software, as the failure in doing so would derail the company's hard-fought success? The following Boeing case study demonstrates the extent to which greed can blind even companies that had a reputation for being primarily engineering-driven. Lessons learned from these events prove that the adherence and prioritization of ethics is a worthy long-term investment.

Boeing Executives

All paths that led to the tragic crashes of the Boeing 737 MAX trace back to decisions made by Boeing executives. By definition, the role of executives in any company is to oversee the operation of a company and ensure it is on track to meet outlined goals and objectives in a manner that aligns with company policy and vision. However, business executives also must satisfy a company's shareholders, typically achieved through the creation of the aforementioned goals and objectives in such a manner that would ideally lead to company growth, market share, and consequently profit. Historically, Boeing was known to be an engineering-first company with a reputation for building the safest and most advanced airplanes. However, after a merger with McDonnell Douglas Corporation, a fellow aerospace manufacturing company, a cultural shift within the company changed its identity into that of a business-oriented company. This is evident through Boeing's relocation of its headquarters to Chicago which served as a way of distancing top management from influences of its production plants and engineers to a "neutral" location allowing for more global growth opportunities.

Airbus's launch of its more fuel-efficient, larger airliner catalyzed the creation of the 737 MAX. Boeing was losing market share as a result, causing Boeing executives to pressure engineers and production plants to "overcome all obstacles and put the 737Max into operation as scheduled" (Zhang et al., 2022). Boeing executives were catering to the wants of the

shareholders and prioritizing profit, even over safety, purposefully bypassing the engineering process and neglecting its moral obligations as Boeing was willing to accept the "trade-off between technical airworthiness and operating costs" to maintain its competitiveness and recapture market share from rival aviation company, Airbus (Zhang et al., 2022). Critically, this meant that executives did everything in their power to cut corners, resulting in key design changes such as MCAS not "[undergoing] a sufficient safety assessment, and the cost of flight tests, manuals update, and pilot training [being] reduced" (Zhang et al., 2022). The pressure exerted by the executives choosing greed over safety had a ripple effect throughout the entire company; engineering work was outsourced at a cheaper cost, known defective parts were still used in assembly, and the deliberate effort to avoid new certification costs, ultimately resulted in the Lion Air and Ethiopian Airlines accidents.

These design decisions made by Boeing executives resulted in technology shaping society. The decisions and pressure exerted by Boeing executives to cut costs wherever possible, and forcing its engineers to provide a software solution to mask the 737 MAX's hardware problems, directly resulted in a faulty system. Software code is deterministic, it does exactly what it is told to do regardless of the developer's intention, meaning the bugs in MCAS were not an unfortunate circumstance. By preventing adequate testing of MCAS, which could have caught and rectified the issues, Boeing executives neglected their ethical duty to the general public to supply a safe product and are directly responsible for the crashes in 2019.

Boeing Engineers

Engineers hold a unique position as they are the group of people with the most intimate knowledge of a product's intricacies and functionality, but often do not have the power to dictate the direction of the product. As a consequence, they hold a moral obligation to all the relevant

stakeholders, but especially to the users, to ensure that what they are working on and the company they work for uphold all necessary ethical considerations. A common tool available to engineers wanting to safely voice their concerns is the act of whistleblowing, which serves as an avenue that focuses on finding root causes to resolve the issue instead of finding someone to blame. For this reason, engineers must take full advantage of whistleblowing, especially when working in the safety-critical aviation industry.

Given the obvious direction Boeing executives were taking by prioritizing company profits, Boeing engineers should have been vocal about safety concerns. However, Boeing has been accused of having a toxic whistleblower culture as many engineers and employees have come out stating that they avoided raising their concerns in fear of retaliation and career hampering. Additionally, engineers have a distrust of the systems in place as they question the effectiveness of such systems, as whistleblower John Barnett stated "[Boeing] culture is all about speed and production and getting airplanes out the door. And any issues, any concerns that you bring up are going to slow them down" (Garcia, 2024). Moreover, to cut costs, Boeing outsourced engineering to temporary workers, "The Max software…was developed at a time Boeing was laying off experienced engineers and pressing suppliers to cut costs" (Robison, 2019). In doing so, Boeing started bypassing the senior engineers who knew the risks and considerations needed to account for when creating aviation software to less experienced, temporary workers who simply created the software to the specification of Boeing. This meant that although the software itself was functional, it was not created with the proper techniques and methods that would have been required of full-fledged aviation software. The outsourcing of work also signaled Boeing's sentiment towards existing engineers as being expendable since the

work could be outsourced for cheaper. Further reinforcing the fear engineers and employees already had towards raising concerns in fear of reprisal.

Boeing engineers did not hold much agency in preventing the creation of the flawed MCAS. The primary tool that should have been available for them to report their concerns was inefficient by design, and any further escalation would have resulted in a hindered career. Additionally, the disconnect between the software engineers who designed MCAS and the standard procedures of the cockpit can be attributed to the decision made by Boeing's upper management to outsource engineering. As a result, the calibration of trust between engineers and pilots was disrupted, as pilots operate under the assumption that engineers build and test systems with safety as a top priority. Furthermore, the lack of communication with pilots regarding MCAS and its functionality, further empowered technology to shape society.

The Federal Aviation Administration

In the aviation industry, the Federal Aviation Administration (FAA) is an important regulating body put in place to serve as a check and balance for aviation companies. This is to ensure the safety of the general public by enforcing strict regulatory guidelines that force companies to uphold quality and safety in their products. The sole purpose of the FAA is to act as a neutral third party to objectively determine if a company has adhered to safety in its designs and manufacturing processes. However, a deep investigation into the development and certification of the Boeing 737 MAX revealed that the FAA "missed its own opportunities to change the direction of the 737 MAX based on its aviation safety mission" (Committee on Transportation and Infrastructure, 2020). Such a catastrophic oversight is not acceptable for the sole governing body put in place to ensure the safety of the public. The Boeing 737 MAX failure reveals that the "current regulatory system is fundamentally flawed and needs to be repaired"

(Committee on Transportation and Infrastructure, 2020). How, then, can the general public be expected to put their trust and faith into the aviation industry, as the agency meant to regulate aviation companies has proved incompetent at ensuring their safety?

Private companies' ability to lobby the government also raises concerns on the question of whether the FAA can truly remain unbiased and objective in its duty. Evident by the FAA's reluctance to ground the 737 MAX airplanes as other aviation administrations around the world had done after the crashes reveals the effect lobbying can have on what is supposed to be an unbiased and objectively neutral regulating body. Additionally, in the certification of the 737 MAX, the FAA "showed excessive trust and reliance" on Boeing to internally ensure the safety of their systems, failing to perform their sole objective and "perform its duty of supervising aircraft safety on behalf of the public" (Zhang et al., 2022). This oversight from the FAA is what allowed Boeing to sweep many errors under the radar and get away with poor manufacturing, quality control, and implementation of faulty software. The FAA failed to meet its mission statement, "to provide the safest, most efficient aerospace system in the world" and failed the public by allowing certification of the 737 MAX by relying on Boeing to report their faults; further allowing them to prioritize profit by relinquishing control (Federal Aviation Administration, n.d.).

The FAA's mismanagement allowed Boeing to act in their interest, undermining the authority and sole purpose of the FAA. This jeopardizes the trust calibration between itself, pilots, and the general public by failing to meet their ethical and moral obligations to them. The FAA could have prevented such a catastrophe on various occasions but failed to do so, and even further empowered Boeing by approving the removal of MCAS references from the flight manual. This decision, ultimately led to pilots being unaware of MCAS and therefore unable to

act accordingly when the system failed. Adding yet another point of failure, leaving society at the mercy of flawed technology.

General Public

It is estimated that there are 22.2 million flights worldwide every year, demonstrating the power the general public holds over the commercial aviation industry as it relies on massive ticket sales to be profitable. This, in turn, requires the general public to put trust in the aviation industry and the checks and balances put into place to ensure their safety. Statistically, this has proven to work as air travel is one of the safest modes of transportation in the world. The problem arises when companies and governing bodies cut corners and put an emphasis on profits and appeasing shareholders that they lose sight of what is truly important, safety. The recent 737 MAX failures have proven this as the general public has grown to distrust Boeing and even the FAA in their ability to create safe flight travel systems. The gravity of Boeing's greed is to such an extent that even its executives, engineers, and employees would not fly on Boeing's planes. A former senior manager publicly stated, "I would absolutely not fly on a Max plane. I've worked in the factory where they were built, and I saw the pressure employees were under to rush the planes out the door," alluding to first-hand knowledge of the poor quality of work that goes into the engineering and manufacturing of these planes (Jones, 2024). Boeing's reputation precedes itself and is far from its once envious prestige highlighted in the slogan "if it's not Boeing I'm not going," emphasizing Boeing's decline in reputation. Unfortunately, the price of the shortcomings of Boeing and the FAA are paid by the general public.

Testing Software

The Boeing 737 MAX case study teaches many valuable lessons relating to the importance of software testing. Especially when it involves software used in sensitive industries,

where loss of life can occur, software must undergo rigorous testing to ensure its reliability and functionality as much as possible. Boeing has demonstrated that a company enthralled in increasing its profit will cut corners and costs, even if it means sacrificing the quality and standards of its products. Although it is hard to test software for all scenarios it will be used in and it is not possible nor practical for companies to catch every bug and error in the software, companies must be held to a high standard and urged to prioritize the ethical creation of their products over sacrificing it in the pursuit of profit. In such safety-critical industries especially, it is the moral and ethical obligation of companies to ensure society drives technology and not the other way around, as technology is imperfect.

Limitations:

Software testing is not a trivial task and this is why such a large portion of a product's development budget is allocated to testing. Ultimately, a piece of software will never be fully tested to the point where the software is "perfect" and exhibits no flaws, errors, or bugs. However, companies can still commit to upholding and meeting the strict regulations set out by regulatory agencies as a way to minimize the potential of software failing catastrophically. This alone, however, will not deter greedy companies from devising strategies that fool and cheat the system.

Future Work:

Software testing is a hot topic in research, as companies and researchers alike are continuously searching for new cost-effective ways to test software at scale. If the difficulty of testing software is lessened not only financially, but also in terms of time, companies would be less likely to circumvent their responsibility to test their software. Additionally, companies need to establish proper, effective channels internally that allow employees to raise their concerns

without fear of retaliation, to catch mistakes earlier in the development process. Finally, companies and regulating agencies need to ensure that they uphold ethical guidelines in every avenue throughout the company.

Conclusion:

Releasing faulty software is detrimental to the well-being of society, with devasting consequences especially prevalent in critical industries. Developers and companies need to consider the potential effects their software can have, no matter how minuscule they may seem. These small allowances of seemingly inconsequential effects will start to accumulate and result in a non-linear impact on complex systems with catastrophic outcomes. Such was the case with Boeing after their merger with McDonnell Douglas that resulted in a cultural shift within the company which ultimately led to dramatic oversight in the creation of safety-critical systems that were at fault for the loss of life in the 2019 crashes. Companies must realize that upholding ethical standards is non-negotiable and should not become an afterthought as a result of greed.

References

Committee on Transportation and Infrastructure. (2020, September 15). *Final committee report: The design, development & certification of the Boeing 737 Max*. U.S. House Committee on Transportation and Infrastructure. Retrieved from https://democrats-transportation.house.gov/imo/media/doc/2020.09.15%20FINAL%20737%20MAX%20Report%20for%20Public%20Release.pdf

Cressman, D. (2009). *A Brief Overview of Actor-Network Theory: Punctualization, Heterogeneous Engineering & Translation*. https://summit.sfu.ca/_flysystem/fedora/sfu_migrate/13593/0901.pdf

Cresswell, K. M., Worth, A., & Sheikh, A. (2010). Actor-Network Theory and its role in understanding the implementation of information technology developments in healthcare. *BMC Medical Informatics and Decision Making*, *10*(1). https://doi.org/10.1186/1472-6947-10-67

Federal Aviation Administration. (n.d.) Mission.Retrieved from https://www.faa.gov/about/mission

Garcia, M. (2024, March 12). *Boeing's flawed "speak up" policy and the death of a whistleblower.* Forbes. Retrieved from https://www.forbes.com/sites/marisagarcia/2024/03/12/boeings-flawed-speak-up-policy-and-the-death-of-a-whistleblower/?sh=7b61c6d133ef

Jones, Callum. (2024, February 1). *Ex-Boeing manager would 'absolutely not' fly 737 Max after cabin panel blowout*. The Guardian. Retrieved from https://www.theguardian.com/business/2024/feb/01/boeing-manager-737-max

LinkedIn. (n.d.). *How do you identify widely accepted software testing?*. LinkedIn. Retrieved from https://www.linkedin.com/advice/3/how-do-you-identify-widely-accepted-software-testing#:~:text=Software%20testing%20standards%20are%20sets,terminology%2C%20processes%2C%20and%20criteria

Robison, Peter. (2019, June 28). *Boeing's 737 Max software outsourced to $9 an hour engineers.* IndustryWeek. Retrieved from https://www.industryweek.com/supply-chain/article/22027840/boeings-737-max-software-outsourced-to-9-an-hour-engineers

Stack Exchange Inc. (n.d.). Is it possible to reach absolute zero bug state for large scale software? Software Engineering Stack Exchange. Retrieved from https://softwareengineering.stackexchange.com/questions/195571/is-it-possible-to-reach-absolute-zero-bug-state-for-large-scale-software

Travis, Gregory. (2019, April 18). *How the Boeing 737 Max disaster looks to a software developer*. IEEE Spectrum. Retrieved from https://spectrum.ieee.org/how-the-boeing-737-max-disaster-looks-to-a-software-developer

Zhang, L., Sun, J., Bardell, N., Liu, Y., Zhang, H., & Dai, X. (2022). Trade-off between Technical Airworthiness and Operating Costs Study Based on B737 MAX MCAS System Failures. *Proceedings of the 5th International Conference on Computer Science and Software Engineering*, 34–38. Guilin, China. https://doi.org/10.1145/3569966.3569975