

Requirements Elicitation: Finding Out What Your Users Need

Discussion About the Reading

- Why do we need nonfunctional requirements?
- What is requirements validation and how might we do it?
- Compare & contrast user requirements, system requirements, and design specifications.
- What should you do when requirements conflict?

A Difficulty

- You have 10 weeks for the project
- Many "real" projects are much bigger (more resources)
- There are lots of aspects of "real" projects that you can't do in class
- Lecture will include some techniques you don't have time to use now

Eliciting Requirements

- We want to meet a **real** user's needs
- What arguments should win?
 - Not the loudest ones
 - Instead, ones that are grounded in the best data.
- HCI mantra: "The user is not like me"

Interviews

- Consider all stakeholders
- Who are the stakeholders of your system?

Interview Structure

- *Semistructured*: Ask questions in a conversational way (potentially out of order)
- Your results will depend on interviewer skill.

Focus Groups

- Focus group: gather 5-7 or 8-12 participants for a group interview
- Pro: more participants, less experimenter time
- Discussions reveal similarities and differences
- Con: quiet people might not get heard
- Skill is needed to manage conversation
- Analysis can be tricky (interruptions, changes of speaker)

Demonstration: Writing Questions

- "A taco is a kind of sandwich, right?"
- Is a taco a sandwich?
- What food categories are the items in the picture in?
- What do you expect to see on the menu at a sandwich restaurant?
- What makes something a sandwich?



Demonstration (2)

- How do you feel about covariant return types?
 - Use terms your participant knows.
- When do you usually decide to start using the debugger?
 - Think of the last bug you fixed. What debugging strategies did you use?

Designing Questions

- Neutral: unbiased, nonjudgmental
- Simple
- Open-ended
- Speak their language
- Ask for demonstrations or recall of concrete events

Contextual Inquiry

- Observe people doing the work as it is done now
- For your project: you can find worksheets, lesson plans, etc. online
- Printing example

Goal-Based Approaches

- Analyze the problem domain
- Example: *automate all aspects of baggage handling from passenger origin to destination.*
 - Goal 1: Automate the tracking of baggage from check-in to pick-up.
 - Goal 2: Automate the routing of baggage from check-in counter to plane.
 - Goal 3: Reduce the amount of lost luggage to 1%.

Prototyping

- Prototype user interface and interview users
- We'll talk about prototyping later

Scenarios

- Informal descriptions of the system in use

A customer walks into the pet store and fills the cart with a variety of items...the cashier scans products using a bar code reader. As each item is scanned, the sale is totaled and the inventory is appropriately updated. Upon completion of product scanning a subtotal is computed. Then any coupons and discounts are entered. A new subtotal is computed and applicable taxes are added. A receipt is printed and the customer pays using cash, credit card, debit card, or check. All appropriate totals (sales, tax, discounts, rebates, etc.) are computed and recorded.

Specifying Requirements

Requirements, User Stories

- Question: how to express requirements?
- Answer (user stories): "As a <stakeholder>, I want <something> so that <need>."
- Example: "As a student, I want to filter recipes by cost so I can keep dinner under \$5 per person."

User Story Criteria: "INVEST"

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

Independent

- Ideally: want to implement requirements in any order
- In practice, there may be dependencies

Negotiable

- Details to be negotiated during development
- Good Story captures the essence, not the details

Valuable

- This story needs to have value to someone (hopefully the customer)
- Especially relevant to splitting up issues

Estimable

- Helps keep the size small
- Need to complete each user story in 1-2 weeks (or less)

Small

- Fit on 3x5 card
- At most two person-weeks of work
- Too big == unable to estimate
- Too big == may not finish in time for delivery

Testable

- Ensures clarity
- If not testable, when do we say the task is done?

Summary

- Write open-ended, high-quality questions to elicit requirements
- Use INVEST criteria to write good user stories