The Requirement Phase

- Requirements describe *what* the system is supposed to do using
  - Natural language
  - Diagrams
  - Other notation
- An important requirement of requirements is that they are understood by the customer/client for concept exploration/feedback/validation
  - Not too much technical jargon

Elicitation Techniques

- Interviews
  - structured, planned, close-ended conversations
  - unstructured, encouraged to speak and brainstorm
  - in our case
    - interview each other
    - make sure you are all on the same page
- System/documentation inspection
  - Look at other similar systems to see if there is functionality that might want/need to incorporate
- Paper prototypes
  - Hand sketches of interface *storyboard* to determine *scenarios*
- Scenarios
  - A scenario is a way a user might use a target product to accomplish some objective
  - An easy way for the client to define functionality in an accessible way
  - Only *major* system functions should be explored/explained with scenarios.
  - Examples of scenarios

Buy a Product Scenario:

The customer browses the catalog and adds desired items to the shopping basket. When the customer wishes to pay the customer describes the shipping and credit card information and confirms the sale. The system checks the authorization on the credit card and confirms the sale both immediately and with a follow-up email.

Elevator Scenario:

1.User A presses the Up floor button at floor 3 to request an elevator. User A wished to go to floor 7.
2.The Up floor button is turned on.
3.An elevator arrives at floor 3. It contains User B, who has entered the elevator at floor 1 and pressed the elevator button for floor 9.
4.The Up floor button is turned off.
5.The elevator doors open.
6.The timer starts. User A enters the elevator.
7.User A presses the elevator button for floor 7.
8.The elevator button for floor 7 is turned on.
9.The elevator doors close after a timeout.
10.The elevator travels to floor 7.
11.The elevator button for floor 7 is turned off.
12.The elevator doors open to allow User A to exit from the elevator.
13.The timer starts. User A exits from the elevator.
14.The elevator doors close after a timeout.
15.The elevator proceeds to floor 9 with User B.

Types of Requirements

- Functional requirements
  - Relate to the functionality of the system
- Non-functional requirements -- non-functional properties of the system:
  - Quantitative constraints
    - Response time
    - Accuracy
  - Language/OS
  - Documentation/process requirements
  - Hardware consideration
    - Compatibility
    - Interaction
  - Error handling
    - System exception
    - Safety considerations
    - Worst case scenarios
  - Quality issues
    - Reliability
    - Availability
    - Robustness
  - Resource issues
    - Constraints on the resources consumed by the system
      - Disk space
      - Network traffic

Desirable Attributes of Requirements

- Correctness
  - Represent the *clients view* of the system
- Completeness
  - Every phenomenon/scenario of interest is described including exceptional behavior
- Consistency
  - The requirements do not contradict each other
- Clarity
  - It is not possible to interpret the requirements in more that one way
- Realism
  - The system can be implemented as specified
- Verifiability
  - A repeatable test can be designed to demonstrate that the system satisfies a particular requirement
  - Not verifiable:

    > The product shall have a good interface.

    Should be:

    > The interface of the product shall conform to standard XYZ.
- Traceability
  - Each system function can be traced to its corresponding requirements
  - To achieve traceability each statement in the requirements documentation needs to be numbered.