A Comedy of Errors: the London Ambulance Service case study

Anthony Finkelstein & John Dowell
School of Informatics, City University, UK
{acwf@soi.city.ac.uk, johnd@soi.city.ac.uk}

Abstract

This paper provides an introduction to the IWSSD-8 case study - the "Report of the Inquiry Into the London Ambulance Service". The paper gives an overview of the case study and provides a brief summary. It considers how the case study can be used to orient discussion at the workshop and provide a bridge between the various contributions.

Introduction

The International Workshop on Software Specification & Design has established a tradition of using "case studies" to focus and provide coherence to its intensive working sessions. These case studies, supplied in advance to participants in the various tracks, have proved a fruitful way of working. Evidence of this can be seen most clearly in the "succeedings" or workshop reports which have followed previous workshops. It was decided for IWSSD-8 that, in order to provide common ground between the tracks, a single shared "case study" should be used, with each track drawing on it in a manner appropriate to their own interests and concerns. After some discussion we settled on the "Report of the Inquiry Into the London Ambulance Service" which is interesting in its own right, reflects aspects of requirements architecture, design, concurrency and distribution, and raises significant issues on the relation between these aspects.

The report is available by ftp, details of how to obtain it can be found at the bottom of the paper. Subsequent comments in this paper assume that you have access to the report. References in this paper are to report paragraph numbers.

Overview & Summary

Like most computing professionals in the UK we were aware of the failure, using this term broadly, of the computer aided despatch (CAD) system deployed by the London Ambulance Service (LAS) in, or shortly after, October 1992. We suspect, as London residents, we were more immediately aware of it than most. At any rate, both of us read the items that appeared in the newspapers with considerable interest and concern.

Neither of us can remember when we first saw a copy of the report, probably in summer 1993, but both remember clearly our initial reactions - a mixture of horror and, we must confess, a certain macabre enjoyment. If not a comedy of errors it is at least a compounding of them. It seemed, on first reading, as if everything had gone wrong - every component of good engineering practice had been ignored, every guideline of software engineering disregarded, basic management principles neglected, even the dictates of common sense overlooked. Subsequent readings have rather changed our understanding of the failure which now emerges as an example of "systemic failure" or "normal accident" of the type identified by Perrow (1984). This having been said it is evident that at the heart of the failure are breakdowns in specification and design common to many software development projects and that the context in which they occurred is far from atypical. Therein lies its particular interest and challenge from our standpoint.

The failure, and the subsequent reaction to it must be understood in a broad political setting. The National Health Service (NHS), the government supported "free-at-the-point-of-use" system of health care provision in the UK, was undergoing considerable changes - in particular the move towards more decentralised and directly financially accountable management. These changes were combined with a lack of prior investment, significant ongoing resource pressures and a reallocation of NHS priorities drawing money away from London. A further feature of the political environment was a strong focus on the "effectiveness" or "performance" of public services. The mix of these changes with a combative political scene and fraught labour relations gave a particular significance and weight to the failure and lead to the establishment of the inquiry which reported in February 1993.

For orientation a short sketch of the report follows. There have been a number of other analyses of the LAS CAD system failure of which Mellor (1994) is probably the most useful.

The LAS despatch system is responsible for: receiving calls; despatching ambulances based on an understanding of the nature of the calls and the availability of resources; and, monitoring progress of the response to the call. A computer-aided despatching system was to be developed and would include an automatic vehicle locating system (AVLS) and mobile data terminals (MDTs) to support automatic communication with ambulances. This system was to supplant the existing manual system.

Immediately following the system being made operational the call traffic load increased (but not it should be noted to exceptional levels). The AVLS could not keep track of the location and status of units. This lead to an incorrect database so that (a) units were being despatched non-optimally (b) multiple units were being assigned to some calls. As a consequence of this there were a large number of exception messages and the system slowed down as the queue of messages grew. Un-responded exception messages generated repeated messages and the lists scrolled off the top of the screens so that awaiting attention and exception messages were lost from view. Ambulance crews were frustrated and, under pressure, were slow in notifying the status of their unit. They could not (or would not) use their MDTs and used incorrect sequences to enter the status information. The public were repeating their calls because of the delay in response. The AVLS no longer knew which units were available and the resource proposal software was taking a long time to perform its searches

The entire system descended into chaos (one ambulance arrived to find the patient dead and taken away by undertakers, another ambulance answered a 'stroke' call after 11 hours - 5 hours after the patient had made their own way to hospital). The CAD system was partly removed and aspects of its function (notably despatch decisions) were performed manually. This part-manual system seized up completely 8 days later. The back-up server did not work since it had not been fully tested. Operators used tape recordings of calls then reverted to a totally manual system. The Chief Executive of the LAS resigned.

A summary of this form cannot do justice to the range of problems identified by the inquiry. Key points which emerged were: the software was incomplete and effectively untested; the implementation approach was 'high risk'; inappropriate and unjustified assumptions were made during the specification process; there was a lack of consultation with users and clients in the development process with knock-on consequences for their "ownership" of the resulting system; the poor fit of the system with the organisational structure of the ambulance service. Subsidiary to these points but nevertheless important were the poorly designed user interfaces; lack of robustness; poor performance and straightforward bugs or errors. Though outside the scope of IWSSD there is a very strong message in the report about the attempt to change working practices through the specification, design and implementation of a computer system.

The report is exceptionally easy to read. It is divided into 6 parts: summary conclusions and recommendations on the part of the inquiry team; the background to the inquiry itself, including an orientation to the LAS and CAD; an account of the development of the CAD system; a discussion of the major system problems and breakdowns (failure in the narrow sense); a strategy for the future of CAD within the LAS; an analysis of the management and operation of the LAS. Another way to view the report is as having two facets - record and recommendation - and two targets - system and organisational context. The recommendations are less important for our purposes than the record though they are, for the most part, sensible and interesting. The discussion of the system is obviously our principal concern but the context is vital if it is to be properly understood. The report is best read in its entirety even if only pieces are to be used.

Inevitably the serious reader will experience some frustration with the report and will want access to parts of the underlying data and related source documents which are not readily available. These lacunae are the price that is paid for dealing with "real" cases - the flip side of the contextual richness of the material.

Using the Report

The report is not typical of specification and design case studies or "exemplars". It is not itself a specification or problem statement (like the lift, central heating system, package router or library system), though it contains significant fragments of such documents. Nor is it a complete account of the system development process, though again, it contains significant fragments of such an account. The particular role of the report, as a postmortem study, does however open some possibilities for analysis which "classical" exemplars do not.

The most obvious use of the report is simply to extract relevant specification-like fragments and use them, in isolation, to demonstrate specification and design techniques. An instance of this might be to model the manual despatch process, a typical office information system with the added complications of safety criticality

and real-time constraints, see 3001 et seq. This has the clear merit of demonstrating the techniques in a real system. A variant of this is to rework some of the models presented in the report such as the communications structure, a sort of system architecture crudely presented in diag 3.1 and associated text.

A more challenging approach is to identify specific problems highlighted by the report and demonstrate, convincingly, that these problems would be avoided by particular specification and design techniques. An example, chosen almost at random, is the false assumption of "near perfect information of vehicle location and crew/vehicle status" on which the developers relied and which is documented in 4008. A related, though significantly more difficult, task is to demonstrate these techniques would work in the context described in the report. In other words that the specification and design techniques are robust with respect to the process, management and organisation which frame them. That is that they possess what psychologists term "ecological validity".

Less work is required to identify problems which lie outside the current state of the art in specification and design. The interplay between procurement and specification processes is a good example, see 3029 et seq. This can be combined with the use of the report to rebalance concerns within the field as a whole. There is, on the face of it, clear blue water between the primary concerns of the report, which line up neatly with those commonly expressed by industrial managers, and those which constitute the main targets of specification and design research. This suggests, we put it no stronger than that, the need for a reappraisal of research priorities.

Somewhat obliquely the report raises questions about how inquiries into system failures ought to be conducted what information should be recorded and how, in general, we can learn from our experience.

Our preference is to treat the report as a whole and to look at recurring themes. An illustration of this is how performance concerns bind together requirements, architecture, usability and testing. Another interesting example is how system integration and the reliability of behaviour and service provision by "bought-in" components continually emerges as a problem. We leave the identification of further themes as an exercise for the reader. This gestalt approach links well to the concept of systemic failure to which the LAS CAD so closely conforms.

Conclusion

Software engineers, and more specifically those concerned with specification and design, have become enamoured of what might be termed a "lachrymose theory" of software engineering - a fixation on errors and bugs. Software engineering can often be said to define itself by reference to problems and failures. The use of the LAS as a case study is not intended to reinforce this. However, "breakdowns" are important as it is only through an understanding of failed systems that we can formulate a view of what a successful system would be and, perhaps, the role of specification and design in this context.

How to Obtain the Report

We would like to thank the Communications Directorate of South West Thames Regional Health Authority for permission to scan and distribute this document electronically. The original printed version is available as ISBN 0-905133-70-6. The electronic version is available as:

Flavour 1: includes scanned images, 529K compressed
ftp://ftp.cs.city.ac.uk/pub/requirements/lascase0.9.ps.gz
ftp://ftp.cs.colorado.edu/users/iwssd8/lascase0.9.ps.gz

Flavour 2: without scanned images, 83K compressed
ftp://ftp.cs.city.ac.uk/pub/requirements/lasnodiags0.9.ps.gz
ftp://ftp.cs.colorado.edu/users/iwssd8/lasnodiags0.9.ps.gz

References

Mellor, P. (1994); CAD: computer-aided disaster; High Integrity Systems; 1, 2, pp101-156.

Perrow, C. (1984); Normal Accidents: living with high-risk technology; Basic Books, New York.