Lecture Overview

- Homework review
- Requirements Engineering
Establishing the *services* that the customer requires from a system and the *constraints* under which it operates and is developed.
Requirements Engineering

- **Top Down Process**
  - Begin with abstract (user) requirements
  - Refine them into more specific (system) requirements

- **User-centered Process**
  - Focus on stakeholders
    - Determine needs, desires
    - Involve them in translating needs into requirements
      - Describe needs
      - Review and evaluate requirements
  - Empirical evaluation
    - Assess quality by collecting data
  - Iteration
Requirements Engineering Activities

- Inception
- Elicitation
- Elaboration/Analysis
  - Negotiation
- Specification
- Validation
Inception

➤ Goal is to develop a very basic understanding of:
  - Problem
  - People that want a solution
  - Nature of desired solution
  - Effectiveness of initial communication between customer and developer

➤ Results in
  - Statement of business need for software to be developed
  - Feasibility of developing software
Inception Actions

- Identify stakeholders
  - Anyone that benefits from system to be developed
  - Typical examples
    - Marketing
    - Customer
    - End-user
    - Software engineers

- Identify and organize viewpoints
  - Further classify stakeholders according to perspectives
  - Broadly:
    - Interactor
    - Indirect
    - Domain
Viewpoint Example

- **Interactor**
  - Library manager
  - Finance
  - Publishers

- **Indirect**
  - User
    - Faculty
    - Student
  - Library staff

- **Domain**
  - Classification standards
  - Copyright regulations
Inception Actions (2)

- Ask context-free questions:
  - Regarding stakeholders
    - Who requested this system?
    - Who will use it?
    - What is the economic benefit?
    - Is there another source for the desired solution?
  - Regarding the problem
    - What would be “good” output generated by the solution?
    - What problem does the solution address?
    - What kind of business environment will this be used in?
    - Are there performance issues or constraints?
Ask context-free questions:

- Regarding communication
  - Are you the right person to answer these questions?
    - Is this answer “official”?
  - Are my questions relevant?
  - Can anyone else provide additional information?
  - Should I be asking you anything else?
Requirements Engineering Activities

- Inception
- Elicitation
- Elaboration/Analysis
  - Negotiation
- Specification
- Validation
Elicitation

- Goal is to understand the system to be developed
  - More specific information is obtained from the stakeholders

- Start with needs elicitation
  - Stakeholders needs and desires define design problem

- Evaluate needs
  - Make sure you’ve captured the needs correctly
  - Make sure you’ve captured the correct needs

- Translate needs into requirements
  - Requirements define the design solution
    - Represent decisions between conflicting needs and desires
    - Ways to satisfy needs and desires
Needs Elicitation Challenges

- Stakeholders are unavailable
  - Can’t leave their jobs
  - Don’t want to participate (suspicious)
- Stakeholders express how they work in their own terms
  - Technical jargon
  - Office politics
  - No big picture
- Stakeholders may have conflicting requirements
- Stakeholders often describe needs as a user interface
  - Confuses concerns
- New stakeholders emerge
  - Business environment changes
- Stakeholders make mistakes
- Stakeholders omit info
Needs Elicitation Heuristics

- Do a little research
  - Try to understand the problem domain first
    - Will ask stakeholder better questions, better understand their responses

- Try to understand the big picture and how needs fit in
  - Knowing the goals of a user helps you to catch needs that a stakeholder may omit

- See it for yourself
  - Study user tasks to help determine the set of needs
Elicitation Approaches

- **Research**
  - Observation
  - Documentation studies
  - Comparative product studies

- **Guided Meetings**
  - Interview
  - Focus groups
  - Elicitation workshops

- **Prototype demonstrations**

- **Scenarios and Use Cases**

More detailed and pointed questions
May propose some services, features
Needs Elicitation Approaches (1)

- Guided Meetings
  - Interview
    - Question the stakeholders
      - With purposeful and directed questions!
  - Focus groups
    - Informal discussion among stakeholders with moderator
  - Elicitation workshops
    - Focused, detailed discussion among certain stakeholders
Needs Elicitation Approaches (2)

- **Research**
  - **Observation**
    - Watch the stakeholders work
  - **Documentation studies**
    - Study problem domain
    - Study development process, policies, practices, techniques
    - Study previous documentation of system (evolution phase)
  - **Comparative product studies**
    - Study the products
      - Tinker with them
      - Reviews
      - Market studies
Needs Elicitation Approaches (3): Scenarios

- Real-life examples of how a system can be used
- They should include
  - A description of the involved stakeholders
  - A description of the starting situation
  - A description of the normal flow of events
  - A description of what can go wrong
  - Information about other concurrent activities
  - A description of the state when the scenario finishes
Scenario Example

- Ask the SafeHome user:
  - How would you use the home security function?
Needs Elicitation Approaches (3.5): Use Cases

- Scenario-based descriptions of how an *actor* interacts with the system
  - Actor (UML terminology)
    - Represents role people or device play when using the system
    - External to system (system is not an actor) Captures functional requirements
  - Captures functional requirements
  - Abstraction of a scenario
    - A scenario is one specific path
    - Use case captures multiple related paths of interaction
- Serve as basis for developing more detailed models
  - Supports traceability of requirements and design
Capturing Use Cases

- **Two forms of use case specification**
  - Full textual description
    - Name
    - Actors
    - Preconditions
    - Trigger
    - Scenario Description
    - Exceptions (a.k.a. Extensions)
    - Postconditions
    - Associated non-functional requirements (optional)
  - UML graphical description
    - We’ll see this later
Use Case Example

- Ask the SafeHome user:
  - How would you use the home security function?
Needs Documentation: Domain Knowledge

- Documenting Understanding of Problem Domain
  - Problem domain glossary
    - Use to record terminology
  - Organization chart
    - Use to record reporting/communication relationships between employees
  - Activity diagrams
    - Use to record business processes
Requirements Engineering Activities

- Inception
- Elicitation
- Elaboration/Analysis
  - Negotiation
- Specification
- Validation
Elaboration/Analysis

- Refine models of requirements
- Organize requirements according to:
  - Stakeholders
  - Viewpoints
  - Service
  - Preliminary decomposition of the system
- Prioritize requirements (and risks)
- Update cost estimates
- Negotiate requirements
  - Customers may ask for more than can be achieved
  - Prioritization information may be in conflict
In Terms of Your Textbook:
Moving from Needs to Requirements

1. Analyze Product Design Problem
2. Elicit/Analyze Needs
3. Generate/Improve Candidate Requirements
4. Evaluate Candidate Requirements
5. Select Requirements
6. Finalize Requirements
Generating Candidates and Alternatives

- Common failing: too few alternatives
- Idea sources
  - Users and other stakeholders
  - Props and metaphors
  - Competitive products
  - Similar products
- Generation techniques
  - Team brainstorming
  - Individual brainstorming
  - Modeling
Candidate Requirement Example

- Aqualush
  - Frequency of irrigation
    - Once a day?
    - Once every $n$ days?
    - Once a week?
    - Once every $n$ weeks?
    - Some particular days of the week?
    - Some particular days of the month?
    - Exclude certain days?
    - Randomly?
Requirements Engineering Activities

- Inception
- Elicitation
- Elaboration/Analysis
  - Negotiation
- Specification
- Validation
Approaches to Specifying Requirements

- Structured natural language
  - Rules guide specification of requirements

- Graphical modeling notations
  - Easy to understand graphical representations

- Mathematical specifications
  - Based on mathematical concepts such as sets or finite state machines
Problems with Natural Language

- **Ambiguity**
  - Readers and writers of the requirement must interpret the same words in the same way

- **Over-flexibility**
  - The same thing may be said in a number of different ways

- **Lack of modularization**
  - Natural language structures doesn’t capture hierarchical requirement relationships
An Example of Ambiguous Requirements

- An escalator in an London Underground station has two signs:
  - Dogs must be carried
  - Shoes must be worn
Guidelines for Using Structured Natural Language

- Rules for writing user and system requirements:
  - Use standard structured format
  - Write complete, simple sentences
  - Use active voice
  - Define terms, use them consistently
  - Group related requirements hierarchically
  - Express all requirements using “must” or “shall”
  - Write requirements in a **verifiable** manner
  - Write **atomic** requirements
    - States a single product function, feature, or characteristic
    - Helps with requirements traceability
Structured Natural Language for Scenarios

Rules for writing scenarios:

- All of the previous rules
- Format:
  - Initial assumption
  - Normal execution
  - Exceptional execution
  - State on completion
Structured Natural Language for Use Cases

- Rules for writing scenarios:
  - All of the previous natural language rules
  - Format:
    - Use-Case Name
    - Actors
    - Pre-conditions
    - Post-conditions
    - Trigger
    - Basic Flow
    - Extensions
    - Non-functional Requirements
Use Cases with UML

**Basic Notation**

Bank Customer \[ \rightarrow \] Withdraw Money

*More on use cases and UML use case notation coming soon…*
Requirements Engineering Activities

- Inception
- Elicitation
- Elaboration/Analysis
  - Negotiation
- Specification
- Validation
Requirements Validation Techniques

- **Requirements reviews**
  - Systematic manual analysis of the requirements

- **Prototyping**
  - Using an executable model of the system to check requirements
    - Covered in Chapter 17

- **Test-case generation**
  - Developing tests for requirements to check testability
    - If difficult to design tests, requirement will be difficult to
      - Implement
      - Validate
Reviewing Requirements

Individual Requirements

- Verifiability
  - Is the requirement realistically testable?

- Comprehensibility
  - Is the requirement properly understood?

- Traceability
  - Is the origin of the requirement clearly stated?

- Adaptability
  - Can the requirement be changed without a large impact on other requirements?
Reviewing Requirements

- Collection of requirements (SRS)
  - Consistency
    - Are there any requirements conflicts?
  - Completeness
    - Are all functions required by the customer included?
  - Realism
    - Can the requirements be implemented given available budget and team?
Requirements Management

- Requirements change during a project
  - Need to understand, control, and track changes
    - Policies
      - Requirements identification
      - Change management process
      - Traceability policy
      - CASE tool support
Requirements Change Management

- Should apply to all proposed changes to the requirements
- Principal stages
  - Problem analysis
    - Discuss requirements problem and propose change
  - Change analysis and costing
    - Assess effects of change on other requirements
  - Change implementation
    - Modify documentation to reflect change
Traceability

- Traceability is concerned with the relationships between requirements, their sources and the system design
- Requirements should be uniquely identified to aid in traceability
  - Naming convention should reflect traceability relationships
- Traceability relationships
  - Source traceability
    - Links from requirements to stakeholders who proposed these requirements
  - Requirements traceability
    - Links between dependent requirements
  - Design traceability
    - Links from the requirements to the design
# Traceability Matrix

- **Relates requirements to**
  - Stakeholders
  - Other requirements
  - System features
  - Design modules
    - Subsystems
    - Interfaces

- **Creation**
  - Manual for small projects
  - Generated from database in large projects

<table>
<thead>
<tr>
<th>Req. id</th>
<th>1.1</th>
<th>1.2</th>
<th>1.3</th>
<th>2.1</th>
<th>2.2</th>
<th>2.3</th>
<th>3.1</th>
<th>3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>D</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td></td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td>D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D</td>
</tr>
<tr>
<td>3.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
</tr>
</tbody>
</table>

Traceability and requirement relationships
CASE Tools for Requirements Change Management

- **Basic productivity tools**
  - Word processors, spreadsheets, databases

- **Workbenches**
  - **CASE Spec**
    - [www.analysttool.com](http://www.analysttool.com)
    - Free demo
  - **Borland Caliber Analyst**
    - Free trial
  - **DOORS (by IBM)**
    - Free trial
The Software Requirements Specification Document

- The official statement of what is required of the system developers

- Should include:
  - a definition of user requirements
  - a specification of the system requirements
Requirements Document Structure

- Preface
  - Intended audience
  - Version
  - Change summary
- Introduction
  - Project overview
- Glossary
- Requirements
  - User requirements definition
  - System requirements specification
- Appendices
  - Hardware, software related requirements