The Ariane 5 Explosion

By: Nicholas YC Chu

The Ariane 5

• The Ariane 5 was a rocket used to bring payloads into orbit. No humans on board.



Event History

- The Ariane 5 was launched 9:34 am, June 4, 1996.
- 36 sec. later, at 3700 meters, the Flight Control System failed.
- Rocket swung in the position of 2 solid boosters.
- 39 sec after launch, the high aerodynamic loads caused a 20 degree offset.
- 40 sec after launch, the self-destruct sequence was initiated.
- The Ariane 5 was totally destroyed.

Inertial Reference System (IRS)

- Used during the launch sequence.
- Contains 7 variables used to store data from various sensors.
- Data sent is in 64-bit floating point number.
- The IRS uses 16-bit signed integers.
- The IRS needs to convert the data from the sensors before execution can proceed.

Technical Mishap

- Cause of explosion:
 - Software failed in the inertial reference system (IRS).
 - Software was taken from the Ariane 4, where it worked successfully.
 - Due to the success rate in the Ariane 4, they wanted to change the IRS as little as possible.
 - An error is caused if the 64-bit number cannot be represented by a 16-bit number.
 - An error handling code was there for up to 4 of the 7 being too large.



Smoke from the explosion June 4,1996 (AP Photo)

Main Problem

- When the IRS software was carried over, it was under the assumption that only four of seven variables required error-handling code.
- This assumption was based on the trajectory data for the Ariane 4.
- The flight behavior for the Ariane 5 was quite different, which in this case lead to information loss.
- The variable BH, which holds the data on the horizontal bias, experienced overflow and was unprotected by the error-handling code. The IRS shut down.

Result Of Failure

- Loss of 500 million invested dollars (US)
- Loss of the payload
- Loss of faith from companies who use the Arianespace company
- Loss of time



Stakeholders

- Arianespace
- ESA, European Space Agency
- Tax payers
- Companies that launched with Arianespace
- Other engineers who worked on the project

The Inquiry Board's Recommendations

- A failure report was produced shortly after the explosion.
- Section 4 contained a number of specific recommendations.
- There were three main category's of recommendations.

The Inquiry Board's Recommendations (1)

- The IRS software testing procedures should involve as much real equipment as technically feasible
- Use as much realistic input data as possible, and get better test coverage.
- If the IRS was not tested in isolation, failure could have been avoided.

The Inquiry Board's Recommendations (2)

- This recommendation relates to the philosophy that was used in dealing with the software.
- The IRS was not treated with mission critical care.
- In the case of error, it just shut down. Should have continued to send "best effort data".

The Inquiry Board's Recommendations (3)

- The third and final class of recommendations deals with all software in general.
- Software should be subject to a software qualification review.
- The industrial architect should take part in the review.

Is This Enough?

- The main recommendation is to do more testing.
- "Testing can show the presence of errors but not their absence" [Dijkstra]
- The Ariane project had software developers when Software Engineers were needed.

The Real Solution

- Have people responsible for software.
- Elevate the process of software development to that of a true engineering discipline.
 - Schools teach students how to develop software using traditional engineering concepts.
 - Have the security that software products are being built with the highest standards.

Parties At Fault?

- No one was singled out
- No groups were blamed
 - The people there did the best job they could.
 - No engineer worked on or approved the software IRS system.
 - The developers followed the standard practices to that date.

Ethics

- The developers were not negligent or unethical in their work.
- The know-how isn't there.
- They did not follow the best practices because they were not professionally educated in software.

Compared To Classical Engineering

- IRS shut down when error occurred.
 In engineering artifacts, redundant safety systems are present.
- No formal mathematical analysis of software components were performed.
- In all other engineering disciplines, mathematical models are created before construction.

Conclusion

- The explosion could have been easily avoided.
- Better practices are needed to prevent amateurish errors.
- Universities and Engineering Societies should work together.
- Professional software engineers are needed to have professional accountability.

References

- Pictures taken from CNN.com and Arianespace.com
- [1] Brooks, Frederick P. "The Mythical Man-Month: Anniversary Edition"
- Published by Addison-Wesley, 1995
- [2] Parnas, David L. "Teaching Programming as Engineering"
- in "Software Fundamentals: Collected Papers by David L. Parnas"
- Published by Addison-Wesley, 2001
- [3] Parnas, David L. "Software Engineering: An Unconsummated Marriage"
- in "Software Fundamentals: Collected Papers by David L. Parnas"

- Published by Addison-Wesley, 2001
- [4] Lions, J.L. "Flight 501 Failure: Report by the Inquiry Board"
- Available on the Internet: java.sun.com/people/jag/Ariane5.html
- [5] Jézéquel, J., and Meyer, B., "The Lessons of Ariane"
- Available on the Internet: archive.eiffel.com/doc/manuals/technology/contract/ariane /page.html
- [6] Baber, R.L. "The Ariane 5 explosion as seen by as software engineer"
- Available on the Internet: cas.mcmaster.ca/~baber/TechnicalReports/Ariane5/Ariane 5.htm
- [7] Garlington, Ken "Critique of 'The Lessons of Ariane"
- Available on the Internet: flash.net/~kennieg/ariane.html#f34

- [8] Arianespace, "Ariane 5"
- Available on the Internet:
- arianespace.com/site/launcher/future_sub_index.html
- [9] Parnas, David L., "Inspection of Safety-Critical Software Using Program-Function Tables"
- in "Software Fundamentals: Collected Papers by David L. Parnas"
- Published by Addison-Wesley, 2001
- [10] Parnas, David L. "The Professional Responsibilities of Sofware Engineers"
- in "Software Fundamentals: Collected Papers by David L. Parnas"
- Published by Addison-Wesley, 2001
- [11] Baber, R.L. "Mathematically Rigorous Software Design"
- Available on the Internet:
- cas.mcmaster.ca/~baber/Courses/46L03/MRSDLect.pdf