Requirements Elicitation: Finding Out What Your Users Need

Project Theme

- Reducing student stress
- Your project should address *some* aspect of stress
- Next class: you will interview experts (each other!) to understand what causes stress

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?

Eliciting Requirements

- We want to meet a **real** user's needs
- HCI mantra: "The user is not like me"
- Let's interview some stakeholders.

Whom To Interview

- 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

Simple Questions

- "What were the strengths and weaknesses of the compiler and IDE?"
    - -> "What did you think of the compiler" & "What did you think of the IDE?"

Netural Questions

- "You find homework pretty stressful, right?"
  - -> "What kinds activities do you find the most stressful?"
- "Why do you like this design?"
  - What if they didn't like the design?

Recording Data

- Write notes
  - Rewrite and summarize after the interview
- Record audio & transcribe
- Screen capture (if there are demonstrations)

Rapport

- Be nonjudgmental — develop a poker face!
- Keep people comfortable. Water? Snacks?

Conducting the Interview

- Start with easy questions
- Listen!
- Provide opportunities to continue: "Is there anything else you wanted to tell me?"
- Ask for clarification when needed: "What exactly do you mean when you say…?"

Activity

- You want to know how people remember things that need to be done (you hope to create a new calendar/to-do system).
- Write two *open-ended* questions asking about how people organize their tasks.
- Ask them of your neighbor. Summarize their answers.

https://www.gradescope.com/courses/940938/assignments/

Specifying Requirements

Requirements, User Stories

- Question: how to express requirements?
- Answer: "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."

https://www.agilealliance.org/glossary/invest/

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