Requirements engineering

- Involves
  - eliciting
  - understanding
  - analyzing
  - specifying
- Focus on
  - *what* qualities are needed, NOT on
  - *how* to achieve them

What is needed

- **Understand interface between the application and the external world**
- **Understand the application domain**
- **Identify the main stakeholders and understand expectations**
  - different stakeholders have different viewpoints
  - software engineer must integrate and reconcile them

The requirements specification document (1)

- Provides a specification for the interface between the application and the external world
  - defines the qualities to be met
- Has its own qualities
  - *understandable, precise, complete, consistent, unambiguous, easily modifiable*

The requirements specification document (2)

- **Must be analyzed and confirmed by the stakeholders**
  - may even include version 0 of user manual
- **May be accompanied by the system test plan document**

The requirements specification document (3)

- As any large document, it must be modular
  - "vertical" modularity
    - the usual decomposition, which may be hierarchical
  - "horizontal" modularity
    - different viewpoints
- Defines both functional and non functional requirements

- Who are the stakeholders?
    - management of the train company
    - train drivers and their unions
    - passengers (customers)
    - contractors
- Each has different goals

Case study
how to classify requirements

- ## Safety requirements
  - the probability of accidents should be less than 10^{-9} per year

- ## Utility requirements
  - level of usefulness of the system as perceived by the various stakeholders

Case study
the produced document

- **Introduction: the "mission" of the system**
- **Architecture: the main structure of the system**
- **Specific requirements associated with each subsystem**
  - discussion of how the subsystems' requirements guarantee that the overall goals are indeed achieved