

## **The London Ambulance System Disaster, 1992**

In October 1992, the London Ambulance Service suffered a disaster that brought their operations to a virtual standstill over 36 hours, and up to 20-30 people may have died as a result of ambulances arriving too late on the scene. Upon further investigation, it was discovered that the new computer aided dispatch (CAD) software was responsible for the crisis. The problem lay in the design of the software, which was completely inadequate for the needs of the London Ambulance Service. So terribly pathetic were this software's services that response times to emergency calls were as high as 11 hours during the 36 hours crisis.

### **LAS Overview**

The major objective of the London Ambulance Service Computer Aided Dispatch (LASCAD ) project was to automate many of the human-intensive processes of manual dispatch systems associated with ambulance services in the UK. The major rationale expressed for such computerization was typically that a number of problems existed with the manual CAD systems. Most such problems related to the time-consuming and error-prone nature of activities such as: identification of the precise location of an incident, the physical movement of paper forms, and maintaining up-to-date vehicle status information.

The basic functionality of the intended LASCAD system was as follows: British Telecom (BT) operators would route all 999 calls concerning medical emergencies as a matter of routine to LAS headquarters (HQ) in Waterloo. 18 HQ 'receivers' were then expected to record on the system the name, telephone number and address of the caller, and the name, destination address and brief details of the patient. This information would then be transmitted over a local area network to a locator. The system would pinpoint the patient's location on a map display of areas within London. The system was expected to

monitor continuously the location of every ambulance via radio messages transmitted by each vehicle every 13 seconds. The system would then determine the nearest ambulance to the patient.

Experienced ambulance dispatchers were organized into teams based on three zones (south, north-west and north-east). Ambulance dispatchers would be offered details of the three nearest ambulances by the system and the estimated time each would need to reach the scene. The dispatcher would choose an ambulance and send patient details to a small terminal screen located on the dashboard of the ambulance. The crew would then be expected to confirm that it was on its way. If the selected ambulance were in an ambulance depot then the dispatch message would be received on the station printer. The ambulance crew would always be expected to acknowledge a message. The system would automatically alert HQ of any ambulance where no acknowledgement was made a follow up message would then be sent from HQ. The system would detect from each vehicle's location messages whether an ambulance was heading in the wrong direction. The system would then alert controllers. Further messages would tell HQ when the ambulance crew had arrived, when it was on its way to a hospital and when it was free again.

The LASCAD system was built as an event-based system using a rule-based approach in interaction with a geographical information system (GIS). The system was built by a small Aldershot based software house called Systems Options using their own GIS software (WINGS) running under Microsoft Windows. The GIS communicated with Datatrak's automatic vehicle tracking system. The system ran on a series of network PCs and file servers supplied by Apricot.

## **The Crash of the System**

The system was lightly loaded at start-up on 26 October 1992. Any problems, caused particularly by the communications systems (such as ambulance crews pressing the wrong buttons, or ambulances being radioed in black spots), could be effectively managed by staff. However, as the number of ambulance incidents increased, the amount of incorrect vehicle information recorded by the system increased. This had a knock-on

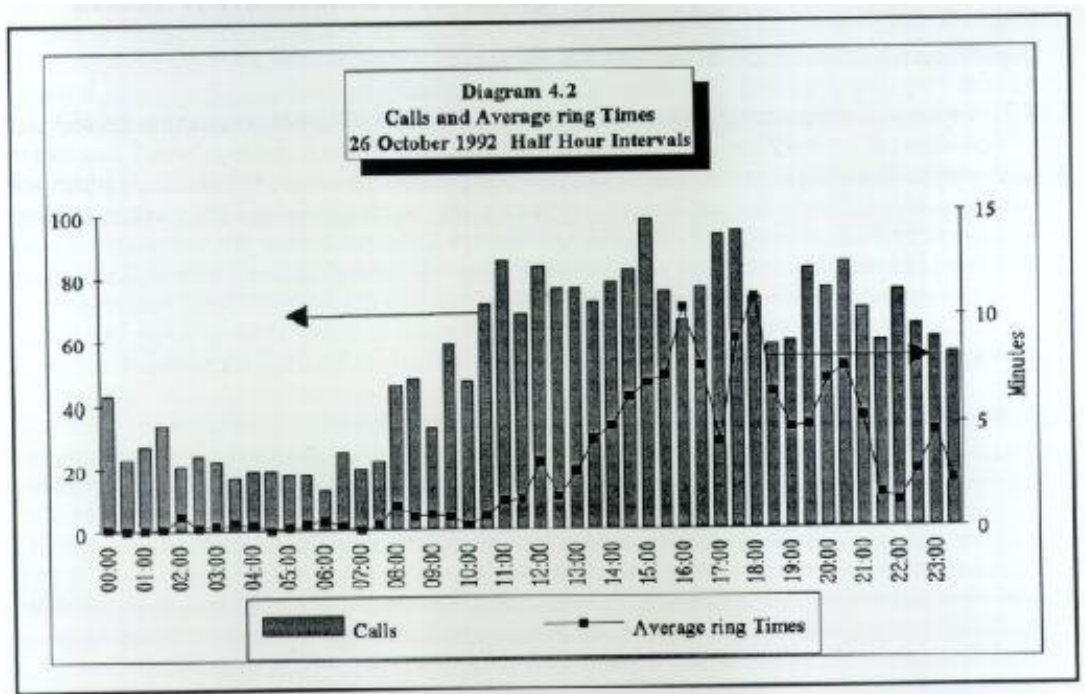
effect in that the system made incorrect allocations on the basis of the information it had. For example, multiple vehicles were sent to the same incident, or the closest vehicle was not chosen for dispatch. As a consequence, the system had fewer ambulance resources to allocate. The system also placed calls that had not gone through the appropriate protocol on a waiting list and generated exception messages for those incidents for which it had received incorrect status information. Indeed, the number of exception messages appears to have increased to such an extent the staff were not able to clear the queue. It became increasingly difficult for staff to attend to messages that had scrolled off the screen. The increasing size of the queue slowed the system.

All this meant that, with fewer resources to allocate, and the problems of dealing with the waiting and exceptional queues, it took longer to allocate resources to incidents. There came two worst problems in the system: 1) Inability of the software to distinguish between duplicate calls from different people pertaining to the same incident. 2) Failure of the software to maintain and keep track of logged calls. One particularly evident case was a woman who called the emergency number after being trapped by the body of her collapsed husband. She repeatedly called the ambulance service every half-an-hour, only to be told that there was no record of her earlier call. When an ambulance eventually arrived nearly 3 hours later, her husband had already died.

At the receiving end, patients became frustrated with the delays to ambulances arriving at incidents. This led to an increase in the number of calls made back to the LAS HQ relating to already recorded incidents. The increased volume of calls, together with a slow system and an insufficient number of call-takers, contributed to significant delays in answering calls which, in turn, caused further delays to patients. At the ambulance end, crews became increasingly frustrated at incorrect allocations. This may have led to an increased number of instances where crews failed to press the right status buttons, or took a different vehicle to an incident than that suggested by the system. Crew frustration also seems to have contributed to a greater volume of voice radio traffic. This in turn contributed to the rising radio communications bottleneck, which caused a general slowing down in radio communications, which, in turn, fed back into increasing crew frustration. The system therefore appears to have been in a vicious circle of cause and

effect. In addition, it was claimed that the organization of sector desks over the preceding weekend may have caused loss of local knowledge.

The following diagram shows us that as the number of ambulance incidents increased, the amount of incorrect vehicle information recorded by the system increased.



## Factors Contributed to Such a Disaster

- The resulting inquiry reported that neither the CAD system itself, nor its users, was ready for full implementation on the 26 October.
- The CAD software was not complete, not properly tuned, and not fully tested.
- The resilience of the hardware under a full load had not been tested.
- There existed outstanding problems associated with the data communications of the system, specifically communication to and from the mobile data terminals.

- There existed some skepticism over the accuracy of the Automatic Vehicle Location System (AVLS).
- Staff, both within the Central Ambulance Control (CAC) and ambulance crews, had no confidence in the system and were not all fully trained.
- The physical changes to the layout of the control room on 26 Oct meant that all CAC staff were working in unfamiliar positions, without paper backup, and were less able to work with colleagues with whom they had jointly solved problems before.
- There had been no attempt to foresee fully the effect of inaccurate or incomplete data available to the system (i.e. late status reporting or vehicle locations).
- These imperfections led to an increase in the number of exception messages that would have to be dealt with and which in turn would lead to more callbacks and enquiries

Systems Options, the company supplying the major part of the software for the system, is reported as having had no previous experience of building dispatch systems for ambulance services. The company had won the 1 million contract for the system in June 1991.

Under the RHA Standing Financial Instructions, which provide the regulatory framework within which such procurements may take place, the basic rule is that contracts such as this have to be put out to open tender. This requirements was complied with alongside accepting the lowest tender unless there are "good and sufficient reasons to the contrary" Over the following weeks several meetings were held with prospective suppliers covering queries on the full specification and resolving other potential technical and contractual issues. These meetings were minuted by the project team and it was clear that most of the suppliers raised concerns over the proposed timetable - which was for full implementation by 8 January 1992. They were all told that this timetable was non-negotiable.

Out of all the proposals there was only one, which met the total LAS requirement, including timetable and price. On the basis of the proposals as submitted, the optimum

solution appeared to be the proposal by the consortium consisting of Apricot, Systems Options and Datatrak.

Amongst the papers relating to the selection process there is no evidence of key questions being asked by the selection team about why the Apricot bid, particularly the software cost (Systems Options), was substantially lower than other bidders. Neither is there evidence of serious investigation, other than the usual references, of Systems Options (or any other of the potential suppliers') software development experience and abilities.

## **Possible Solutions**

*First, choose the team members properly*

Purchased system projects are successful when the organization has selected a product and a vendor that is able to satisfy the firm's current and future system needs. This requires an effective project team with representative members committed to the project goals at the outset. The right business managers, end users, and IS specialists need to be a part of the project team to ensure that the best package is purchased from the best vendor and that both technical and business risks have been adequately considered.

But in this case, the selection team apparently did a poor job. The team was consisted of wrong members.

In this case, the prime responsibility for the technical evaluation of the tenders fell upon the contract analyst and the Systems Manager. The representative from Regional Supplies was unable to evaluate the tenders on technical merits, as her experience was in procurement in its most general sense rather than specific to Information Technology. A contractor and an arguably unsuitably qualified systems manager (who knew that he was to be replaced and made redundant) were put in charge of the procurement of an extremely complex and high risk computer system with no additional technical expertise available to them.

In addition, some special roles are needed for packaged application purchase. Successfully implementing a packaged application typically requires a major

commitment on the part of business managers because of the extensive changes in an organization's processes and procedures that are needed to effectively implement the purchased solution. As a result, it is not uncommon for the project manager for a packaged application system project to be a business manager. Because IS expertise is still required in order to assess the technical feasibility of implementing a package, IS managers may share the project management role or be relied on as IS specialists for the project management role or be relied on as IS specialists for the project team. Organizations that have no IS specialists will need to rely on their own business managers as well as outside consultants, to provide the necessary IS expertise.

Because of this initial and ongoing dependence on the software vendor, other important roles can also be critical to the success of a packaged system implementation: purchasing specialists and contract specialists. Both types of expertise should be involved in the purchasing project, whether or not they are formal members of the project team.

#### *Second, improve the selection process*

The organization's requirements are used to eliminate all but a few of the most promising candidate packages that were identified in the feasibility analysis. Further research on the vendor's capabilities can be undertaken to eliminate vendors due to problems experienced with other users of the package, an inadequate track record or firm size of the vendor, or other concerns about long-term viability.

In this case, Systems Options, the company supplying the major part of the software for the system, is reported as having had no previous experience of building dispatch systems for ambulance services. Systems Options was completely out of its depth in writing such software. It had even no prior experience in writing large and complex systems, much less mission critical CAD systems on which people's lives would depend. However, allegations arose that LAS settled for the lowest-bid from competing software vendors, rather than ensuring a contractor with the necessary experience and depth to provide a reliable end product. Remember that lowest-bid does not necessarily mean the highest quality; rather, it may provide a product with bad quality.

### *Third, refine requirement specifications*

In order to prepare the requirements specification for the proposed new system, a team was assembled under the chairmanship of the Director of Support Services with the then Systems Manager, a contract analyst, and the Control Room Services Manager. Other individuals were also involved representing training, communications and other areas. Because of the problems at the time with the staff consultation process there was little involvement at this stage from the ambulance crews, although invitations to participate were given to union representatives. Apparently, the team had lacked the participation by end users or IS specialists which is essential in the development of any system.

### *Fourth, assess project risks in advance.*

It's important to identify all risks that are obvious to both managers and practitioners. Project risks threaten the project plan. That is, if project risks become real, it is likely that project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel, resource, customer, and requirements problems and their impact on a software project.

One method for identifying risks is to create a risk item checklist. The checklist can be used for risk identification and focuses on some subset of known and predictable risks in the following generic subcategories:

- Product size--risks associated with the overall size of the software to be built or modified.
- Business impact—risks associated with constraints imposed by management or the marketplace.
- Customer characteristics—risks associated with the sophistication of the customer and the developer's ability to communicate with the customer in a timely manner.
- Process definition—risks associated with the degree to which the software process has been defined and is followed by the development organization.



- Development environment—risks associated with the availability and quality of the tools to be used to build the product.
- Technology to be built—risks associated with the complexity of the system to be built and the newness of the technology that is packaged by the system.
- Staff size and experience—risks associated with the overall technical and project experience of the software engineers who will do the work.

These provide useful insight into generic risks for software projects and should be used whenever risk analysis and management is instituted.

The LAS case is absolutely a high-risk project, which definitely needs proper analysis of all potential risks. Unfortunately, the project team had failed to do this before they got started. In addition, purchased system projects introduce several new types of risks. First, the success of the project is highly dependent on the performance of a third party. The quality of the implemented system will depend not only on the software engineering capabilities of the vendor, but also how well the package's capabilities are understood by the implementing organization and the vendor's training and installation capabilities. As discussed earlier, a key aspect of the vendor selection process is accurate assessment of the vendor's capabilities, not just an evaluation of the current software package.

The initial success of the project, as well as the long-term effectiveness of the system being installed, is also highly dependent on the contract negotiation process. Vendor expertise may be required to install the package, build interfaces to existing systems, and perhaps modify the package itself to better match the needs of the purchasing organization.

*Fifth, make a proper schedule and a backup copy*

The CAD system implemented in 1992 was over ambitious and was developed and implemented against an impossible timetable.

LAS applied pressure to the contractor to advance the timeline more quickly than was prudent, resulting in a lower quality product. Perhaps due to the compressed timeline, the LAS implemented the system, allowing it to “go live” on October 26, 1992 with its staff

having little or no training in the new system. This one-phase deployment lacked testing and critical oversight, and was a contributing factor in subsequent disaster.

User training needs to be planned and carefully scheduled so that people are prepared to use the system when it is installed, but not trained so far in advance of the installation that what they learned is forgotten if user resistance to proposed changes is anticipated, this potential situation needs to be addressed during training or earlier.

Even worse, in the face of such a rushed system, LAS had no backup plan, no fail over mechanism to handle problems in the CAD system. The ultimate protection against many system failures is to have a backup copy. Periodically a file can be copied and saved in a separate location such as a bank vault. Then, when a file becomes contaminated or destroyed, the most recent version can be restored.

*Sixth, pay attention on the references from outside agencies.*

Independent consultants with expertise on specific types of applications or specializing in a given industry can be key resources here and may be able to help the project team eliminate inappropriate candidates.

LAS management chose to ignore advice from outside agencies. The managing director of a competing software house wrote various memoranda to LAS management in June and July 1991 describing the project as 'totally and fatally flawed'. Another consultant described LAS's specifications as poor in leaving many areas undefined. This undoubtedly is one of the mistakes which had eventually led to the wrong choice.

## **REFERENCES**

Flowers, Stephen (1996), “software failure: management failure”, Chichester: John Wiley and Sons, 1996, ISBN: 0471951137

E. Wainright Martin, Carol V. Brown, Daniel W. DeHayes, Jeffrey A. Hoffer, William C. Perkins (1998), “Managing Information Technology”, Fourty Edition, 2002, Prentice Hall, ISBN 0-13-064636-9

R. Pressman (2001), “Software Engineering”, fifth Edition, 2001, McGraw Hill, ISBN 0-07-249668-1