ICS 132: Organizational Information Systems

Planning, Building and Maintenance

where are we?

- project
  - due next Monday...
- meantime...
  - continuing with topics from Alter
  - today: planning, building and maintenance
  - previously encountered in ICS 52, 121...
    - this is meant to supplement, not replace, that material

why look at these?

- the traditional technical view:
  - we’re system developers. we write code. we don’t need to worry about these management issues.
- the unfortunate reality:
  - most system failures are project management failures
  - the cost of correcting errors increases as the project proceeds
- so the 132 view:
  - IS development encompasses the whole lifecycle
  - from planning to retirement

four roles for planning

- determining feasibility
  - is it even possible to do this?
  - let’s see step by step how we’ll get there
- determining costs
  - breaking down costs
  - estimating benefits
- predicting resource needs
  - what else will have to be in place, and when?
- getting political buy-in
  - often the most important....
    - software architects and Powerpoint....

planning: challenges

- difficulty foreseeing and assessing opportunities
  - hindsight is 20-20
  - organisations and technologies co-evolve
    - systems afford new organisational opportunities
    - organisations adapt to capitalise upon systems
  - people find new uses for technology
    - e.g. information systems may indirectly communicate information
      - Apple’s meeting reservation system

planning: challenges

- difficulty assuring consistency with organisational plans and objectives
  - no single point of view
  - unexpected conflicts
  - distributed effort
  - individual benefit versus overall benefit
planning: challenges

• difficulty building large systems
  – long-term development efforts
    • changing circumstances
    • problems of consistency
  – distributed effort
    • coordinating all the players
    • getting agreement
  – project failure is a huge problem
    • both common and costly
    • once you have a plan, sticking to it is hard
    • when your plan starts to fail, recovery is even harder

• difficulty maintaining information systems performance
  – as usual, performance can have many meanings
    • throughput, efficiency, quality, costs, reliability...
  – two challenges
    • delivering performance
    • maintaining performance
      – the environment is continually changing
      – the organisation is continually changing
      – new approaches can yield short-term benefit
    • many factors (as shown by WCA)

planning: challenges

• difficulty collaborating with system builders
  – one of the reasons for 132!
    • creating more informed project managers
    • creating more informed system builders
  – different models of performance
    • business performance
    • system performance
      – system can perform well but business goals fail
      – business goals can succeed without system optimisation
        » ensuring that your effort is worthwhile

planning: principles

• support the firm’s business strategy with appropriate technical architecture, standard and policies
  – this is more than “do it right”
  – focus on business strategy
    • you need to be able to articulate this!
  – the importance of scale
    • but... watch out for issues of maintenance and evolution

planning: principles

• evaluate technology as a component of a larger system
  – the best technology does no good without an infrastructure to make it work
  – don’t ask what can this technology do?, but rather, what kind of use are we in a position to make of it?

planning: principles

• recognise life cycle costs, not just acquisition costs
  – “Total Cost of Ownership”
    • support, administration, training, running costs, consumables, ...
    • infrastructure (and it’s own knock-on costs)
planning: principles
• design information systems to be maintainable
  – supporting monitoring, control, evolution
  – making it possible to find out what’s going on!
  – “IS maintenance” might actually be a response to changes in the organisation or the work itself

planning: principles
• recognise the human side of technology use
  – human issues
    • training
    • growth
    • motivation
    – all those good “Human Relations” school ideas...
  – this is not just a question of design, but a question of engagement
    • ethnographic techniques
    • the Scandinavian “Participatory Design” movement

planning: principles
• support and control the technical side
  – 132 may emphasize human issues, but the technology doesn’t look after itself...
  – continual monitoring
  – functional maintenance
    • making sure hardware is reliable
  – preventative maintenance
    • software
    • data
  – who’s responsible?

cost-benefit analysis
• evaluating specific plans
• comparing alternatives
• do the benefits outweigh the costs?
  – seems obvious, but not always
  – measurement may not be straightforward
  – remember the time value of money
    • unused monies don’t just sit around collecting dust
  – and don’t forget the cost of CBA!

cost-benefit analysis
• costs and benefits stated objectively
  – but of course, they’re not! issues of perspective
    • statement of purpose
    – decision-making? background information?
    • time period
    – before the fact? monitoring an ongoing project?
    • scope
    – considering radical alternatives?
    • criteria

cost-benefit analysis
• costs
  – resources required to procure a solution
    • not always financial, but expressed financially
      – equipment
      – wages for work
      – rent for space
cost-benefit analysis

- benefits
  - cost savings
    - inc. better utilization of assets, reduced inventories...
  - cost avoidance
  - improved performance
  - "intangibles"
    - e.g. better information

- some problems
  - how much time to spend identifying alternatives?
  - cost accounting problems
    - double counting, omitting costs, hidden costs, spillovers
  - quantifying benefits
    - intangibles
      - e.g. morale, improved decision-making
  - underestimating cost, overestimating benefit
  - temporal effects too...

development models

- traditional system development
  - this is the conventional approach CS discusses
  - e.g. the SE model at the heart of ICS 52, 121
- prototyping
  - an iterative model
    - quickly build a mock-up or basic functional system
    - put it into limited use, see what works and doesn’t
  - preparatory to full system development, or standalone
  - various things to learn:
    - what’s easy or hard technically
    - what’s easy or hard organisationally
  - only as good as your ability to evaluate it

- application
  - better to buy than to build
  - formal relationships for quality, support, maintenance
  - customisation generally needed
    - but not always possible...
- end-user development
  - a Holy Grail of interactive system development
  - example: spreadsheets
  - EUD is normally a response to individual problems
    - not a strategic organisational approach
    - but, may favour distributing the ability to solve problems

the blame lifecycle

- idiot managers
- idiot users
- lousy vendors and their damnable lies
- government
- capitalist avarice
  - lack of the right tools
  - lack of the right methods
  - lack of sufficient willpower
the blame lifecycle

Initiation
Identifying needs; general description of problems

Development
Transform requirements into artifacts; create documentation

Implementation
Make system operational; train, covert, consult

Operation/Maintenance
Ongoing use; bug fixes; enhancements

progress

summary

- This class talks about technology in context
  - In general, organisational context
  - Today, temporal context
    - What comes before... planning
    - What comes after... maintenance
- Planning isn't just deciding what to do
  - Evaluating options & managing resources
  - You will be called on these!
- Maintenance is the major part of use
  - Meeting changing needs
  - Accommodating new opportunities

next time

- Security
- Read Alter ch 13

maintenance

- Most of a project lifetime is maintenance
  - If it takes longer to build than to use, you're doing something wrong...
- Varieties of maintenance
  - Regular maintenance
  - Accommodating changing needs
  - Accommodating changing technologies
- Designing for maintenance
  - Modularity
  - Scalability
  - Flexibility