



Project Failures

Consuelo Cogrossi
Raffaello Martini



LAS

London Ambulance Service

- ⌘ Managed by South West Thames Regional Health Authority.
- ⌘ Largest ambulance service in the world
 - Covers geographical area of over 600 square miles;
 - Resident population of 6.8 million people;
 - Carries over 5,000 patients every day;
 - 2,000-2,500 calls received daily, of which 1,300- 1,600 are emergency calls.



The LAS fiasco

- ⌘ The London Ambulance Service (LAS) Computer Aided Dispatch (CAD) system failed dramatically on October 26th 1992, the same day it was introduced:
 - The system could not cope with the load placed on it by normal use;
 - The response to emergency calls was several hours;
 - Ambulance communications failed and ambulances were lost from the system.
- ⌘ A series of errors were made in the procurement, design, implementation, and introduction of the system.



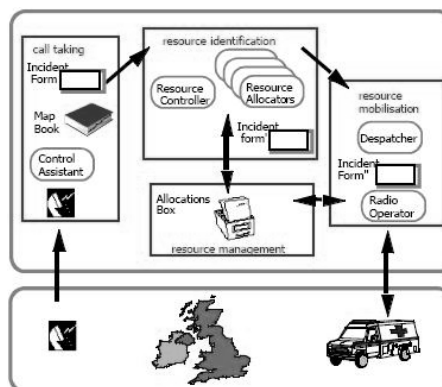
CAD system

- ≠ Provide one or more of the following:
 - Call taking;
 - Resource identification;
 - Resource mobilisation;
 - Ambulance resource management.
- ≠ Consist of:
 - CAD software & hardware;
 - Gazetteer and mapping software;
 - Communications interface (RIFS).
 - Radio system;
 - Mobile data terminals (MDTs);
 - Automatic vehicle location system (AVLS).



The manual system

- ≠ CAC Central Ambulance Control
- ≠ Call taking
 - Recorded on form; location identified in map book; forms sent to central collection point on conveyor belt;
- ≠ Resource identification
 - Form collected; passed onto resource allocator decides on which resource to be mobilised;
- ≠ Resource mobilisation
 - Dispatcher telephones relevant ambulance station, or passes instructions to radio operator if ambulance already on road;



Concept/design of the CAD system

- ⌘ Existing systems dismissed as inadequate and impossible to modify to meet LAS's needs
- ⌘ Desired system:
 - To consist of Computer Aided Dispatch; Computer map display; Automatic Vehicle Location System (AVLS);
 - Must integrate with existing MDTs and RIFS (Radio Interface System).
- ⌘ Success dependent upon:
 - Accuracy and reliability of technology;
 - Absolute cooperation from all parties including CAC staff and ambulance crews.



Problems: Procurement (i)

- ⌘ Contract had to be put out to open tender:
 - Regulations emphasis is on best price;
 - 35 companies expressed interest in providing all or part of the system;
 - Most raised concerns over the proposed timetable of less than 1 year until full implementation.
- ⌘ Previous Arthur Andersen report largely ignored:
 - Recommended budget of £1.5M and 19 month timetable for packaged solution. Both estimates to be significantly increased if packaged solution not available;



Problems: Procurement (ii)

- ✗ Successful consortium:
 - ✗ Apricot, Systems Options (SO), Datatrak; bid at £937k was £700k cheaper than the nearest bid;
 - ✗ SO's quote for the CAD development was only £35k
 - ✗ Their previous development experience was only for administrative systems.
 - ✗ Ambiguity over lead contractor.
- ✗ 2 key members of evaluation team:
 - ✗ Systems manager: Career ambulance man, not an IT professional;
 - ✗ Analyst: Contractor with 5 years experience working with LAS.



Problems: Project management

- ✗ Lead contractor responsible
 - ✗ Meant to be SO, but they quickly became snowed under, so LAS became more responsible by default;
 - ✗ No relevant experience at LAS or SO.
- ✗ SO regularly late in delivering software
 - ✗ Often also of suspect quality, with software changes put through 'on the fly'.



Problems: Human resources (i)

- ⌘ Generally positive attitude to the introduction of new technology.
- ⌘ Ambiguity over consultation of ambulance crews for development of original requirements.
- ⌘ Circumstantial evidence of resistance by crews to Datatrak equipment, and deliberate misleading of the system.
- ⌘ Large gap between when crews and CAC staff were trained and implementation of the system.
- ⌘ Inability of the CAC and ambulance staff to appreciate each others' role



Problems: Human resources (ii)

- ⌘ Management 'fear of failure'.
- ⌘ CAD system seen as solution to management's desire to reduce 'outdated' working practices.
- ⌘ System allocated nearest resource, regardless of originating station.
- ⌘ System removed flexibility in resource allocation.



System problems (i)

- ⌘ Need for near perfect information
 - ⌘ Without accurate knowledge of vehicle locations and status, the system could not allocate optimum resources.
- ⌘ Poor interface between crews, MDTs & the system
 - ⌘ There were numerous possible reasons for incorrect information being passed back to the system.



System problems (ii)

- ⌘ Unreliability, slowness and operator interface
- ⌘ Numerous technical problems with the system, including:
 - ⌘ Failure to identify all duplicated calls;
 - ⌘ Lack of prioritisation of exception messages;
 - ⌘ Exception messages and awaiting attention queues scroll off top of screen;
 - ⌘ Visual Basic on Window 3.0.



Configuration changes

- ⌘ Implementation of the system on October 26th involved a number of significant changes to CAC operation, in particular:
 - ⌘ No paper backup system;
 - ⌘ Going 'pan London' rather than operating in 3 divisions;
 - ⌘ Using only the system proposed resource allocations;
 - ⌘ Allowing some call takers to allocate resources;



So, what happened?

- ⌘ Changes to CAC operation made it extremely difficult for staff to intervene and correct the system.
- ⌘ As a consequence, the system rapidly knew the correct location and status of fewer and fewer vehicles, leading to:
 - ⌘ Poor, duplicated and delayed allocations;
 - ⌘ A build up of exception messages and the awaiting attention list;
 - ⌘ A slow up of the system as the messages and lists built up;
 - ⌘ An increased number of call backs and hence delays in telephone answering.



Why did it fail?

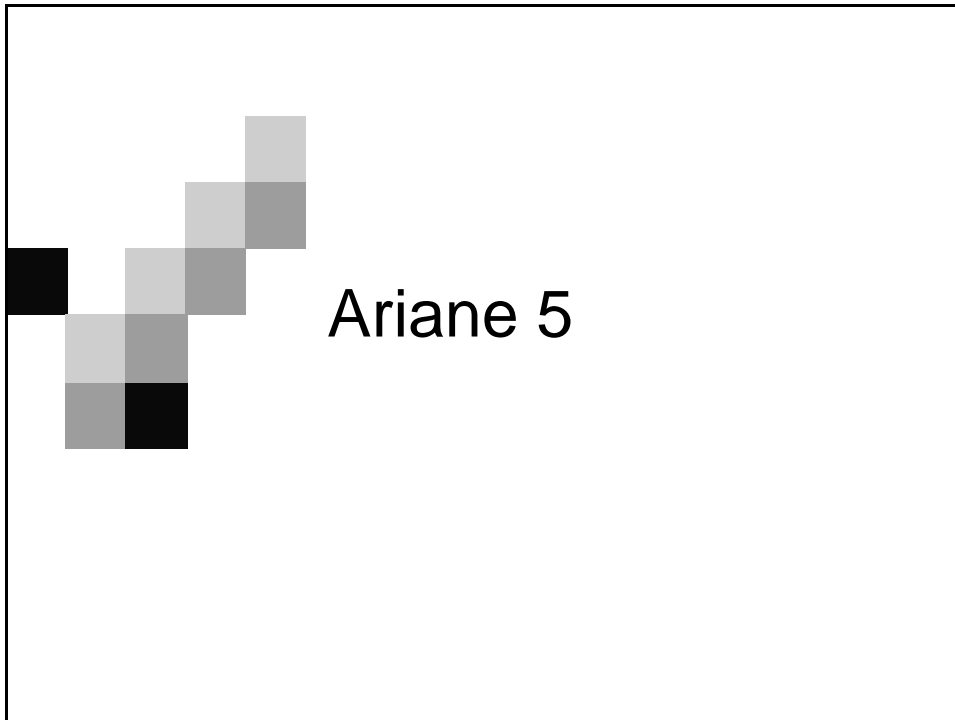
- ⌘ Technically, the system did not fail on October 26th
 - ⌘ Response times did become unacceptable, but overall the system did what it had been designed to do!
 - ⌘ Failed 3 weeks later due to a program error - this was a memory leak where allocated memory was not completely released.



The way forward for the CAD

- ⌘ Inquiry report makes detailed recommendations for future development of the LAS CAD system, including:
 - ⌘ Focus on repairing reputation of CAD within the service;
 - ⌘ They still believe that a technological solution is required;
 - ⌘ Development process must allow fully for consultation, quality assurance, testing, training;
 - ⌘ Management and staff must have total, demonstrable, confidence in the reliability of the system;
 - ⌘ Any new system should be introduced in a stepwise approach.





Ariane 5

What is Ariane?

- ✍ The name Ariane refers to a series of civilian European expendable launch vehicles for space launch use;
- ✍ France first proposed the Ariane project in the 1970s;
- ✍ The project was Europe's second attempt to develop its own launcher following the unsuccessful Europa project;
- ✍ Several versions of the launcher.



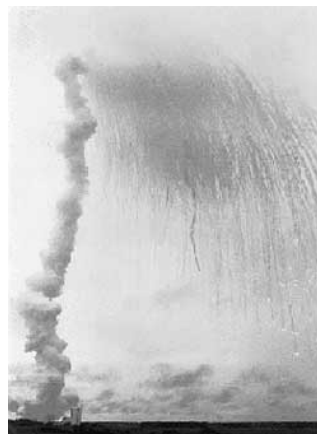
Design by contract

- ⌘ Essential for reliability;
- ⌘ It is the principle that interfaces between modules of a software system should be governed by precise specifications;
- ⌘ Similar to a contract between humans or company;
- ⌘ The contracts will cover mutual obligations, benefits and consistency constraints.



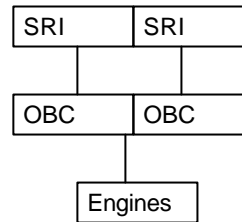
What happened

- ⌘ On June 4th 1996, the maiden flight of the Ariane 5 launcher ended in a failure;
- ⌘ Only about 40 seconds after initiation of the flight sequence the launcher veered off its flight path, broke up and exploded;
- ⌘ The system failure was a direct result of a software failure.



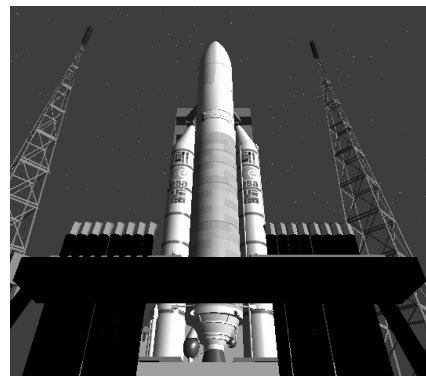
The subsystem

- ✍ SRI: computer-based inertial reference system, computes attitude and trajectory of the rocket and sends them to OBC. Redundant.
- ✍ OBC (on board computer): executes flight program, controls engines. Redundant.



The problem

- ✍ Software failure on SRI. Occurred when, in function F, an attempt to convert a 64-bit floating point number to a signed 16-bit integer caused the number to overflow.
- ✍ There was no exception handler associated with the conversion so the system exception management facilities were invoked. These shut down the SRI.
- ✍ The backup SRI had the same software, and behaved in exactly the same way.
- ✍ The OBC received diagnostic commands from shutting down SRI, and interpreted them as normal data, commanding engines to extreme position, resulting in unforeseen stresses on the rocket, that caused separation of the boosters from the main stage, in turn triggering the self-destruct system of the launcher.



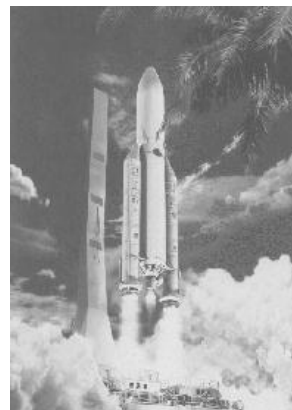
Why?

- ⌘ Why the overflow?
- ⌘ Why no exception handling?
- ⌘ SRI was reused from Ariane 4. The physical characteristics of Ariane 4 (A smaller vehicle) are such that it has a lower initial acceleration and build up of horizontal velocity than Ariane 5. The value of the variable on Ariane 4 could never reach a level that caused overflow in function F during the launch period.
- ⌘ Besides, function F was NOT needed in Ariane 5 (was in Ariane 4). Decisions were made:
 - Not to remove F as this could introduce new faults;
 - Not to catch overflow exceptions because the processor was heavily loaded. For dependability reasons, it was thought desirable to have some spare processor capacity.



Validation failure

- ⌘ As the function that failed was not required for Ariane 5, there was no requirement associated with it.
- ⌘ As there was no associated requirement, there were no tests of that part of the software and hence no possibility of discovering the problem.
- ⌘ During system testing, simulators of the inertial reference system computers were used. These did not generate the error as there was no requirement!



The key problem

- ✍ Assumptions and requirements to reuse a subsystem were not properly stated.

Lessons learned

- ✍ Don't run software in critical systems unless it is actually needed.
- ✍ As well as testing for what the system should do, you may also have to test for what the system should not do.
- ✍ Do not have a default exception handling response which is system shut-down in systems that have no fail-safe state.



Lessons learned

- ⌘ In critical computations, always return best effort values even if the absolutely correct values cannot be computed.
- ⌘ Wherever possible, use real equipment and not simulations.
- ⌘ Improve the review process to include external participants and review all assumptions made in the code.



Avoidable failure

- ⌘ The designer's of Ariane 5 made a critical and elementary error.
- ⌘ They designed a system where a single component failure could cause the entire system to fail.
- ⌘ As a general rule, critical systems should always be designed to avoid a single point of failure.

