CAPSA: a technical review

OR "Physician Heal Thyself"

Anthony Finkelstein
University College London
Department of Computer Science

a.finkelstein@cs.ucl.ac.uk
http://www.cs.ucl.ac.uk/staff/A.Finkelstein

UCL

Contents

- A brief preamble

- Why
- How
- What

- A technical analysis
- A research prospective

Report to the Audit Ctee & Board of Scrutiny

Personal speculation

Preamble

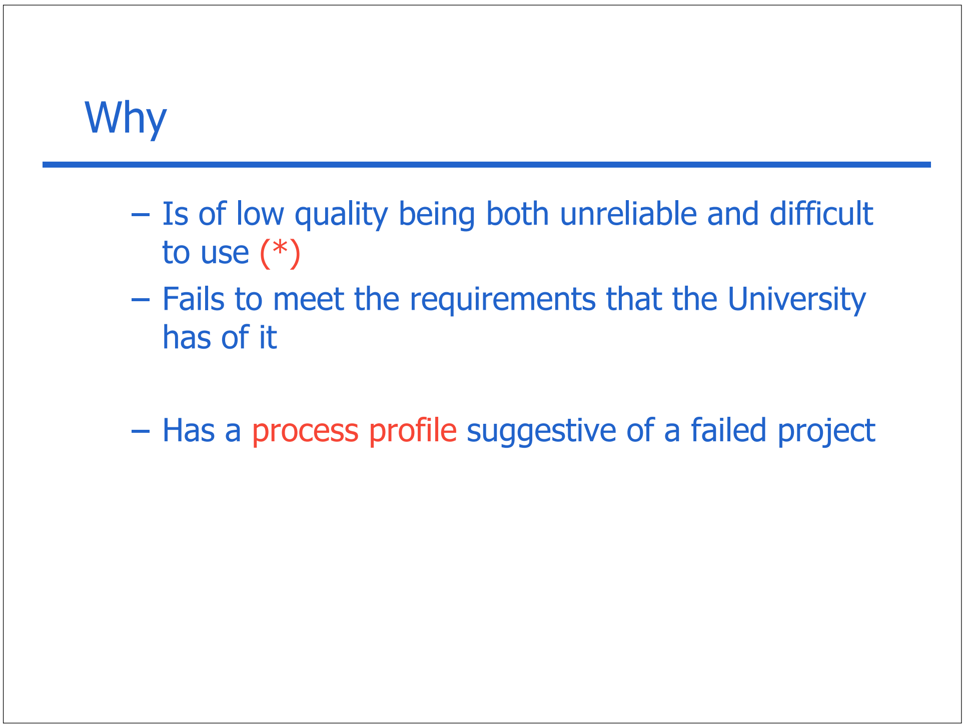
- Status of seminar …
- Scope of comments …
- Why I wanted to give this talk …
- What I would rather not talk about …

- I will be happy to talk about the recommendations in the report at the end of the seminar

{what I do the rest of the time …}

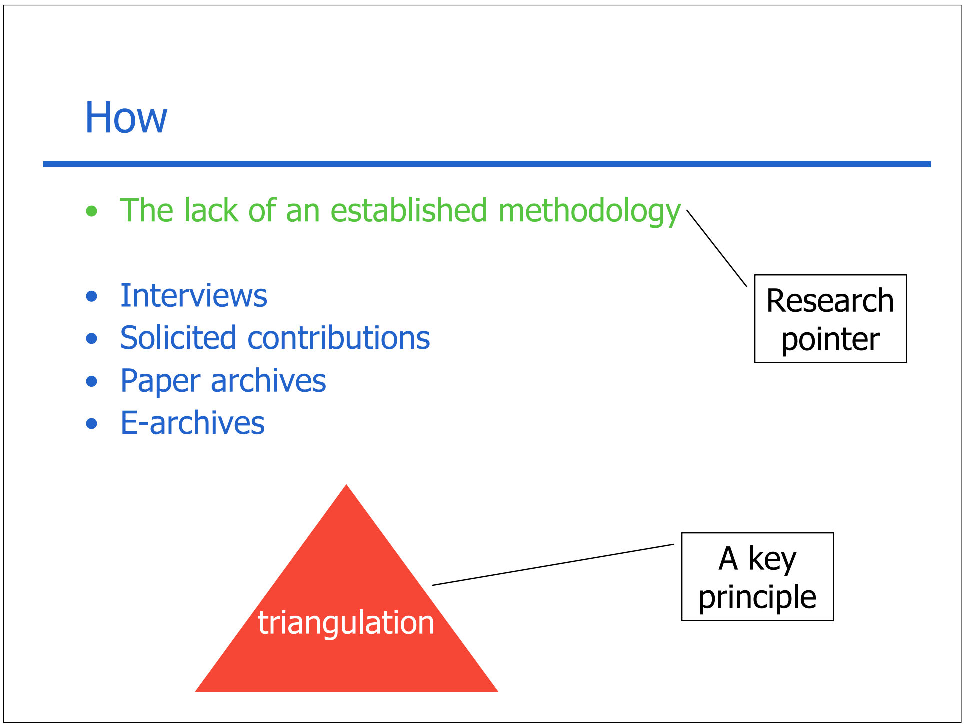
Why

- CAPSA (a convenient shorthand) a ‘classic’ system failure …
  - Significantly more costly than had been anticipated (worse than it appears because of hidden costs)
  - Substantial disruption to working of the University
  - Placed staff under undue pressure
  - Placed the finance of the University at risk and may have prevented the University and its staff from fulfilling their legal responsibilities



Why

- Is of low quality being both unreliable and difficult to use (*)
- Fails to meet the requirements that the University has of it
- Has a process profile suggestive of a failed project



How

- The lack of an established methodology

- Interviews
- Solicited contributions
- Paper archives
- E-archives

Some Intermediate Observations

- ‘The systems perspective’
  - Consequences for development processes
- ‘Systems fail systemically’
  - Consequences for analysis
- ‘System ownership’
  - Consequences for recommendations

What

< a narrative is inserted here>

if you want to follow it in more detail
READ THE REPORT

Excluding ‘managerial’ issues

Technical Analysis

Research triage

- ‘No brainers’
  - We know how to do this, basic good practice, no professional should make this mistake
- Tricky
  - Experts can do this, requires some experience and judgement, a reasonable expectation of a skilled professional
- Interesting
  - We do not know how to do this, it causes problems in other settings, there is no treatment of this problem in the literature

No Brainers

- Configuring an appropriate process
  - In almost all situations based on incremental delivery
  - Establishing clear pre and post conditions for each stage and managing to them
- Establishing and maintaining proper project management controls in order to monitor resource use wrt that process
- Explicitly managing project risks

No Brainers

- Producing and maintaining clear documentation giving views of the system at all levels
- Ensuring document traceability
- Ensuring and maintaining a project library
- Policing the ‘quality interface’ with suppliers
  - Dealing with vapour-ware
- Establishing a testing programme and ensuring the integrity of that programme
- Maintaining configuration control with disciplined issuing and application of patches

No Brainers

- Using a method in an ‘intelligent’ manner
- Planning for training, handover and operation

  …

Tricky

- Knowing when to ‘evolve’ a system
- Establishing the ‘right’ system architecture
- Interplay of requirements and architecture (NFRs)
- Identifying and recruiting/engaging ‘competent’ computing professionals
- Costing and estimation
  - Yes, it is still a big problem!
- Identifying (mapping) stakeholders
- Managing the engagement of ‘end-users’ in the development process

Research

- Package acquisition
  - Specification
  - Search

    What is wanted and what is possible
  - Matching
  - The ‘balance of mutability’
  - Workarounds and tailoring

- Intertwining business process reengineering and software engineering

Conclusions (sort of)

- Lets not be too quick (as CS researchers) to mark some things as ‘done and dusted’. As a profession we are failing too often. Many core pieces of enterprise software are of amazingly poor quality.
- We need to review much established software engineering practice in the context of the widespread use of commercial-off-the-shelf software.
- Effective in a real setting system development, project management and strategic ‘business’ management are woven together and need to be treated holistically

Conclusions (sort of)

- Yes, putting big ERP systems in place is difficult
- Yes, lots of people make the same mistakes as Cambridge did in the CAPSA project
- But following known good practice would have made the ride more comfortable and made the fall less painful …