Objectives

- To introduce the concepts of user and system requirements
- To describe functional and non-functional requirements
- To explain how software requirements may be organised in a requirements document

Last lectures
Requirements analysis and definition
- The process of establishing what services are required and the constraints on the system's operation and development.
  - What is the system about?
- Requirements engineering process
  - Feasibility study;
  - Requirements elicitation and analysis;
  - Requirements specification;
  - Requirements validation.

Requirements Engineering Process
- The requirements themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process.
- It may range from a high-level abstract statement of a service or of a system constraint to a detailed functional specification.
Requirements: "General" Statement

- The system will maintain records of all library items including books, serials, newspapers, magazines, video and audio tapes.
- No item shall be removed from the library without the details of its borrowing being recorded in the system.
- All items shall have a bar code containing a unique reference number.

Could be "general" statements showing how the system should achieve, interact with the user, other systems, and environment...

Requirements: "Detailed" Statements

- Could be "detailed" statements of the system’s functionality

**FUNCTIONAL REQUIREMENTS**

- The system shall permit all users to search for an item by title, by author, by ISBN.
- Borrowed items that are one day overdue shall cause a reminder e-mail to the borrower.

- Could be statements of the practical constraints or limitations within which the system must operate --- Non-FUNCTIONAL REQUIREMENTS

- The system shall respond to a transaction requests from a user within 1.5 seconds.

- Could be statements on how the system to be implemented --- Implementation REQUIREMENTS

- When an item is borrowed or returned, it should be scanned through a card reader.

Eliciting Requirements

- Process of "capturing" or "discovering" requirements
  - Stakeholder consultations (interviews)
  - Scenarios (i.e., showing state of the system and flow of activities and events)
  - Observations
  - Revising existing documentations, manual system etc.
Definitions and Specifications

- Statements in natural language...
- Perhaps, expressed in mathematical model...
- Noted in Diagrams showing what the system provides and its operational constraints, showing behaviour, interaction...
- Perhaps, written in a formal language
- Or perhaps, written in Structured English

Problems with Requirements Specification

- **Ambiguity**
  - The readers and writers of the requirement must interpret the same words in the same way. NL is naturally ambiguous so this is very difficult.
- **Over-flexibility**
  - The same thing may be said in a number of different ways in the specification.
- **Lack of modularisation**
  - NL structures are inadequate to structure system requirements.
- **Others** - inconsistency, missing requirements, stakeholders' bias, incompleteness, redundancy, irrelevance, overloaded statements...
**Functional & Non-Functional Requirements**

- **Functional requirements**
  - Statements of services the system should provide, how the system should react to particular inputs
- **Non-functional requirements**
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.

**Examples of Functional Requirements**

- The library system *shall* provide a facility for identifying the identity of a library user
- The library system *shall* provide a reminder when the book is overdue

**Non-functional Requirements**

- These define system properties and constraints
  - E.g., reliability, security, availability, etc., response time
  - Storage requirements. Constraints are I/O device capability, distribution paradigm, etc.
  - Use of particular systems, programming language or development methods, etc.
  - Non-functional requirements *may be more critical* than functional requirements. If these are not met, the system is useless.

\[A\ constraint\ on\ how\ the\ functional\ requirements\ may\ be\ implemented\]
**Non-functional Requirements**

- The library system *shall* authenticate a library customer in five seconds or less
- The library system *shall* be available 24/7
- The library system *shall* abide to ISO standards

**Non-functional Classifications**

- **Product requirements**
  - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- **Organisational requirements**
  - Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc.
- **External requirements**
  - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

**Non-functional Requirement Types**

Diagram showing various types of non-functional requirements, including:
- Performance requirements
- Security requirements
- Robustness requirements
- Portability requirements
- Interoperability requirements
- Reliability requirements
- Usability requirements
- Information requirements
- Technical requirements
Non-functional Requirements: Dimensions of Dependability

- Availability: The ability of the system to deliver services when requested.
- Reliability: The ability of the system to deliver services as specified.
- Safety: The ability of the system to operate without catastrophic failure.
- Security: The ability of the system to protect itself against accidental or deliberate intrusions.

Other Dependability Properties

- Repairability: Reflects the extent to which the system can be repaired in the event of a failure.
- Maintainability: Reflects the extent to which the system can be adapted to new requirements.
- Survivability: Reflects the extent to which the system can deliver services whilst under hostile attack.
- Error tolerance: Reflects the extent to which user input errors can be avoided and tolerated.

Requirements Engineering Process
Requirements Document

- A structured document setting out detailed descriptions of the system’s functions, services and operational constraints.
- Should include both a definition of user requirements and a specification of the system requirements.
- It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it.

Defines what should be implemented so may be part of a contract between client and contractor.

Users of a Requirements Document

- System developers
- Users
- Managers
- Project managers
- Documentation
- Management
- Software engineers

Definitions and Specifications

User requirement definition

1. The software must provide a means of representing and accessing external files created by other tools.

System requirement specification

1.1 This user should be provided with facilities to define this type of external file.
1.2 Each external file type may have an associated tool which may be supplied by the file.
1.3 Each external file type may be represented as a specific icon on the user's display.
1.4 Facilities should be provided for the user representing an external file type to be defined by the user.
1.5 When a user selects an icon representing an external file, the effect of that selection is to display the tool associated with that type of the external file to the file represented by the selected icon.
MoSCoW Criteria

Often used in requirements prioritization and as a language for specifying requirements
- **M**: Must have - mandatory requirements that are fundamental to the system
- **S**: Should have - important requirements that could be omitted
- **C**: Could have - optional requirements
- **W**: Want to have - these requirements really can wait (i.e. bells and whistles)

### Example Format

<table>
<thead>
<tr>
<th>ID</th>
<th>Functional Requirements</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The system shall...</td>
<td>M</td>
</tr>
<tr>
<td>2</td>
<td>The system shall ....</td>
<td></td>
</tr>
</tbody>
</table>

### Non-functional Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Non-functional Requirements</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The system shall...</td>
<td>M</td>
</tr>
<tr>
<td>2</td>
<td>The system shall ....</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Performance</td>
<td></td>
</tr>
</tbody>
</table>
Why are Requirements Important?

- "The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later."
  
  [Fred Brooks in the "Mythical Man Month"]

Read this article!

Exercise

- Determine a set of functional and non-functional requirements for a library system
- Work in a group of two…
- What you have to do
  - Express the functional & non-functional requirements with a unique ID for traceability
  - Group your requirements into sensible sets (e.g., user interface, borrowing, browsing)
- Prioritise your requirements