Requirements elicitation
Requirements specification
Requirements validation

Four aspects of **requirements elicitation**:

*Application domain understanding* – understanding of the general characteristics of all applications that are similar. Understanding general characteristics of the problem space.

Example: to do a card catalog system, need to generally understand how libraries work

*Problem understanding* – specific details of the customer’s problem. Organizationally specific details.

*Business understanding* – how the system contributes to the development of the business, or organization.

*Needs and Constraints of Stakeholders* – Stakeholders are people affected by the system in some way. Can be end-users, department managers, people outside the organization, etc. Understand how each group interacts with the system.
Requirements elicitation techniques:

**Interviews**
- closed interviews, with engineers asking a pre-defined set of questions
- open interviews, with engineers conducting an open-ended discussion about requirements
Both techniques are valuable. Useful to have a predefined set of interview questions. Also useful to have the flexibility to follow up on areas where more detail is necessary.

**Scenarios**

Easier for people to relate to real-life examples, rather than abstract discussions.

A scenario is a short story about use of the system.

Scenarios include:

- Description of the state of the system before entering the scenario
- normal flow of events in the scenario
- exceptions to the normal flow of events
- information about other activities that might be going on at the same time
- a description of the state of the system after the scenario

Several styles of scenario:

- Textual stories involving real people using the system to perform a task
- Graphical depictions of the flow of activity in the system

In the Web Engineering context, **storyboarding** is an alternate elicitation technique, combining the content of a scenario, with preliminary ideas about the layout.

Can also use ethnographic techniques to understand people’s work practices in detail. This is most likely too time consuming for Web Engineering projects, though it might be applicable for a system redesign.

Key differences for Web application projects (from reading):
- developed in shorter timeframes
- act as the direct interface between multiple stakeholders
- meet a more generic set of requirements
- serve a less specific user group
- often developed quickly from templated solutions

One approach to address this is to use a fine-grained incremental and iterative development model.

Design artifacts play a crucial role in the development of client’s understanding. As a result, getting client feedback from early prototypes is crucial in getting the requirements correct. That is, the specification emerges from the design, rather than vice-versa.


- clients did not well understand the technologies
- clients did not understand their own needs as they related to the technology
  - in part due to lack of understanding of existing processes, and how they might be changed to make use of the technology
- most interviewed clients (96%), fewer felt interviewing users was useful (58%)
- most felt (84%) that look and feel issues were secondary to the business case
- Most attempt to get requirements right before signing final contract
- Most (94%) recognized that client needs will change and evolve over the life of the project, despite a well-written requirements specification
- Clients have an initially poor understanding of their requirements, that evolves over the life of the project.