The ARIANE 5 Software Failure

Mark Dowson
Marlstone Software Technology
e-mail:dowson@marlstone.com

On 4 June 1996, the Ariane 501 satellite launch failed catastrophically 40 seconds after initiation of the flight sequence, incurring a direct cost of approximately $370 million. The Inquiry Board Report (IBR), clearly identifies the proximate cause of the disaster as a software failure; but in other respects is one of the more astonishing engineering documents of our time. In summary, the sequence of events was as follows:

1. An outside-expected-range value, derived from the launcher's actual horizontal velocity, led to an unhandled exception in the Inertial Reference System (SRI) software. This caused the backup "hot standby" SRI to fail approximately 37 seconds after launch initiation (H0).
2. Seventy two milliseconds later, the active SRI (running identical software) failed for the same reason.
3. In the absence of a working backup, the failure of the active SRI caused it to transmit diagnostic bit patterns to the main On Board Computer (OBC).
4. The OBC, misinterpreting the diagnostic input as valid data, ordered full nozzle deflections of the solid boosters and the Vulcain main engine.
5. These deflections created aerodynamic loads that separated the boosters from the main stage, (correctly) triggering the launcher's self-destruct system at approximately H0+40 seconds.

The Inquiry Report concludes that, in essence, poor software engineering practices were responsible for the "software failure" that caused the catastrophe, and makes a number of general and specific recommendations including:

- "Although the failure was due to a systematic software design error, mechanisms can be introduced to mitigate this type of problem. ..." (IBR page 5).
- "Do not allow any sensor ... to stop sending best effort data." (IBR Recommendation 3, page 13).
- "... perform complete, closed-loop, system testing. Complete simulations must take place before any mission. ..." (IBR Recommendation 4, page 13).

So – a software problem, with software engineering solutions. Closer analysis of the Inquiry Report reveals a rather different picture:

1. "... the early part of the flight trajectory of Ariane 5 differs from that of Ariane 4 and results in considerably higher horizontal velocity values." (IBR page 4).
2. "The design of the Ariane 5 SRI is practically the same as that of an SRI which is presently used on Ariane 4, particularly as regards software." (IBR page 3).
3. "There is no evidence that any trajectory data were used to analyze the behavior of the unprotected variables, and it is even more important to note that it was jointly agreed not to include the Ariane 5 trajectory data in the SRI requirements and specification." (IBR page 5, emphasis added).

Excuse me? This was a software failure?

As should now be obvious, the software functioned precisely according to its specification and design, and no additional software-related measures, such as ensuring that all code explicitly included a handler for "others" conditions, improved QA , e.g., better test coverage, or use of improved approaches to software fault tolerance, could have ensured that some similar catastrophe did not occur.

These, then, are the two astonishing aspects to the Inquiry Report:

1. The report reveals that expected flight trajectory of Ariane 5 was deliberately excluded from consideration in the design of a key component involved in its control.
2. The Inquiry Board (which included software engineers), characterized the cause of the consequent flight failure as a software problem.

Unfortunately, it is all to easy to imagine how the first of these happened in an industrial context: pressures of time and budget, if-its-not-broke-don't-fix-it arguments, etc. The second is harder to comprehend, and perhaps it is wiser not to speculate on its causes without better insight into the procedures followed by the Inquiry Board.

So, how should we characterize the causes of failure of Ariane flight 501, and how can future such failures be avoided?

At the very least, the failure was a systems engineering failure (I'm grateful to Gerard Le Lann, of INRIA, France for drawing this perspective to my attention). As we all know, software can't be designed in isolation, and the system that it is part of includes its operating environment.

Furthermore, the failure was a process failure. A well defined (and well designed) process for engineering systems such as Ariane should make it impossible to exclude any significant aspect of the system operating environment from consideration.

Ariane 5 should teach us that there are "political" facets of engineering processes. A good process needs to regulate not only how systems are designed and developed, but also how high-level decisions about that design and development are arrived at. Only by adopting and following a process that encompasses these considerations, will sound engineering practice be able to triumph over immediate pressures of time and budget.

Reference:

ARIANE 5 Flight 501 Failure, Report by the Inquiry Board, Paris 19 July 1996