IS 214 Needs Assessment and Evaluation of Information Systems

Usability Testing Plan & Conduct

Usability Testing

- Compared to design exploration
  - More observation, less conversation
  - Lower-level interaction issues
  - Should include performance measures
  - Evaluate a single design, not a comparison

Usability Testing - When

- Later stages of the design cycle
- When low-level design issues arise
- When a high-fidelity prototype is available

Usability Inspection - Objectives

- Identify usability problems early enough for design changes
- Measure how well users can complete real tasks in the near-complete user interface
- Measure user satisfaction with the experience

Usability Testing – Method

- Write test plan
- Conduct the test
- Analyze and report findings (will be covered in the next class)

Usability Testing – Plan

- Identify purpose and objectives
- Select participants
- Write scenarios / tasks
- Choose measures and benchmarks
- Prepare test materials
- Identify the test equipment & team

Usability Testing – Conduct

- Practice test
- Welcome user, overview test procedure
- Administer pre-test questionnaire
- Administer consent & confidentiality form
- Provide tasks and/or scenarios
- Final questionnaire and debriefing

Purpose and objectives

- Project focus – what do you want to learn?
- What parts of the user interface will you test?
- Do users understand how to start <task>?
- Can users successfully complete <task>?
- Is the response time a problem?
- Are users satisfied with <task> user interface?

Select Participants

- Use your user definitions to write a screener questionnaire for prospective users
  - Experience, education, job description
- Recruit 3-5 users from each user group
- Start recruiting early, set appointments, and confirm appointments the day before

Write Tasks / Scenarios

- Use your task analysis to identify the most frequent, important, and problematic tasks
- Prototype tasks that you want to test or test the tasks that have been prototyped
- Provide task context, motivation for the user
- If possible, allow the user to provide context by using their own examples
- What is successful completion of the task

Write Tasks / Scenarios (Continued)

- Examples of leading tasks:
  - Select File -> New… and then Insert -> Chart… to create a chart, then type a new column heading …
  - Create a new presentation, then insert a chart.
- Examples of goal-oriented tasks:
  - Create a presentation with a bar graph.
  - Create a chart that looks like this (sample chart).

Measures

- Performance measures
  - Time to complete tasks, percentage completed
  - Number of errors, time to recover
  - Number of uses of help, manuals, support calls
- Preference measures
  - Rating importance, usefulness, capability
  - Rating ease of use, learning, installation

Benchmarks

- Comparison to user performance in competitor, previous release, or earlier user tests
- Maximum limit can be used to halt task
- Don't commit to arbitrary goals

Test Materials

- Orientation script
- Background / pretest questionnaire
- Consent and non-disclosure forms
- Data collection forms
- Task scenarios
- Post-test questionnaire
- Debriefing topics guide

Orientation Script

- Introduce yourself and other observers
- Offer refreshments, they can break anytime
- Inform them of what they will be doing
- Assure them they are not being tested
- Ask them to think aloud, ask questions

Background questionnaire

- Confirm the right users are being tested
- Gather education and experience data that might be relevant in evaluating results
- Possibly use screener questionnaire and ask users to fill it out in advance

Pretest questionnaire

- Measure pretest knowledge / terminology
- Measure pretest opinions of product features
- Use for comparison with post-test
- Be careful not to bias participants
  - Retrospective invention of work practices
  - Opinions without context

Consent and Non-Disclosure

- Consent to being observed and videotaped
- Release of comments and suggestions
- Confidentiality agreement
- Protection of participants' intellectual rights
- Freedom to withdraw

Data Collection

- What performance and preference data will you collect and how will you record it
- Don't collect data for its own sake, think about what data will you use in your analysis
- Forms, online loggers, and automated loggers
- Coded, hand-written notes
- Make data collection assignments

Task Scenarios

- Read to the user to encourage interaction
- Provide on paper for consistency
- One at a time if later tasks might bias earlier
- Provide sample data, don't make them invent it
- If necessary, provide completed task data to enable users to continue to next task

Post-test Questionnaire

- Preference ratings
- Room for comments to explain ratings
- Minimize open-ended questions
- Keep the number of questions small or ask them in the context of specific tasks
- Discuss the users' answers with them

Debriefing Guide

- List of issues to discuss with users
- Predefined project focus issues
- Issues identified by observers
- Questions raised by users

Identify Equipment

- What hardware and software will you use?
- How will observers observe (desktop sharing, monitor cameras, one-way mirror, in room)?
- Will you record (video taping, ScreenCam)?
- Test the setup before the test – dry run

Identify Team

- Monitor – interaction with users
- Logger – gathers specific types of data
- Timers – measure time on task
- Video – starting, stopping, mixing video
- Technical – troubleshoot product
- Observers – take notes on user behavior

The Monitor Role

- Orient users and put them at ease
- Be impartial and receptive to user comments
- Nudge users past problems
- Avoid bias through reaction or assistance

What to Say Before the Test

- We're testing the product, not you. Any problems you encounter will help us improve it.
- You can quit at any time.
- Tell us your thoughts as you are working. If you forget to think aloud, we may remind you.
- Please ask questions as they occur to you, but we won't be able to answer them until the end of the test.

What to Say During the Test

- As little as possible.
- Reflect back what the user has said.
- Remind the user you can't help with problems.
- Probe verbal or non-verbal cues to problems:
  - What are you thinking right now?
  - What were you expecting to happen there?
- Provide gradual hints to get user "unstuck"

What Not to Say to Users

- Don't show users how to do it
- Don't show surprise
- Don't imply that the user did something wrong
- Don't rescue users at first struggle
- Don't ask users how they would design it
- Don't badger the user to think aloud

When to Help Users

- User has past maximum time for task
- User is uncomfortable with task
- User is distressed and ready to quit
- User needs information normally available
- A bug causes problems for the user

Debriefing the User

- Review the post-test questionnaire
- Give the user a chance to give their overall reaction and ask questions
- Return to the user interface and recreate tasks to discuss task problems or issues
- Discuss user's design ideas

Usability Test Exercise

- Six or nine volunteers to be users
- Need to be able to access their email
- Three person user groups A, B, & C
- Number users 1, 2, & 3
- Leave the room for test preparation

Usability Test Exercise (Continued)

- Two observers for each user
  A1, A2, A3, B1, B2, B3, and so on
- Monitor and logger responsibilities
- Demo of expected user behavior
- Measures to be gathered
- Issues to be observed
- Post-test questions and debriefing

What's Next

- Analyzing and reporting test results