

# Software Failures

## Software Architecture, Process, and Management

Describing a litany of software project failures, trying to tease out the sources of failure. Ultimately the same reasons crop up again and again, but there are many of them, ranging from social/organisational through unmanaged technical risk, poor design and implementation to bad delivery, rollout and support. Awareness of some these possibilities might, one hopes, mitigate their likelihood in future.

# Confirmation Bias

- A study, carried out in the USA, and actually seeking to investigate perceived similarity, concerned two pairs of countries:
  1. East Germany and West Germany
  2. Sri Lanka and Nepal
- The study asked two groups of participants respectively:
  1. Which two countries are more **similar** to one another?
  2. Which two countries are more **different** to one another?
- In the first group the majority said that East and West Germany were more **similar** to each other
- In the second group the majority likewise said that East and West Germany were more **different** from each other.

# NHS Connecting for Health

- “Originally expected to cost 2.3bn GBP over three years, in 2006 the total cost was estimated ... to be 12.4bn over 10 years...”
- “Officials ... estimate[d] the final cost to be as high as 20bn, indicating a cost overrun of 440% to 770%...”
- “This is the biggest IT project in the world and it is turning into the biggest disaster.” ... despite a probable expenditure of 20 billion pounds “it is unlikely that significant clinical benefits will be delivered...”

# Claims of Catastrophic Failure

- You should of course be skeptical of anything that claims pound;12bn has been wasted
- Other sources were more sober, it seems that there is a move to decentralisation allowing for local NHS trusts to make appropriate local decisions rather than imposing a top-down system
- However, the announcement was described as “an acceleration of the dismantling of the National Programme for IT”
- Sir David Nicholson “The NPfIT has provided us with a foundation but we now need to move on if we are going to achieve the efficiency and effectiveness required in today’s health service”
- So whilst the entire pound;12bn cannot be written off as entirely wasted it is also quite clear that the project, by our criteria, is an abject failure

# Why Software Projects Fail

- Software appears, by its nature, to be difficult to engineer on a large scale
- Nevertheless, there is an insatiable demand for sizeable, well-engineered software
- We continue to be dogged by large numbers of project failures, on small and large projects

# Why Software Projects Fail

- Many (most?) of these are due to mistakes in project management
- We include going over budget and past scheduled delivery date in our definition of failure
  - However we have seen in previous lectures that this is often due to an inability to estimate the cost and time required for a project in the first place
  - So here we will mostly be discussing projects which fail to deliver quality software
  - Most times the entire system is scrapped with no discernible project deliverable outcomes

# FBI: Virtual Case File

- The US Federal Bureau of Investigation has often been criticised for not sharing leads between agents and divisions
- Just before the 2001 terrorist attacks, the FBI hired Science Applications International Corp (SAIC) to develop Virtual Case File software (VCF)
- VCF was designed to manage case files electronically, so that any agent with suitable permissions can find relevant information
- Originally scheduled for completion in 2003

# FBI: Virtual Case File

- After repeated delays, a version was delivered in December 2004, but:
  - Was about one tenth of originally promised
  - Was eventually scrapped altogether
  - Does not approach functionality of existing commercial packages
  - Used as an extremely expensive prototype
  - About dollar;170 million wasted



# FBI: Virtual Case File

- Apparent causes:
  - Changing requirements (after the September 11 attacks)
  - Ambitious project, run as an emergency fix
  - 14 different managers over project lifetime
  - Poor oversight of external contractor
  - Not paying attention to new, better commercial products
  - Hardware purchased already; waiting on software

# Supply Chain Management

- **Background:** Supply chain management crucial in price war between Sainsbury's and Tesco
- **pre-2000:** Sainsbury's had "mainframe-based warehouse management system"; "400 different supply chain software applications"
- **2000:** new CEO Peter Davis authorised outsourcing IT to Accenture, aiming to get an "agile IT infrastructure built on an open, adaptive, scalable architecture with hardware and software systems that would give very high performance, strong data security, and low total cost of ownership." - Key supplier Sun.

# Supply Chain Management

- **May 2004:**
  - “The dollar;1.8 billion overhaul is well under way, and will be completed in 2005.”
  - “The relationship with Accenture has worked so well that Sainsbury’s has chosen to extend its IT outsourcing contract for another three years, until 2010 - a move that should allow the retailer to net additional cost reductions of more than dollar;230 million by 2007.”
- **July 2004:** Davis resigns - poor financial performance

# Supply Chain Management

- **October 2004:**
  - New system unable to track stock correctly
  - Shops go short
  - Sainsbury's recruits 3000 shelf stackers to handle crisis, writes off pound;260m IT spend, renegotiates contract with Accenture
  - Accenture blames poor reliability of four fully automated depots, not covered by their agreement
  - new Sainsbury's CEO Justin King blames Accenture
- **October 2005:** Outsourcing cancelled, IT brought back in house

# Supply Chain Management

- Apparent causes of problems with the Accenture attempt:
  - Weak outsourcing governance
  - Loss of staff with knowledge about legacy systems
  - Risky “big bang” approach
  - Political in-fighting
  - Generally poor business management

Main source: Douglas Hayward in a 2005 silicon.com article

# Customer Database System

- In 1996 a US consumer group embarked on an 18-month, dollar;1 million project to replace its customer database
- The new system was delivered on time but didn't work as promised, handling routine transactions smoothly but tripping over more complex ones
- Within three weeks the database was shut down, transactions were processed by hand and a new team was brought in to rebuild the system

# Customer Database System

- Problems:
  - The design team was over-optimistic in agreeing to requirements
  - Developers became fixated on deadlines, ignoring errors

# Customer Tracking System

- In 1996 a San Francisco bank was poised to roll out an application for tracking customer calls
- Reports provided by the new system would be going directly to the president of the bank and board of directors
- An initial product demo seemed sluggish, but telephone banking division managers were assured by the designers that all was well
- But the system crashed constantly, could not support multiple users at once and did not meet the bank's security requirements
- After three months the project was killed; resulting in a loss of approximately dollar;200,000 in staff time and consulting fees



# Customer Tracking System

- Problems:
  - The bank failed to check the quality of its contractors
  - Complicated reporting structure with no clear chain of command
  - Nobody “owned” the software

# Payroll System

- The night before the launch of a new payroll system in a major US health-care organisation, project managers hit problems
- During a sample run, the off-the-shelf package began producing cheques for negative amounts, for sums larger than the top executive's annual take-home pay, etc.
- Payroll was delivered on time for most employees but the incident damaged the relationship between information systems and the payroll and finance departments, and the programming manager resigned in disgrace

# Payroll System

- Problems:
  - The new system had not been tested under realistic conditions
  - Differences between old and new systems had not been explained
    - E.g. dollar;8.0 per hour was entered as dollar;800 per hour
  - “A lack of clear leadership was a problem from the beginning”

# Systems of Systems

- A key component in many of these large software failures is that of integration
- Integration appears to be more challenging than original work
- Despite these, often very visible failures, there appears to be no let up in demand for such “megasoftware”

# Critical Failure Factors

## Warning Signs

- Organisation: hostile culture, poor reporting structures
- Management: over-commitment, political pressures
- Conduct of the project:
  - Analysis and design phase:
    - design by committee
    - technical fix for management problem
    - poor consultation and procurement

# Critical Failure Factors

## Warning Signs

- Conduct of the project (cont):
  - Development phase:
    - Staff turnover
    - poor communication (e.g. split sites)
  - Implementation phase:
    - receding deadlines
    - inadequate testing
    - inadequate user training

# Reasons for Failure

- Unrealistic or unarticulated project goals
- Inaccurate estimates of needed resources
- Badly defined system requirements
- Poor reporting of the project's status
- Unmanaged risks
- Poor communication among customers, developers, and users
- Use of immature or outdated technology
- Inability to handle the project's complexity
- Sloppy development practices
- Poor project management
- Stakeholder politics
- Commercial pressures

# Most of these are not specifically developer issues

- Unrealistic or unarticulated project goals
- Inaccurate estimates of needed resources
- Badly defined system requirements
- Poor reporting of the project's status
- Unmanaged risks
- *Poor communication among customers, developers, and users*
- Use of immature technology
- ~~Inability to handle the project's complexity~~
- ~~Sloppy development practices~~
- Poor project management
- Stakeholder politics
- Commercial pressures



# Irrational Project Escalation

- This is caused when, despite all indications to the contrary a project is allowed to continue or is even “*stepped up*”
- The national audit office reported numerous cases of government IT projects proceeding despite a recommendation to halt
- The UK even has a government department whose *one job* is to prevent IT failures
  - Unfortunately, more than half of the agencies the department oversees *routinely* ignore its advice