HCI 3/5

Reagan Shirk

March 5, 2020

Designing Usability Tests

What

- Tasks: a specification of test activities involving design aspects of concern
  - Select a variety of frequent, typical, critical, problematic, and/or new tasks
  - You want to focus on the things that matter the most to the success of your application
  - You also want to choose open-ended tasks that lead particpants to use a specific functionality
    - Don't tell them "click this then click that", tell them to "accomplish x task"
  - Tasks should be explicit, unambiguous, relevant, reasonable, and doable
  - Tasks should be arranged in a logical and progressive way (easy → hard)
  - You want to choose an appropriate number of tasks for the available time
    - If a task takes about 30 seconds to do, the user should get about 5 minutes
  - For a typical multistep task (like what we've been doing in our assignments), we should give 10 minutes
  - Tests shouldn't go over 45 minutes, so realistically there'll only be about 4 tasks per test
  - Do dry runs and reasonably pad the expected time
    - Lets us see confusion or error in instructions
  - We need to supply any external resources needed (esp information)
- Scenarios: an organization of tasks for presentation to participants
  - Create non-leading stories without explicit directions (i.e. minimal bias)
  - imply one or two tasks in the form of a goal pursued by personas
  - stories should be familiar, brief, complete, unambiguous, and clear
  - Just a few sentences, doesn't need to be super complicated
  - Don't want to be "here is what you to specifically" but rather "here is the motivation for what we want you to do"
- Measurements: an enumeration of the results that are being obtained either formally or informally, can be qualitative or quantitative
  - Elicit qualitative observations such as user opinion and subjective feedback
    - Can be hard to analyze the responses because you get a lot of feedback and a lot of it conflicts
    - Sometimes it lets you know that something is good/bad but doesn't tell you how to fix it
    - Can identify problems to look at more objectively
  - Collect quantitative data as well
    - task completion time
    - error count
    - error repitition patterns
    - error recovery time
    - We want to define success **before** we gather the results. What results satisfy which goals?