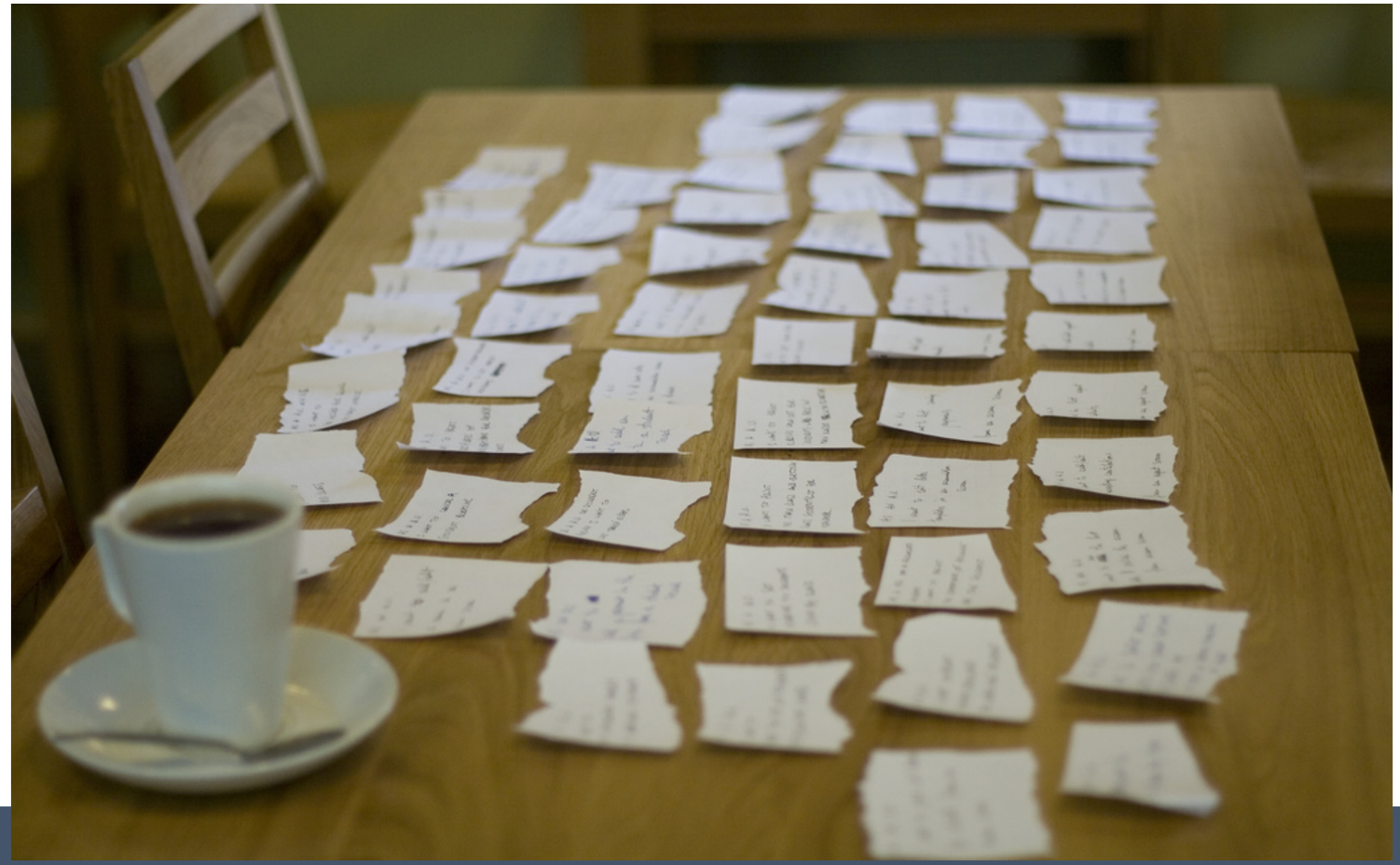
Credit: Michael Hilton, CMU 17-313

Requirements: user stories



Source: https://www.flickr.com/photos/jakuza/2728096478

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

- "Did you like the language you used?"
  - -> "What did you think of the language?"
- "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

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…?"

Are User Stories Enough?

- User stories capture what the system should do (functional requirements)
- What about *how well it does it*?

Functional Requirements vs. Quality Attributes

- Functional requirements: things the system must do
  - "The system shall support receiving email from an IMAP server."
- Quality attributes: requirements concerning how the system meets its functional requirements
  - "The system should fetch 1 GB of email in under 1 minute."
  - Distinguish between different implementations that meet the functional requirements
  - Should be testable

Quality Attribute Examples

- Informal goal: "the system should be easy to use by experienced controllers, and should be organized such that user errors are minimized."
  - Verifiable usability quality requirement: "Experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day, on average."
- Availability: The system shall be available 99.99% of the time.

A Common Mistake

- Avoid writing requirements in terms of implementation details
- Design is a separate process; don't do it now
- "The database server shall handle at least 4M records."
  - "Oh, there's a database server? Just one?
  - → "The system shall support at least 4M users."

Exercise

- Write two open-ended interview questions that focus on behavioral health and wellness needs for students.

Exercise

- Write a quality requirement pertaining to an entertainment system for airplane passengers.
  - Careful: what assumptions are you making?

User Stories

- Succinctly express functional requirements (and sometimes quality attributes)
- Facilitate conversations with stakeholders
- Define acceptance criteria (so you know when you've succeeded)

User story cards (3”x5”)

“As a [role], I want [function], so that [value]”

How to evaluate user story?

Follow the INVEST guidelines for good user stories!

Source: http://one80services.com/user-stories/writing-good-user-stories-hint-its-not-about-writing/

one 80

Independent

- Schedule in any order.
- Not overlapping in concept
- Not always possible

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
- Ensure we negotiated correctly
- “Plans are nothing, planning is everything” -Dwight D. Eisenhower

Small

- Fit on 3x5 card
- At most two person-weeks of work
- Too big == unable to estimate

Testable

- Ensures understanding of task
- We know when we can mark task “Done”
- Unable to test == do not understand

Activity

Follow the INVEST guidelines for good user stories!