Software Engineering

Computer Science Tripos 1B
Michaelmas 2011

Richard Clayton

Lecture One

Aims

- Introduce software engineering, and in particular:
  - the problems of building large systems
  - the problems of building safety-critical systems
  - the problems of building real-time systems
- What goes wrong will be illustrated with case histories
- We will study software engineering practices as a guide to how mistakes can be avoided

Objectives

- At the end of the course you should know how writing programs with tough assurance targets, or in large teams, or both, differs from the programming exercises done so far.
- You should appreciate the waterfall, spiral and evolutionary models of development and be able to explain which kinds of software development might profitably use them.
- You should appreciate the value of tools and the difference between incidental and intrinsic complexity
- You should understand the basic economics of the software development lifecycle.
- You should also be prepared for the organizational aspects of your Part 1B group project, for your part 2 project, and for courses in systems, security etc.

Resources

- Recommended reading:
  - S Maguire, Debugging the Development Process
  - N Leveson, Safeware (also her System Safety Engineering online)
  - SW Thames Regional Health Authority, Report of the Inquiry into the London Ambulance Service
  - RS Pressman, Software Engineering
  - the Usenet newsgroup comp.risks (also at http://www.risks.org)
- Additional reading:
  - FP Brooks, The Mythical Man Month
  - J Reason, The Human Contribution
  - P Neumann, Computer Related Risks
  - R Anderson, Security Engineering 2e, ch 25–26, or 1e ch 22–23
- Read more than this! Whichever application areas interest you.

Outline of this course

1. Large systems and the ‘Software Crisis’
2. How to organise software development
3. Failures and critical software
4. Human factors
5. People and tools
6. Guest lecture on current industrial practice (Tue 25 October)

The ‘software crisis’

- Software lags far behind the hardware’s potential!
- Many large projects fail in that they’re late, over budget, don’t work well, or are abandoned (LAS, CAPSA, NPfIT, …)
- Some failures cost lives (Therac 25) or cause large material losses (Arianne 5)
- Some cause expensive scares (Y2K, Pentium)
- Some combine the above …
- … London Ambulance Service (LAS), 1992
  - commonly cited example of project failure because it was thoroughly documented at the time (and the pattern has been frequently repeated)
  - attempt to automate ambulance dispatch failed conspicuously with London being left without effective service for a day.
  - hard to say how exactly how many deaths could have been avoided; estimates ran as high as 20.
  - led to CEO being sacked and public outrage.

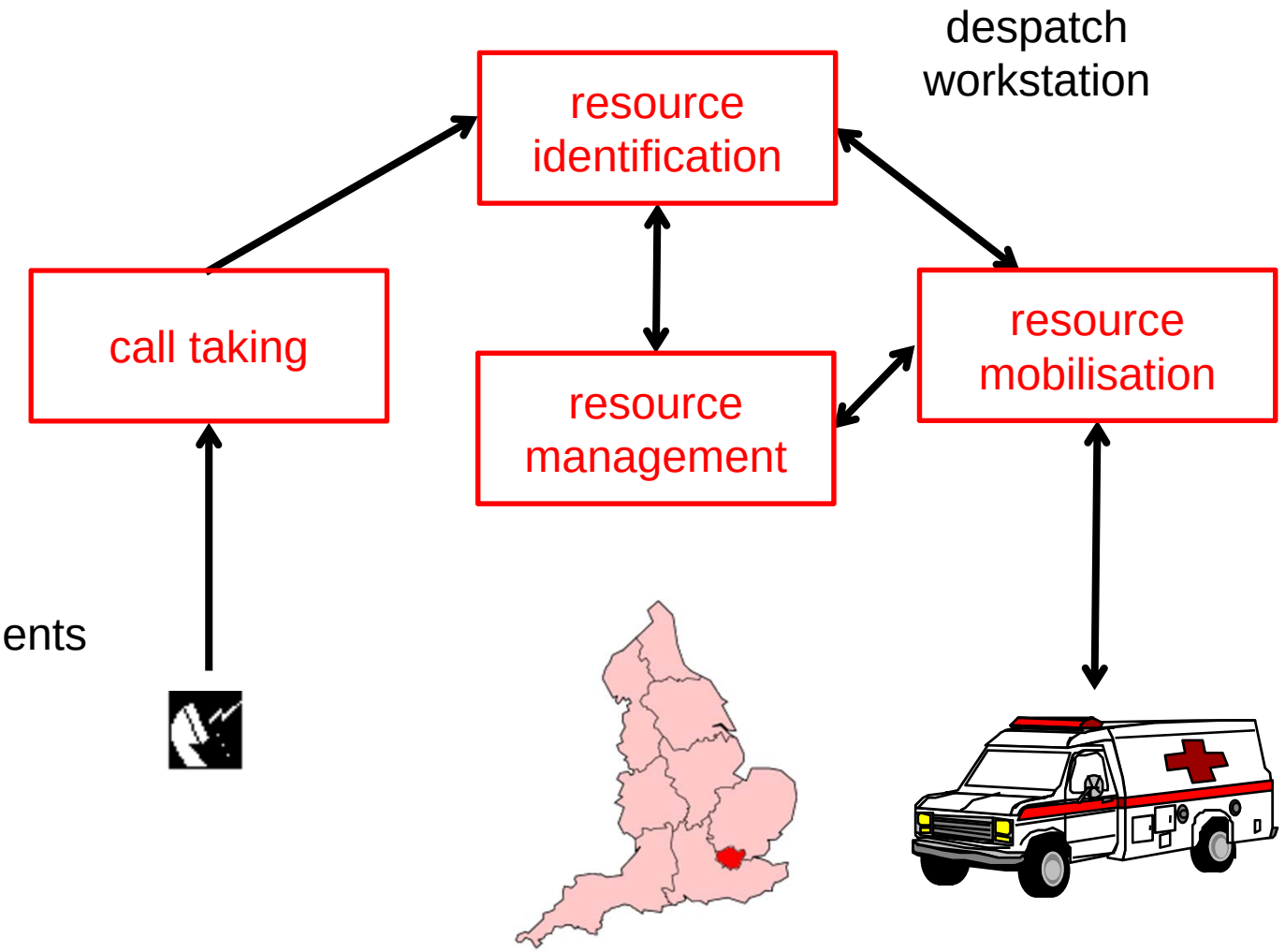
London ambulance service manual system

- Service receives 2,000 to 3,000 calls a day (1992 data)
  - around 2/3 of these are emergency (999) calls
- Details of the 999 and urgent calls written onto paper tickets
  - map reference is looked up in a street atlas
  - tickets passed by conveyor belt to central point
  - controller de-duplicates tickets and passes them on to three divisions – NW / NE / S
  - division controller identifies vehicle to be given the task and puts note in its activation box
  - phone call is made to ambulance station; or radio call made to an ambulance if it is not at it's home base
- This all takes about 3 minutes and 200 staff of 2700 total.
  - some errors occur (esp. de-duplication), some queues arise (esp. for radio traffic), and call-backs (public asking where the ambulance has got to) are tiresome

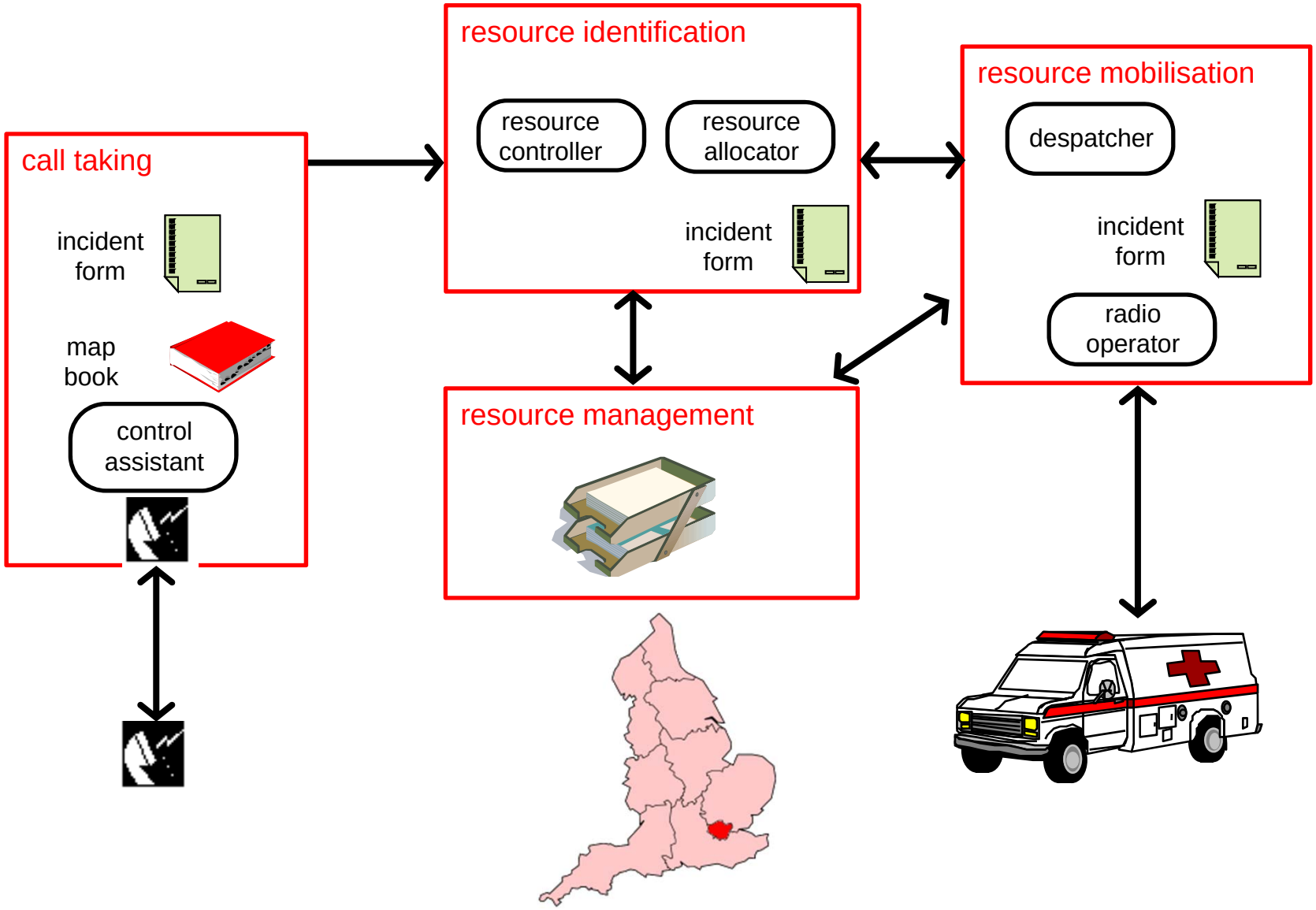
The manual dispatch system

PROPERTIES:

- Large
- Real-time
- Critical
- Data rich
- Embedded
- Distributed
- Mobile components



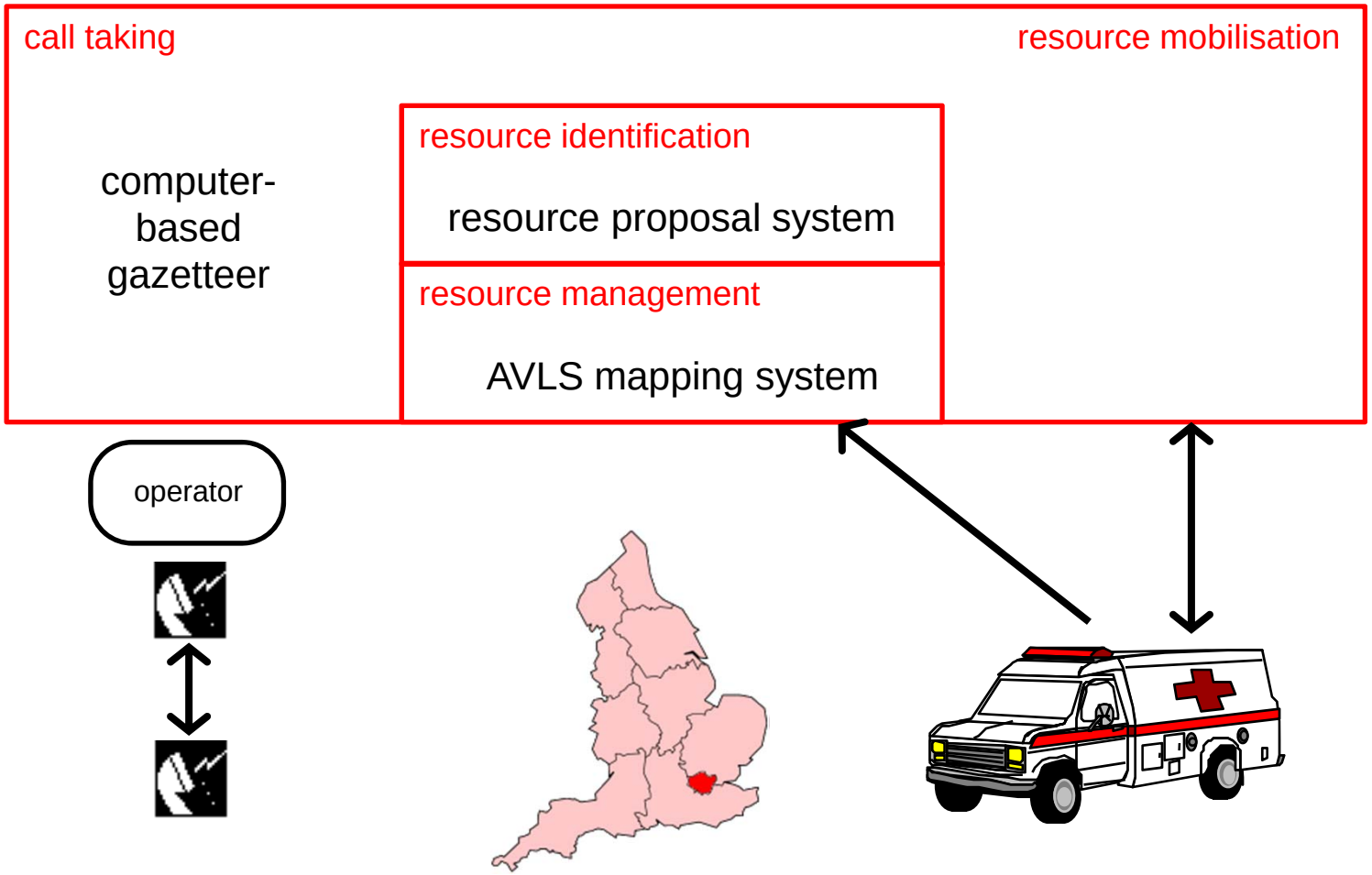
The manual implementation



Project context

- Industrial relations poor; service under pressure to cut costs; public concern over service quality
  - attempt to automate in 1980s failed – system failed load test
- LAS decided on fully automated system
  - call taker to fill in form on screen and ambulance would be emailed
  - consultants said this might cost £1.9m and take 19 months – if a packaged solution could be found. AVLS would be extra.
  - idea of a £1.5m system stuck; idea of AVLS (Automatic Vehicle Location System) added; proviso of a packaged solution forgotten; new IS director hired; timescales perceived of as inviolate
- Tender sent out Feb 1991 with completion deadline Jan 1992
  - 35 firms looked at tender; 19 proposed; most said timescale unrealistic, only partial automation possible by deadline
  - tender awarded to consortium of Systems Options Ltd, Apricot and Datatrak for £937,463 – £700K cheaper than next bidder

The goal


call taking
resource mobilisation
computer-based gazetteer
resource identification
resource proposal system
resource management
AVLS mapping system
operator

Project timeline I

- Design work ‘done’ July 1991; main contract signed in August
- LAS told in December that only partial automation possible by January deadline – front end for call taking, gazetteer, with printers added to create tickets to feed into manual system
  - progress meeting in June had already minuted a 6 month timescale for an 18 month project, a lack of methodology, no full-time LAS user representative, and SO’s reliance on ‘cozy assurances’ from subcontractors
- Server never stable in 1992; client and server lockups
- Phase 2 introduced radio messaging – blackspots, channel overload, inability to cope with ‘established working practices’
- Yet management decided to go live 26 Oct 1992
  - CEO: “No evidence to suggest that the full system software, when commissioned, will not prove reliable”

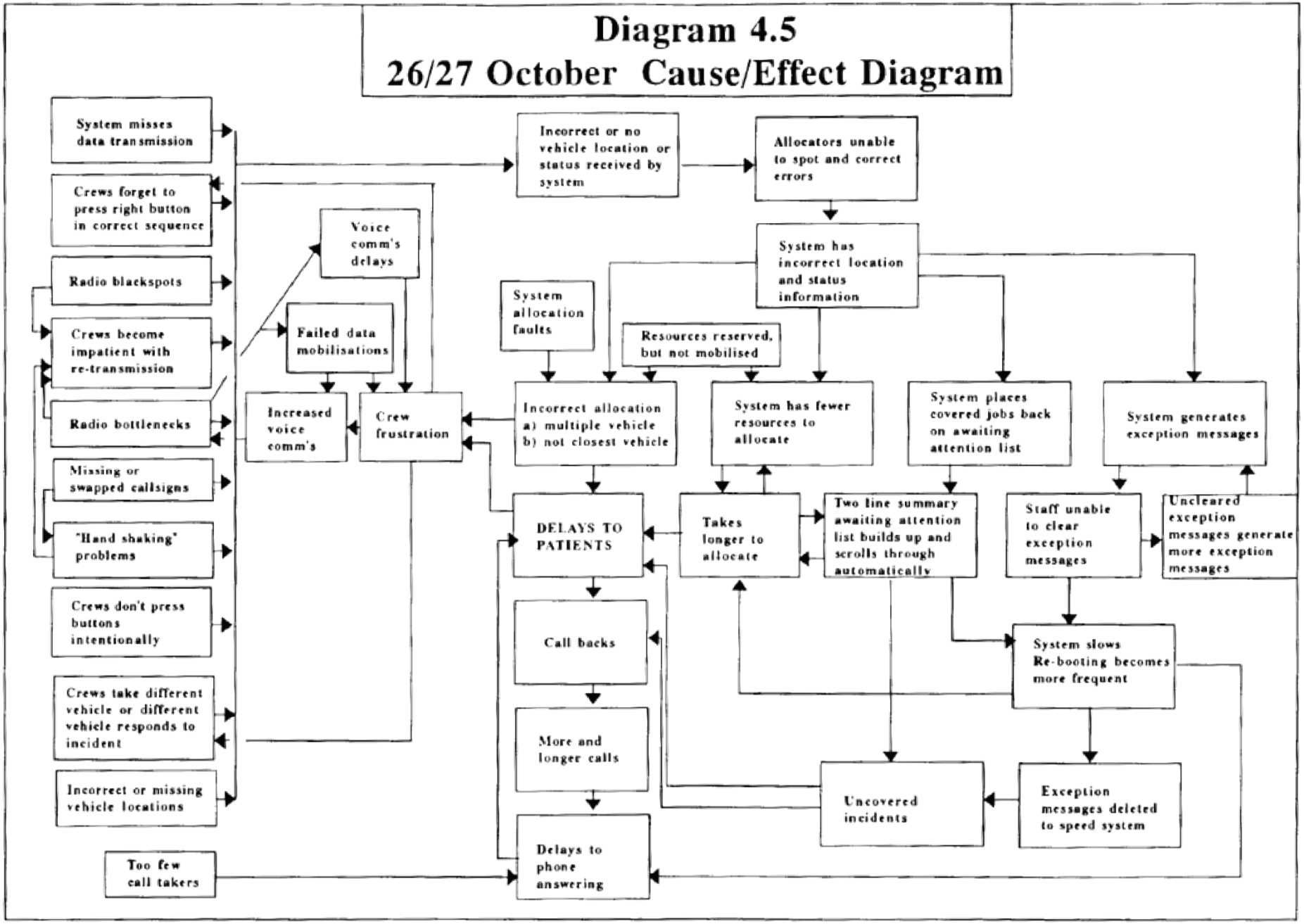
Project timeline II

- Independent review had called for volume testing, implementation strategy, change control … It was ignored!
  - On 26 Oct, the call handling room was reconfigured to use terminals, not paper. There was no backup …
- 26/27 October vicious circle:
  - system made incorrect allocations or sent multiple vehicles
  - manual intervention corrected problems while system lightly loaded
  - system progressively lost track of vehicles and allocations became even less appropriate, swamping ability of humans to correct
  - exception messages scrolled up off screen and were lost
  - incidents held as allocators searched for vehicles
  - callbacks from patients increased; data delays; radio congestion;
  - crew frustration led to errors in pressing buttons to show status
  - crews were taking ‘wrong’ vehicles or `wrong’ vehicle responded to a call so that system status was even less correct

Project timeline III

- Entire system descended into chaos:
  - e.g., one ambulance arrived to find the patient dead and taken away by undertakers
  - e.g., another answered a ‘stroke’ call after 11 hours, 5 hours after the patient had made their own way to hospital
- Some people probably died as a result. Estimates at the time were from 10-20, albeit no coroner’s inquest held LAS to blame
- Switched back to semi-manual operation on 27 October
- John Wilby (CEO) resigned on 28 October (& inquiry set up)
- System slowed and stopped altogether at 2am on Nov 2
  - memory leak (poor QA on a change made to deal with the printers)
  - fallback to second system did not happen (it had never been tested). Staff tried switching off and on again but system failed to run. Management decided to return to full manual system.

Diagram 4.5
26/27 October Cause/Effect Diagram



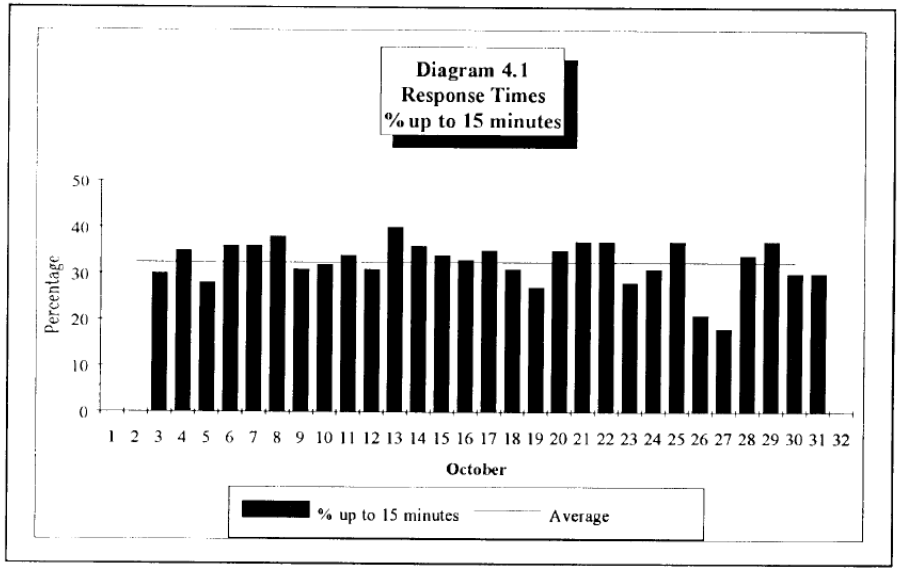


Diagram 4.1
Response Times
% up to 15 minutes
Percentage
50
40
30
20
10
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
October
% up to 15 minutes
Average

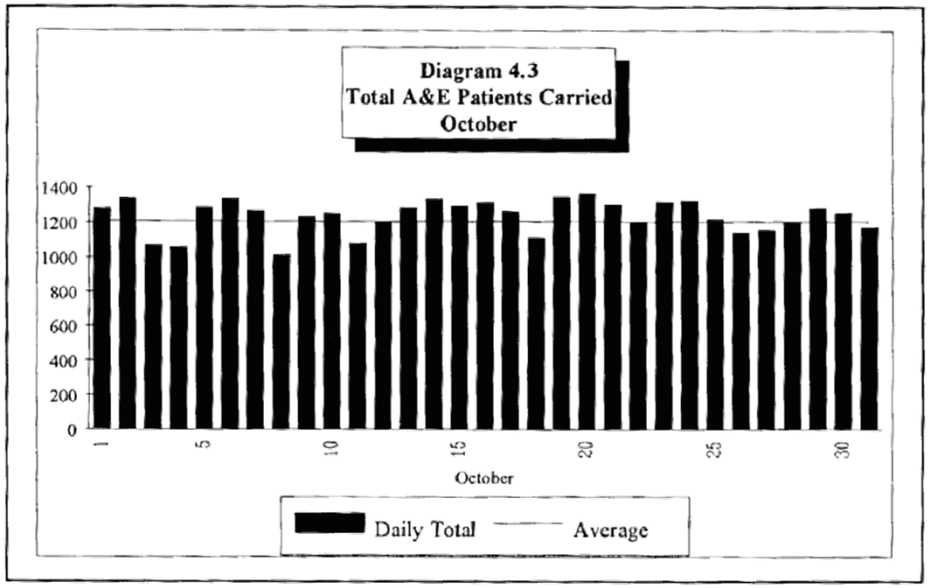


Diagram 4.3
Total A&E Patients Carried
October
1400
1200
1000
800
600
400
200
0
1
5
10
15
20
25
30
October
Daily Total
Average

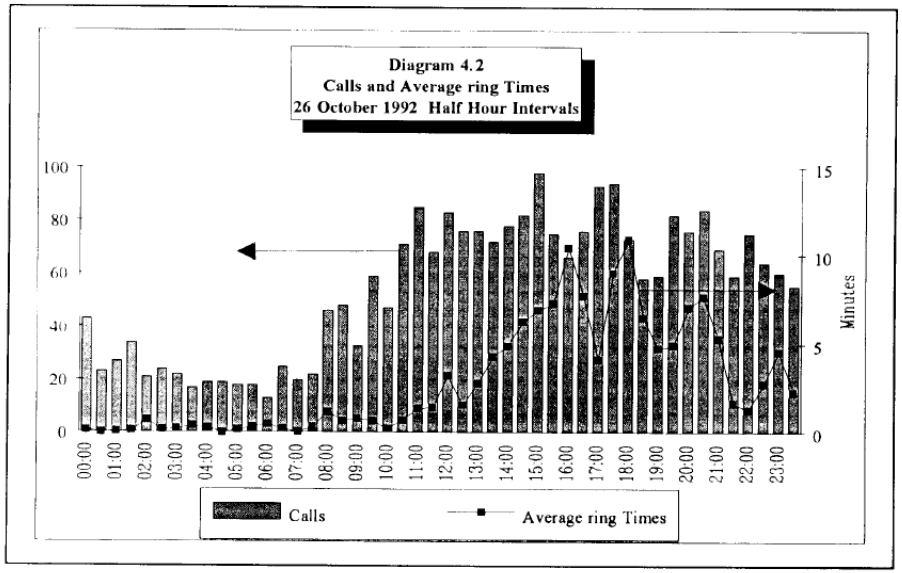


Diagram 4.2
Calls and Average ring Times
26 October 1992 Half Hour Intervals
100
80
60
40
20
0
15
10
5
0
Minutes
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Calls
Average ring Times

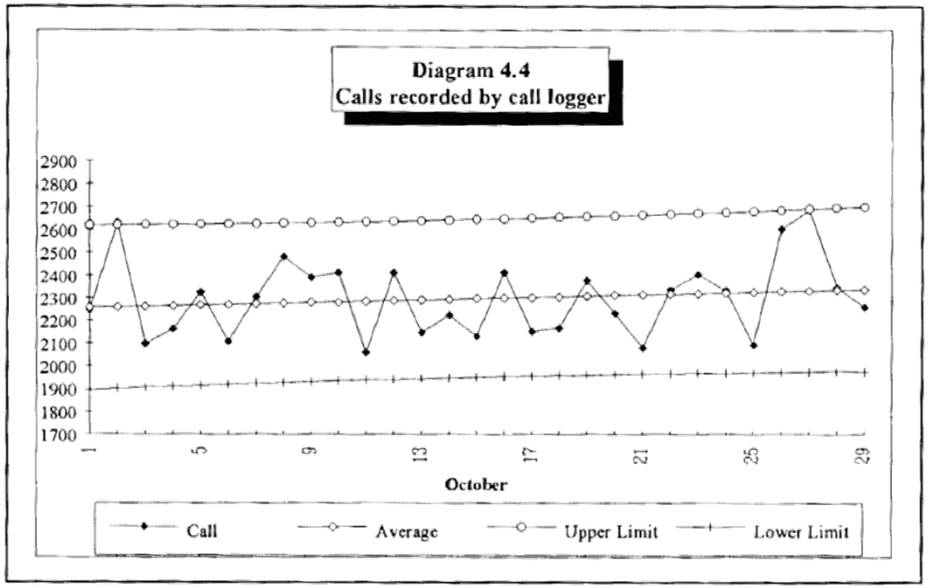


Diagram 4.4
Calls recorded by call logger
2900
2800
2700
2600
2500
2400
2300
2200
2100
2000
1900
1800
1700
1
5
9
13
17
21
25
29
October
Call
Average
Upper Limit
Lower Limit

What went wrong – specification

- “They made virtually every mistake in the book”
  - Paul Williams LAS Inquiry Member
- LAS ignored advice on cost and timescale
- No ‘systems’ view
- Specification was inflexible but incomplete: it was drawn up without adequate consultation with staff
- Ignored established work practices and staff skills
- Timescale was unrealistic and portrayed as inviolate
- Attempt to change organisation through technical system
  - e.g. crews deciding which resources to send to incident
  - see #3116 of the report
- Procurement team insufficiently qualified and experienced

What went wrong – project

- Inexperienced software house (arguably they breached ethical standards in doing the project at all)
- Confusion over who was managing it all
- No professional project manager
- Poor change control, no independent QA function, some suppliers misled on their progress
- Inadequate software development tools
- Flaws in communications kit – effects of which not foreseen
- Poor interface for ambulance crews
- Poor control room interface

What went wrong – go live

- System commissioned with known serious faults
- Still had slow response times and workstation lockups
- Software not tested under realistic loads or as a complete integrated system
- Inadequate staff training (and some done too early)
- System failures meant it was not trusted by the staff
- No back up plan (and no tested back up system)
- In period up to 26 October frustrated crews had used voice more. Instructions on 26 Oct were to minimise voice traffic, but this meant that mistakes were not corrected.
- Too few call takers (especially once callbacks escalated)

Things are still far from perfect

- LAS was a long time ago… but disasters still occur
- CAPSA (mid 1990s)
  - Cambridge wanted ‘commitment accounting’ for research grants
  - Oracle system implemented but could not cope with the volume
  - review concluded it was an “unmitigated” failure
  - 2011: Research grant accounting systems still a mess (and slow)
- NHS National Program for IT: NPfIT (2002…)
  - £11.4 billion (maybe more!)
  - much delivered late or yet to be made to work
  - goals changing throughout
  - “worlds biggest software project, and the world’s biggest disaster”
- Smart meters
  - £10 billion, and now to be delivered earlier than 2017…