Lecture 07: Techniques for Gathering Requirements

Valentin Razmov

Outline
- Techniques:
  - Use Cases / Usage Scenarios
  - Commonality and Variability Analysis
  - Frequent Customer Feedback
  - Throwaway Prototyping
- Risks from Inadequate Requirements Processes
- Discussion Questions

Note: The list of techniques is necessarily incomplete.

Resources
- "Software Requirements", by Karl Wiegers
- "Use Cases and The Ever Unfolding Story", seminar by Dan Rawsthorne

Requirements Engineering
"The goal of requirements engineering is to develop high quality – not perfect – requirements that allow you to proceed with construction at an acceptable level of risk."
-- from "Software Requirements", Karl Wiegers

Use Cases
- Describe the typical paths for a user to interact with a system
- Increasing level of detail depending on the complexity of the interaction
  - "The ever unfolding story" (Dan Rawsthorne)
- Questions to consider (in that order):
  1. Who are the actors?
  2. What are they (normally) doing?
  3. What can go wrong with that?
  4. How do we handle this situation?

Use Cases
- Example: Reading your web-based mail (e.g., @ hotmail, yahoo, gmail, etc.)
Commonality and Variability Analysis

- Commonalities are the enduring concepts of the domain you’re modeling.
- They would give stability to your designs.
- Variabilities are only defined in the context of existing commonalities.

07 Jul 2005  CSE410, Summer’05, Lecture 07

Commonality and Variability Analysis

- Example: Computing the price for a purchase at an e-commerce site

07 Jul 2005  CSE410, Summer’05, Lecture 07

Frequent Customer Feedback

- Are requirements going to change?
  - When are they final?
  - Are they ever exact and clear?
- Frequent communication with customers => no need to make dangerous assumptions about the finality and completeness of requirements

07 Jul 2005  CSE410, Summer’05, Lecture 07

Throwaway Prototyping

- Using a rough sketch / diagram to show your understanding and to evoke customer response
- Example: © Busta’ Sandwich Co.
  - Caution: Do not overdo it! It must look and remain throwaway.

07 Jul 2005  CSE410, Summer’05, Lecture 07

Risks from Inadequate Requirements Processes

- Insufficient user involvement => ?
- Creeping user requirements => ?
- Ambiguous requirements => ?
- Gold-plating by developers and users => ?
- Minimal specifications => ?
- Overlooking the needs of certain user classes => ?
- Incompletely defined requirements => ?

07 Jul 2005  CSE410, Summer’05, Lecture 07

Risks from Inadequate Requirements Processes

- Insufficient user involvement => unacceptable products
- Creeping user requirements => overruns and degraded product quality
- Ambiguous requirements => ill-spent time and rework
- Gold-plating by developers and users => unnecessary features
- Minimal specifications => missing key requirements
- Overlooking the needs of certain user classes => dissatisfied customers
- Incompletely defined requirements => accurate project planning and tracking impossible

07 Jul 2005  CSE410, Summer’05, Lecture 07
Questions for Discussion

- Is there a way to test requirements?
  - If so, how?
  - If not, how does one know that they've been fulfilled at delivery time?

- How does one evaluate the presence and level of fulfillment of non-functional requirements?
  - E.g.: reliability, robustness, efficiency, security, safety, maintainability, availability, ...

The V-Model and Testing throughout the Lifecycle

Your Questions on Gathering Requirements