Software Engineering Theory and Practice

School of Computing	
Title	Software Engineering Theory and Practice
Module Coordinator	Steven Ossont
Email	steven.ossont@port.ac.uk
Code	M30819
Moodle	https://moodle.port.ac.uk/course/view.php?id=11429

U30819: Software Engineering Theory and Practice

London Ambulance Service (System Failure)

https://moodle.port.ac.uk/course/view.php?id=11429

Steven Ossont steven.ossont@port.ac.uk

The London Ambulance Service (1992)

- Services ~6.8 million people
- 318 emergency ambulances,
- 212 in service at all times
- 70 ambulance stations
- ~ 2700 staff members

The London Ambulance Service

- 2000 – 2500 calls received daily
  - About 100 calls/hour or 1.7 calls per minute
- 60% request emergency services
- 75% of budget dedicated to emergency response

Mid 1980's

- Entire system managed from a central location in Waterloo
- LAS emergency dispatch system was run completely manually:
  - Call taking
  - Resource identification
  - Resource mobilization

Example (Manual processing)

1. Call requesting emergency ambulance service is received
2. The call taker fills in the form with the caller’s details, and passes to form to someone else
3. A second LAS employee analyses the location of the ambulances in the caller’s region
4. The call is assigned to an available ambulance and the form is updated with the id of the ambulance
5. A dispatcher contacts the ambulance to which the call was assigned and provides the ambulance crew with the details of the call

Manual processing

The manual process has some issues

- Time consuming
- Error prone
  - Location identification
  - Physically moving paper forms
  - Updating vehicle status

Fast forward to the 1990's

- The government stipulated that calls be responded to within three minutes
- LAS management seeks-out a computer-based alternative

The Development Process

- Step 1: LAS looks at adopting an existing system, but no acceptable option available
- Step 2: LAS decides to go for developing a new system from scratch
- Step 3: Requirements elicitation without any input from ambulance crews or dispatchers
- Step 4: Completely computerized system, nearly everything automatic

Step 5: Architectural design

- Automatic Vehicle Location system (AVL)
  - Radio location updates (~ every 13 seconds)
- Mobile Data Terminals (MDT)
  - Inside emergency vehicles to facilitate communication
  - Crew can confirm they are on route
  - Crew must confirm messages (or HQ are notified)
- Event based system with rules and Geographical Information System (GIS)

Step 6: Behavioral design

- A person answers the phone, enter incident data into a terminal
- A person responds only if the system does not find any ambulances available within 11 minutes
- The locations of all calls are automatically mapped by the software
- The system finds and dispatches the available ambulance closest to the caller’s location

The Development Process

- Step 7: Project cancelled
- Step 8: Project re-designed
- Step 9: Project deployed on October 26, 1992

Release style

Big bang release

- Turn off existing system , turn on new system

First Day of Use

- AVLS was unable to keep track of the ambulances and their status in the system
  - Radio black spots
- Sent multiple units to some locations and no units to others
- Closest ambulances were not always selected
  - Resource starvation
- The efficiency of the system was substandard
- A large number of exception messages were generated on the dispatcher’s terminals
- Calls got lost
- People called back multiple times
  - The system became clogged

Key problems

- Inability of the software to distinguish between duplicate calls from different people pertaining to the same incident
- Failure of the software to maintain and keep track of logged calls
- Crew got very frustrated and pressed incorrect buttons or swapped vehicles / incidents

> This generated more calls, swamping the call handlers

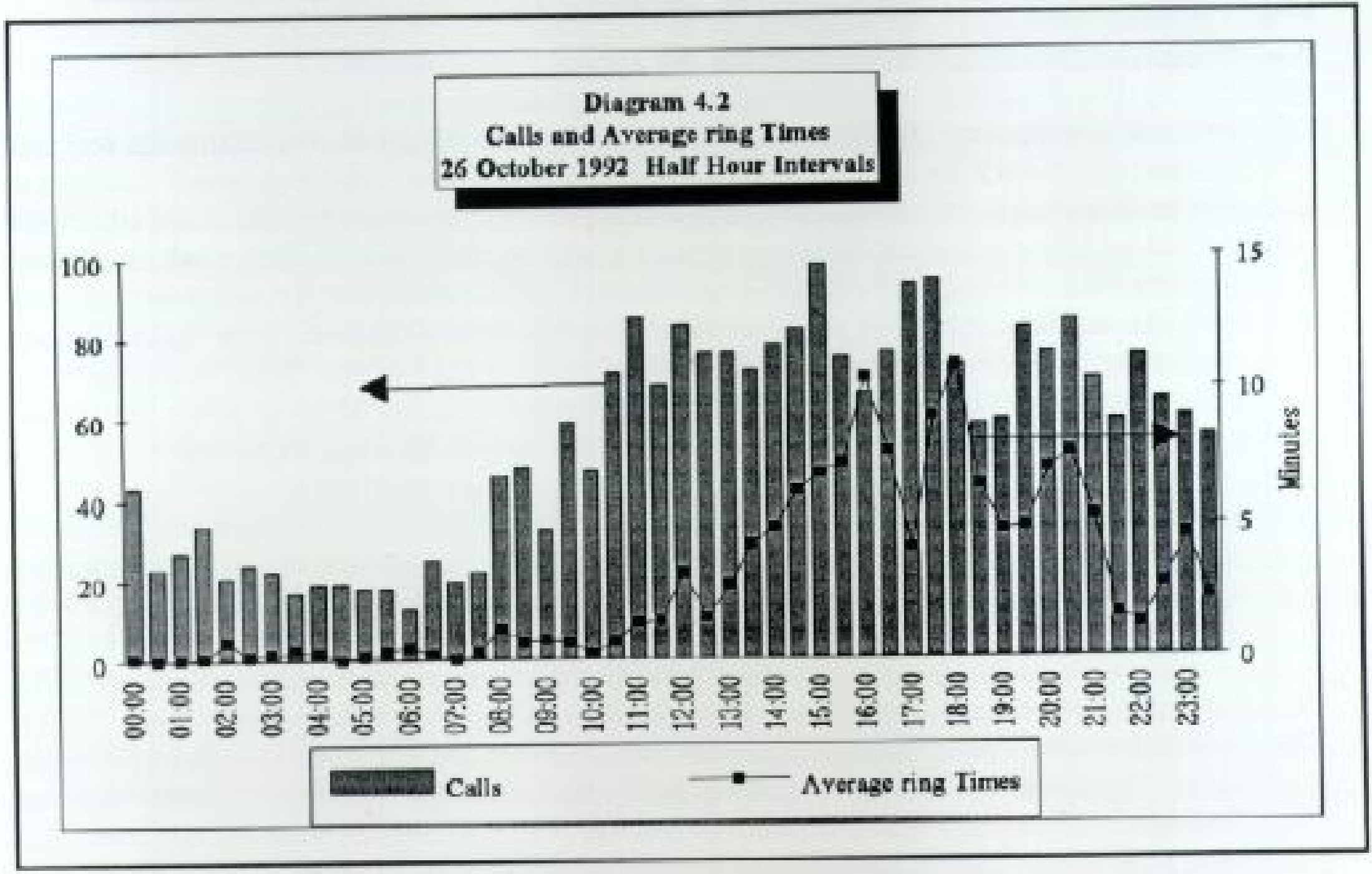
Diagram 4.2
Calls and Average ring Times
26 October 1992 Half Hour Intervals
100
80
60
40
20
0
15
10
5
0
Minutes
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Calls
Average ring Times

After the first day

- LAS switched back to a part-manual system

After 8 Days

- The system stopped working altogether
- LAS shut it down completely

The Cost

46 deaths

- 1 heart attack patient waited for 6h for an ambulance
- 1 lady called LAS every 30 minutes for 3 hours before an ambulance arrived.
  - It was too late

Immediate Causes

- System went live with 81 known issues and no load-tests run
- No provisions for a backup system
- 10 months between dispatchers being trained and system being deployed
- System did not work with incorrect or incomplete data on the ambulances
- Undo operation not available on MDTs
  - Ambulance staff not able to correct wrong inputs
- Memory leak in a small portion of the code
  - calls resurfaced long after beign dealt with
- Parts of the MDT screens had black spots
  - Ambulance staff could not read all data on the screen (Scroll issue)

Deeper Causes

- 1987 – original design proposed
- 1989 – designed modified
- 1990 – project cancelled due to 300% overspending
- 1992 – national mandate to reduce emergency response times

Hardware Causes

- Hardware used on the failed project was reused in the development of the newer version
- No research in more up-to-date and suitable hardware

Vendor Selection

“a manager expecting to become redundant and a contractor who was a temporary addition to the organization”

- All bids for longer than 11 months not considered
- All bids greater than £1.5 million not considered
  - Previous system failed after a £7.5 million investment

Vendor Selection

- LAS accepts £1m bid from a conglomeration of companies

Software Development Company

- Never worked on a large product
- Had no experience with “real-time, *safety* critical, command and control systems”
- Concerns brought up by selection process audit ignored

Software Development Process

- No key users of the system were consulted during the requirement elicitation phase
- Software Requirements Spec. described the **how** not the **what**
- No sign-off on the design specification
- Design changed once implementation began

Lifecycle Model

- No quality assurance
- Configuration management absent
- Agreed-upon changes not tracked
- No test plans documented

Who is Responsible for the Failure?

- *System Options*?
  - Management?
  - Developers?
  - Testers?
  - Project management?
  - Requirements analysts?
  - Designers?

Who is Responsible for the Failure?

- Software developer introducing the memory leak?
- LAS for imposing an 11 month development time constraint?
- Vendor selection committee for going with an inexperienced software company?
- Project manager for not following the right lifecycle model?

Who is Responsible for the Failure?

- Testers for poor testing?
- QA for not ensuring software quality?
- *System Options* for accepting to do the job knowing they lack the needed experience?
- *System Options* for not pointing out and agreeing to the unrealistic time constraint?
- Requirements analysts for not eliciting the key users' requirements?
- *System Options* for not following a professional code of conduct and protecting the public?

Questions ?