Chapter 2: Requirements Elicitation

Requirements Engineering
Objectives

In this chapter, you will learn about:

- Eliciting Requirements
- Your Stakeholders
- Sample stakeholder’s analysis template
- Case Study
Eliciting Requirements

Requirements Elicitation is the first step in Requirements Engineering.

You need to use a variety of techniques to determine what the end-users and customers really want, because:

- Customers are not always good at describing their needs (litany of business jargons/assumptions)
- Technologists/Computer Scientists/SW Engineers are not always good at understanding/appreciating someone else’s business concerns (we have jargons too!)
- Words/jargons are contextual
Eliciting Requirements
Interviewing Techniques

1. Discuss requirements with all who have a **stake** in the system then:
   - Create a coherent set of requirements that reflect the different stakeholder views
     - **Who are your stakeholders?**
   - Next, review the requirements document with stakeholders [to reach a consensus]
     - Each stakeholder has a particular view of the system and how it should work
     - RE captures concerns of each stakeholders. **How do you resolve conflicting stakeholders’ views?**
   - **Examples of Stakeholders:** People who have something to contribute to a new system
     - Customer (Clients) – Ultimately stakeholders pay for the development
     - End-users – Interact with and Purchase the software after it is developed
     - Users already familiar with the current system and will use the new system
     - Problem Domain Experts – People who are familiar with the problem that the software must solve
     - Subject Matter Experts – Ensures that product is technically feasible; understand innovative software/hardware technologies, protocols; can educate the customer
     - Market Researchers – People who have conducted surveys to determine trends and customer needs
     - Software Engineers
Eliciting Requirements
Interviewing Techniques

How to resolve conflicting stakeholders’ views

- Ask customer to prioritize requirements into categories
  - Example Prioritization Scheme:
    - Requirements that absolutely must be met – Essential
    - Requirements that are highly desirable but not necessary – Desirable
    - Requirements that are possible, but could be eliminated – Optional

- Be a good listener
- Do not argue with the customer
Eliciting Requirements
Interviewing Techniques

Sample questions when you interview stakeholders

- **Functional Requirements**
  - **Functionality:**
    - What will the system do?
    - When will the system do it?
    - Are there several modes of operation?
    - What kinds of computations or data transformations must be performed?
    - What are the appropriate reactions to possible stimuli?
  - **Data**
    - For both input and output, what should be the format of the data?
    - Must any data be retained for any period of time?

- **Design Constraints**
  - **Physical Environment**
    - Where is the equipment to be located?
    - Is there one or several location?
    - Are there constraints on size of the system (Handheld/Server/PC etc)?
    - Are there any COTS or other constraints on programming language, OS because of existing software components?
  - **Interfaces**
    - Is input coming from one or more other systems (“upstream”)?
    - Is output going to one or more other systems (“downstream”)?
    - What is the protocol for the upstream and downstream systems?
  - **End-Users**
    - Who will use the system?
    - Will there be several types of users?
    - What is the skill level of each user?
Sample questions when you interview stakeholders

- **Quality Requirements**
  - **Performance**
    - Are there constraints on *execution speed, response time* or *throughput*?
    - How much data will flow through the system?
    - How often will data be received or sent?
  - **Usability and Human Factors**
    - What kind of training will be required for each type of user?
    - How easy should it be for a user to understand and use the system?
  - **Security**
    - Must access to the system or information be controlled?
    - Should each user’s data be isolated from other users?
    - Should user programs be isolated from other programs and from the OS?
  - **Reliability and Availability**
    - Must the system detect and isolate faults?
    - What is the prescribed Mean Time between Failures?
    - Is there a maximum time allowed for restarting the system after a failure?
    - How often will the system be backed up?
    - Must backup copies be stored at a different location?
  - **Maintainability**
    - When and in what ways might the system be changed in the future?
    - How easy should it be to add features to the system?
    - How easy should it be to port (or migrate) the system from one platform to another?
  - **Precision and Accuracy**
  - **Timeline /Cost**
2. **Review available documentation**
   - Procedures of manual tasks, specifications, manuals

3. **Observe current system**
   - Gather info about how end-users perform their tasks

4. **Apprentice with users**
   - Learn users tasks in detail as they are performed

5. **Use domain-specific strategies**
   - Joint Application Design (JAD) for information systems

6. **Use Requirements template**
   - Volere stakeholders requirements template
It is Ok (but not necessary) to adapt the Volere Template to your requirements elicitation process:

Let’s review the template:
- [http://www.personal.kent.edu/~asamba/tech61095/voleretemplate.xls](http://www.personal.kent.edu/~asamba/tech61095/voleretemplate.xls)

- **Additional Reading:**

Elicit Requirements

Summary

- Stakeholders wants/needs
- Domain Models
- Current organization and systems
- Existing Documents
- Current situation Model
- Requirements Template
- Reuse library
Eliciting Requirements

Case Study: Source of requirements

- Existing Documents
- Current situation Model
- Stakeholders wants/needs
- Current system

Source of Requirements
Eliciting Requirements
Example: Case Study

Case Study

Case Study materials will be presented in class