cisc3650
human-computer interaction
spring 2012
lecture # II.1
evaluation techniques

topics:

- evaluation techniques
- usability testing

references:

- Human-Computer Interaction, by Alan Dix, Janet Finlay, Gregory D. Abowd and Russell Beale, *ch 9, Evaluation Techniques*
- Web Usability: A user-centered design approach, by Jonathan Lazar, *Ch 9, Usability testing*

evaluation techniques

- evaluation occurs throughout the software development lifecycle
- note that we focus here on *user interface evaluation*
  however many of the lessons and guidelines can be applied to software system evaluation in general

- there are two main types of evaluation:
  - evaluation by system designers or experts
    typically conducted early in the development lifecycle, though may give late feedback on system performance
  - evaluation by end users
    typically conducted late in the development lifecycle, though may give early feedback on system design

goals of evaluation

- there are 3 main goals of user interface evaluation:
  - system functionality
    does the system meet the user's requirements?
    is the system clear to operate?
    does the system help make the user effective at her task?
  - user's experience
    is the interface *usable*?
    is the user satisified?
    is the user's experience using the interface pleasant?
    is the user happy/angry/frustrated when using the interface?
    take into account the user's task, for example, if the interface is for a game, then the user should have fun using the system (playing the game).
  - problem identification
    does the system produce errors?
    is the user confused when using the system?

evaluation by experts

- there are 4 types of evaluations conducted by experts or system designers:
  - *cognitive walkthrough*
  - *heuristic evaluation*
  - *model-based*
  - *based on prior studies*

experts: cognitive walkthrough

- "walkthrough" means a sequence of steps
- e.g., "code walkthrough" is when a programmer shows her source code to others and explains, line by line, how the code works
- *cognitive walkthrough* is when a user explains how she is using an interface, talking as she performs each action and explaining as she goes
- then, experts analyze the user's process
- for each step:
  - is the effect of the user's action the same as the user's goal?
  - will the user see that the necessary action is available (in the interface)?
  - will the user know which widget in the interface enables the desired action?
  - after the action is performed (e.g., clicking on a button), will the user receive understandable feedback?
- in order to conduct a cognitive walkthrough, experts need:
  - system specifications or system prototype

- description of users' task(s)
- list of user actions to accomplish the task(s)
- profile(s) of target user(s) (i.e., expected background, experience, etc.)

experts: heuristic evaluation

- typically performed on a design specification, but can be performed on a prototype or full system
- there are 10 heuristics that are frequently assessed
- each is assessed on a scale of 0 to 4, as follows:

$0 =$ not a problem
$1 =$ cosmetic problem; fix if time
$2 =$ minor usability problem; low priority fix
$3 =$ major usability problem; important to fix
$4 =$ catastrophic usability problem; must be fixed before system/interface is released

- the 10 heuristics are:
  1. visibility of system status (i.e., inform users with feedback)
  2. match between system and real world (i.e., "speak" user's language; is system understandable by the user?)
  3. user control and freedom (i.e., can user "undo" and "redo" actions?)

4. consistency and standards (does system follow standard conventions for symbols, menu items, etc?)
5. error prevention (does system make it hard for the user to make mistakes?)
6. recognition rather than recall (does system make it easy for the user to remember what to do? are there no or few commands to memorize?)
7. flexibility and efficiency of use (does the system let users tailor the interface to accommodate frequent actions or sequences of actions, e.g., macros?)
8. aesthetic and minimalist design (e.g., dialogues shouldn't contain extra information; interface shouldn't be cluttered)
9. help users recognize, diagnose and recover from errors (are error messages in clear language?)
10. help and documentation (does the system have any documentation and/or a help facility?)

experts: model-based

- high-level models (such as GOMS) are used to predict user's performance
- low-level models (such as Fitts law—keystroke, dialogue timing) are used to assess limits of user's performance
- see lecture notes from feb 6 (2nd lecture) for a discussion of types of user interface design models, such as GOMS

experts: based on prior studies

- empirical models evolved from previous studies give indicators of generic interface elements
- for example, menu types, command name recall, color schemes, button labels, icon designs, etc.

evaluation by users

- can be done at different stages of development
- system developers can simulate missing (undeveloped) pieces of interface using techniques like "Wizard of Oz" where a human takes the part of the pieces that will later be automated
- elements of user evaluation:
  - styles of evaluation
  - experimental evaluation design
  - observational techniques
  - query techniques
  - evaluation through physiological responses

users: styles of evaluation

- laboratory studies
  - conducted in controlled settings
  - advantages: no distractions
  - disadvantages: but no context
- field studies
  - conducted "in situ", i.e., in situated settings
  - advantages: context
  - disadvantages: but can add distractions

users: experimental evaluation design

- for conducting "controlled experiments" where the goal is to support a claim or a *hypothesis*
- participants:
  - choose carefully, to "simulate" actual end users
  - "sample size" (number of users): usability studies recommend 3-5 users, also called "subjects"; controlled experiments recommend at least twice this number; even more when statistical significance is sought from results
- variables:
  - manipulated to create different conditions for experimental comparison
  - *independent* versus *dependent*
    - independent: variable that changes; that is manipulated (e.g., menu composition in an interface)
    - dependent: variable that doesn't change; that is measured (e.g., speed of selecting items on the menu)

  - a valid experiment is one in which the dependent variable is effected by the changed value(s) of the independent variable
- hypothesis:
  - prediction of outcome
  - framed in terms of independent and dependent variables
  - *null hypothesis*—the experimental results are random; not caused by the value(s) of the independent variable
  - *alternate hypothesis*—the experimental results are not random, but are caused by the changes to the independent variable (i.e., what we are testing)
- experimental design:
  - choose the hypothesis
  - choose the experimental method
    - "between subjects"
      each participant is assigned a different experimental condition (i.e., one value of the independent variable)
      all participants are divided into groups, so that each group tries one experimental condition
      then the results for each group are averaged; and the averages are compared

    - "within subjects"
      each participant is assigned all experimental conditions
      all particpants try all conditions, though typically the order in which they try the different conditions should be varied
      — to control for the *transfer learning effect* which results when users run the same experiment multiple times, with conditions (i.e., independent variable) varied slightly
  - decide how to analyze results (see below)
- hypothesis testing asks questions like:
  - is there a difference? (between outcomes, experimental results, with different values of the independent variable)
  - if yes, how big is the difference?
  - is the estimate (results) accurate?
  - if yes, how accurate is the estimate?
- statistical measures
  - big subject!
  - variables are either *discrete* or *continuous*
  - if dependent variable is continuous, then variations in the value of this variable are called its *distribution*

  - if the distribution is "regular", then it is *parametric*; e.g., *normal distribution* = the classic "bell curve"
  - some data can be transformed to a normal distribution
  - *non-parametric* data can be "ranked" (ordered)
  - *contingency* data can be "clustered" (grouped according to feature values)
  - statistics is a big subject, and we won't have time to go into detail in this class—suffice it to say that the design of an experiment and collecting the experimental data are not the only important aspects; determining how to analyze the data is also very important

users: observational techniques

- protocols:
  - "think aloud" — user talks through what she is doing, uninterrupted by evaluator
  - "cooperative evaluation" — user talks through what she is doing, but evaluator can interrupt and ask questions (like "why did you do that?", "what if ...?")
- protocol analysis:
  - evaluator takes notes during the experimental session (e.g., using "paper and pencil")
  - evaluator records session using video and/or audio
    but this presents a "transcription" problem, where someone has to transcribe the experimental session from audio/video media to text (like a script)
  - computer logs user data
    but this presents a data filtering problem, where typically a program has to be written that will filter out unwanted actions so that analysis can focus on the actions that are being assessed in the experiment
  - user logs their activity
    but this can present a problem of incompleteness if user forgets to record everything

- automatic protocol analysis tools
  - there are some tools that allow evaluators to annotate video/audio, by watching the video/audio in one window and making notes that are recorded and tagged to positions in the video/audio stream
  - can be used to synchronize data from multiple recording sources
- post-task walkthrough
  - evaluator interviews users after they have completed the experimental interaction with the system, and shows them a (video) recording of their session and asks them to narrate what they did and why

users: query techniques

- evaluator asks user directly about the interface
- methods:
  - interview
  - questionaire
- types of questions in a typical questionaire:
  - general (e.g., user background information, demographics)
  - open-ended
  - scalar (e.g., "rate this on a scale of 1 to 5...")
  - multiple choice
  - ranked (e.g., "pick the best of ...")

users: evaluation through physiological responses

- use medical devices to record user's physiological responses while using interface
- types of bodily functions tracked:
  - eye tracking
    - record the number of *fixations* (times when the user's gaze is fixed on the same location for a minimum period of time)
    - record the length of each fixation
    - record the user's *scan path* (the trajectory across the interface where the user's gaze moves)
  - heart rate, pulse
  - breathing
  - skin secretions (sweat)
  - muscle activity
  - brain activity (e.g., EEG)

choosing evaluation techniques

- there are 8 factors to take into consideration when choosing the appropriate evaluation technique:
  1. stage in development cycle
     design (early) versus implementation (late) stage
     early stage should be quick and cheap
  2. style of evaluation
     i.e., laboratory versus field study
  3. level of subjectivity or objectivity
     how much will results rely on the (subjective) interpretation of the evaluator?
     versus quantitatave measures, like time to complete a task
  4. types of measures
     quantitative versus qualitative versus *mixed methods*
  5. information provided
     is high-level information about the interface sought—e.g., is it usable? which would require interviews, questionaires, etc.

   is low-level information sought—e.g., which font looks better?
6. immediacy of responses needed
   are evaluation results needed immediately, at the time of the experiments? then use something like a "walkthrough" or interview
   are results needed later? then use techniques that require analysis, such as computer logs or user logs
7. level of interference
   how intrusive is the evaluation process?
   the evaluation should NOT impact how the user uses the interface...
8. resources required
   how much time, money, personnel is required to conduct the evaluation?