Fitting software to the organisation

Reducing time to value for new software systems

Prof. Ian Sommerville
School of Computer Science
University of St Andrews
Scotland

St Andrews

- Small Scottish town, on the north-east coast of the UK
- Home of golf
- Scotland's oldest university (founded in 1413)
- Small university focusing on research and teaching excellence

About me

- Background in traditional software engineering
- Author of widely-used textbook in this area
- Now primarily interested in human, social and organisational influences on software systems design and dependability

Takeaway knowledge

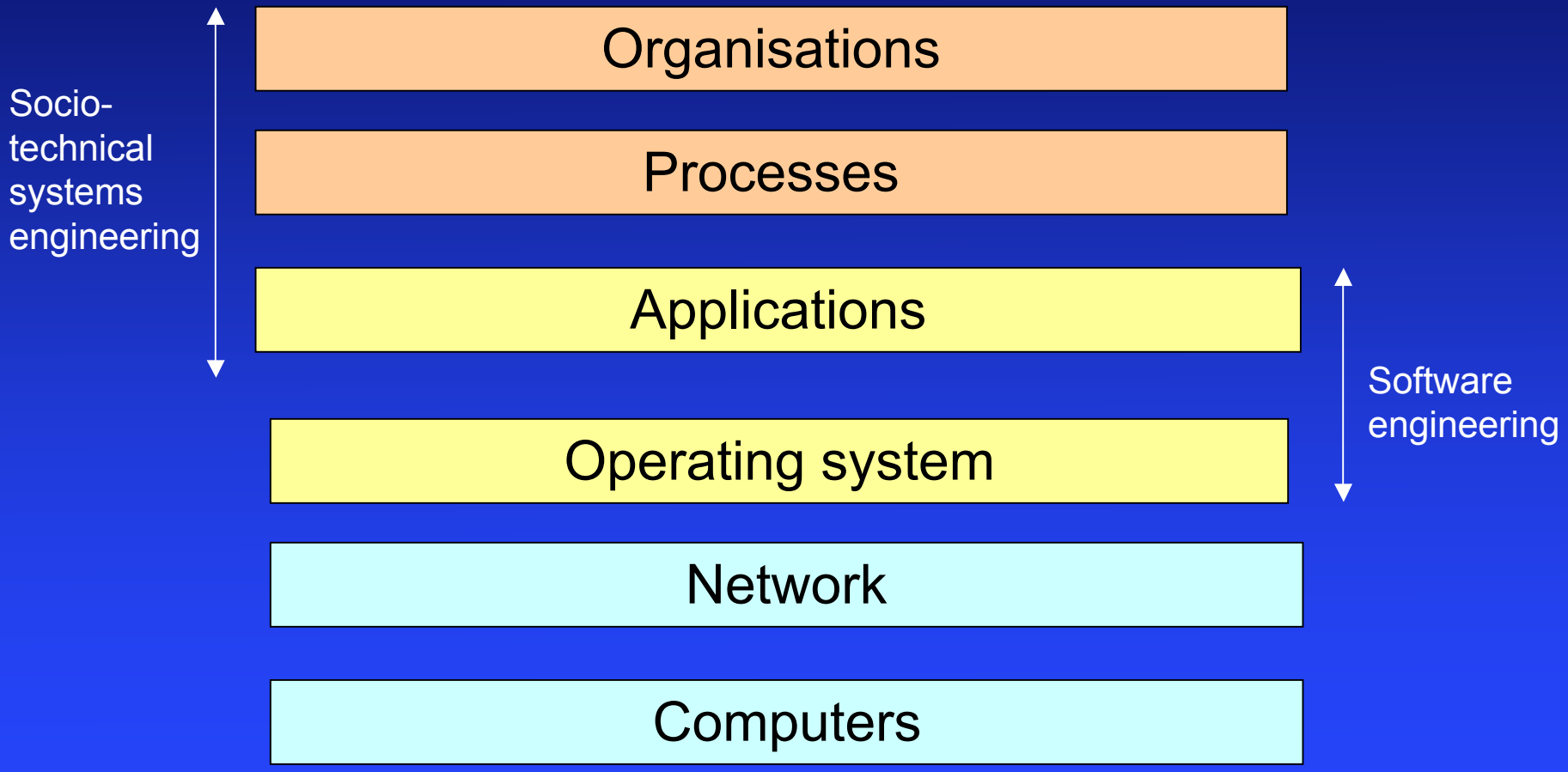
- Socio-technical systems
- Time to value
- Socio-technical systems engineering
  - Why this approach is useful in reducing time to value
  - How to get started with STSE
  - Some things we've learned from STSE

Socio-technical systems

- A socio-technical system is a system that exists to serve some organisational purpose. It includes:
  - Computers
  - Software
  - Business processes
  - Organisational rules and regulations
  - Human operators
- Examples
  - A system to support admission and discharge of hospital patients
  - A system to support purchasing in a company

Socio-technical systems

Systems engineering



Enterprise systems

- When designing enterprise or organisational information systems, we are not just designing software. We are concerned with the design of socio-technical systems
- Most modern systems, especially those based on ERP or COTS systems, are based around the notion of co-design of business processes and software
- Socio-technical issues have a profound influence on the ways that processes are enacted

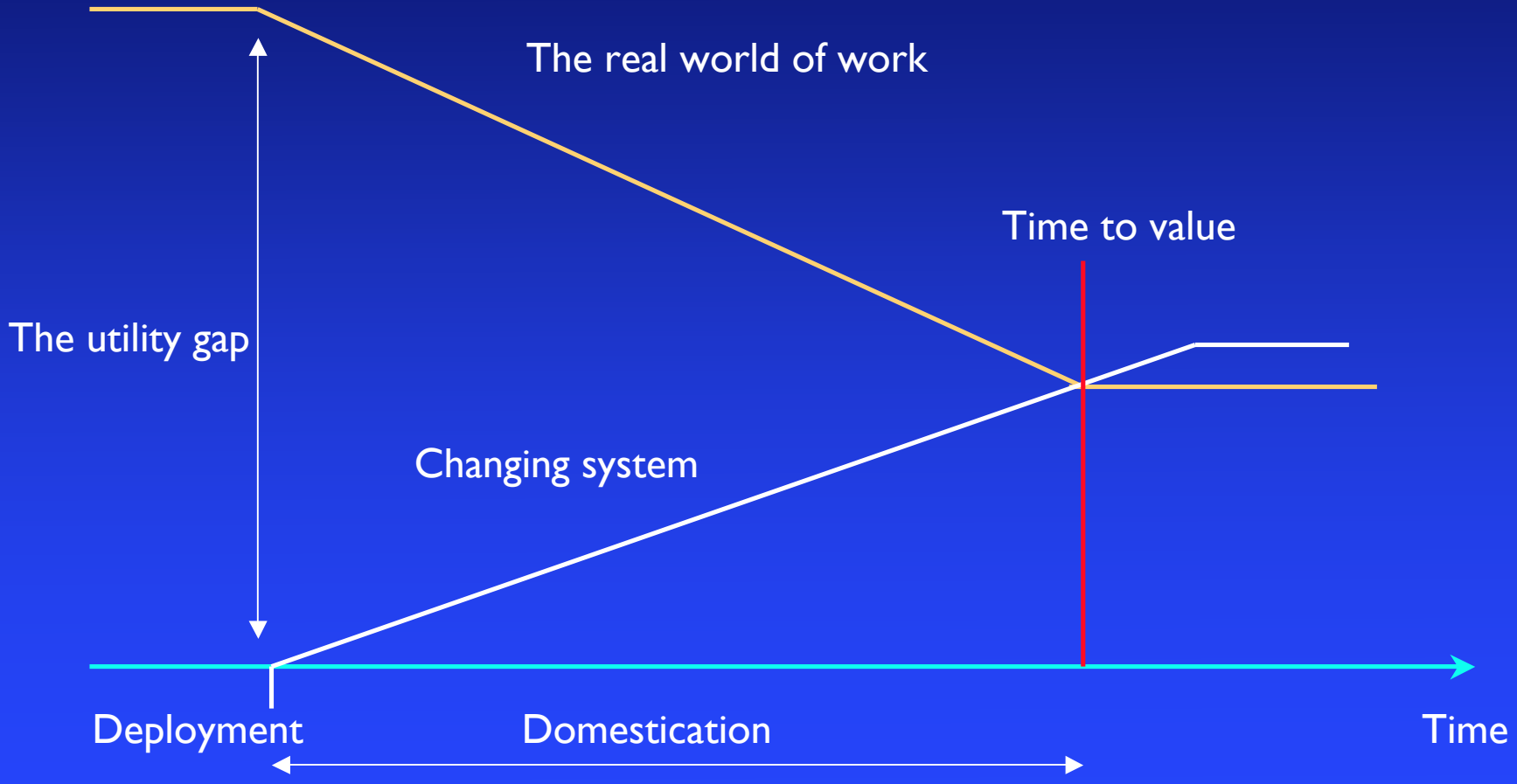
Time to value

- The length of time between the decision that some new system is needed and the time when that system is effectively used in an organisation

Time to value components

- Time to procure the system
- Time to develop the system
- Time to deploy the system
  - Deployment may have to fit with rhythms of the organisation e.g. end of accounting periods
- Time to 'domesticate' the system
  - Time required to embed the system in organisational processes

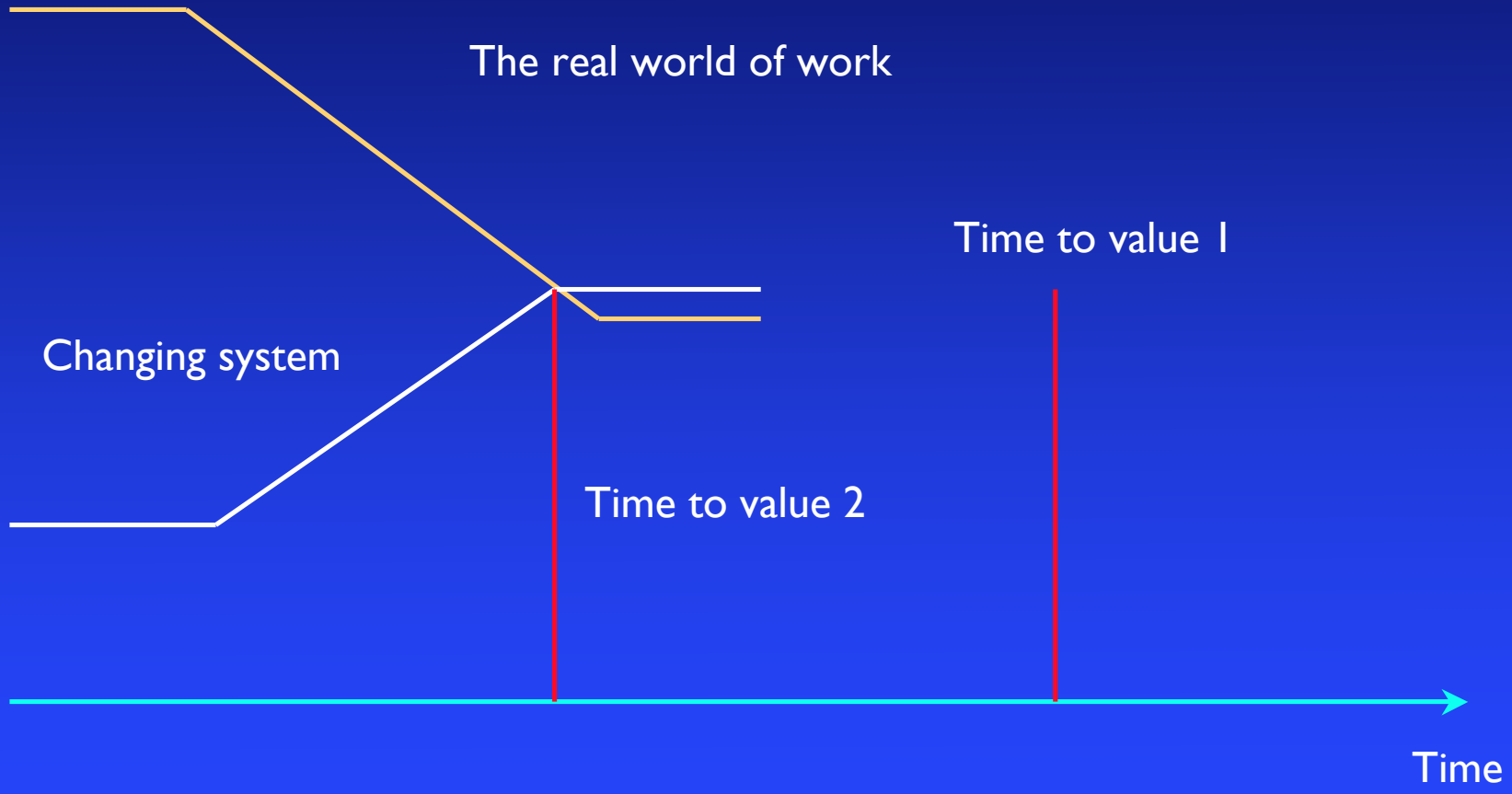
Domestication



Reducing time to value

- Reduce the time it takes to make
  - the system fit the organisation
  - the organisation fit the system
- To reduce time to value, we need to
  - Narrow the utility gap when the system is deployed
  - Plan the deployment to fit with other activities
  - Reduce the domestication time

Narrowing the utility gap



Socio-technical systems engineering

- The analysis, specification, design and evaluation of socio-technical systems

Motivation

- Many years of research have demonstrated the importance of socio-technical issues (human, social and organisational) on the success or otherwise of large organisational systems
- However, conventional development processes lack any systematic approach to considering how socio-technical issues affect the system requirements, use and evolution

Value from STSE

- Effectiveness
  - Deployed systems are more effective in supporting business processes
  - In many cases, value from new systems is not realised because these are not used at all or part of their functionality is not exploited
- Dependability
  - Reduced probability of usage errors
  - More effective error recovery
- User satisfaction
  - Better user acceptance of new systems
- Faster ‘time to value’
  - Shorter assimiliation period for new systems. Fewer mismatches between system and work

Key features of STSE

- Focuses on the social and the technical together rather than consider technical issues in isolation
- Based on a pragmatic acceptance of the world as it is, populated by imperfect people
- Accepts that systems will always be a compromise, with multiple, often conflicting, notions of what is meant by ‘success’ and where the system boundaries lie
- Relies on observation of work and an understanding of the context of use of software

Stages of STSE

- Sensitisation
  - Helping analysts and engineers understand the significance of socio-technical issues
- Fieldwork
  - Going into the organisation and collecting data about the real world of work
- Analysis
  - Analysing this data to develop a deeper understanding of what is really required for the system to be effective
- Synthesis
  - Using the analysis systematically to design and implement better software

Getting started with STSE

- Sensitisation and Fieldwork
- The outcome of these activities is insight into the social and organisational issues that influence the use of a system
- With this insight, you are in a better position to design more effective and efficient systems
- Analysis and synthesis will not be covered as these require more time than we have available. See the URLs on the last page for more information

Sensitisation

- Many technical people believe that:
  - People will follow processes
  - Users are all the same
  - Design is about meeting requirements rather than providing an efficient and effective system
- In many projects, developers never meet or observe the users of a system
- Sensitisation is about taking the development team into the world of work and showing them what really happens
  - 10 minutes in a ward explains why you should never have a system that doesn't allow for simple restart after interruptions

Process diversity

- ERP-supported business processes - my way or no way!
  - Standardising business processes is sometimes seen as one of the principal benefits of ERP systems
  - Sometimes, more money is spend on process engineering rather than software engineering
  - But process mismatch and a wider utility gap is common
- Real processes are diverse and contingent on their environment.
  - A process that works when a unit is fully staffed has to change when there is a flu epidemic and only 50% of the staff are at work
  - Different parts of an organisation have local goals that processes adapt to
  - People with different knowledge and experience adapt process in line with their own skills

Hospital admissions system

- Patients arrive at a hospital and are checked in at some central point
- However, when they leave it is the responsibility of the nurses in a ward to check out the patients
- Maternity ward
  - Two (at least) patients normally leave but only 1 arrives
  - Emergencies are rare, nurses have time to check out patients
- Cardiac ward
  - Emergencies are routine and must be dealt with immediately
  - Nurses give this priority over the check out of patients

Fieldwork

- Systematic observation of work in the context where the work is performed
- The aim is to develop a rounded understanding of the details of how the work is done and the contextual influences on the work

You need to get out more

- To understand process diversity and what really happens, you need to observe work as it is really done and to talk to users in the context in which they work
- Look at and compare how the same work is done in different parts of an organisation
- Focus on differences between people and groups
  - Find out if the difference is accidental or essential
  - Accidental: Processes have simply developed in different ways because of individual preferences
  - Essential: There are good local reasons why different processes co-exist.

Being a good observer

- Avoid judgements
  - Your job is to understand what is going on, not to judge whether or not people are doing the right thing. Managers should not be involved
- Be humble
  - You don't understand people's job better than they do and you shouldn't impose your preconceptions about the work on them
- Focus on details
  - Details are important and can significantly affect the effectiveness, efficiency and dependability of work
- Ask open questions
  - "Tell me about…" rather than "Why do you do …"
- Find out something about the domain before you start

4 key questions

- What's done at the moment and can't be done in any other way?
  - E.g. externally mandated compliance activities
- What's done at the moment but can be done in a different way?
- What might disappear with the introduction of a new technology and what will be the consequences?
  - E.g. paper records of activities
- What should be designed out of the system?
  - Stupidity has a long history!

Fieldwork outcomes

- Debriefing reports to the project team on what's been observed and, wherever possible, a discussion of differences that exist within groups and between different parts of an organisation
- Wherever possible, take these back to the people observed and discuss the observations with them
- Do not have closed reports for any group
  - The point of fieldwork is learning not assessment

Practical STSE

- Requirements sanity checking
  - Doing some simple STSE
- Supporting domestication
  - Some design lessons learned from STSE

The requirements delusion

- Many people involved in system requirements and business process engineering believe that they are doing user-centred design
  - They ask users for their opinions
  - They invite users to attend meetings where they present their ideas and discuss their intentions
  - They conscientiously pass information to users and tell them about changes
- The requirements delusion(s)
  - Users care about IT systems
  - Users can articulate what they do and why

Requirements insanity!

- Requirements insanity arises when some proposed system requirement or process proposal is simply so stupid that it is unworkable
- Requirements insanity often arises because the designers of the system focus on a single goal without considering the conflicts with personal, local or organisational goals, with how the system will actually be used in practice
- This is often compounded by assuming that people will never forget things, will always concentrate on what they are doing, will read and follow instructions, will never be ill, pregnant, etc.

Requirements sanity checking

- One effective way to reduce time to value is to reduce the 'stupidity factor' of new systems
- Look at the proposed requirements for a new process/system in the context where work is actually done
- Find the answers to the following questions
  - How does the process interact with other processes going on?
  - Does the new process require resources that have to be taken from other activities? If so, what are the consequences?
  - Does the process have any benefits for the people involved in enacting the process? Are there any sanctions that can be imposed for non-compliance?
  - Are the benefits of the new system local to one or more workgroups or organisational?

Customer sales leads

- A retail financial institution had a system in place where, during a transaction, tellers could note information about potential sales opportunities. These were recorded on paper and passed to the back office for processing. They planned to introduce a new system where the teller entered information into a sales opportunity database directly rather than record it on paper
- When we observed the system, we noted that this process meant taking resources from the process of serving customers
- Queues would lengthen, customers would be unhappy, customer service promises would not be met, managers would be unhappy
- The system was abandoned before development

Domestication

- The meshing of a new system with the way that work is done in a particular context
- Initial problems
  - These are simply due to misunderstandings about the use and purpose of the system, often due to inadequate training
  - Performance issues are common because it is very difficult to predict actual loads on a system
- Longer-term assimilation
  - Processes change to accommodate the new system and the system changes to reflect practical experience
  - People find workarounds for problems and embed the system into their everyday work

Design for domestication

- Local knowledge
- Process flexibility
- Flexible information schemas

Local knowledge

- Local knowledge includes knowledge of who does what, how authority structures can be bypassed, what rules can be broken, etc.
- Local knowledge is used to find ways around problems. When new systems are introduced, some of this becomes obsolete and needs to be reconstructed
  - Maintain information about the provenance of data
    - Who provided the data, where the data came from, when it was created, edited, etc.
  - Maintain organisational models in the software
    - Who is responsible for what, contact details, etc.

Process flexibility

- Make workflows explicit rather than embedding them in the software
  - Not just ‘continue’ buttons! Users should know where they are and where they are supposed to go
- Support workflow navigation/interruption/restart
- Design systems with an ‘emergency mode’ where the the system changes from enforcing policies to auditing actions
  - This allows the rules to be broken but the system would maintain a log of what has been done and why so that subsequent investigations could trace what happened
- Support ‘Help, I’m in trouble!’ as well as ‘Help, I have a problem?’

The database delusion

- It is possible to design a database structure that will cover all possible eventualities
- This assumes a homogeneity in users, customers, processes which simply does not exist in practice
- Database structures are inevitably a compromise between the ways in which different processes are enacted or reflect organisational rather than local information needs

Extensible schemas

- Provide a free-format text box with every form where users can enter any information they wish
  - In a museum cataloguing system, this was used in 50% of the catalogue entries
  - This allows people to cope with deficiencies in design and to use the database to record a conversation rather than simply maintain information
- Support a ‘not relevant’ value that can be used instead of leaving a field blank
- Support ‘come back later’ form filling to help cope with interruptions

If you do nothing else….

- Take ALL system developers and spend a couple of hours walking around, watching how users really do their jobs
- Find out why there are different processes to do the same thing BEFORE you standardise the process
- When you design your data schema and forms for data input, ALWAYS allow for a comments field where people can add extra information that may be useful

Conclusion

- Don't think that you can get it right first time
  - this is impossible for complex, enterprise systems
- Plan for domestication
- If you can, think about trade-offs between procurement/development and domestication
  - Rapid delivery doesn't necessarily mean rapid time to value

For more information

www.comp.lancs.ac.uk/computing/research/cseg/projects/coherence/

www.comp.lancs.ac.uk/computing/research/cseg/projects/pointer/patterns.html

www.comp.lancs.ac.uk/~rouncefi/Tutout.html

http://www.dirc.org.uk/research/DIRC-Results/Ethnography.html