Requirements Engineering

1 Gathering Requirements

- A software requirement is a functional or non-functional need to be implemented in the system
- Functional means providing particular service to the user
  - For example, in context to an online purchase application the functional requirement will be when customer selects ”Checkout” they must be able to look at the latest items in their basket before paying
- A software requirement can also be non-functional, it can be a performance requirement
  - For example, a non-functional requirement is where a click on a link results in a <0.5s delay to give a response

2 Functional Requirements

- Descriptions of data to be entered into the system
- Descriptions of operations performed on each system
- Descriptions of work flows performed by the system
- Descriptions of system reports or other outputs
- Who can enter data into the system
- How the system meets applicable regulatory requirements
- The user can get some meaningful work done

3 Non-Functional Requirements

- Characteristics of the system which cannot be expressed as functions:
  - Maintainability, portability, usability, security, safety, reliability, performance ..
  - These characteristics can be measurable
  - E.g a system sub-component that needs to have a response time or <1s 95% of the time
  - Constraints are NFR
  - PM issues are NFR (costs, schedule, logistics)

4 RE: Elicitation/discovery

Develop a plan

- What information should be discovered?
- What sources should be used?
- What mechanisms or techniques should be employed

The information to discover:

- A description of the problem domain
- The basic type of application
- The identity of the stakeholders
- The main motivation behind the development
- List of problems requiring solutions (the requirements)
- Any stakeholder imposed constraints upon behaviour or structure of solution system

5 Techniques for discovery

Interviewing:

- Different types of interviews
- Good for getting an overall understanding of what, how they interact with the system and any difficulties
- Not good for understanding specific domain requirements

Scenarios

- Useful for adding detail to an outline requirements description
- Starts with an outline of the interaction and adds detail to create a complete description - must include a description of:
  - What system and users expect when scenario starts
  - Normal flow of events
  - What can go wrong and how it is handled
  - Information on other activities which might be going on at the same time
  - System state when scenario finishes

6 RE: Analysis

Having determined the basic list of requirements we now analyse them

- Check completeness
- Remove inconsistencies
- Conflict resolution
- Revisit assumptions
- Prioritise

7 Prioritise

- Must have
  - Fundamental requirements, without which the system will not work
  - Define the MVP
- Should have
  - Considered important but the lack of these can be worked around
- Could have
  - Extended functionality but can be left out without undermining the project
- Won't have
  - The waiting list, requirements that can wait until a later development date