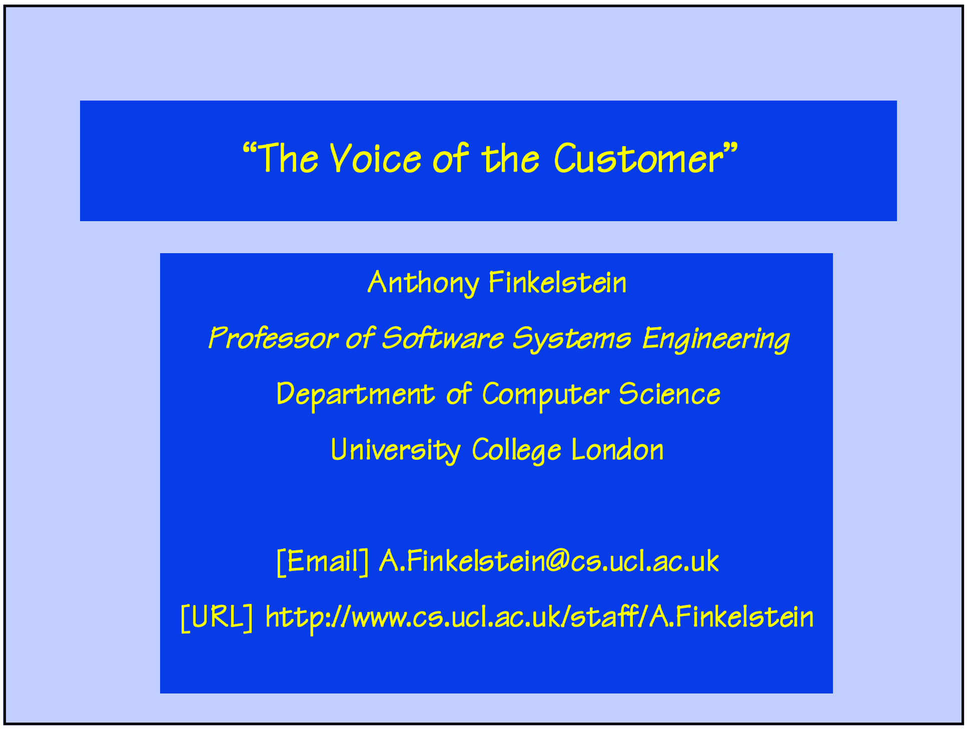


"The Voice of the Customer"

Anthony Finkelstein

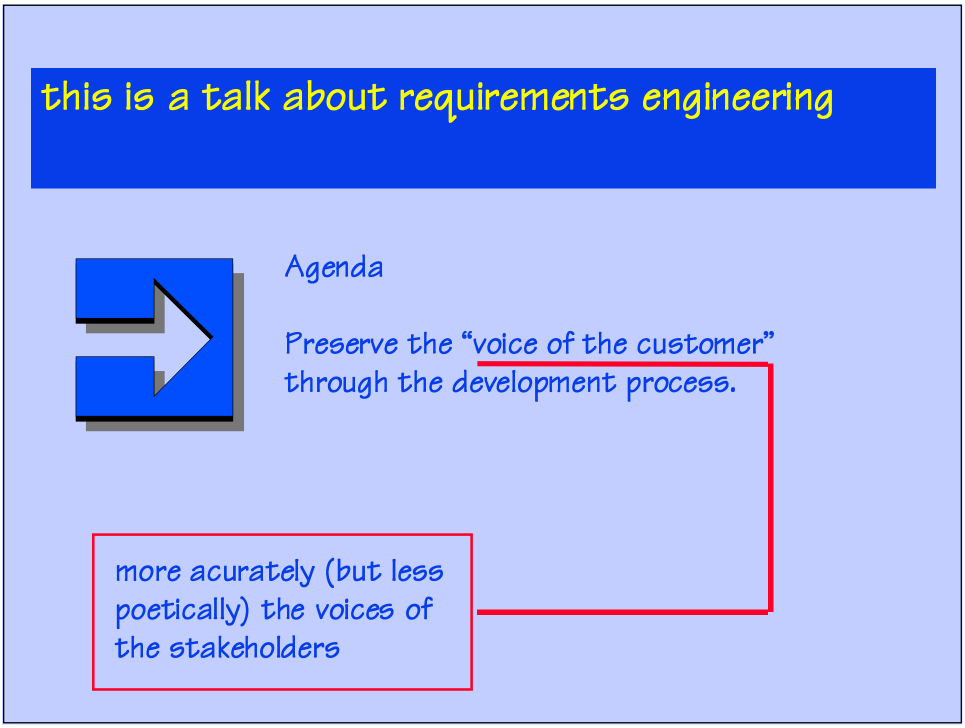
Professor of Software Systems Engineering

Department of Computer Science

University College London

[Email] A.Finkelstein@cs.ucl.ac.uk

[URL] http://www.cs.ucl.ac.uk/staff/A.Finkelstein



this is a talk about requirements engineering

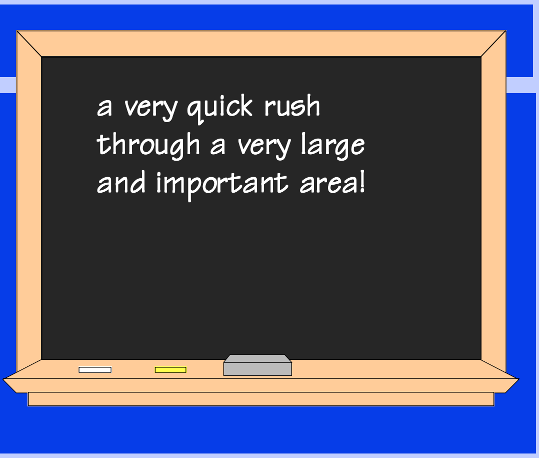
Agenda

Preserve the "voice of the customer" through the development process.

more acurately (but less poetically) the voices of the stakeholders

Outline

- introductory remarks
- incomplete motivation
- a few problems
- some partial solutions
- an interim conclusion


a very quick rush
through a very large
and important area!

Orientation

- "Requirements engineering is the branch of systems engineering concerned with the real-world goals for, services provided by, and constraints on a large and complex software-intensive system. It is also concerned with the relationship of these factors to precise specifications of system behaviour, and to their evolution over time and across system families."
- Establishing the needs that have given rise to the development process and organising this information in a form that will support system conception and implementation.

Why is Requirements Engineering Important?

- the negative case
  - client contact
  - time and effort expended
  - error removal cost
  - risk minimisation

—

- the positive case
  - ensures user focus
  - supports adaptation and evolution

+

Survey - 17 countries - 4000 responses

- Software developers in the IT, production and service sectors consistently ranked "requirements specification" and "managing customer requirements" as the most important problems they faced. In the case of requirements specification more than 50% of respondents rated it as a "major problem" and 35% of respondents rated as a "minor problem". Less than 12% of respondents rated it as "never a problem", the lowest rating in the survey. Similar responses were obtained for managing customer requirements.

> “Requirements were regarded as significantly more problematic than documentation, testing, quality systems, standards, design, configuration management, and programming.”

The Bottom Line

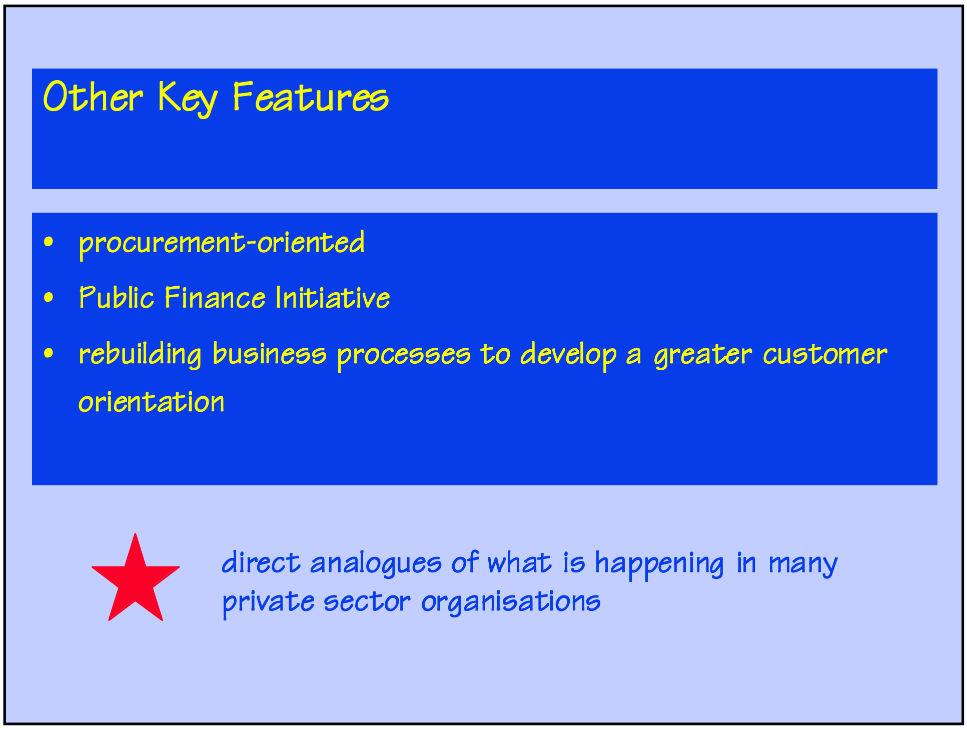
quality makes no sense without reference to requirements

quality-oriented development is requirements-driven development

"A Large Organisation"

at our sponsors request!

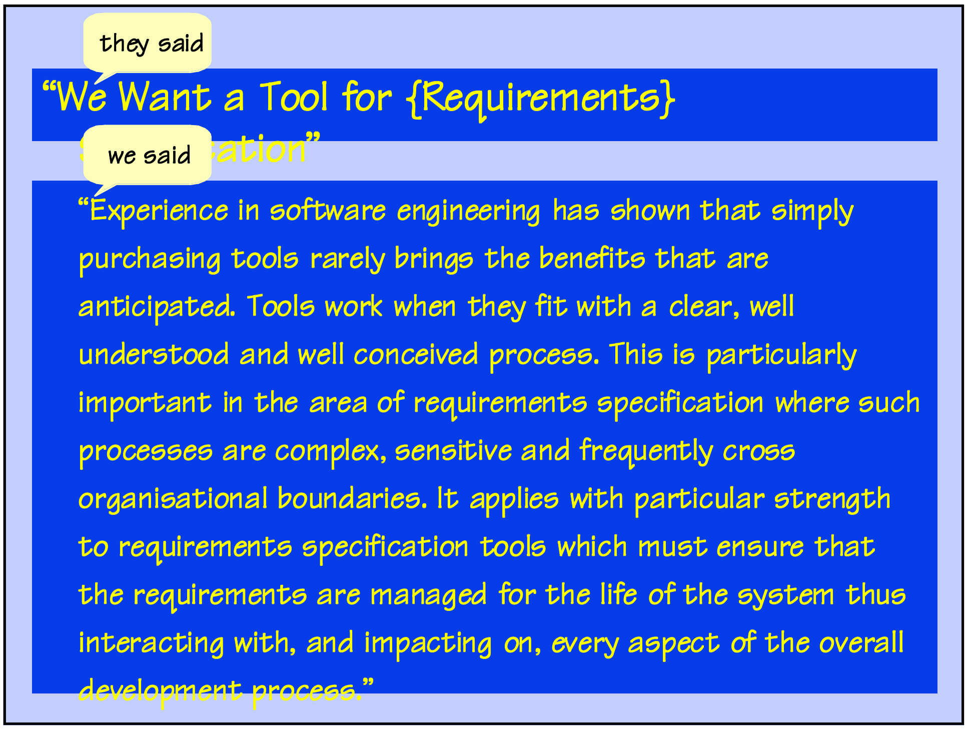
- government owned (at the moment)
- provides services in the transport sector
- procures, operates and maintains very large software-intensive systems, some of which are safety-critical
- projects with a budget of up to half a billion pounds
- development processes for these systems are long-term and complex
- processes involve internal and external organisations, some of which are located in different European countries.



Other Key Features

- procurement-oriented
- Public Finance Initiative
- rebuilding business processes to develop a greater customer orientation

★ direct analogues of what is happening in many private sector organisations



they said

“We Want a Tool for {Requirements} [illegible]ation”

we said

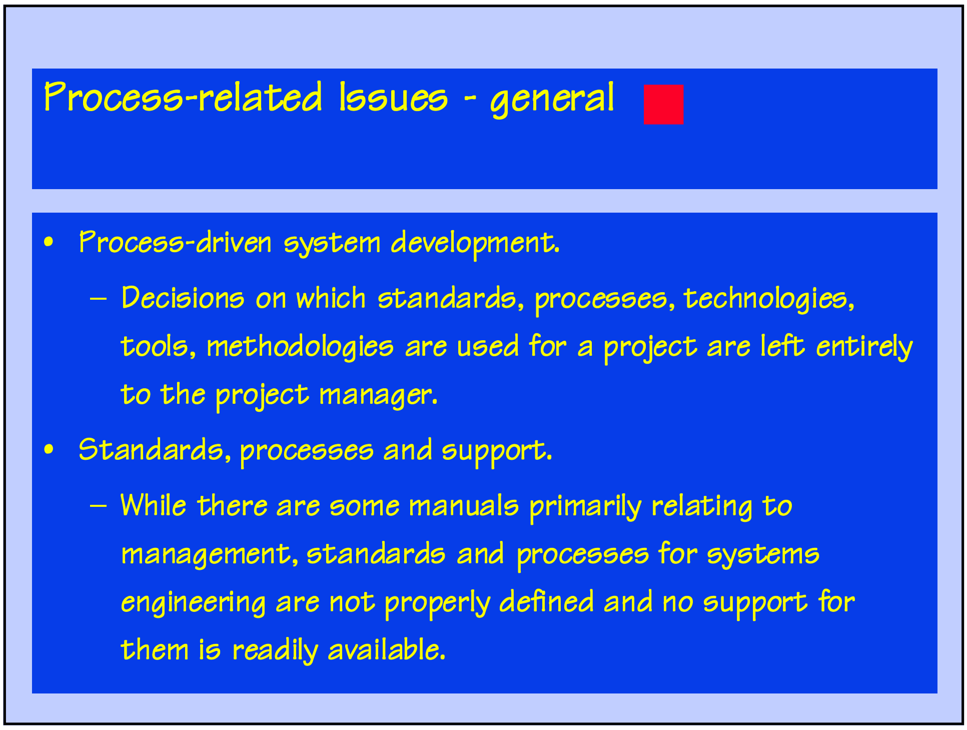
“Experience in software engineering has shown that simply purchasing tools rarely brings the benefits that are anticipated. Tools work when they fit with a clear, well understood and well conceived process. This is particularly important in the area of requirements specification where such processes are complex, sensitive and frequently cross organisational boundaries. It applies with particular strength to requirements specification tools which must ensure that the requirements are managed for the life of the system thus interacting with, and impacting on, every aspect of the overall development process.”

Issue Classes

- Processes 🟥
  - Issues associated with this class are related to processes and standards on an organisational rather than project level.
- Requirements Content 🟧
  - Issues associated with this class are directly related to the content of requirements documents.
- Domain 🟨
  - Issues associated with this class are related to capturing the specialised domain in which the organisation operates.

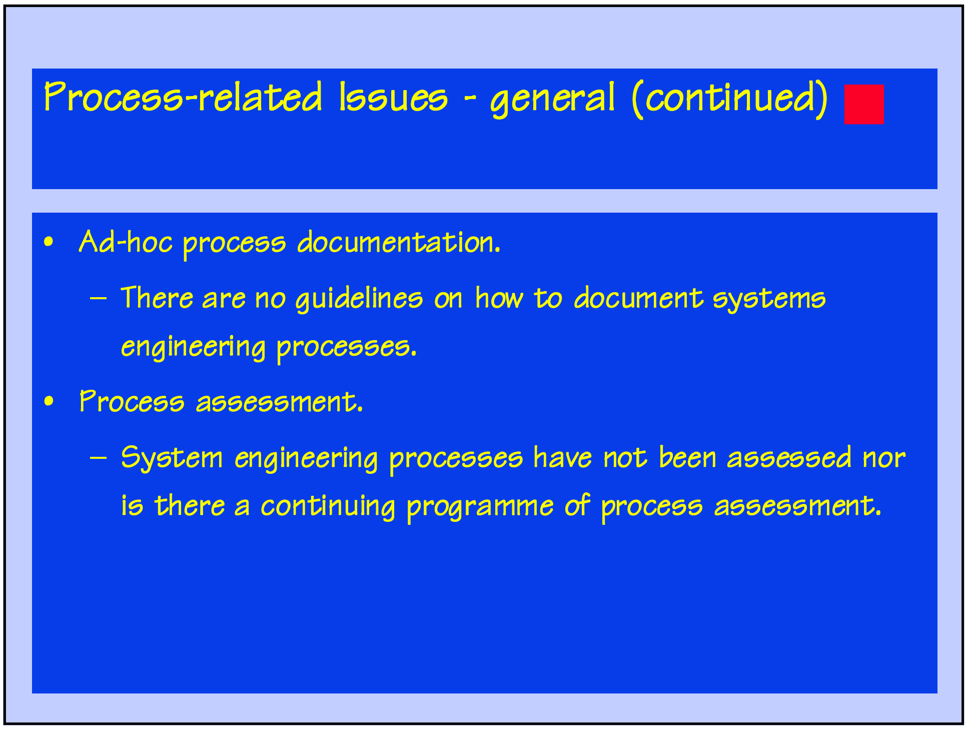
Issue Classes (Continued)

- Organisational Infrastructure 🟩
  - Issues associated with this class are about organisational level support for systems engineering.
- People 🟪
  - Issues associated with this class are related to people, for example education, motivation, or training.
- Tools ⬜
  - Issues associated with this class are concerned with tools and their use.



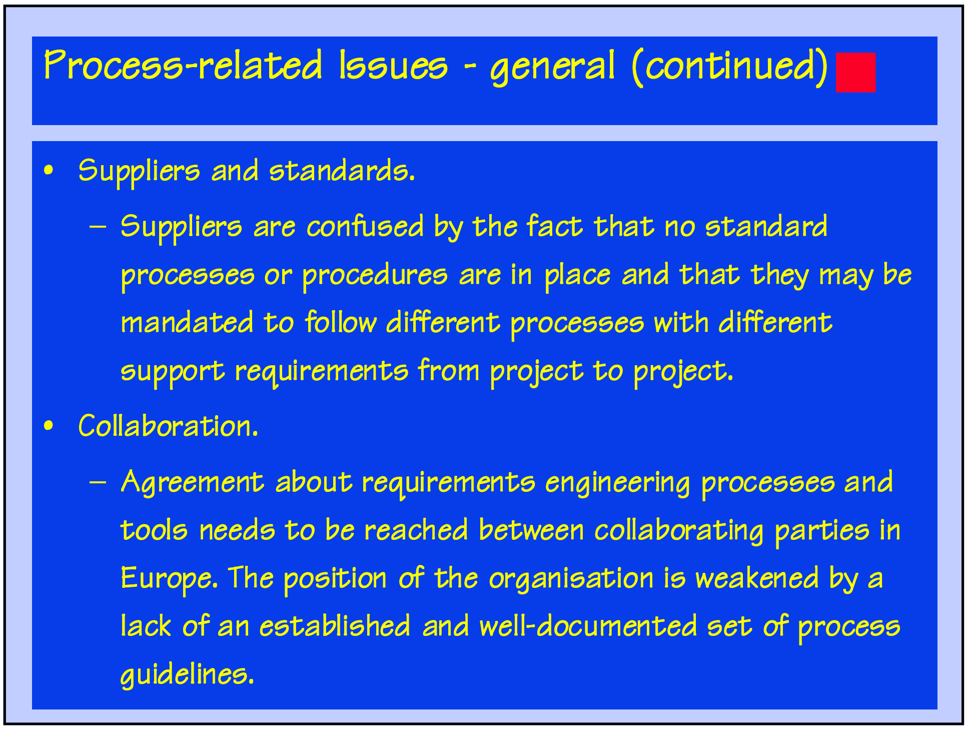
Process-related Issues - general

- Process-driven system development.
  - Decisions on which standards, processes, technologies, tools, methodologies are used for a project are left entirely to the project manager.
- Standards, processes and support.
  - While there are some manuals primarily relating to management, standards and processes for systems engineering are not properly defined and no support for them is readily available.



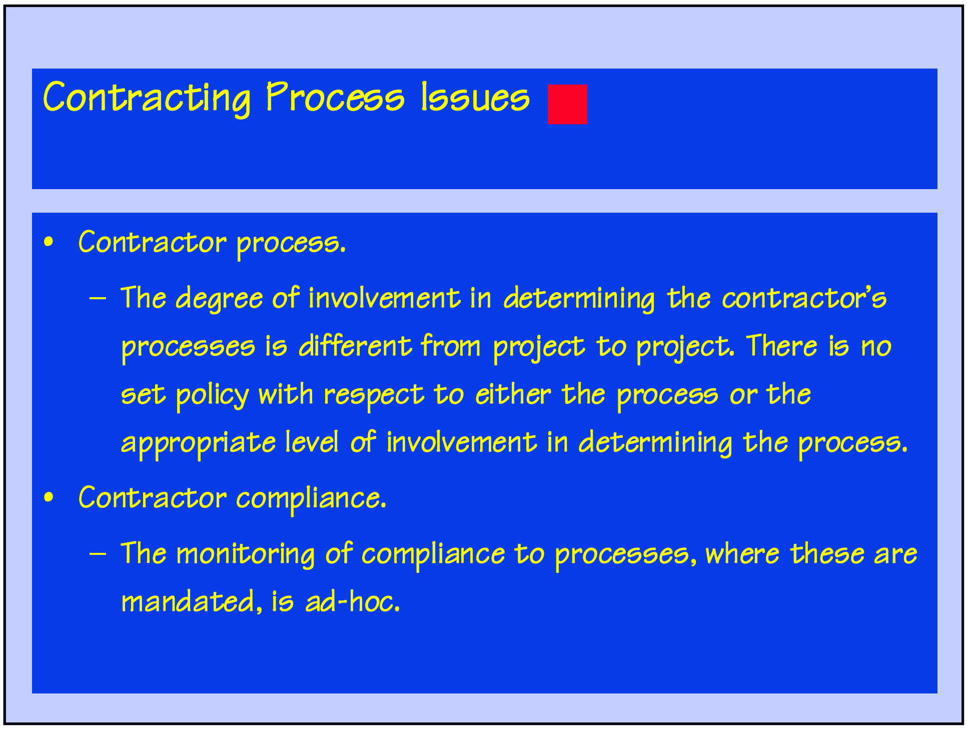
Process-related Issues - general (continued)

- Ad-hoc process documentation.
  - There are no guidelines on how to document systems engineering processes.
- Process assessment.
  - System engineering processes have not been assessed nor is there a continuing programme of process assessment.



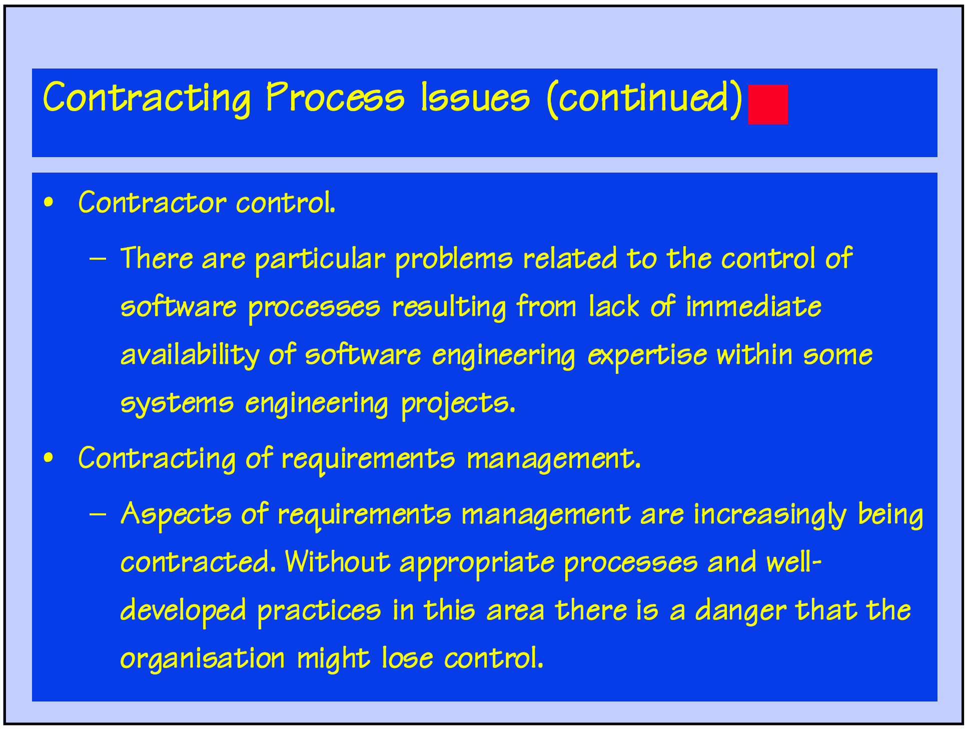
Process-related Issues - general (continued)

- Suppliers and standards.
  - Suppliers are confused by the fact that no standard processes or procedures are in place and that they may be mandated to follow different processes with different support requirements from project to project.
- Collaboration.
  - Agreement about requirements engineering processes and tools needs to be reached between collaborating parties in Europe. The position of the organisation is weakened by a lack of an established and well-documented set of process guidelines.



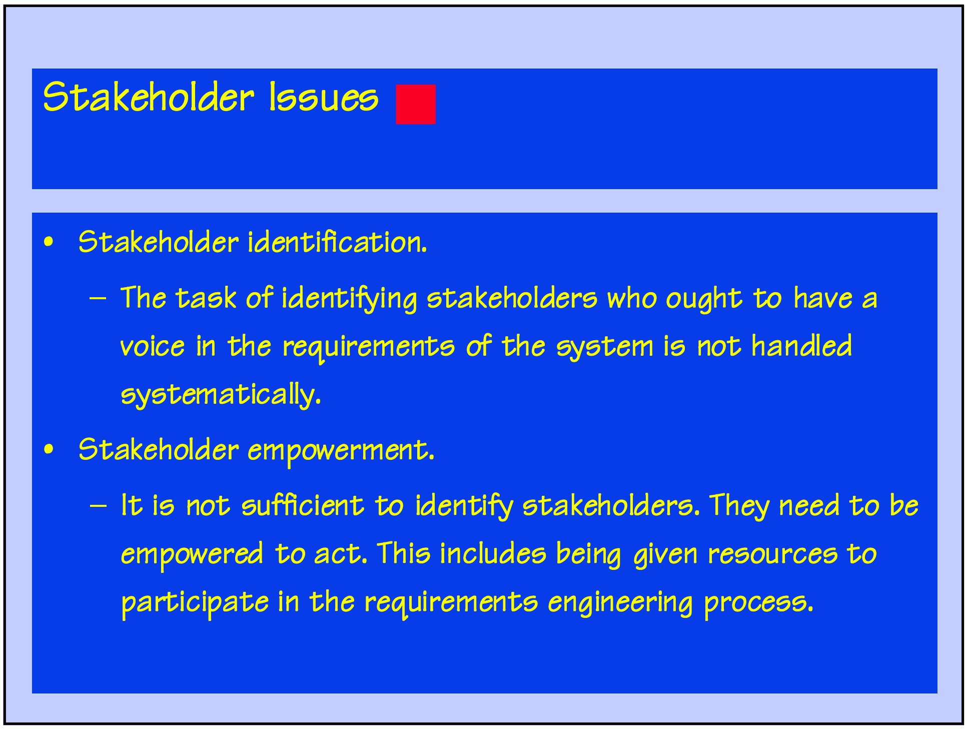
Contracting Process Issues

- Contractor process.
  - The degree of involvement in determining the contractor’s processes is different from project to project. There is no set policy with respect to either the process or the appropriate level of involvement in determining the process.
- Contractor compliance.
  - The monitoring of compliance to processes, where these are mandated, is ad-hoc.



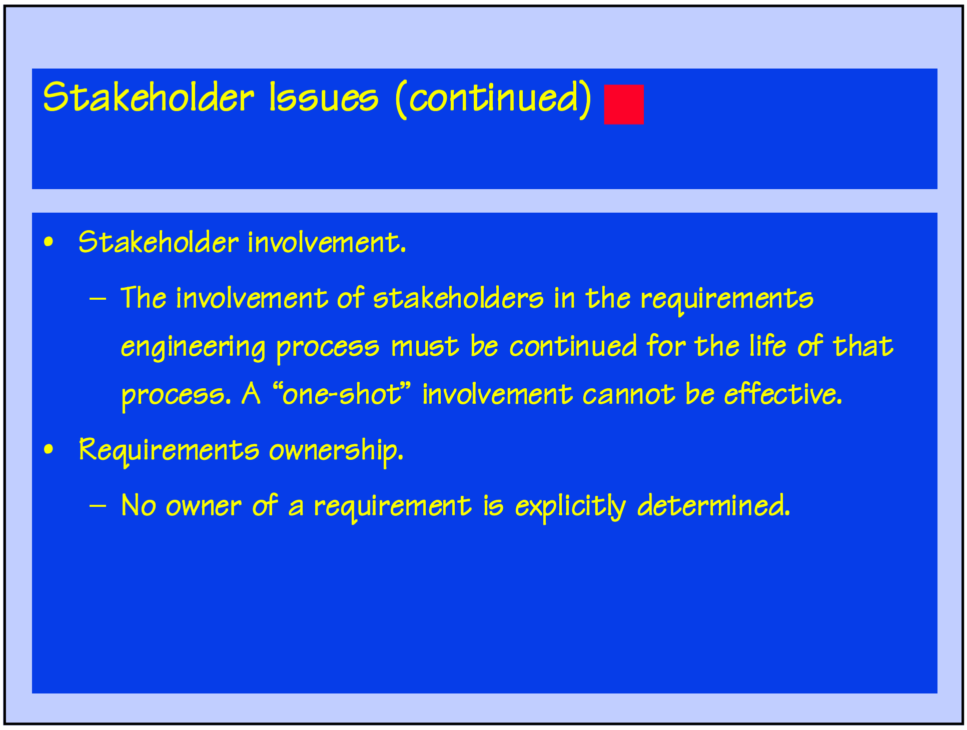
Contracting Process Issues (continued)

- Contractor control.
  - There are particular problems related to the control of software processes resulting from lack of immediate availability of software engineering expertise within some systems engineering projects.
- Contracting of requirements management.
  - Aspects of requirements management are increasingly being contracted. Without appropriate processes and well-developed practices in this area there is a danger that the organisation might lose control.



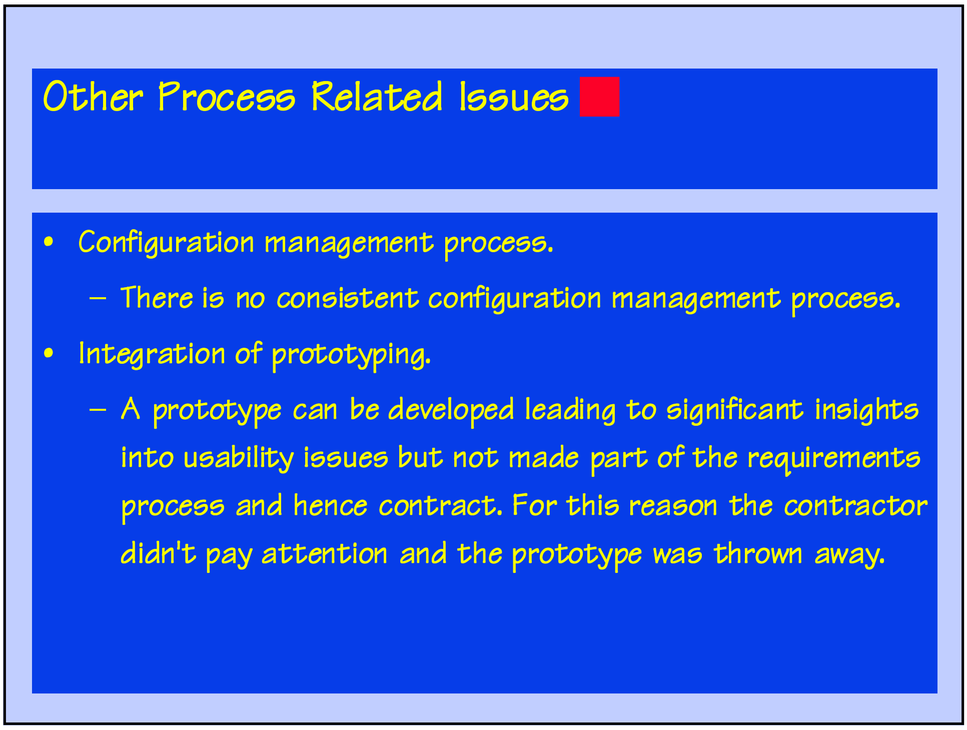
Stakeholder Issues

- Stakeholder identification.
  - The task of identifying stakeholders who ought to have a voice in the requirements of the system is not handled systematically.
- Stakeholder empowerment.
  - It is not sufficient to identify stakeholders. They need to be empowered to act. This includes being given resources to participate in the requirements engineering process.



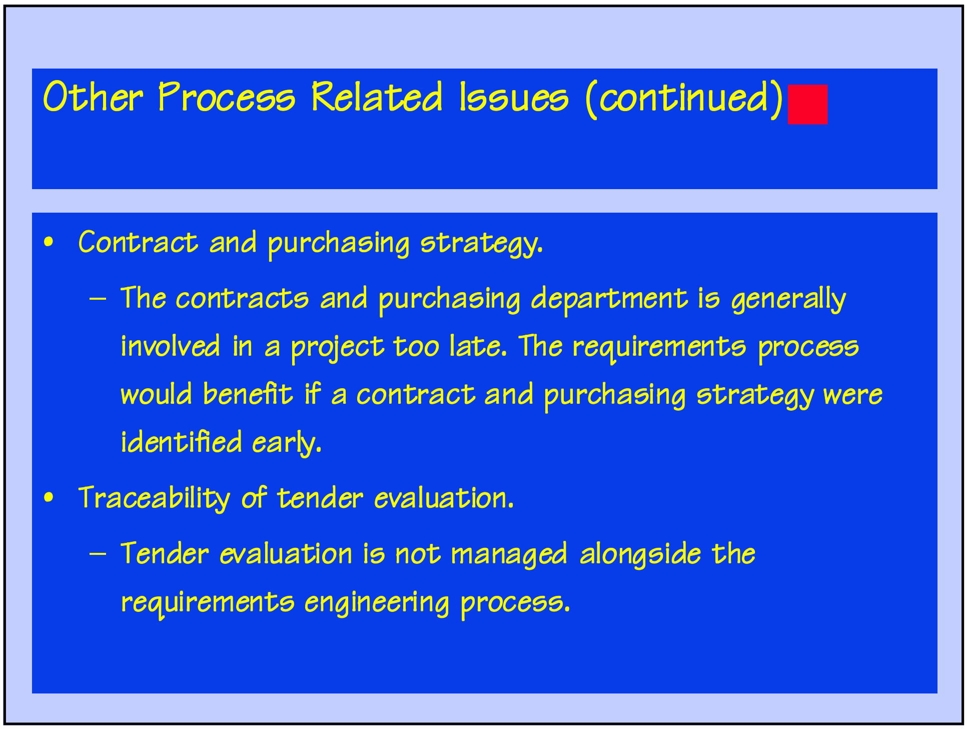
Stakeholder Issues (continued)

- Stakeholder involvement.
  - The involvement of stakeholders in the requirements engineering process must be continued for the life of that process. A “one-shot” involvement cannot be effective.
- Requirements ownership.
  - No owner of a requirement is explicitly determined.



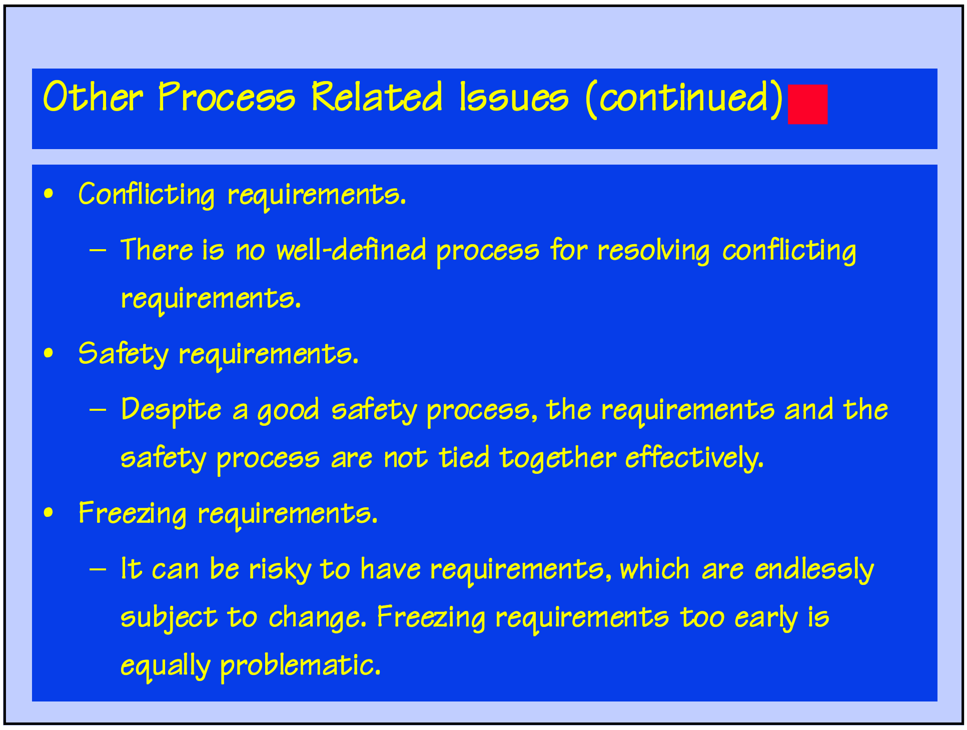
Other Process Related Issues

- Configuration management process.
  - There is no consistent configuration management process.
- Integration of prototyping.
  - A prototype can be developed leading to significant insights into usability issues but not made part of the requirements process and hence contract. For this reason the contractor didn't pay attention and the prototype was thrown away.



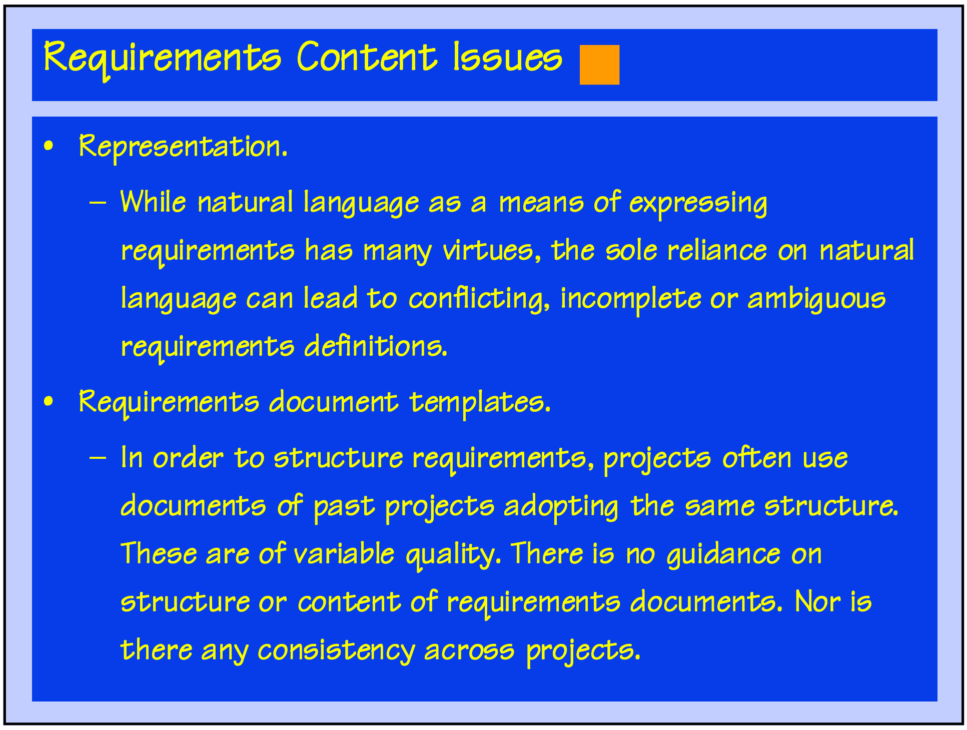
Other Process Related Issues (continued)

- Contract and purchasing strategy.
  - The contracts and purchasing department is generally involved in a project too late. The requirements process would benefit if a contract and purchasing strategy were identified early.
- Traceability of tender evaluation.
  - Tender evaluation is not managed alongside the requirements engineering process.



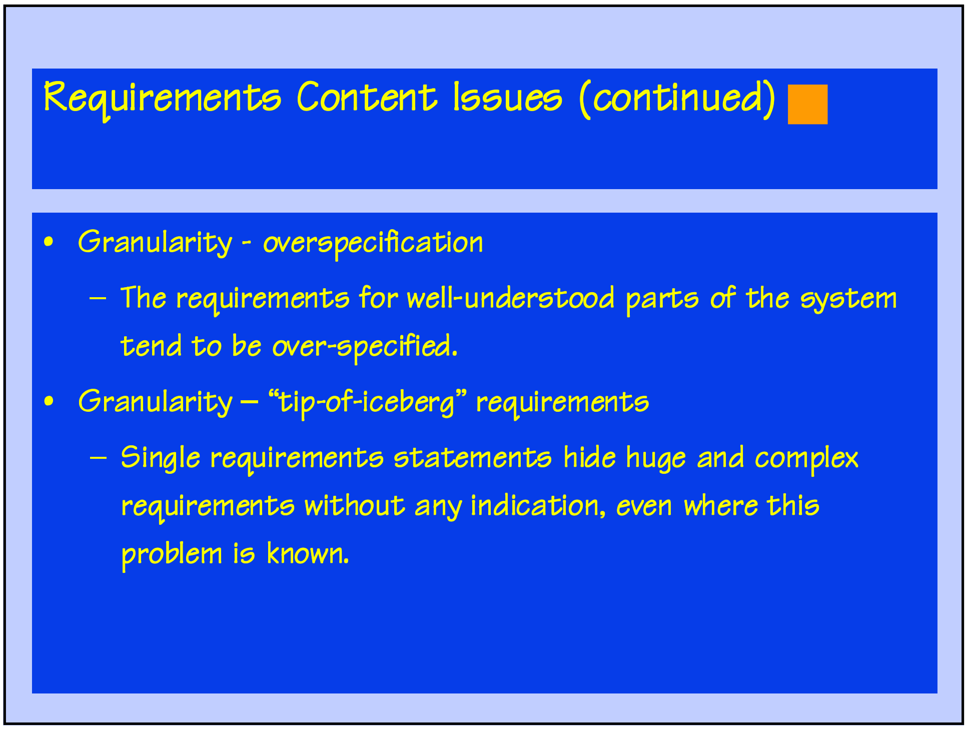
Other Process Related Issues (continued)

- Conflicting requirements.
  - There is no well-defined process for resolving conflicting requirements.
- Safety requirements.
  - Despite a good safety process, the requirements and the safety process are not tied together effectively.
- Freezing requirements.
  - It can be risky to have requirements, which are endlessly subject to change. Freezing requirements too early is equally problematic.



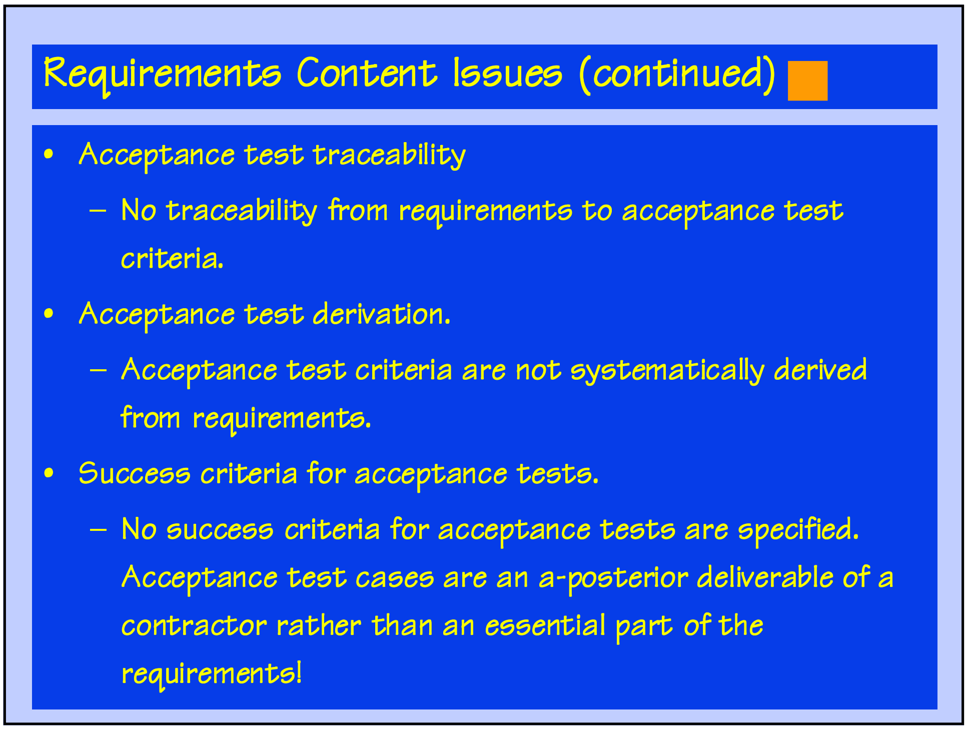
Requirements Content Issues

- Representation.
  - While natural language as a means of expressing requirements has many virtues, the sole reliance on natural language can lead to conflicting, incomplete or ambiguous requirements definitions.
- Requirements document templates.
  - In order to structure requirements, projects often use documents of past projects adopting the same structure. These are of variable quality. There is no guidance on structure or content of requirements documents. Nor is there any consistency across projects.



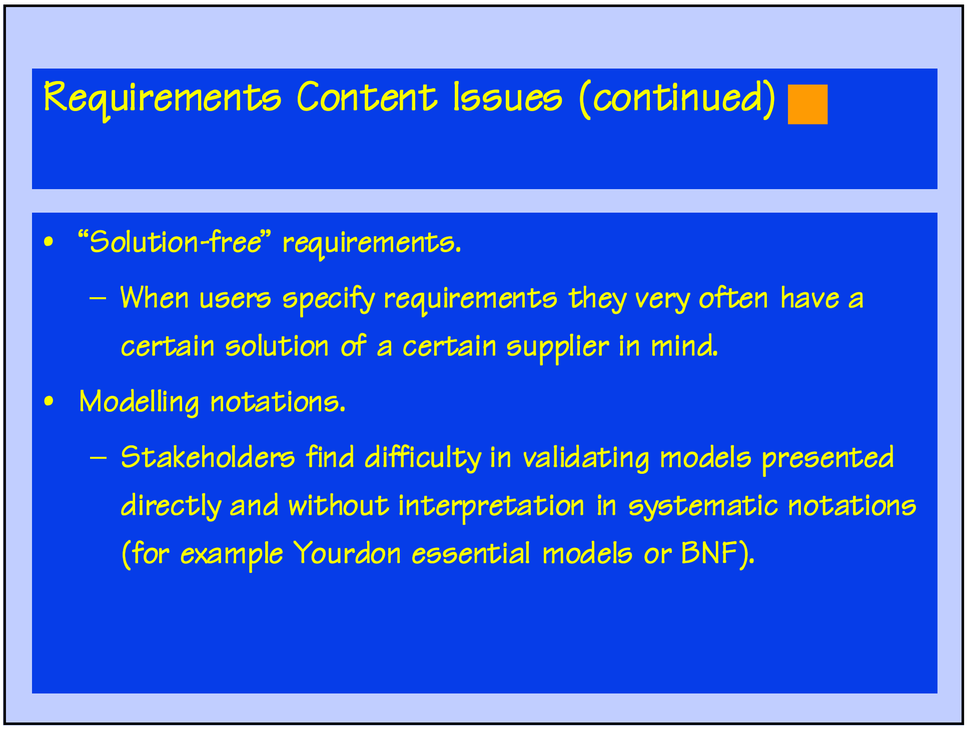
Requirements Content Issues (continued)

- Granularity - overspecification
  - The requirements for well-understood parts of the system tend to be over-specified.
- Granularity – "tip-of-iceberg" requirements
  - Single requirements statements hide huge and complex requirements without any indication, even where this problem is known.



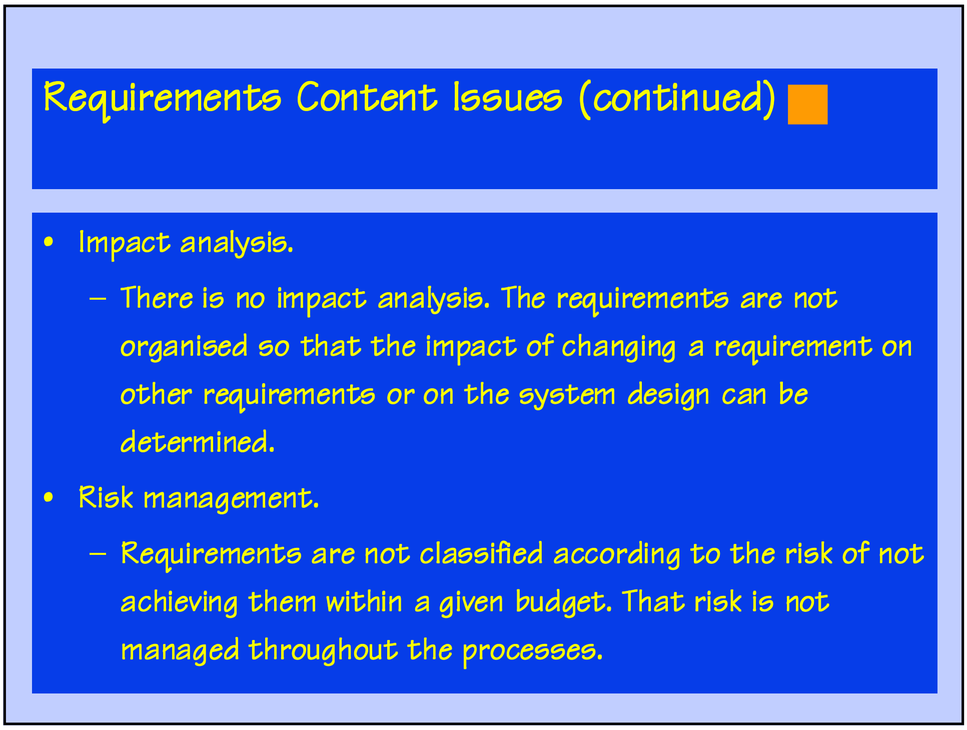
Requirements Content Issues (continued)

- Acceptance test traceability
  - No traceability from requirements to acceptance test criteria.
- Acceptance test derivation.
  - Acceptance test criteria are not systematically derived from requirements.
- Success criteria for acceptance tests.
  - No success criteria for acceptance tests are specified. Acceptance test cases are an a-posterior deliverable of a contractor rather than an essential part of the requirements!



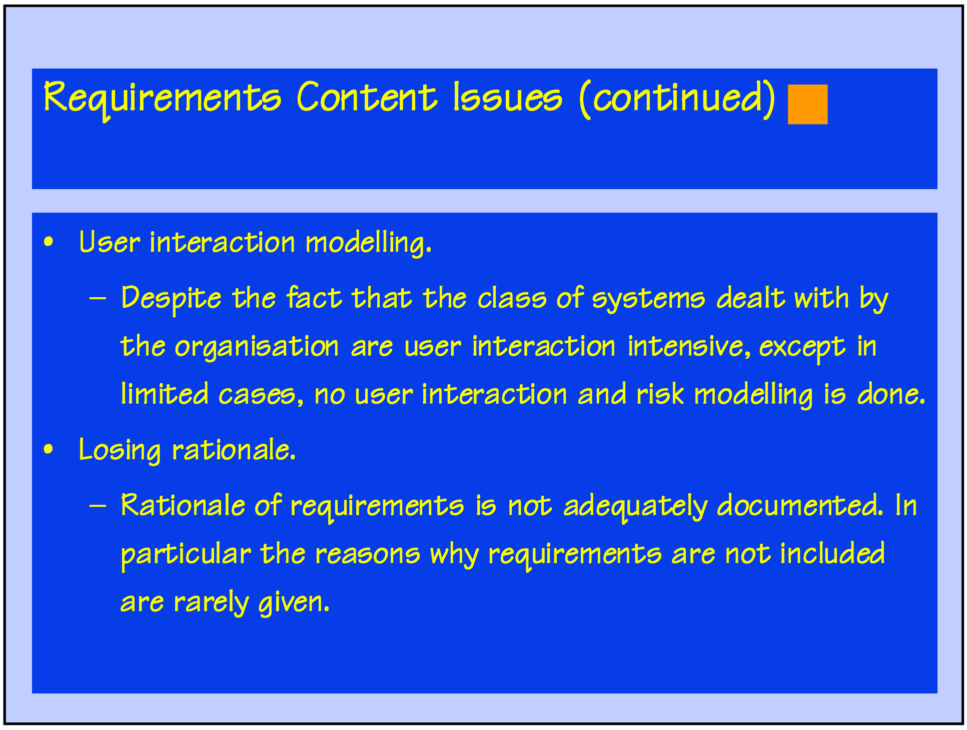
Requirements Content Issues (continued)

- “Solution-free” requirements.
  - When users specify requirements they very often have a certain solution of a certain supplier in mind.
- Modelling notations.
  - Stakeholders find difficulty in validating models presented directly and without interpretation in systematic notations (for example Yourdon essential models or BNF).



Requirements Content Issues (continued)

- Impact analysis.
  - There is no impact analysis. The requirements are not organised so that the impact of changing a requirement on other requirements or on the system design can be determined.
- Risk management.
  - Requirements are not classified according to the risk of not achieving them within a given budget. That risk is not managed throughout the processes.



Requirements Content Issues (continued)

- User interaction modelling.
  - Despite the fact that the class of systems dealt with by the organisation are user interaction intensive, except in limited cases, no user interaction and risk modelling is done.
- Losing rationale.
  - Rationale of requirements is not adequately documented. In particular the reasons why requirements are not included are rarely given.

Requirements Content Issues (continued)

- Relationships between requirements.
  - Relationships between requirements are not identified and maintained.
- Non-Functional-Requirements and system-wide properties.
  - The requirements specifications are organised around functional blocks. Non-functional requirements and system-wide properties are not managed or tied into the requirements process. They are frequently omitted or repeated inappropriately for each functional block.

Requirements Content Issues (continued)

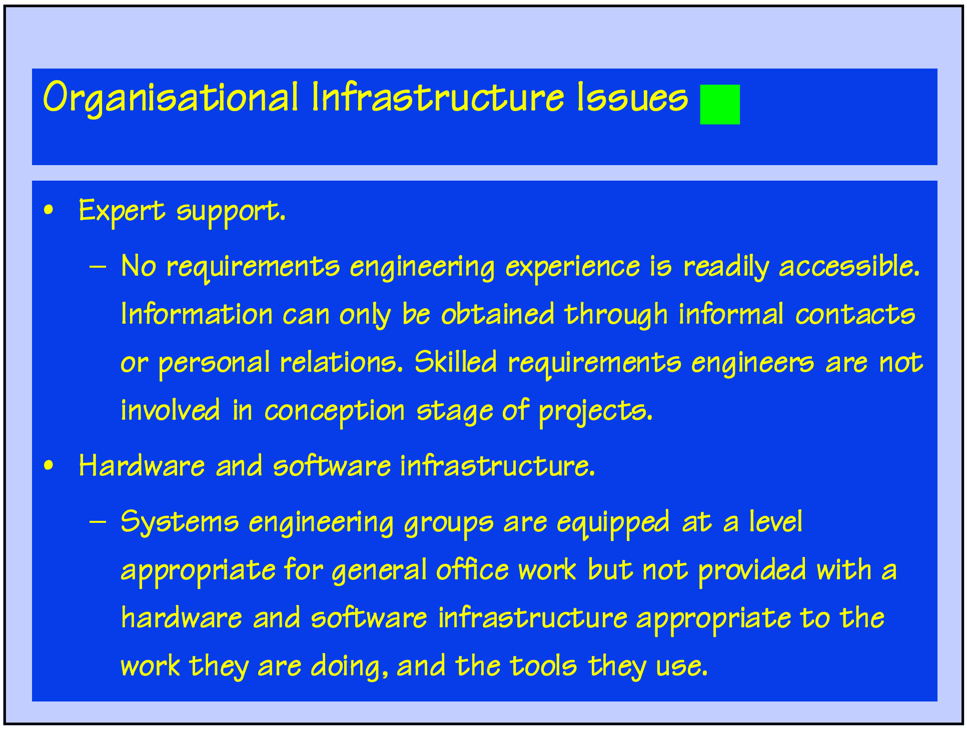
- Precision vs. readability.
  - Because the natural language text is the only carrier used to express requirements information, the text is forced to be more precise than it can be naturally achieved.

Requirements Content Issues (continued)

- Prioritisation of requirements.
  - Some requirements are more important than others in terms of the benefits they deliver to stakeholders. The importance of requirements is not identified.
- Yourdon essential models.
  - The development of Yourdon essential models submerged the system architecture, which was represented by functional blocks before.

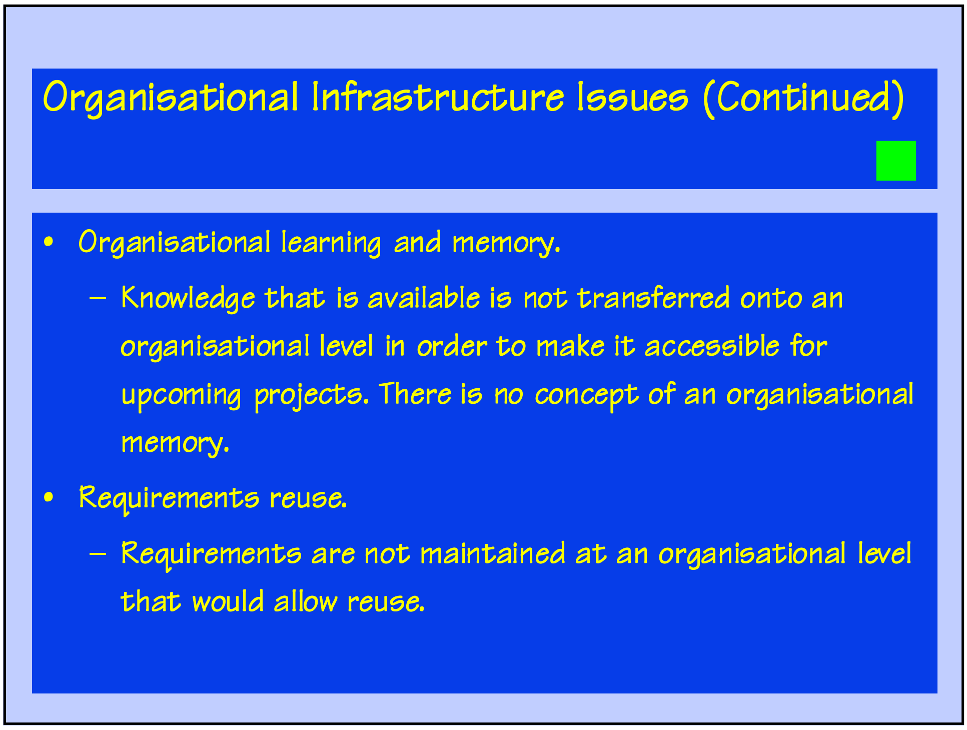
Domain-related Issues

- Concept of Operation.
  - Lack of a documented concept of operation or “domain model”.
- System Architecture.
  - The system architecture for the overall service the organisation provides, regarded by all analysts as an important organising principle of requirements documents, is implicit and it is not consistently shared across projects.



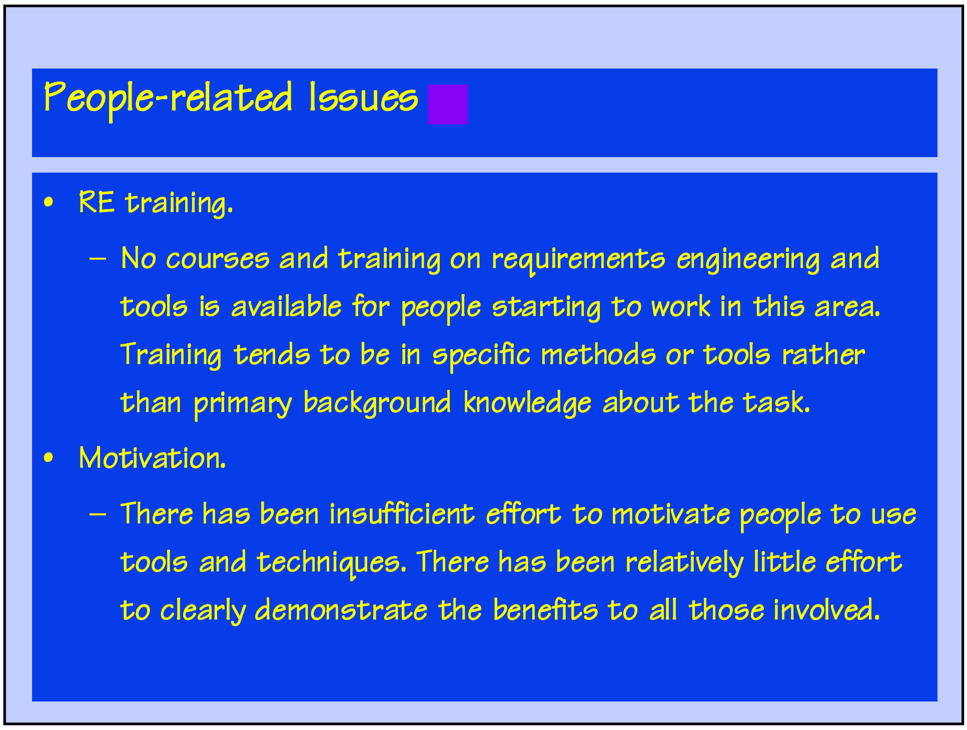
Organisational Infrastructure Issues

- Expert support.
  - No requirements engineering experience is readily accessible. Information can only be obtained through informal contacts or personal relations. Skilled requirements engineers are not involved in conception stage of projects.
- Hardware and software infrastructure.
  - Systems engineering groups are equipped at a level appropriate for general office work but not provided with a hardware and software infrastructure appropriate to the work they are doing, and the tools they use.



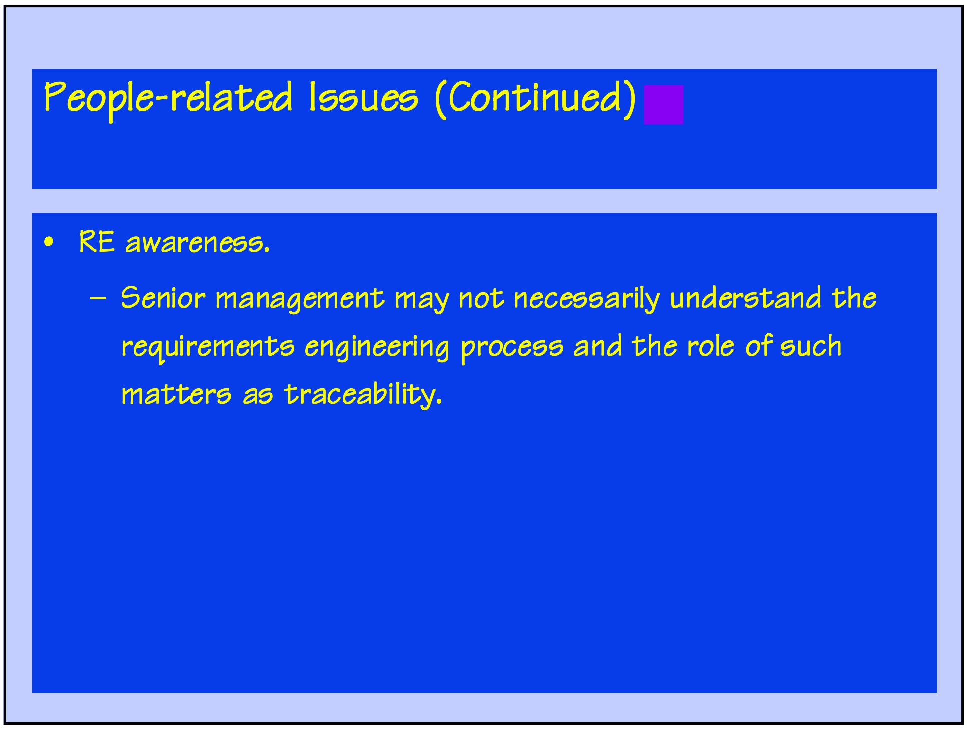
Organisational Infrastructure Issues (Continued)

- Organisational learning and memory.
  - Knowledge that is available is not transferred onto an organisational level in order to make it accessible for upcoming projects. There is no concept of an organisational memory.
- Requirements reuse.
  - Requirements are not maintained at an organisational level that would allow reuse.



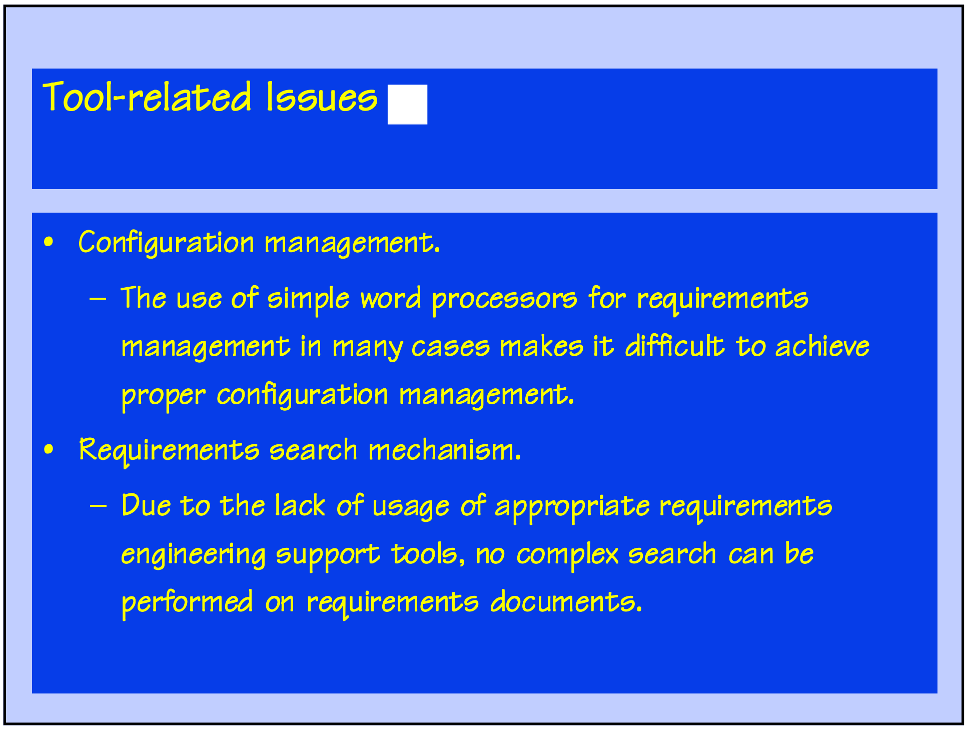
People-related Issues

- RE training.
  - No courses and training on requirements engineering and tools is available for people starting to work in this area. Training tends to be in specific methods or tools rather than primary background knowledge about the task.
- Motivation.
  - There has been insufficient effort to motivate people to use tools and techniques. There has been relatively little effort to clearly demonstrate the benefits to all those involved.



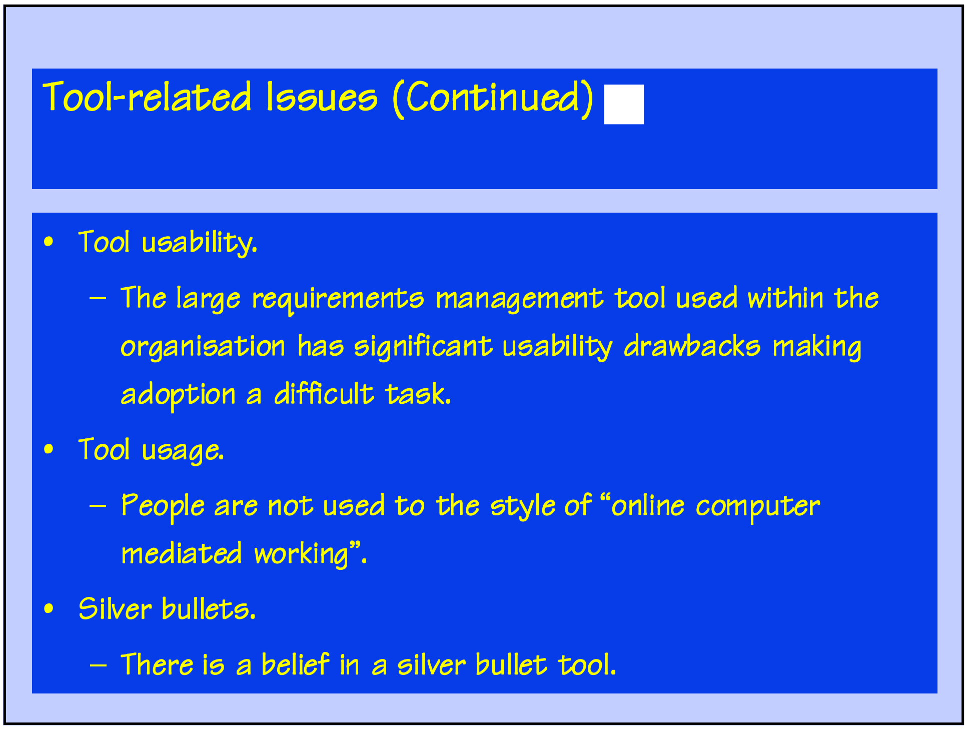
People-related Issues (Continued)

- RE awareness.
  - Senior management may not necessarily understand the requirements engineering process and the role of such matters as traceability.



Tool-related Issues

- Configuration management.
  - The use of simple word processors for requirements management in many cases makes it difficult to achieve proper configuration management.
- Requirements search mechanism.
  - Due to the lack of usage of appropriate requirements engineering support tools, no complex search can be performed on requirements documents.



Tool-related Issues (Continued)

- Tool usability.
  - The large requirements management tool used within the organisation has significant usability drawbacks making adoption a difficult task.
- Tool usage.
  - People are not used to the style of “online computer mediated working”.
- Silver bullets.
  - There is a belief in a silver bullet tool.

What we Know about RE [1]

management context
contract & procurement

- The lesson of the CMM! Software process improvements are interlocking. If you don't have "commitment control", don't waste your time and money on requirements engineering!
- Contract and procurement procedures are about managing the relationships between customer and supplier. The work of requirements engineering takes place in the "space" between customer and supplier. It may be necessary to "reengineer" contract and procurement procedures to: promote relationships based on partnership and risk sharing; support continuing and direct interaction between customer and supplier.

What we Know about RE [2]

individual performance

- There is good empirical evidence for the existence of striking differences in programming ability, similar results are reported for software design. There is strong anecdotal evidence of significant differences in individual performance at requirements engineering tasks. The most important step we can take in order to improve performance at requirements engineering is to select the right people. The key appears to be personal communication and group facilitation skills, generally accompanied by a sensitive appreciation of organisational politics.

What we Know about RE [3]

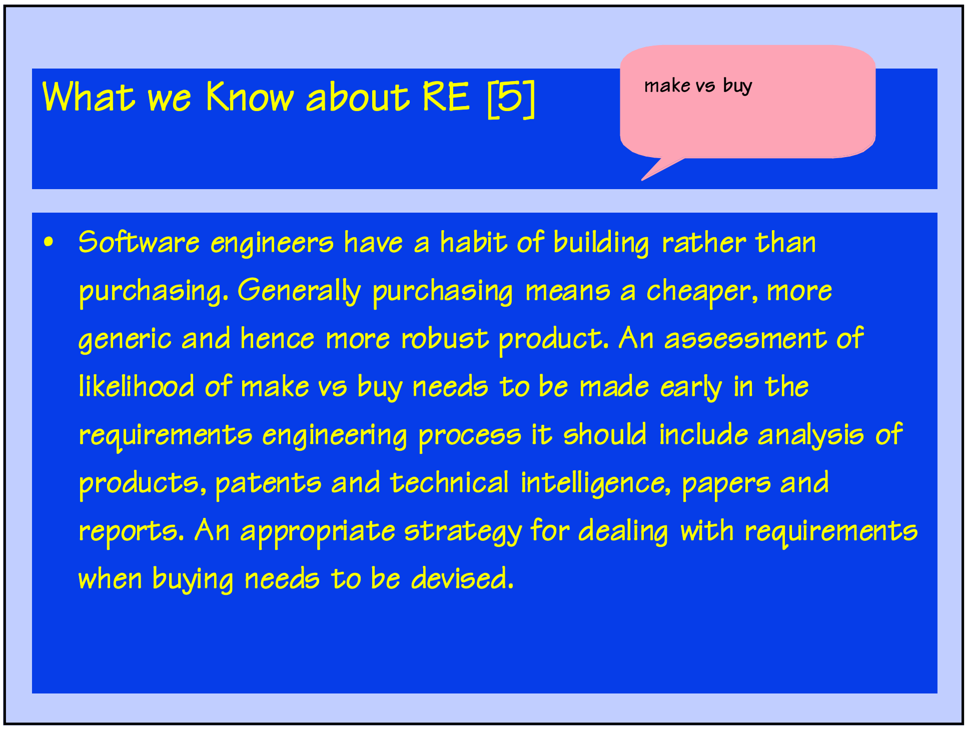
organisational setting

- Requirements engineering takes place in different organisational settings (internal, bespoke, customisation, cooperative, generic/market). The problems of requirements engineering are different in each of these settings. You cannot expect to use methods and processes from one setting in another without significant adaptation. When the setting changes you need to change your requirements engineering practices.

What we Know about RE [4]

bounding

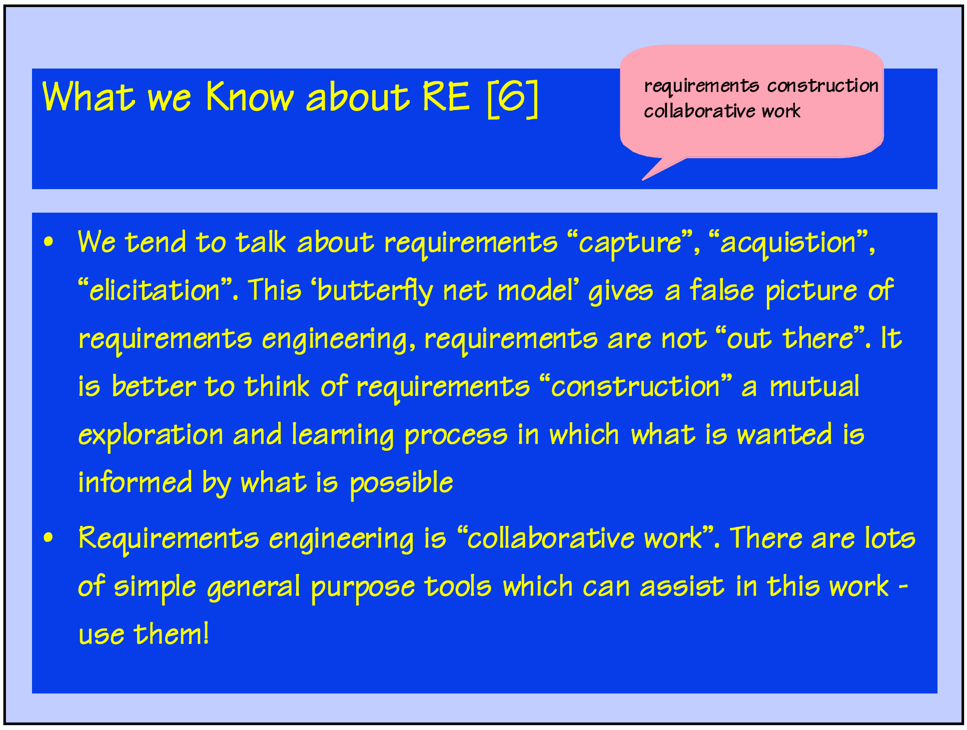
- Establishing the scope and delineating the bounds of the requirements and design spaces is the most difficult and critical part of requirements engineering. Decisions on bounding must be explicit and clearly rationalised. The following tests should be applied and documented: investigation resources; competence; freedom of action; missed solution cost; robustness.



What we Know about RE [5]

make vs buy

- Software engineers have a habit of building rather than purchasing. Generally purchasing means a cheaper, more generic and hence more robust product. An assessment of likelihood of make vs buy needs to be made early in the requirements engineering process it should include analysis of products, patents and technical intelligence, papers and reports. An appropriate strategy for dealing with requirements when buying needs to be devised.



What we Know about RE [6]

requirements construction
collaborative work

- We tend to talk about requirements “capture”, “acquistion”, “elicitation”. This ‘butterfly net model’ gives a false picture of requirements engineering, requirements are not “out there”. It is better to think of requirements “construction” a mutual exploration and learning process in which what is wanted is informed by what is possible
- Requirements engineering is “collaborative work”. There are lots of simple general purpose tools which can assist in this work - use them!

What we Know about RE [7]

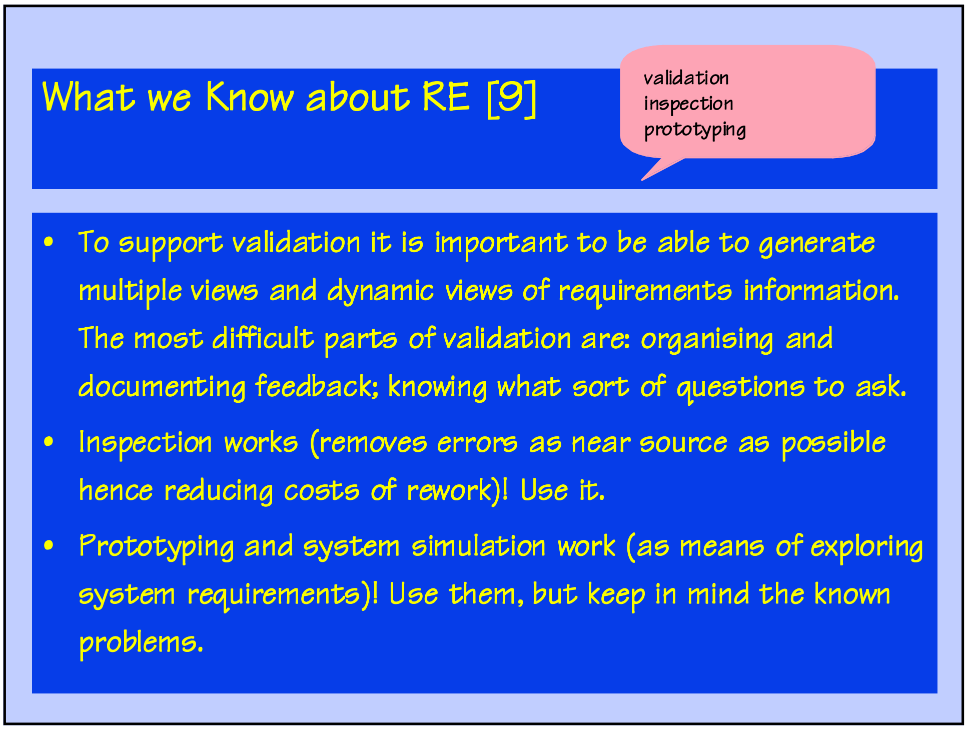
stakeholder identification
information gathering

- The most common error in requirements engineering is to forget/omit/lose important stakeholders. It is important to create a “map” as a means of identifying stakeholders and interpreting the information provided by stakeholders and its status. This map should identify responsibilities, capacities and organisational relations.
- Gathering information on the requirements and on the domain in which they are situated is the characteristic activity of requirements engineering. There are better ways of gathering requirements than simply asking for them.

What we Know about RE [8]

modelling

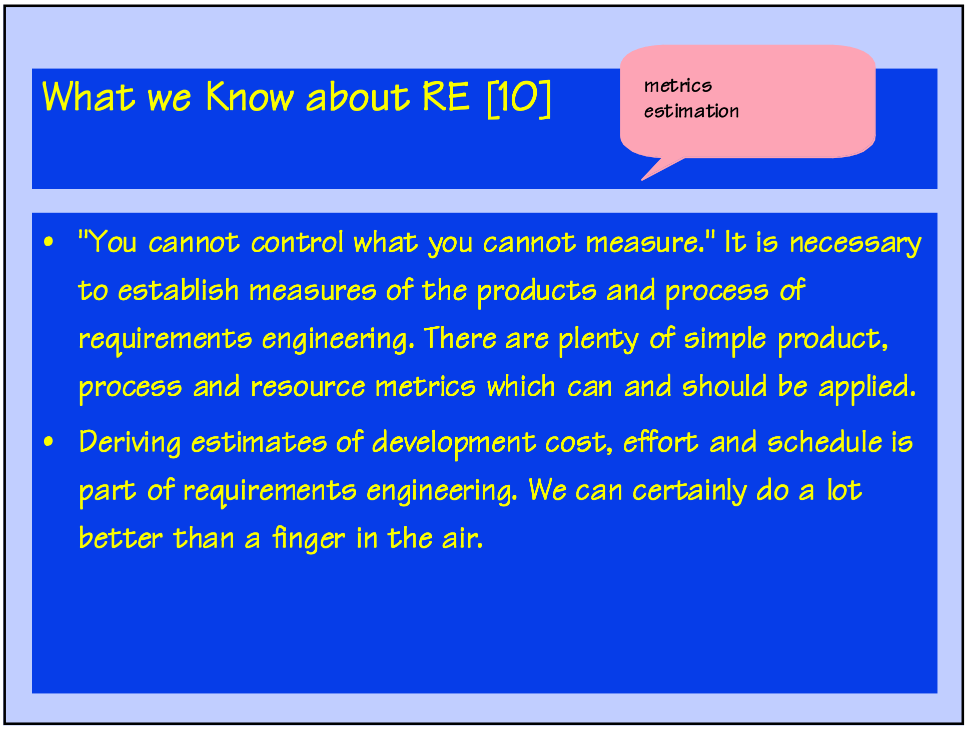
- In order to render "raw" requirements usable they need to be organised through a process of modelling and specification. We need to produce 3 types of model: a system model which identifies the services the system is to provide and the assumptions that have to be made about the operational domain in order to provide those services; a task model which identifies the users and the tasks the users perform; a value model which identifies those properties of the service relevant to the fulfilment of the stakeholder requirements. To build these models we need to be able to talk about agents, goals, events, actions, objects and preferences.



What we Know about RE [9]

validation
inspection
prototyping

- To support validation it is important to be able to generate multiple views and dynamic views of requirements information. The most difficult parts of validation are: organising and documenting feedback; knowing what sort of questions to ask.
- Inspection works (removes errors as near source as possible hence reducing costs of rework)! Use it.
- Prototyping and system simulation work (as means of exploring system requirements)! Use them, but keep in mind the known problems.



What we Know about RE [10]

metrics
estimation

- "You cannot control what you cannot measure." It is necessary to establish measures of the products and process of requirements engineering. There are plenty of simple product, process and resource metrics which can and should be applied.
- Deriving estimates of development cost, effort and schedule is part of requirements engineering. We can certainly do a lot better than a finger in the air.

What we Know about RE [11]

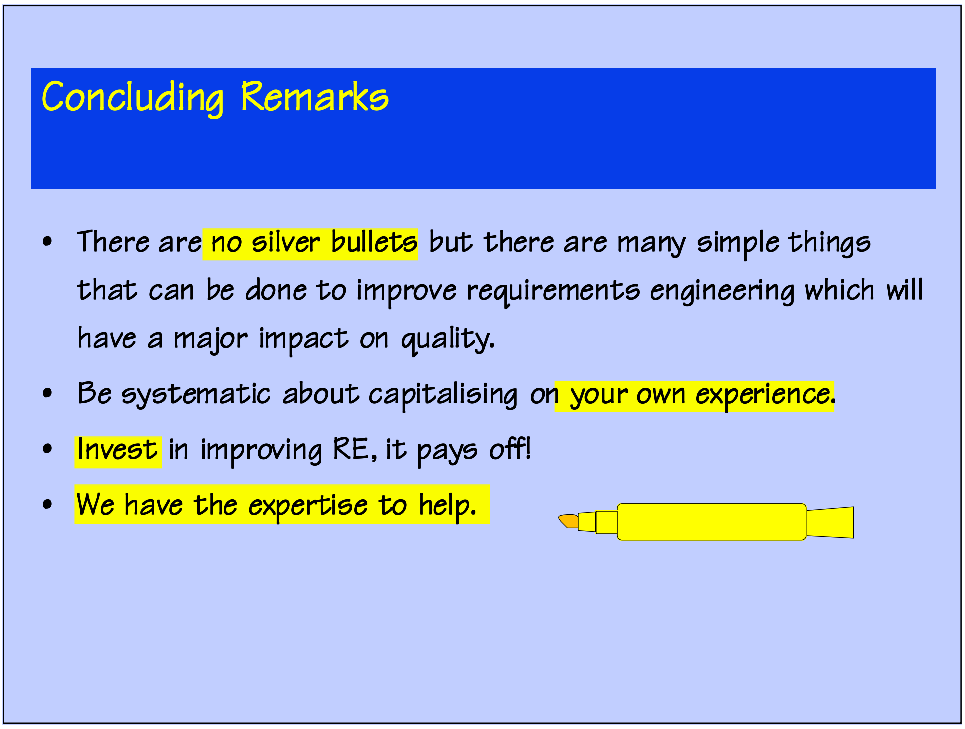
rationale

- The current RE process is artifact-oriented great emphasis is placed on the creation and tracking of “products” however more than 70% of software development costs are in maintenance and rework and half the effort in these activities is about *understanding* the process which lead to those products so as to make effective corrections and enhancements. In order to achieve this understanding you need to know what decisions were considered, assumptions made, alternatives posited. This information is rationale, it may be remembered but with time and staff turnover it soon gets lost. There are many schemes for recording rationale which are simple, proven and available.

What we Know about RE [12]

traceability

- Requirements traceability (RT) refers to the ability to describe and follow the life of a requirement in both a forwards and backwards direction. A variety of highly effective tools to support RT have been developed however the majority of RT problems are not technical problems, but human and organisational problems. Effective RT requires: one hundred percent commitment from all the stakeholders; dedicated up-front resourcing; process wide coordination. We need to find ways to overcome the provider - end user conflict.



Concluding Remarks

- There are no silver bullets but there are many simple things that can be done to improve requirements engineering which will have a major impact on quality.
- Be systematic about capitalising on your own experience.
- Invest in improving RE, it pays off!
- We have the expertise to help.