ELSEVIER

Interacting with Computers 11 (1999) 699–720

Interacting with Computers

Human error and information systems failure: the case of the London ambulance service computer-aided despatch system project

Paul Beynon-Davies*

Computer Studies Department, University of Glamorgan, Pontypridd CF37 1DL, Mid-Glamorgan, Wales, UK

Abstract

Human error and systems failure have been two constructs that have become linked in many contexts. In this paper we particularly focus on the issue of failure in relation to that group of software systems known as information systems. We first review the extant theoretical and empirical work on this topic. Then we discuss one particular well-known case — that of the London ambulance service computer-aided despatch system (LASCAD) project — and use it as a particularly cogent example of the features of information systems failure. We maintain that the tendency to analyse information systems failure solely from a technological standpoint is limiting, that the nature of information systems failure is multi-faceted, and hence cannot be adequately understood purely in terms of the immediate problems of systems construction. Our purpose is also to use the generic material on IS failure and the specific details of this particular case study to critique the issues of safety, criticality, human error and risk in relation to systems not currently well considered in relation to these areas.

Keywords: Human error; Information systems failure; Risk; Safety critical systems

1. Introduction

‘‘Most of the well-publicised failures of large computer systems have not been caused by simple breakdowns in their functioning, but by breakdowns in the larger web of computing in which the equipment residues’’ [1].

On the 27th October 1992 an information system (IS) made the lead story on the BBC’s Nine-O’Clock News. It was reported that a new computerised system established at the headquarters of the London Ambulance Service (LAS) (The London Ambulance Service’s

* Corresponding author. Tel.: 01443-482243; fax: 01443-482715.
E-mail address: pbdavies@glamorgan.ac.uk (P. Beynon-Davies)

PII: S0953-5438(98)00050-2

Computer-Aided Despatch System — hereafter referred to as the LASCAD system) failed, and that as a direct result of this failure the lives of 20–30 people may have been lost. In the aftermath, claims were made that the system was losing information essential to the effective performance of ambulance command and control. A counter-claim was made that a breaking up of established ways of working over the preceding weekend may have caused loss of local knowledge, contributing to a poor response by control staff.

This particular project is interesting in a number of ways:

1. It is a particularly cogent example of the features of IS failure in that it contains within it elements both of development problems and problems in use. Therefore, it is extremely useful as a means of exemplifying issues of human error both in relation to the construction of software systems and in terms of the ways in which such systems are meant to support human activity.
2. It is a particularly good example of the way in which the tendency to analyse IS failure solely from a technological standpoint is limiting. It highlights the way in which the nature of information systems failure is multi-faceted, and hence cannot be adequately understood purely in terms of the immediate problems of systems construction. To this purpose we emphasise the importance of 'web' models to explaining technical systems failure, i.e. explanations of failure that attempt to encapsulate the complex intertwining of relationships in the context of computing.
3. It is particularly useful in relation to our primary purpose here, which is to use both the generic material on IS failure and the specific details of this particular case study to critique the issues of safety, criticality, human error and risk in relation to systems not currently well considered in relation to these areas, namely, information systems.

The structure of the paper is as follows. Section 2 defines the concept of IS failure in terms of the extant theoretical and empirical work on this topic. Section 3 discusses the LASCAD case in terms of an organising framework based on the idea of web-explanations of failure. Section 4 analyses this case in terms of the issues of criticality, human error and risk in IS development. Section 5 concludes the paper with a summary of the key features of our discussion.

2. Information systems failure

IS failure is a topic that has become extremely newsworthy both in the general and the computing press in recent years. As a consequence, there is some evidence of an increasing concern amongst organisations in the UK and elsewhere with the large amounts of money that appears to have been devoted to software projects with little apparent organisational benefit. Available literature suggests that failure is a ubiquitous feature of both the UK and International experience of IS engineering [2].[1]

[1]Note we use the term IS engineering here in the broad sense used by the British Computer Society (BCS). In one sense, IS engineering is a discipline which broadens the area of software engineering from 'programming in the large' to include issues of people and organisations. In another sense, IS engineering has a more specific focus than software engineering in concentrating on software written to support human activity (particularly decision-making) within organisations.

Given its prominence it is perhaps not surprising to find that the topic of IS failure has been a fervent area of debate for academics in the information systems, software engineering and computer science areas for a number of years. A considerable amount of published theoretical work has helped academics and practitioners achieve a better understanding of the multi-faceted nature of IS failure. Perhaps surprisingly, however, what empirical data we have on this topic is limited to either anecdotal evidence, case studies of different orders and indeed quality, and a limited amount of survey work. This evidence has helped in validating, in a very broad manner, some of the theoretical approaches that aim to explain the phenomenon of IS failure.

2.1. Theoretical work

Since the 1970s, a number of frameworks have been proposed for understanding the concept of IS failure, for example: [3–6]. Two recent approaches to IS failure seem particularly important because of the way in which they relate IS failure to social and organisational context: Lyytinen and Hirschheim's concept of expectation failure and Sauer's concept of termination failure.

Lyytinen and Hirschheim [5], in conducting a survey of the literature on IS failure, identify four major theoretical categories of such phenomena:

1. Correspondence failure. This is the most common form of IS failure discussed in the literature and typically reflects a management perspective on failure. It is based on the idea that design objectives are first specified in detail. An evaluation is then conducted of the information system in terms of these objectives. If there is a lack of correspondence between objectives and evaluation the IS is regarded as a failure.
2. Process failure. This type of failure is characterised by unsatisfactory development performance. It usually refers to one of two types of failure. First, when the IS development process cannot produce a workable system. Second, the development process produces an IS but the project runs over budget in terms of cost, time, etc.
3. Interaction failure. Here, the emphasis shifts from a mismatch of requirements and system or poor development performance to a consideration of usage of a system. The argument is that if a system is heavily used it constitutes a success; if it is hardly ever used, or there are major problems involved in using a system then it constitutes a failure. Lucas [3] clearly adheres to this idea of failure.
4. Expectation failure. Lyytinen and Hirschheim describe this as a superset of the three other types of failure. They also describe their idea of expectation failure to be a more encompassing, politically and pluralistically informed view of IS failure than the other forms. This is because they characterise correspondence, process and interaction failure as having one major theme in common: the three notions of failure portray a highly rational image of IS development; each views an IS as mainly a neutral technical artefact [7]. In contrast, they define expectation failure as the inability of an IS to meet a specific stakeholder group's expectations. IS failures signify a gap between some existing situation and a desired situation for members of a particular stakeholder group. Stakeholders are any group of people who share a pool of values that define what the desirable features of an IS are, and how they should be obtained.

Lyytinen [8] broadens this analysis by making the useful distinction between development failure and use failure. Stakeholder groups can face problems in IS either in development terms or in terms of use. In the former case, a stakeholder's main concern is to mould the future IS to fit its interests. In the latter case, the main concern is to align the IS with the stakeholder's going concerns. In terms of development failure Lyytinen lists a number of categories of common problems: goals, technology, economy, organisational impact, participation and control of development, perception of development. In terms of use failures the following categories are held to be important: technical solutions, data problems, conceptual, people's reactions, complexity problems.

Sauer [6] has recently criticised the model proposed by Lyytinen and Hirschheim for its plurality. Sauer's model posits a more conservative definition of IS failure. According to his account an IS should only be deemed a failure when development or operation ceases, leaving supporters dissatisfied with the extent to which the system has served their interests. This means that a system should not be considered a failure until all interest in progressing an IS project has ceased. This definition of *termination failure* is hence stricter than Lyytinen and Hirschheim's concept of *expectation failure*.

Sauer develops a model of IS failure based on exchange relations. He portrays the development of information systems as an innovation process based on three components: the project organisation; the information system; and its supporters. Each of these components is arranged in a triangle of dependencies working within the context of an environment. The information system depends on the project organisation, the project organisation depends on its supporters, and the supporters depend on the information system. The information system requires the efforts and expertise of the project organisation to sustain it; the project organisation is heavily dependent on the provision of support in the form of material resources and help in coping with contingencies; supporters require benefits from the IS. Fig. 1 illustrates this triangle of dependencies.

One of the key ways in which Sauer distinguishes termination failure from expectation failure is in terms of the concept of a flaw. Information systems are the product of a process that is open to flaws. Every information system is flawed in some way. However, flaws are

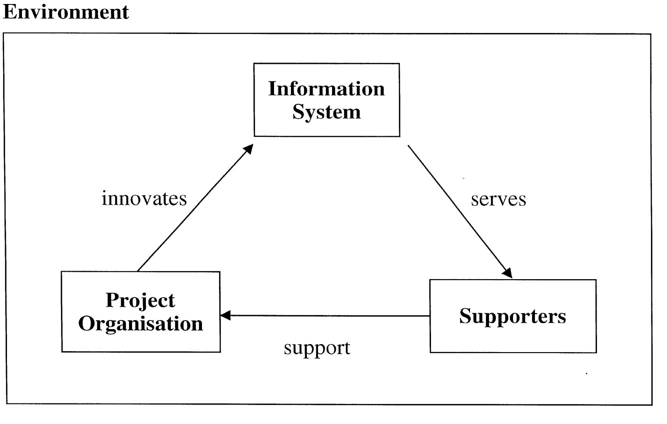

Fig. 1. Sauer's model of IS development (adapted from Ref. [6]).

different from failures. Flaws may be corrected within any innovation process at a cost, or accepted at a cost. Flaws describe the perception of stakeholders that they face undesired situations that constitute problems to be solved. Examples of flaws are program bugs, hardware performance, organisational changes, etc. Unless there is support available to deal with flaws they will have the effect of reducing the capacity of some IS to serve its supporters and may result in introducing further flaws into the innovation process. At some stage, the volume of flaws may trigger a decision to remove support and hence to terminate a project.

2.2. Empirical studies

For convenience it is useful to divide empirical investigations of IS failure into three categories: anecdotal evidence, case studies, survey research.

2.2.1. Anecdotal evidence

It has become something of an orthodoxy, or what Harel [9] calls a folk theorem, to assume that no IS project, particularly a failed project, can be understood in isolation from its context. Harel describes a folk theorem as an accepted wisdom with three major characteristics: popularity, anonymous authorship, and apparent age. Folk theorems seem to rely for much of their validity on anecdotal evidence. For a number of years a collection of anecdotal descriptions of IS failure has been accumulating in the ACM's *Software Engineering Notes*. McKenzie [10] has analysed this material and found that of computer-related accidents (examples mainly of use failures) reported, 92% involved failures in what McKenzie calls human–computer interaction.

''More computer-related accidental deaths seem to be caused by interactions of technical and cognitive/organisational factors than by technical factors alone.''

2.2.2. Case studies

Benbasat et al. [11] describe case research as being particularly important for those types of problems where research and theory are still at their early, formative stages. They see a key use of case studies in the generation of theory from practice. The topic of IS failure has been seen to be a particularly formative research area and, therefore, one particularly amenable to the case study approach.

The London Ambulance Computer-Aided Despatch System project has been one of the most frequently quoted UK examples of information systems failure in recent times [12]. However, in terms of the estimated financial cost of failure LASCAD (£1.1–£1.5 million) is dwarfed by other British IS 'failures' such as Wessex Regional Health Authorities' RISP project (£63 million) ([13]), and the UK Stock Exchange's TAURUS settlement system (£75–£300 million) [14].

IS failure is of course not specifically a British malaise. For instance, Oz [15] takes a similar documentary approach to describing the important case of the CONFIRM reservation system in the US ($125 million). Also, Sauer [6] comprehensively describes a large Australian government IS project — Mandata (A$30 million) — that was abandoned during the 1970s.

2.2.3. Survey research

Lucas [3] describes a series of quantitative investigations used to verify a number of general hypotheses on IS failure. Unfortunately, the data collected does little to illuminate the complex reality of organisational IS and indeed the factors contributing to IS failure.

Lyytinen [8] describes an exploratory study of the expectation failure concept by looking at systems analyst's perceptions of IS failure. Systems analysts are discussed in terms of being a major stakeholder group in systems development projects. Their view of IS and IS failures differs from users or management views. Interestingly, Lyytinen found that systems analysts believed that only 20% of projects are likely to turn out to be failures. They also preferred to explain failures in highly procedural and rationalistic terms. Reasons mentioned for failure include inexact development goals and specifications, inadequate understanding of user's work and inadequate understanding of system contingencies. This, Lyytinen concludes is concomitant with a set of professional expectations of the IS development process that conceives of it as an overtly rational act involving high technical and professional competence.

Ewusi-Mensah and Przasnyski [16], distinguish between what they call IS project abandonment and IS failure. IS failure in their terms deals with the failure of usage and/or operation of the IS, whereas IS project abandonment is concerned with the process of IS development. This is similar to Lyttinnen's distinction between development and use failure. In terms of project abandonment, Ewusi-Mensah and Przasnyski [17] delineate three major types:

1. *Total abandonment*. Complete termination of all activities on a project prior to full implementation.
2. *Substantial abandonment*. Major truncation or simplification of the project to make it radically different from the original specification prior to full implementation.
3. *Partial abandonment*. Reduction of the original scope of the project without entailing significant changes to the IS's original specification, prior to full implementation.

They suggest that from their small survey study, total abandonment is the most common type of development failure experienced in the US. They also found that organisational factors, particularly the amount of senior management involvement and the degree of end-user participation in the project development were the most widespread and dominant factors contributing to IS success–failure.

3. The LASCAD project

On the night of Monday 26th October to the morning of Tuesday 27th October 1992 things started to go wrong at the HQ of LAS. It was reported that a flood of 999 calls (some 2900 instead of the usual 2300) apparently swamped operators' screens. It was also claimed that many recorded calls were being wiped off screens. This, in turn, caused a mass of automatic alerts to be generated indicating that calls to ambulances had not been acknowledged.

Claims were later made in the press that up to 20–30 people may have died as a result of ambulances arriving too late on the scene. Some ambulances were taking over 3 h to answer a call. The government's recommended maximum is 17 min for inner-city areas [18]. A counter-claim was made that a breaking up of sector desks over the preceding weekend may have caused loss of local knowledge.

Arguably the LASCAD project was the most visible UK information systems failure in recent years. It is therefore not surprising to see that the situation described above triggered a whole series of responses. In many ways such responses seem reminiscent of a moral panic in IS work [19].

3.1. Web explanations of information systems failure

This paper uses what has been called a web description of information systems failure [11]. That is, it explains failure in terms of the complex intertwining of relationships in the context of computing. Web explanations such as these are necessarily complex. They do not offer a simple linear explanation of phenomenon and consequently provide no simple answers to the problem of preventing information systems failure.

We shall also use Sauer's triangle of dependencies model (see Section 2.1) as a means of hanging key elements of our description of the LASCAD case. A more detailed description of this project is given in [12].

3.2. The information system

LASCAD was an attempt to build a computer-aided despatch system for the London Ambulance Service. In this section we set the scene by describing:

1. elements of computer-aided despatch systems;
2. how the LASCAD system was intended to work;
3. the human activity system within which LASCAD was placed.

3.2.1. Elements of a computer-aided despatch system

A computer-aided despatch (CAD) system for an ambulance service would normally expect to provide one or more of the following system functions [20].

1. *Call taking*: acceptance of calls and verification of incident details including location.
2. *Resource identification*: identifying resources, particularly which ambulance to send to an incident.
3. *Resource mobilisation*: communicating details of an incident to the appropriate ambulance.
4. *Resource management*: primarily the positioning of suitably equipped and staffed vehicles to minimise response times.
5. *Management information*: collation of information used to assess performance and help in resource management and planning.

Until quite recently most despatch systems were manual in nature. It is useful to understand some of the rationale for the LASCAD project if we briefly consider the workings of a manual despatch system. Such a manual system would ideally consist of, amongst others, the following functions [20].

1. Call taking. Emergency calls are received by ambulance control. Control assistants write down details of incidents on pre-printed forms. The location of each incident is identified and the reference co-ordinates recorded on the forms. The forms are then placed on a conveyor belt system that transports all the forms to a central collection point.
2. Resource identification. Other members of ambulance control collect forms, review details on forms, and on the basis of the information provided decide which resource allocator should deal with each incident. The resource allocator examines forms for his/her sector and compares the details with information recorded for each vehicle and decides which resource should be mobilised. The status information on these forms is updated regularly from information received via the radio operator. The resource is recorded on the original form that is passed on to a despatcher.
3. Resource mobilisation. The despatcher either telephones the nearest ambulance station or passes mobilisation instructions to the radio operator if an ambulance is already mobile.

Many UK ambulance services have now put some form of computerisation in place. Such systems particularly address the call-taking and resource identification functions described above. The major rationale expressed for such computerisation is typically that a number of problems are seen to exist with the manual CAD systems. Most such problems relate 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; maintaining up-to-date vehicle status information. A CAD system is seen by many within the ambulance service as a means of overcoming a number of these problems, and, in particular, improving the service to patients. In this light, one particularly contentious area of computerisation that has been undertaken by many ambulance services over the last few years is the incorporation of so-called triage (despatch in terms of medical priority) systems into ambulance command and control.

3.2.2. How LASCAD was intended to work

The major objective of the LASCAD system was to automate many of the human-intensive functions described above. A diagram illustrating the essential features of the proposed system is provided in Fig. 2. The basic functionality of the intended LASCAD system is described below:

1. BT operators route all 999 calls concerning medical emergencies as a matter of routine to LAS headquarters (HQ) in Waterloo, London.
2. Eighteen 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.
3. This information was then transmitted over a local area network to an ‘allocator’. The system would pinpoint the patient’s location on a map display of areas of London.
4. The system was expected to continuously monitor the location of each ambulance via radio messages transmitted by each vehicle every 13 s. The system would then determine the nearest ambulances to the patient.
5. Experienced ambulance despatchers were organised into teams based on three zones (south, north-east and north-west). Ambulance despatchers would be offered details by the system of the three nearest ambulances and the estimated time each would need to reach the scene.

6. The despatcher 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 they were on their way.
7. If the selected ambulance was in an ambulance depot then the despatch message would be received on the station printer.
8. The ambulance crew would always be expected to acknowledge a message. The system would automatically alert the HQ of any ambulance where no acknowledgement was made. A follow-up message would then be sent from HQ.
9. The system would detect from each vehicle's location messages if any ambulance was heading in the wrong direction. The system would then alert controllers.
10. 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) [21]. The system was built by a small Aldershot-based software house called Systems Options using their own GIS software (WINGS) running under Microsoft Windows [22]. 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.

3.3. The human activity system

Checkland [23] has argued that any information system can only really be understood and analysed in relation to the human activity system that it supports. Human activity systems consist of people, conventions and artefacts designed to serve human needs.

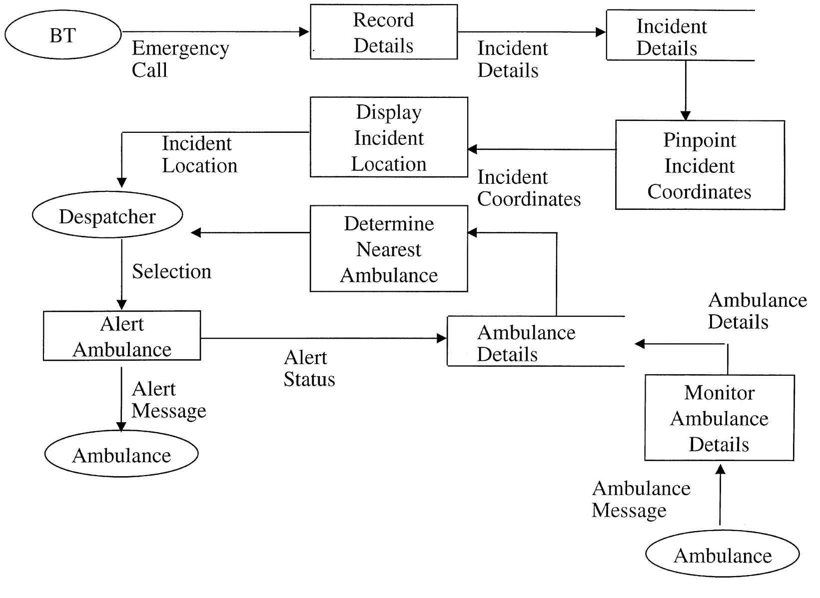

Fig. 2. How LASCAD was intended to work.

It must be understood that LAS is unlike any other ambulance service in the UK. The service receives 10 times as many emergency calls as any other ambulance service in the country. The organisation covers a geographical area of just over 600 square miles and handles emergencies for an area with a resident population of 6.8 million people.

Questions were also raised in the subsequent public inquiry about the complexity of the technical system. A typical ambulance despatch system, like the ones employed at the time in Surrey, West Yorkshire or Berkshire, merely acted as a repository of details about incidents. Communication between HQ and ambulances was conducted via telephone or voice radio links (see Ref. [24]). In the LASCAD system, links between communication, logging and despatching via a GIS were meant to be automated.

It is therefore tempting to adopt a stance of explaining this ‘failure’ purely in terms of technical problems. However, the report of the public inquiry (see Ref. [20]) portrays a more complex picture of the so-called technical problems experienced by the LASCAD system than that reported either in the computing or general press. It is interesting that they conclude:

> On 26th and 27th October the computer system did not fail in a technical sense. Response times did on occasions become unacceptable, but overall the system did what it had been designed to do. However, much of the design had fatal flaws that would, and did, cumulatively lead to all of the symptoms of systems failure.

Discussions with a number of people have revealed a range of opinions about this important statement. However, if we take this statement at face value, it does beg the question of what did happen to the system to cause response times to become unacceptable?

According to the public inquiry, the system was lightly loaded at start-up on 26th October 1992. Any problems caused particularly by the communications system such as ambulance crews pressing wrong buttons, or ambulances being in radio blackspots, 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 despatch. 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 numbers of exception messages appear to have increased to such an extent that staff were not able to clear the queue and 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 exception queues, it took longer to allocate resources to incidents.

At the receiving end, patients became frustrated with the delays to ambulances arriving at incidents. This lead 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 phone-answering. This caused further delays to patients.

At the ambulance end, crews became increasingly frustrated at incorrect allocations. The inquiry [20] believes that this may have led to an increased number of instances where crews did not press the right status buttons, or took a different vehicle to an incident than that suggested by the system. Crew frustration also appears to have contributed to a greater volume of voice radio traffic. This in turn contributed to the rising radio communications bottleneck. The bottleneck caused a general slowing down in radio communications which fed-back into increasing crew frustration.

The system, therefore, appears to have been in a vicious circle of cause and effect. This vicious circle is illustrated diagrammatically in Fig. 3. The arrows on this diagram are meant to represent causal dependencies between critical elements of the human activity system during the period in question.

3.4. Project organisation

As with any project of this nature, the LASCAD system was shaped by the prior history of IS innovation within the organisation in question.

Firstly, it is interesting that, Systems Options, the company supplying the major part of the software for the system, is reported as having had no previous experience of building despatch systems for ambulance services. The company had won the £1.1 million contract for the system in June 1991. However, it appears that the London Ambulance Service had

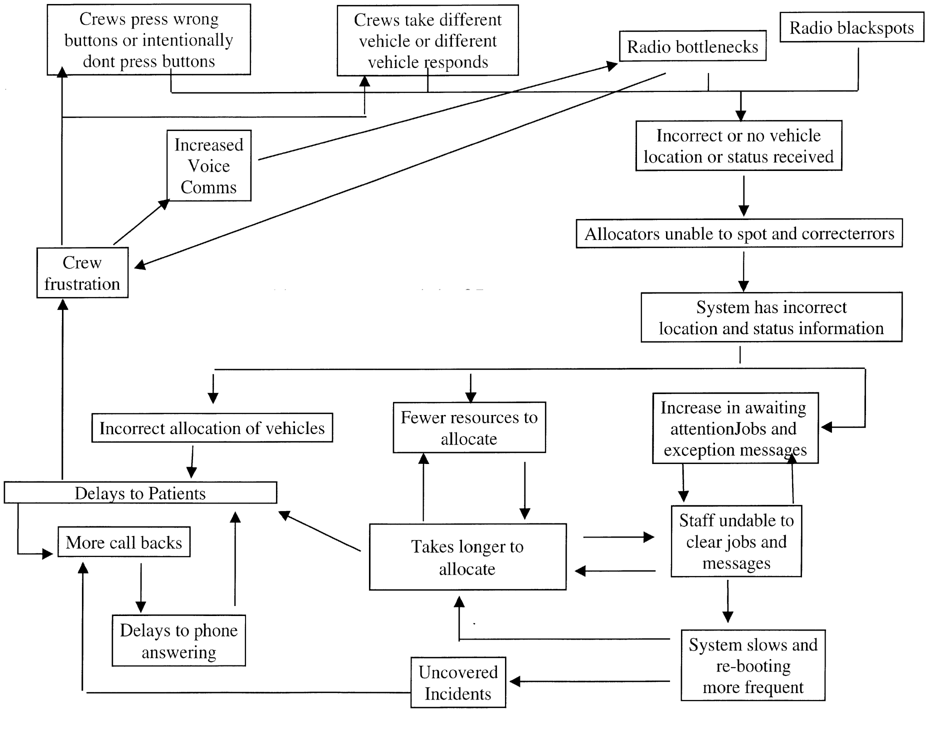

Fig. 3. A simplified model of LASCAD failure (after Ref. [20]).

previously scrapped a BT subsidiary IAL development at a cost of £7.5 million in October 1990. This project is reported to have been late starting in May 1987 after a delay of 1 year. The reason for scrapping this earlier project seems to have revolved around a debate over faulty software. The LAS sought damages from IAL for a faulty despatch module in October 1990 [25].

Secondly, Systems Options substantially underbid an established supplier McDonnell–Douglas and were put under pressure to complete the system quickly. The managing director of a competing software house wrote a number of memoranda to LAS management in June and July 1991 describing the project as ‘totally and fatally flawed’ [26]. Another consultant described LAS's specifications as poor in leaving many areas undefined [27].

In January 1992 phases one and two of the project began live trials. In March 1992, phase two of the trials were temporarily suspended following claims, particularly from the union NUPE, of fatal delays caused by system errors. In October 1992 phase three was terminated after two days of chaos (see Section 3.3).

A number of the findings of the public inquiry report [20] directly relate to project organisation:

First, it was claimed that the LAS chiefs ignored what amounted to an overambitious project timetable. The original procurement document, which was drafted within the guidelines provided by the regional health authority, put price before quality. A report by Anderson Consulting in late 1990, that called for more finance and longer timescales on the project, was suppressed by project managers.

Second, the LAS board were misled by the project team over the experience of Systems Options. The references supplied by Systems Options were not thoroughly investigated. Also, confusion seems to have arisen over who constituted the main contractor in the project. Systems Options, being an Apricot reseller, appear to have originally believed that Apricot would lead the contract.

Third, that the management of the project was inadequate. The project team failed to use the PRINCE [28] project management method as prescribed for public sector projects.

Fourth, the software was incomplete and unstable. In particular, the emergency backup system remained untested. Questions were also raised about the choice of Visual Basic as a development tool and Microsoft Windows as the host operating system.

Fifth, training in the use of the system was incomplete and inconsistent.

3.5. The supporters

Sauer's use of the term supporter (see Section 2.1) is somewhat unfortunate. We prefer the use of the term stakeholder [29] in the sense that not all groups with an interest in the development of an information system necessarily support that development. Some stakeholder groups may have a definite negative interest in the success of a given project. There is some evidence of this in the LASCAD case, particularly in the apparent mismatch of perspectives between LAS management and HQ and ambulance staff.

The system has been described as being introduced in an atmosphere of mistrust by staff stimulated by the many problems experienced with various system components in the preceding months. Consequently, there was incomplete ‘ownership’ of the system by the majority of its users.

Hardware and software suppliers dealing with the LAS spoke of disorganisation, low staff morale, friction between management and the workforce, and an atmosphere of hostility towards computing systems. An ambulance crew member is reported as saying:

> whatever system you have people have to have confidence in it. We want to go back to the simple system, telephone and radio. Anybody can use it. Crews have confidence in it [30].

One of the reasons for this low staff morale may be that control room staff had virtually no previous experience of using computers [31]. The union NUPE continually made aspersions to what they considered a 'macho' style of management at LAS. The Labour party's health spokesman at the time, David Blunkett, demanded a public inquiry into the system in September 1992, a month before the events described above, after receiving a report from NUPE [32].

3.6. The environment

The environment of the LASCAD project can be considered in the macro sense of the constraints imposed by the overarching organisation of the NHS and in the micro sense in terms of the labour relations history at the LAS.

The political and economic context of the NHS has clearly influenced the current shape of the organisation's computing, including most of the information systems projects conducted within the remit of the organisation. Indeed, the LASCAD system is only one example of supposed IS failure within the NHS. Wessex Regional Health Authority's Regional Information Systems Plan is another case in point [13].

Firstly, it must be understood that there is no demonstrable and unitary power-structure within the NHS. The NHS is a body made up of a complex network of autonomous and semi-autonomous groups concerned with health matters. Actual delivery of health care is in the hands of powerful clinical professionals who are naturally concerned with preserving their professional autonomy [33].

One consequence of this is that any project carried out in the NHS, such as LASCAD, has to consider what relationships in the network are affected by the project and what activities have to be undertaken to enable or encourage those relationships [34]. For instance, in a related paper we have discussed some of the constraints this enabling network has placed upon information management in the NHS [35].

Computing within the NHS is therefore complicated by the fact that no one body has overall responsibility for information technology (IT). IT is exploited and controlled at a number of different levels: region, trust, hospital, department, speciality and general practice. Each stakeholder has a different perception of IT. Region and trust tend to emphasise administrative systems. Hospital and GP surgery emphasise clinical applications. Regions emphasise management information. Trusts emphasise operational systems.

The lack of a clear organisation for IT has meant the absence of a clear strategic vision for IT. To build a strategy there must be first some agreement on objectives. Clearly, many of the objectives of the various stakeholders are in conflict. This situation is has not changed with the moves towards a market of providers and purchasers in the NHS.

A great deal of the shape of the LASCAD project was determined by the internal tensions within the NHS. For example, members of the public inquiry [20] reflect on some of the stresses and strains that had been placed on the LAS by changes in the NHS in the period prior to the implementation of LASCAD.

> Under the NHS reforms, all parts of the NHS have gone through major cultural changes in the past few years and it is evident that the LAS could not bury its head in the sand if it was to provide a professional and successful service in the 1990s. However, the result of the initiatives undertaken by management from 1990–1992 did not revitalise management and staff as intended, but actually worsened what was already a climate of mistrust and obstructiveness. It was not a case of management getting the agenda wrong. The size of the programme and the speed and depth of change were simply too aggressive for the circumstances. Management clearly underestimated the difficulties involved in changing the deeply ingrained culture of LAS and misjudged the industrial relations climate so that staff were alienated to the changes rather than brought on board.

However, it is misleading to portray the management problems of the LAS purely in the context of the two years prior to the events of 1992. Many of the pressures on the LASCAD project can be seen to be the result of a protracted climate of conflict in the ambulance service between management, unions and the government of the day. The public inquiry maintains that during the 1980s there was clear evidence that management failed to modernise the service. This was reflected in a lack of investment in the workforce (such as paramedic training and career advancement), the fleet and the estate. By the end of 1990, at the end of a protracted national dispute over pay, the LAS stood in need of major modification and change. During the period between January and April 1991 the numbers of senior and middle-management posts within the LAS were reduced by 53. There appears to have been little consultation with staff over this restructuring and the whole process caused a great deal of anxiety in the organisation.

Therefore, the public inquiry [20] cites an important reason for the unstable industrial relations climate within LAS as the ‘fear of failure’ on the part of management. Changes in structure created a climate in which management was continually under pressure to succeed. This may have put undue pressure on management to ensure that the LASCAD system was implemented on time and within budget. However, it may also have blinded them to some of the fundamental difficulties of the system implementation.

The inquiry team believe that most of the operational management at LAS was of the opinion that LASCAD would act as an essential means of overcoming what they saw as outmoded practices. Such practices included the ability of crews themselves or the ambulance stations to decide which resource to mobilise in response to an incident. These practices were to be replaced with what management saw as a system that would decide in an objective and impartial way the optimum mobilisation of resource.

Clearly, the management was naive in assuming that the simple introduction of a computer system would automatically result in changes in working practices. Crews and stations, if they wished, could still accommodate older practices by employing strategies such as failing to mobilise, sending a different resource, or failing to acknowledge or report status.

4. Analysis of the LASCAD failure

Clearly, it is impossible to point to any one element of the description above as being the cause of the LASCAD failure. The description hopefully demonstrates how the explanation of a particular IS failure must be multi-faceted or web-like in nature. Indeed, the LASCAD project displays many of the features of Perrow's [36] concept of a 'normal accident'. That is, an accident that is particularly hard to foresee in detail because it involves failures in several parts of a system that are often linked in complex and subtle ways.

It is interesting, of course, that there is even some debate about whether the LASCAD case even constitutes a failure. It is notable that LASCAD was not purely a technical failure. It is better characterised as an expectation failure on the part of most stakeholder groups, particularly the LAS HQ staff and ambulance drivers. Also, LASCAD is not a termination failure. Following the events described, the LASCAD project organisation was re-structured and consequently had not lost the support of the major stakeholders, the LAS, South West Thames RHA and the NHS. The new head of IT at LAS was reported as having until August 1997, with a provisional budget of £13.5 million, to deliver a despatch system for LAS. The first stage of the work, a system for call-logging, call transfer, address-finding and vehicle location is reported as having reached completion in April 1995 and was introduced successfully in January 1996 [37]. A number of additional functions were added to this system in 1996 and the head of IT at LAS was recently awarded a British Computer Society project management award [38].

4.1. Development and use failure

The LASCAD case is useful for the way in which it bridges the idea of IS failure as development failure and the idea of IS failure as use failure. Sauer's analysis of IS failure tends to concentrate on the failure of a development project. The analysis tends to stop at the point at which the system is delivered and used. This may be because most of the well-studied examples of failures were never fully delivered.[2] LASCAD is particularly interesting for the way in which it demonstrates the failure of systems in the context of use.

Hence, Sauer's triangle of dependencies continues after a system has been delivered. After delivery, an IS is subject to use and maintenance. The further development and maintenance of a given information system we can describe as the post-implementation trajectory of the IS. Kling and Iacono [40] discuss the way in which organisational politics affect the post-implementation trajectory of an information system and illustrate how structural and ideological concerns influenced the trajectory of development at one US company. They make an interesting comment:''An information system which is 'successful' from the point of view of its implementation can be problematic for its users. We often find that practitioners are embarrassed about systems which have imperfect architectures, but need not be viewed as failures''.

Fig. 4 relates the ideas of IS failure and project trajectory.

One of the key ways in which organisational politics may affect the post-implementation trajectory of an IS is through user resistance. Hirschheim and Newman [41], for

[2]See [12] and [39] for a discussion of some of the recent cases of IS failure.

instance, provide a case study that illustrates how user resistance differed in departments with differing rates of participation in implementation. Keen [42] details a number of counter-implementation strategies that users may take to impede the development of the system or its implementation:

1. 'Lay low'. If you do not want a system to succeed, then the more you keep out of the way and do not give help and encouragement, the more likelihood there is of failure.
2. 'Rely on inertia'. If you can be too busy when asked then the implementation process may come to a halt.
3. 'Keep the project complex, hard to coordinate and vaguely defined'. If the goals are ambiguous or too ambitious there is every chance of failure as energy is dissipated in many different directions.
4. 'Minimise the implementers legitimacy and influence'. If the designers are kept as outsiders, other users will probably not allow them to work effectively.
5. 'Exploit their lack of inside knowledge'. The design team probably know very little about the detailed nature of the work and if they are denied this knowledge, the system will probably prove to be inadequate when it is implemented.

4.2. Human error

As Leveson [43] argues, the most common over-simplification in reports of accidents such as IS failures is to lay the 'cause' at the foot of human error. It is therefore not surprising to find that the concept of human error clearly has a bearing on both development and use failure. Error presupposes some divergence from agreed standards of behaviour. Again, it is useful to clarify the issue of error in relation to IS projects in terms of development error or use error.

4.2.1. Development error

On the development front, most of the cited reasons for the failure of the LASCAD system were expressed in terms of a divergence from a set of principles of best practice established in the domain of software engineering. For instance, the project is cited as lacking clear project management, did not use any accepted systems development methodology, was hurried and over-ambitious, and no thorough testing was carried out.

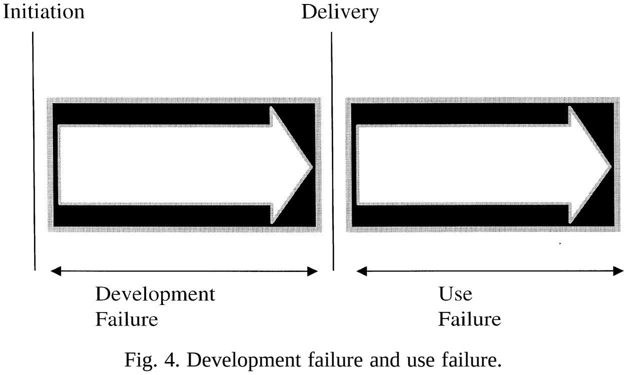

Fig. 4. Development failure and use failure.

In this sense the project is a potent example of a development breakdown. In other disciplines, the detailed study of 'breakdowns' has proven significant in changing established practice. Boddie [44] cogently captures this idea in the following quote: ''We talk about software engineering but reject one of the most basic engineering practices: identifying and learning from our mistakes. Errors made while building one system appear in the next one. What we need to remember is the attention given to failures in more established branches of engineering.'' In structural engineering, for instance, the study of bridge failures has revealed important information leading to the re-design of bridges [45].

However, there is some evidence to suggest that LASCAD is not unusual in the way in which developers diverged from the behaviours prescribed in the development literature. For instance, Button and Sharrock [46] in a study of one development project describe various ways in which software engineers negotiate an order of work that is distinctly different from that described in standard development methodologies. The key difference here was between following the development methodology and saying they were following the development methodology.

In terms of the extant normative or prescriptive literature one might be tempted to propose that IS failures would be prevented if best practice was followed. However, this is a rather simplistic argument. Rather than prescribing what developers should do, perhaps we should first attempt to describe what they currently do.

The link between following precisely the tenets of a given development methodology and the 'success' of the eventual systems produced via this approach remains largely unproved. Perhaps this explains something of the low-takeup of methodologies amongst commercial organisations. Fitzgerald [47], for instance, has found a relatively low utilisation of large-scale methodologies in the current development domain in Eire. This is confirmed by research conducted by Moynihan and Taylor [48] in the UK. Frequent reasons cited in both studies is the perceived cumbersome nature of methodologies such as SSADM.[3]

It is probably true to say that best software engineering practice has more to say about technical construction than socio-technical construction. Whereas there is evidence that IS practitioners are at least aware of the principles of best practice in relation to issues of technical construction, there appears to be a lack of awareness of best practice issues in relation to social–technical interaction. This might partly explain the apparent low take-up of socio-technical approaches such as Checkland's Soft Systems Methodology [23] in the commercial arena [50].

It may be that many aspects of best practice seems not to be followed by practitioners because they are seen to be in some way interfering with the effective development of systems. The current interest in rapid application development (RAD) [51], for instance, might be seen as a reaction against the large amounts of documentation required by traditional methodologies.[4]

[3] Structured Systems Analysis and Design Method [49].

[4] Rapid Applications Development is a method which emphasises rapid delivery of systems through the use of high-level production tools, prototyping and user involvement.

4.2.2. Errors of use

In terms of errors of use, there were clear areas of ambivalence and ambiguity in the context of use in the LASCAD project. At the organisational level, for instance, the prior history of poor relationships between management and workforce at LAS and the previous history of failed IS projects clearly contributed to the 'failure' of this system. There was also clear evidence of an attempt by management to utilise the IS for introducing new patterns of work. Consequently, there is some evidence of resistance on the part of the workforce to the introduction of the new system, further exacerbated by poor user involvement in the design and implementation of the system.

A key element of error in relation to the development of information systems lies in the idea of escalation in the decision-making process associated with IS projects. Sauer's concept of an IS flaw is interesting in the context of the literature on escalation in organisational decision-making. Drummond [52] defines escalation as:

> the predicament where decision-makers find themselves trapped in a losing course of action as a result of previous decisions. Costs are incurred; there is an opportunity to withdraw or persist; and the consequences of withdrawal or persistence are uncertain. Typically the response to such dilemmas is 'irrational persistence'.

The important point about the escalation concept is that support for an IS project can continue even in the face of major system flaws. Major stakeholders in a project may be reluctant to withdraw support because of the heavy investment in personnel and other resources devoted to a project. Newman and Sabherwal [53] have argued that commitment on the part of project participants is a necessary determinant of the success of an IS project. However, commitment is a two-edged sword in that if the 'right' level of commitment is not struck in a project, 'support' in Sauer's terms, may be an escalating process, and certainly may have many features of an irrational and potentially destructive process.

Elements of irrational persistence certainly seem to have been evident in the LASCAD project. This is particularly the case in relation to decisions made to follow a previously failed implementation with a project so tightly constrained in terms of budget and time-scale.

4.3. Criticality

The scale of the LASCAD project is clearly much less than other IS projects such as TAURUS and Wessex RISP, using any measure such as expenditure or man-years of effort. The scale of a failure is clearly one aspect that contributes to the visibility of the failure. However, in terms of LASCAD, the visibility of the project probably had more to do with factors such as the high profile of the LAS and its problems in previous news reporting, the fact that public funds were spent on this project, and the 'safety-critical' nature of the IS. The prominence of this particular case is probably as a result of the claim that 20–30 people may have lost their lives as a result of this failure. This factor is significant in the sense that prior to LASCAD, command and control systems for ambulance services do not seem to have been discussed in the context of safety-critical systems, at least within the UK.

In this sense, the LASCAD project appeared to act as useful fuel for expanding the concept of critical safety to include a larger number of systems than those previously classified as such. This raises the question of whether other types of IS can be considered safety critical. For example, a patient administration system that stores details of information such as people's addresses and GP's details would probably not be regarded as a safety critical system. However, when this information begins to be combined in an integrated hospital information with clinical details such as prognosis and treatment details, clearly errors in entering or manipulating such information may have a direct bearing on the quality of treatment and consequently mortality of patients. Examples such as these demonstrate that criticality is a socially constructed dimension framed through the inter-subjective agreement as to the risk of failure and its consequences in terms of human safety.

4.4. Risk

Perhaps because of the apparent ubiquity of software systems failure, the area of risk and risk assessment has become particularly prominent in the software engineering literature in recent times [54]. The folk theorem here [9] is clearly that risk is involved in all IS projects. Risk might be defined as a negative outcome that has a known or estimated probability of occurring based on some experience or theory. The idea of IS failure is clearly the negative outcome most prominent in most people's minds. However, our analysis above clearly supports the viewpoint expressed by Wilcocks and Margetts [55] that:

> Risk of a negative outcome only becomes a salient problem when the outcome is relevant to stakeholder concerns and interests. Different settings and stakeholders will see different outcomes as salient.

Risk assessment is clearly the process involved in estimating the degree of risk associated with a given project, usually at the feasibility stage of development. A number of frameworks have been generated which suggest a list of characteristics indicative of risky IT projects. For instance, Cash et al. [56] suggest that there are at least three important dimensions that influence the risk of a development project: project size, experience with the technology and project structure. In general, the smaller, more experienced and more highly structured the project the less risk is likely to be associated with it.

However, a web model of IS failure such as the one discussed in this paper has difficulty in melding with a risk assessment framework such as Cash et al.'s that ignores context, history and organisational processes. A 'failure' framework such as Sauer's has more in common with a recent risk assessment framework proposed by Wilcocks and Margetts. In this approach, six interplaying categories are brought into analysing the development, introduction and use of IS:

1. *History*. Prior organisational developments, e.g. prior IS success or failure.
2. *Outer context*. The givens that an organisation and its members need to respond to and accommodate, e.g. government, the economy, markets etc.

3. *Inner context*. The characteristics of the organisation itself, e.g. strategy, structure, reward systems.
4. *Content*. The changes involved in and substance of a project, e.g. size of project, difficulty.
5. *Processes*. How things are done and the issues perceived, e.g. project management, staffing etc.
6. *Outcomes*. Planned or unanticipated, e.g. cost, time, etc.

There are clear links here between Sauer's triangle of information system, project organisation and supporters working within a historical context and environment and Wilcocks and Margett's collection of interplaying factors contributing to risk.

5. Conclusion

To conclude, we discuss some of the important lessons for the concepts of human error, criticality and risk.

In this paper we have presented a case history of a prominent UK IS project which many people report as a 'failure'. Case histories of this nature are a valuable means of helping us to understand the complexity of IS development [11].

We make no claims that we have provided the definitive account of this particular project. This is resonant of Pinch's [57] study of the space shuttle Challenger incident in which he argues that although technical inquiries search for technical scenarios as causes, it is extremely difficult to come up with a definitive scenario for explaining technical systems failure. Technical experts frequently disagree about the contributing factors to a failure. There appears to be a great deal of 'interpretative flexibility' available to people wishing to explain such failures. This seems to apply equally to ideas of the contribution of human error in IS failure. On the one hand, developers can be portrayed as having erred in relation to promoted ideals of development practice. On the other hand, users can be portrayed as having erred in relation to some established norms of commitment, co-operation and use. However, both notions of best practice and the rules of systems engagement are socially constructed and hence open to a number of different points of interpretation.

Explanatory frameworks such as Lyytinnen and Hirschheim's and Sauer's are particularly useful in broadening the notion of technical systems failure and highlighting the political, economic and social nature of IS failure. Lyytinnen and Hirschheim's concept of expectation failure clearly locate the idea of such failure in the area of human interpretation and expectations. Sauer's framework is useful in proposing that it is only when relationships between crucial elements of an information systems project break down irretrievably can the project be said to have failed. In this light, it is interesting that Ian Tighe, the new head of IT at LAS is reported as portraying his prescription of success as being: ''to recognise the risks, allow people sufficient time to absorb change, and not be lulled into taking the cheapest solution'' [58].

McKenzie [10] in his comprehensive empirical analysis of computer-related accidental death has some equally interesting comments about the relationship between failure, risk and 'human factors'.

> To make computer systems safer, we need to address not merely their technical aspects, but also the cognitive and organisational aspects of their 'real-world' application… If this does not happen, there is a risk that purely technical efforts to make computer systems safer may fail. Not only are they addressing only part of the problem, but they may conceivably even increase risks, through their effects on beliefs about computer systems.

References

[1] T. Winograd, F. Flores, Understanding Computers and Cognition, Ablex Publishing, NY, 1986.
[2] Coopers and Lybrand Management Publications, Managing Information and Systems Risks: results of an international survey of large organisations, 1996.
[3] H.C. Lucas, Why Information Systems Fail, Columbia University Press, New York, 1975.
[4] V. Bignel, J. Fortune, Understanding Systems Failures, Manchester University Press, Manchester, 1984.
[5] K. Lyytinen, R. Hirschheim, Information systems failures: a survey and classification of the empirical literature, Oxford Surveys in Information Technology 4 (1987) 257–309.
[6] C. Sauer, Why Information Systems Fail: A Case Study Approach, Alfred Waller, Henley-On-Thames, 1993.
[7] H.K. Klein, R.A. Hirschheim, A comparative framework of data modelling paradigms and approaches, The Computer Journal 30 (1) (1987) 8–14.
[8] K. Lyytinen, The expectation failure concept and systems analysts view of information systems failures: results of an exploratory study, Information and Management 14 (1988) 45–55.
[9] D. Harel, On fold theorems, CACM 23 (7) (1980) 379–389.
[10] D. McKenzie, Computer-related accidental death: an empirical exploration, Science and Public Policy 21 (4) (1994) 233–248.
[11] I. Benbasat I, D.K. Goldstein, M. Mead, The case research strategy in studies of information systems, MIS Quarterly (Sept.) 1987.
[12] P. Beynon-Davies, Information systems 'failure': the case of the London ambulance service's computer-aided despatch system, European Journal of Information Systems 4 (1995) 171–184.
[13] Public Accounts Committee, Wessex Regional Health Authority Regional Information Systems Plan, HMSO, London, 1993.
[14] R. Waters, The plan that fell to Earth, Financial Times, 12th March, 1993, p. 19.
[15] E. Oz, When professional standards are lax: the confirm failure and its lessons, CACM 37 (10) (1994) 29–36.
[16] K. Ewusi-Mensah, Z.H. Przasnyski, Factors contributing to the abandonment of information systems development projects, Journal of Information Technology 9 (1994) 185–201.
[17] K. Ewusi-Mensah, Z.H. Przasnyski, Learning from abandoned information system development projects, Journal of Information Technology 10 (1995) 3–14.
[18] The Guardian, Ambulance chief resigns, 29th Nov. 1992. pp. 1–2.
[19] S. Cohen, Folk Devils and Moral Panics: the Creation of the Mods and Rockers, new ed., Martin Robertson, London, 1980.
[20] D. Page, P. Williams, D. Boyd, Report of the public inquiry into the London ambulance service, South West RHA, 1993.
[21] Daily Telegraph, 'Flood of Calls' 999 chaos, 29th Oct. 1992, p. 1.
[22] Computer Weekly, Ambulance revert to paper-based backup, Nov. 5th, 1992, p. 6.
[23] P. Checkland, Systems Thinking, Systems Practice, Wiley, Chichester, 1987.
[24] Financial Times, Dire effects of a failure to communicate: what caused the London Ambulance Service computer system to crash?, 30th October, 1992, p. 1.
[25] Computing, LAS Chief slammed over system disaster, 4th March 1993, p. 6.
[26] Computing, LAS bosses ignored warnings, 5th Nov. 1992, p. 1.

[27] Computer Weekly, LAS system to stay down until February, 12th Nov. 1992, p. 4.
[28] C. Bentley, Introducing PRINCE: The Structured Project Management Method, NCC Blackwell, Oxford, 1992.
[29] F. Land, Evaluation of systems goals in determining a design strategy for a computer-based information system, Computer Journal 19 (4) (1976) 290–294.
[30] Financial Times, Staff revert to trusted radio and telephones, 30th October 1992, p. 4.
[31] New Scientist, Ambulance computer system was 'too complicated', 14th Nov. 1992, p. 7.
[32] Computing, Labour calls for inquiry into ambulance system, 24th Sept. 1992, p. 5.
[33] R. Leavitt, A. Wall, The Reorganised National Health Service, 4th ed., Chapman and Hall, London, 1992.
[34] P. Checkland, J. Scholes, An application of soft systems methodology in the NHS, in: Soft Systems Methodology in Action, John Wiley, Chichester, 1990.
[35] P. Beynon-Davies, Information management in the British National Health Service: the pragmatics of strategic data planning, International Journal of Information Management 14 (2) (1994) 84–94.
[36] C. Perrow, Normal Accidents: Living with High-risk Technologies, Basic Books, New York, 1984.
[37] I. Tighe, All Systems Go! , Computer Bulletin 8 (4) (1996) 20–21.
[38] Computing, Emergency room, 12th June 1997.
[39] S. Flowers, Software Failure/Management Failure: Amazing Stories and Cautionary Tales, Wiley, Chichester, 1996.
[40] R. Kling, S. Iacono, The control of IS developments after implementation, CACM 27 (12) (1984) 1218–1226.
[41] R. Hirschheim, M. Newman, Information systems and user resistance: theory and practice, Computer Journal 31 (5) (1988) 398–408.
[42] P. Keen, Information systems and organisational change, CACM 24(1) (1981).
[43] N.G. Leveson, Safeware: System Safety and Computers, Addison-Wesley, Reading, MA, 1995.
[44] J. Boddie, The project post-mortem, Computerworld, 7th Dec. 1987.
[45] H. Petrowski, To Engineer is Human: the Role of Failure in Successful Design, Macmillan, London, 1985.
[46] G. Button, W. Sharrock, Practices in the work of ordering software development, in: A. Firth (Ed.), The Discourse of Negotiation, Oxford, Pergamon, 1993, pp. 159–180.
[47] B. Fitzgerald, An investigation into the use of systems development methodologies in practice, Proc. ECIS '96, Lisbon, 1996.
[48] E. Moynihan, M. Taylor, The evolution of systems methodologies and why they are sometimes not used, Proc. BCS Information Systems Methodologies Conf, Wrexham, 1995.
[49] SSADM Management Board, Introducing SSADM 4 + , NCC/Blackwell, Oxford, 1996.
[50] R.A. Hirschheim, Assessing participatory systems design: some conclusions from an exploratory study, Information and Management 6 (1983) 317–327.
[51] J. Martin, Rapid Application Development, Prentice-Hall, Englewood Cliffs, NJ, 1992.
[52] H. Drummond, Escalation in organisational decision-making: a case of recruiting an incompetent employee, Journal of Behavioural Decision-Making 7 (1994) 43–55.
[53] M. Newman, R. Sabherwal, Determinants of commitment to information systems development: a longitudinal investigation, MIS Quarterly March, 1996, pp. 23–54.
[54] B.W. Boehm (ed.), Software Risk Management, IEEE Computer Society Press, Washington, DC, 1989.
[55] L. Wilcocks, H. Margetts, Risk assessment and information systems, European Journal of Information Systems 3 (2) (1994) 127–138.
[56] J.I. Cash, F.W. Mcfarlan, J.L. McKeney, Corporate Information Systems Management, 3rd ed., Richard Irvin, Homewood, IL, 1992.
[57] T.J. Pinch, How do we treat uncertainty in systems failure: the case of the space shuttle challenger, in: T.R. La Porte (Ed.), Social Responses to Large Technical Systems: Control or Anticipation, Kluwer Academic, Dortrecht, 1989.
[58] C. Walker, Ambulance System to Finally go Live in April, Computer Weekly, 9 January 1995.