MBA9009 - 12

Information Systems Failures
[but]
"does IT Matter?"

Strategic Significance of IT

- paper for discussion later:
    - this article caused a furore when published, particularly among IT academics
    - they didn't agree!
      - [see Stewart et al (Letters to the Editor), HBR, June 2003]

some of Carr's points

- IT is not strategic - it is
  - infrastructural
  - commoditised
  - comparatively cheap
  - replicable
  - etc.

Why Consider IT Failures?

- despite the danger of appearing negative, failure studies are often the best way to make important points
  - a set of rules for "how to manage well" only gets you part of the way towards IT management success
  - eg. no decision to purchase and implement an enterprise system is taken lightly, yet the failure rate is >50%

References

- Vaughan, D. 1996. The Challenger Launch Decision: Risky Technology, Culture and Deviance at NASA.
Failure “Statistics”

- I put the word “statistics” in scare quotes because the figures are pretty rubbery - no organizations like to publicise (or even admit to) failures if they don’t have to
  - BPR 70% failures (p322)
  - CRM 60-85% (p322)
  - Data warehouses/data mining “many” failures (p431)
  - OSS “many” failures (p434)
  - e-Commerce failures are “common” (p174)
  - ERP 60% (p110)
  - m-computing “many failures of applications as well as entire companies” (p233)


- even allowing for the roughness of the statistics, these are not the most impressive set of numbers going round,…

Textbook Coverage

- It is irritating that textbooks do not provide any systematic treatment of IT failures, given that they often outnumber the successes
  - in fact there appear to be some common factors at work in many cases of failure – it is notable that
    - technical difficulties are rarely insuperable, and so the technology is almost never the primary cause of a failure
    - the root causes of failure are generally management errors and defects of knowledge and understanding

A Foundational Issue

- there is a widespread tendency to take a “rational” perspective on IT (to see it as the means to a predefined end)
  - but at the time an IT-based project/strategy is initiated, the only available perspective is (usually) very high-level
    - it is not so much that the difficulties at the implementation level are underestimated
    - there is often no understanding that they are possible
  - planning discussions are in terms of “have we got a solution for you” and not “what is the problem you are trying to solve”

?? IT Failures

- thus there are some very positive reasons for analysing failures
  - 1 IT failures are disturbingly frequent (see earlier overhead)
  - 2 it is [relatively] straightforward to describe good management practice but…if that was all it took, everybody would do it!
  - 3 there is reason to believe that the seeds of failure are present in most IT-based business initiatives
  - 4 once an activity is over, it is both easy and natural to rewrite history and forget the errors - failures tend to be the only cases where some genuine soul-searching takes place

An illustrative example

- the London Ambulance Service LASCAD system

From Failure to Disaster

- it is rare for an IT failure to turn into a disaster
  - but some do (“Normal Accident Theory” - Perrow)
    - experiencing a certain number of catastrophic accidents is “normal” as systems become more complex and interrelated
    - every change has some potential to cause unforeseen problems
    - the causes of accidents are often obvious in hindsight but…
    - a disaster is hard to foresee because “it usually involves failures in several parts of a system which are linked in complex and subtle ways” (Perrow)
The “LASCAD” Disaster

- an attempt was made at the inquiry (and the claim was repeated on a film made about it) to blame the disaster on a technical failure - in effect a queue became full and could not be cleared
  - it can be understood as typical of the tendency to look for technical causes of disasters
  - in this case, an inquiry revealed quite a different picture

London Ambulance Service (1)

- anatomy of a disaster
  - the system was lightly loaded at start-up
  - as the # of ambulance incidents increased, the amount of incorrect vehicle information recorded also increased
  - the system then made incorrect allocations (a “knock-on” effect) in which the closest vehicle wasn’t sent to an incident or multiple vehicles were sent
  - ambulance crews also took the “wrong” ambulance to some scenes (thus the system “lost” ambulances)
  - the system thus had fewer and fewer vehicles to allocate

London Ambulance Service (2)

- exception messages were generated but became so numerous the queue could not be cleared
  - this caused further delays in answering calls, further delays in assigning ambulances, and further errors in data recording
  - some calls were taking over 3 hours to answer - recommended maximum was 17 mins for the inner-city
  - a radio communications bottleneck was generated as the ambulance crews’ level of voice traffic increased

London Ambulance Service (3)

- the system was withdrawn from production on 27/10/92
- claims were later made that 20-30 people died as the result of ambulances being late on the scene
  - the initial assumption was that it was a technical disaster: stemming from a software bug (and there was a bug in the system that prevented a queue of “waiting” cases being cleared properly)

the Real Problems

- in the background was a tale of mismanagement on a grand scale
  - which culminated in the decision to go live with a system
    - which had not undergone full (live) testing
    - for which there was no contingency plan
    - in the use of which few operators had been fully trained
**An Analytical Typology of Failure Factors**

- environmental
- organizational
- project
- human/psychological

---

**1 - Environmental Factors**

- the NHS was under huge pressure from the UK Government to cut health costs - this contributed to
  - the LAS being under huge pressure from the NHS to improve its performance and to operate more efficiently - this contributed to
  - the LAS being under huge pressure from the NHS to improve its performance and to operate more efficiently - this contributed to

- the LASCAD tender being for an LAS-specific system to be built within a fixed timeframe
- the tender being issued against a directive that the lowest cost tender should be accepted in the absence of clear justification otherwise

- OUTCOME
  - the successful vendor bid was half that of the next lowest (which was from McDonnell Douglas - a vendor experienced with ambulance systems) - Apricot had no experience with ambulance systems and the next company was an established supplier
  - "you get what you pay for"

---

**2 - Organizational Factors**

- the LAS workforce was in bitter dispute with LAS management (the relationship was described as "toxic"), and many staff refused to attend system training sessions (claiming these were too far in advance of system use)

- LAS management was instituting a range of cost-cutting initiatives against fierce resistance

- OUTCOME
  - morale was very low at implementation time
  - workers had no belief in management and vice-versa; faith that "working together" would solve the initial problems was non-existent

---

**3 - Project Factors**

- no PM was appointed to be in overall charge of the activity (this is not as obvious a mistake as you might think - brief discussion) so lines of authority and responsibility were blurred

- no formal project methodology was used

- there were ambulance "packages" around at the time but the LAS saw itself as "special" (it was the largest in the world at the time in terms of area and # of incidents

- the timetable was defined as "absolutely" fixed

- OUTCOME
  - there was great pressure on the project from the outset

---

**4 - Human (Psychological/Social) Factors**

- management were "determined" not to be seen to fail under any circumstances (they had previously been described as "weak")

- workers were adamant that any problems could not be their fault

- the contractor wishing to deliver on time "against the odds"

- OUTCOME
  - at a time when everybody connected with the project must have known it was simply not ready to go, it seems that nobody spoke up, or if they did, nobody listened (cf the "Challenger" disaster)

---

**“Core” IT Risks**

- based on analysing this and other cases, five “core” IT risks have been identified as follows:
  - 1 fundamentally flawed schedule
  - 2 scope creep
  - 3 employee turnover
  - 4 specification breakdown
  - 5 poor productivity
in the LAS case
  1 fundamentally flawed schedule
    - the schedule was set by fiat very early in proceedings and before all parties could have understood all the issues
  2 scope creep
    - because the LAS had "special" requirements, there was a tendency to interpret requirements very broadly
  3 employee turnover
    - not a documented problem in the LAS case
  4 specification breakdown
    - the specification was at a very high level (the most common error) and required constant discussion
  5 poor productivity
    - the contracting supplier was new to ambulance systems (they were a game developer previously)

ERP (enterprise resource management) Systems
  - now often known just as "enterprise" systems
    - proprietary packages are sold by companies like SAP, Oracle, Baan and Peoplesoft
  - the avowed aim is to have systems that integrate organizations' entire information processing needs (relating to routinizable activities at least)

ERP Implementations
  - there has been some research into ERP failures but no really well documented individual cases
  - but the general outline of how they fail is reasonably clear and can be analysed in terms of the core risks

ERP Initiation
  - management motivation (re improved operational control and efficiencies) meets vendors claims for just those things
    - ERP is seen as a solution not as a task with genuinely problematic aspects
  - the situation is understood in logical terms as a forced improvement in practices (employees will have to change their practices but the software package will "manage" that - management need not be involved)
  - many (most) of the subsequent difficulties stem from this

Core Risk 1 - Flawed Schedule
  - a schedule is set based on the "logical" requirements for implementation
    - testing, process changes, retraining etc.
  - the schedule is based on management’s understanding of what will be required
    - but managers rarely if ever understand operational complexities in the way that staff do
    - the schedule is over-optimistic (it is based on inadequate knowledge)

Core Risk 2 - Scope Creep
  - the single most common cause of ERP failures
    - the organization discovers that it is a bit more "unique" than it had thought
  - as the implementors probe below the surface of the company, and as managers and staff discover what is really involved, so the requests for changes emerge
  - in many cases, these requests are acted on, and the package is "customised" to some extent
Core Risk 3 - Employee Turnover

- turnover was a crippling problem in the early ERP days
- skills, experience and knowledge of ERP systems were at a premium
  - experienced staff were lured away by competitors desperate for rare skills
  - stories filtered back of Monash graduates with ERP knowledge earning $150K+ within a year of graduating
- ERP is not a site for this now so much as other specialised areas can be (eg. M-commerce)

Core Risk 4 - Specification Breakdown

- in the ERP case, this typically means that the original mismatch between the expectations of management and what could actually be delivered by an ERP system comes to dominate proceedings
  - the implementation process starts to implode, relations between staff and the vendor deteriorate, and debates become the norm

Core Risk 5 - Poor Productivity

- decreases in productivity tend to be a consequence of problems elsewhere in most cases
- with the exception that the available pool of skilled resources may be too low - hence requirements emerge for
  - on-the-job training
  - reworkings
  - additional quality control procedures

a Note on Customisation

- the process of “customisation” is the tailoring of commercial software to meet a specific objective
- usual reasons?
  - our company is “special” or has some special requirements
  - it is just too hard to change all our processes to accommodate what the package requires
  - there has been unexpected user resistance that we can’t resolve
  - “new” requirements have emerged since the package was bought
  - other

but why is customisation such a problem?

- customisation is unwise for a variety of reasons
- if you must customise, a good rule of thumb is to modify no more that 5% of the package (difficult to measure anyway)
- but why is customisation such a problem?
  - discussion

Customisation Risks

- list
Endemic Factors “encouraging” IS Failures

- cost-cutting and efficiency pressures
- the pace of technological change/pressure to respond
- interdependencies between different systems
- the costs of quality management and risk management
- skills shortages
- constant stress

these are all risks which can spawn complex interrelationships