Intro to Software Requirements

- What question do software requirements answer?
  - Who, what, when, where, why, how

    What is the system to do?

    Who are the system user groups?

    Business case tells us why (and perhaps who, when, where).

    Project plan tells us when and who.

    Architecture tells us how.

Why do we care?

- Requirements issues are among the most common reasons cited for project failures and challenges
- Project Success Factors

Project Success Factors	% of Responses
1. User Involvement	15.9%
2. Executive Management Support	13.9%
3. Clear Statement of Requirements	13.0%
4. Proper Planning	9.6%
5. Realistic Expectations	8.2%
6. Smaller Project Milestones	7.7%
7. Competent Staff	7.2%
8. Ownership	5.3%
9. Clear Vision & Objectives	2.9%
10. Hard-Working, Focused Staff	2.4%
Other	13.9%

Standish Group’s Chaos Report (1995)

Why do we care?

• Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

Standish Group's Chaos Report (1995)

Why do we care?

- Cost of avoiding/fixing defects increases as project progresses
  - Cheapest in requirements development
  - Boehm’s cost of fixing a defect curve
- Anecdote: Sprint LARS

IEEE Definition of Requirement

- IEEE Standard Glossary of SE Terminology
  1. A condition or capability needed by a user to solve a problem or achieve an objective.
  2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.
  3. A documented representation of a condition or capability as in 1 or 2.

Other Definitions

- Weiger
  - A property that a product must have to provide value to a stakeholder
- Sommerville and Sawyer
  - A specification of what should be implemented. They are descriptions of how the system should behave, or of a system property or attribute. They may be a constraint on the development process of the system.
- Any others?

Types of Requirements

- Discussion Question:
  - Explain the difference between business, user, system, and functional requirements.
  - What are nonfunctional requirements?

Types of Requirements

- Business
  - High-level objectives of the organization or customer who requests the system
  - Documented in a Vision and Scope document
- User
  - User goals or tasks that the users must be able to perform with the product
  - Use-cases often used to capture these
  - Ex. Make a reservation
- System
  - High-level requirements for a product that contains multiple subsystems

Types of Requirements

- Functional
  - Specify the software functionality that the developers must build into the product to enable users to accomplish their tasks.
  - Ex. The system *shall* mail a confirmation to the user
- Non-functional
  - Quality attributes, performance goals, reliability, …
  - Ex. Reservation request submissions should receive a response in less than 10 seconds

Requirements Documents

- “different organizations might call any of the following a ‘requirements document’”:

1. Half-page software product vision
2. Two page list of key features
3. 50 page list of detailed end-user-level requi

4. 250 page exhaustive listing of every visual element on every screen, input-field-by-input-field descriptions of all possible input conditions, all possible system state c description of every persistent data elem

Product Vision

Feature List

Functional Requirements Document

Functional Requirements Specification

1. McConnell, IEEE Software, Sept/Oct 2000, http://www.stevemcconnell.com/ieeesoftware/eic13.htm

Requirements Engineering

- Discussion questions:
  - What is Software Engineering?
  - How does Requirements Engineering fit with Software Engineering?
  - Why do both include the word "engineering" and is this fair and appropriate?
  - Recall Chaos report; why is requirements engineering hard?

Requirements Problems

- Insufficient User Involvement
- Creeping User Requirements
- Ambiguous Requirements
- Gold Plating
- Minimal Specification
- Overlooked User Classes
- Inaccurate Planning
- Discussion Questions:
  - Have you seen any of these occur?
  - What could have been done to avoid the problem?

Excellent Requirements

- Statements
  - Complete, correct, feasible, necessary, prioritized, unambiguous, verifiable
- Specification
  - Complete, consistent, modifiable, traceable
- Discussion Question:
  - What would you add to the list?