# CITS5502 – Requirements engineering process

Unit coordinator: Arran Stewart

## Sources

- Sommerville
- Sawyer et al, “Improving the Requirements Process”

# Requirements engineering

# Requirements engineering

What is requirements engineering?

- The process of establishing the services a customer requires from a system and the constraints under which it operates and is developed.

# Types of requirement

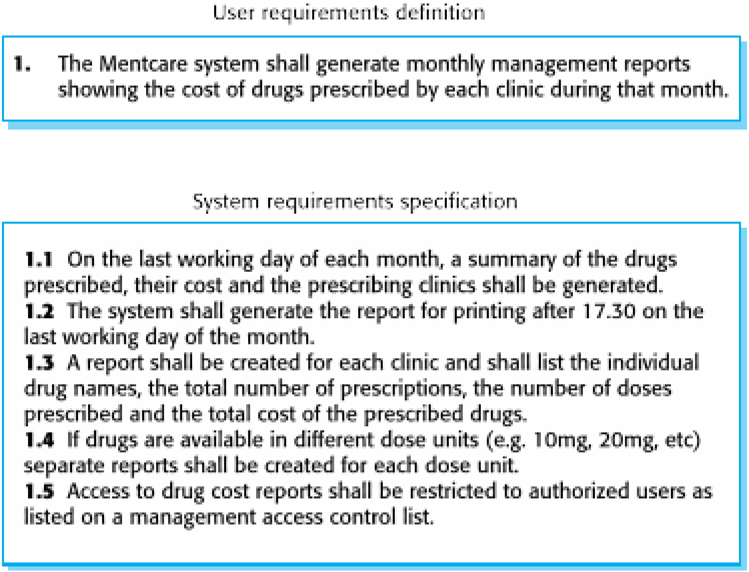
User requirements

- Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.

System requirements

- A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.

# Types of requirement – examples



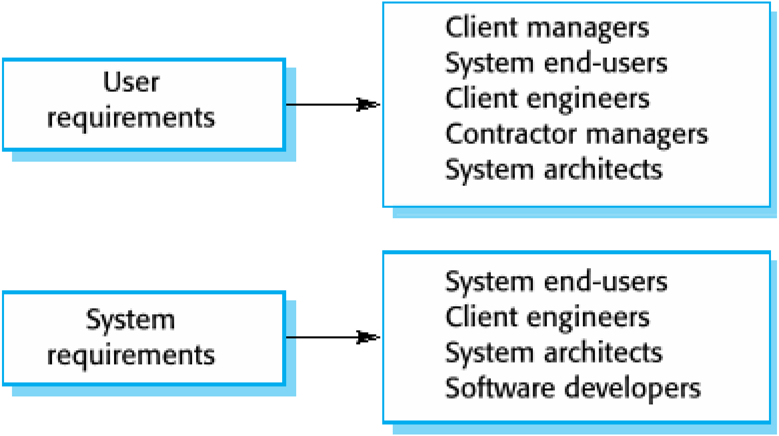
User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

**1.1** On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
**1.2** The system shall generate the report for printing after 17.30 on the last working day of the month.
**1.3** A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
**1.4** If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
**1.5** Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

# Audience for requirements



# System stakeholders

- Any person or organization who is affected by the system in some way and so who has a legitimate interest
- Stakeholder types
  - End users
  - System managers
  - System owners
  - External stakeholders

# The requirements engineering process

The first step in any software process is to find out what the system is intended to achieve.

- This is true irrespective of the development paradigm and model in use.
- Some use an iterative (prototyping) method to define the requirements
- Agile methodologies adopt an evolutionary approach of determining and then building small pieces of functionality at a time.

Regardless, this ends up involving some sort of iteration.

# Agile methods and requirements

- Many agile methods argue that producing detailed system requirements is a waste of time as requirements change so quickly.
- The requirements document is therefore always out of date.
- Agile methods usually use incremental requirements engineering and may express requirements as "user stories"
- This is feasible for business systems but problematic for systems that require pre-delivery analysis (e.g. critical systems) or systems developed by several teams.

# Functional vs non-functional requirements

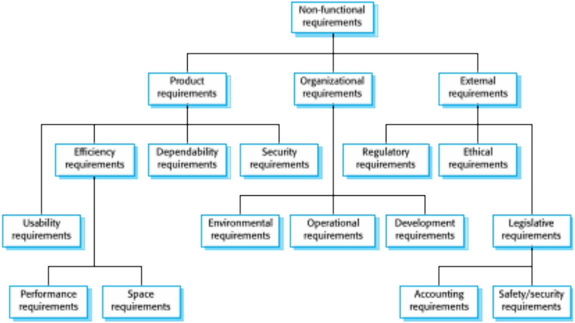
Functional requirements:

- Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- May state what the system should *not* do

Non-functional requirements:

- *Constraints* on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- Often apply to the system as a whole rather than individual features or services.

# Types of non-functional requirements



# Implementing non-functional requirements

- Non-functional requirements may affect the overall architecture of a system rather than the individual components.
    - For example, to ensure that performance requirements are met, you may have to organize the system to minimize communications between components.
- A single non-functional requirement, such as a security requirement, may generate a number of related functional requirements that define system services that are required.
    - It may also generate requirements that restrict existing requirements.

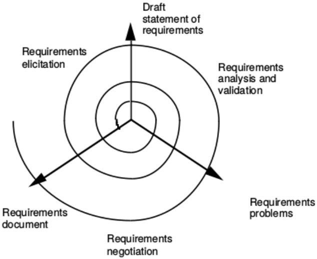
# Requirements engineering processes

# Requirements engineering processes

The processes used for RE vary widely depending on the application domain, the people involved and the organisation developing the requirements. However, there are a number of generic activities common to all processes Requirements elicitation; Requirements analysis; Requirements validation; Requirements management. In practice, RE is an iterative activity in which these processes are interleaved.

# The requirements engineering process

- The requirements process tends to be *cyclical*.
- Requirements are *elicited*; then are integrated and analysed; and problems discovered during anallysis need to be resolved; which may lead to a need for further elicitation.



# Elicitation

Requirements elicitation:

Given a statement of organisational *needs*, various requirements sources are consulted to understand the problem and the application domain.

Requirements sources can include:

- stakeholders
- domain experts
- operating regulations

But the resulting requirements may be incomplete, vaguely expressed and unstructured.

# Analysis and validation

Requirements analysis and validation:

Requirements discovered during the elicitation phase are integrated and analysed.

Problems such as missing information, ambiguities, vagueness and inconsistencies are identified.

# Negotiation

Requirements negotiation:

The problems discovered during analysis need to be resolved.

- Analysts and stakeholders clarify their understanding and consider possible solutions.
- This may require negotiation to establish the necessary trade-offs
- Additional cycles may be needed.

# Difficulties with eliciting requirements

Requirements are hard to elicit.

- The customer may not have a clear view of what they need – assistance may be needed to identify and state the requirements.
- Often, “the customer” consists of a number of different stakeholders with different concerns
- Some requirements only emerge after analysis

# Difficulties with eliciting requirements, cont'd

Requirements change over time.

- The customer's understanding of what they need may change
- The business may evolve during the course of a project
- Changes in the business environment may force reassessment

# Difficulties with eliciting requirements, cont'd

Time and cost constraints.

The requirements phase has a finite budget – usually only a small proportion of the overall budget

For large projects, the work may need to be *partitioned* early on, before requirements are fully understood.

(Meaning errors may be costly to rectify.)

# Difficulties with eliciting requirements, cont’d

As a consequencce, requirements often need to be re-worked – one estimate is that 40% of requirements require rework during the course of a development project.

# Techniques

Many techniques and good practices exist for requirements engineering.

We will consider some of the techniques used in *elicitation*.

- Interviews – can be
  - unstructured (interviewer places only limited control on direction)
  - structured (using predetermined questions)
  - some combination of these
- Questionnaires – most useful after terms and concepts in the domain are already established and understood by participants and analysts
- Task analysis – decomposing tasks into subtasks until all actions and events are described.

# Techniques, cont'd

- Domain analysis – examining existing documentation and applications, identifying reusable concepts and components.
    - Especially important when intending to replace a legacy system
- Brainstorming – stakeholders from different groups meet to rapidly generate many ideas

# Improving a requirements engineering process

Many many other techniques and good practices which exist, and these can resolve some of the challenges of the requirements engineering process.

But requirements engineering is still seen as difficult, because:

- Analysts may be unaware of what techniques exist.
- Analysts may be unsure how to evaluate the suitability of new techniques
- Even if a technique is identified as potentially useful, adopting it and integrating it into the existing requirements process can be challenging.

# Improving a requirements engineering process, cont'd

Literature exists which attempts to place techniques in the context of a *process improvement model*.

For different techniques, it's useful to identify:

- What is the technique or practice?
- What is the cost of introduction?
- What is the cost of application (i.e. use)?
- What sort of expertise is required to introduce?
- What are the key benefits?

# Improving a requirements engineering process, cont'd

*Example:*

Technique: Identify *volatile* requirements.

Cost of introduction: Low
Cost of application: Low
Expertise needed: Advanced
Key benefits: Simplifies requirements change management

# Improving a requirements engineering process, cont'd

We will look in more detail at process improvement in later weeks.