Software can take you to the Moon, and Software can take you to the ground

Bilal Malik, 104435995, January 6, 2020

Abstract- In recent history, Software has frequently been used as one of the tools in helping reach discovery. Software has helped ease the user's job, whether by problem solving or completing a minute task that may be much to repetitive yet crucial for a human to take. As software has come into the picture, more and more as a helpful instrument; its flaws have been that much more magnified. Many projects have been structured to include software as a fundamental base, and many of these as well fail because of this same software.

I. Introduction

An instance where this situation is shown is the Ariane 5 disaster. On June 4, 1996, the Ariane 5 rocket lifted off from French Guiana. This seven-billion-dollar project including a separate cargo valued at five-hundred-million dollars in [1], exploded a mere 40 seconds [2] after its lift off. A failed project that would have been even more devastating had there been a flight crew aboard, thankfully this rocket was unmanned as shown in [1].

I. Overview

In the following weeks, a report was issued by the board of inquiry. This report outlined a specific software error in the in the inertial reference system as the cause of failure. The software had originally been written for the Ariane 4, in the "Ada" [3] programming language. Efficiency considerations for the computer's workload, had led to a decision of "handling four variables", while leaving the other three as "unprotected" [4]. One of these unprotected variables was the horizontal bias variable.

II. Result of Failure

Compounding to these issues was the decision to reuse an entire Ariane 4 subsystem in the Ariane 5 in [3], the Ariane 4 model "being brought forward more then ten years [3]" after its completion in the Ariane 5, which included the flaw spoken about previously. This again can be drawn to the decision for a more cost-effective approach. Now can be seen the trade-off between safety and cost efficiency, to which this organization had sought fit to choose the latter. Ironically this desire to take short cuts in the hope of reducing cost, led to a 7-billion-dollar project exploding in the air without so much as breaking the atmosphere.

III. Specifications

Specifically, this variable - relating to the horizontal velocity of the rocket with respect to the platform, was to be converted from "a 64-bit floating point number - to a 16-bit signed integer [1]". It was deemed safe in the earlier rocket model, as it was "physically limited or that there was a large margin off safety [3]". The failure in Ariane 5 was because the floating-point value was too large to be represented by a 16-bit signed integer. "The Ariane 5's faster engines trigger a bug in an arithmetic routine inside the rockets flight computer [2]." In twos-complement the behavior is to discard the higher order bits, this will lead to crucial information being lost in translation. An integer is 32 bits and a short integer is 16 bits in Ada, converting the value 0x1248642 will result in a value of 0x8642. This 16 bit integer was signed and "the most common practice in systems is to use the twos-complement [4]", which would entail the highest order bit, or the right most bit, would decide the sign of the value and not be used in representing the number as a digit. In this case the resulting numeric value is -31166 calculated in [4]. In the case of the rocket, "the number [being converted] was larger than 32,767, the largest integer storable in a 16-bit signed integer [1]".

IV. Further

This conversion failure led to an operand error which led to the entire inertial reference systems ceasing all forms of functionality. This is because once the operand error had occurred, the exception handling mechanism alerted was tasked to firstly communicate this error to the data bus, then to shut down the SRI processor in its entirety [3]. This second step to proved fatal for the rocket and the mission itself. At this stage the Inertial Reference System became useless. This decision to shut down the entirety of the system instead of allowing it to be handled by a backup system can faulted to the organization. The document report concludes that "The reason behind this drastic action lies in the culture within the Ariane programme of only addressing random hardware failures. From this point of view exception - or error - handling mechanisms are designed for a random hardware failure which can quite rationally be handled by a backup system. [3]"

VII. Conclusion

We can conclude many different mistakes were made at many levels of the organization which in turn led to the failure of Ariane 5. At a higher level, the organization and its management teams promoted a culture within the organization that only realized random hardware failures and failed to see them as reactions to other failures in the system. The hardware might not have been the issue, instead the software manipulating it, but that made no difference to the management teams. Decisions to reuse old systems for a new project, all the while containing bugs in the earlier model, can be pointed again at management and higher levels of the organization. This logic comes from the idea of cost efficiency at he expenses of safety. At a lower level the developer can be held responsible for the faulty conversion, though they were not given tools through testing and whatnot to conclude that the code written is correct. The fatality of this project occurred not by a single conversion, but then by the mishandling of this conversion by the rest of the system.

VIII. Possible Procedures of Avoidance

This software failure could have been avoided by the obvious mishandling of the variable representing the horizontal bias. This issue could have been resolved by either handling the error thus protecting the variable, or by outlining both the value before conversion and after as unsigned to get more appropriate readings, though this may still have caused loss in information. Developers should have been given tools to test and realize that the conversion was not correct.

XI. Further

Though stopping the correction there is another mistake, the "Ariane culture led to the idea of simply deeming faulty hardware as just that and ceasing operations" [3]. Though the variable was computed incorrectly, there could have been a backup system utilized to find and recompute the error without shutting down the entire system. At a higher level, the management team's decision to reuse the subsystem from the Ariane 4 instead of developing new code, for a more cost-effective approach is another mistake worth correcting. As one of the upper level management coordinators, ones who are in control of expenses, they should have thought of redesigning the subsystem for a truly safer project. Yes, correcting the mistake with conversion would have allowed for the rocket to continue its mission, though this work culture would have continued until another rocket failed. These are the many different solutions that should have been implemented together to ensure that the Ariane 5's failure could have been avoided.

Bibliography

General Internet Site

[1] Douglas N. Arnold , “The Explosion of Ariane 5,” in http://www-users.math.umn.edu/~arnold/disasters/disasters.html explosion of Ariane 5 link, Some disasters attributable to bad numerical computing, link II, August 1998, Available: http://www-users.math.umn.edu/~arnold/disasters/ariane.html [Accessed: January 6, 2019]

Internet Journal

[2] Simon Garfinkel , “History’s Worst Software Bugs”, wired.com, June 4, 1996 – Ariane 5 Flight 501, November 8, 2005, Available: https://www.wired.com/2005/11/historys-worst-software-bugs/?currentPage=2 [Accessed: January 7, 2019]

Internet Archive

[3] Prof. J. L. Lions , “Ariane 5 Flight 501 Failure”, web.archive.org , Available: https://web.archive.org/web/20000815230639/http://www.esrin.esa.it/htdocs/tidc/Press/Press96/ariane5rep.html [Accessed: January 7, 2019]

Professional Internet Site

[4] Keith Thompson , “When an int is cast to a short and truncated, how is the new value determined?”, stackoverflow.com, January 20, 2016, Available: https://stackoverflow.com/questions/34885966/when-an-int-is-cast-to-a-short-and-truncated-how-is-the-new-value-determined [Accessed: January 6, 2019]