Evaluating the Design without Users

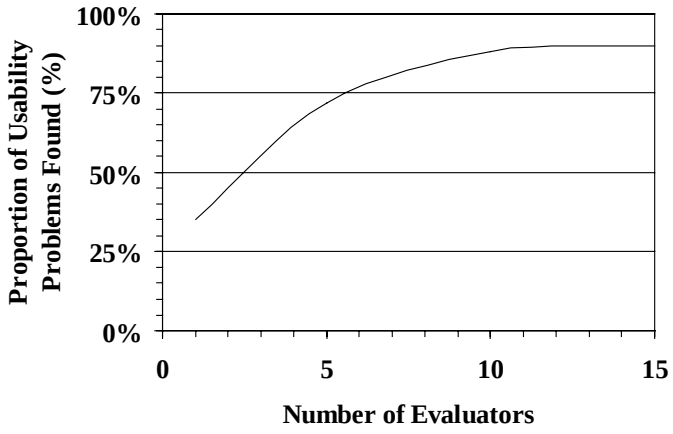
- Cognitive Walkthrough
- Action Analysis
- **Heuristic Evaluation**

Heuristic Evaluation

- Usability Engineering -- Jakob Nielson
  - find the usability problems in a user interface design so that they can be attended to as part of an iterative design process
  - systematic inspection of a user interface design for usability

Heuristic evaluation process

- You need a number of evaluators
  - individual evaluators will miss most of the usability problems
  - different evaluators tend to find different problems
  - You can achieve substantially better performance by aggregating the evaluations from several evaluators
  - cost-benefit analysis
  - recommendations:



Heuristic evaluation process

- each individual evaluator inspects the interface alone
- only after ALL evaluations have been completed are the evaluators allowed to communicate and have their findings aggregated
- results of an evaluations can be recorded in two ways:

Heuristic evaluation process

- Evaluation session:
  - the evaluator goes through the interface several times and inspects various dialogue elements and compares them with a list of recognized usability principles
  - general recommendation:

Heuristic evaluation process

- Domain-dependent systems
  - evaluators may be naïve with respect to the domain of the system, and therefore must be assisted to enable them to use the interface
  - supply the evaluators with a typical usage scenario, listing the various steps a user would take to perform a few realistic tasks
- Output
  - list of usability problems in the interface, annotated with references to those usability principles that were violated by the design
  - This does not provide a systematic way to generate fixes to the usability problems or a way to assess the probably quality of any redesign

Heuristic evaluation process

- Debriefing session
  - extend the heuristic evaluation method to provide some design advice
  - participants include the evaluators and any observers used in the evaluation sessions
  - conducted in brainstorming mode

Usability guidelines:

- simple and natural dialogue
- speak the users' language
- minimize the users' memory load
- consistency
- feedback
- clearly marked exits
- shortcuts
- good error messages
- prevent errors
- help and documentation

Simple and natural dialogue

- Ideal:
  - present exactly the information that the user needs and no more, at exactly the time and place where it is needed
  - information that will be used together should be displayed together
- Simplify the user interfaces as much as possible
- Information should appear in a natural and logical order

Simple and natural dialogue (continued)

- Graphic design and color
  - screen layout
  - colour
- Less is more
  - limit extraneous information
    - adding information can distract the user from the primary information, can confuse the novice user, and can slow down the expert user

Simple and natural dialogue (continued)

- Less is more (continued)
  - limit extraneous functions
    - People often think that “by providing lots of options and several ways of doing things we can satisfy everybody”
    - However, each additional feature is one more thing to learn, one more thing to potentially get wrong, and the manual gets bigger, more intimidating and harder to search

Speak the users' language

- Terminology
  - should be based on users' language, not on system-orientated terms
  - should be in the users' native language, including nonverbal elements like icons
  - view interactions from the users' perspective
  - try not to use words with non-standard meanings

Speak the users' language (Continued)

- Mappings and metaphors
  - good mapping between the computer display of information and the user's conceptual model of the information
  - user interface metaphors
    - mapping between the computer system and some reference system known to users in the real world

Minimize user memory load

- Easier to recognize something than to recall from memory
  - computer should display dialogue elements and let the user choose
  - also easier to modify information rather than generate it all from scratch
  - the objects must be visible to the user but too many objects will result in a loss of visibility for the objects of interest
  - system should describe the required format and provide an example of legal and sensible input (i.e. a default value)
  - system should supply the range of legal input and unit of measurement

Minimize user memory load (Continued)

- System should be based on a small number of rules that apply throughout the user interface
  - if there are a large number of rules
  - if there are no rules
  - generic commands
    - support the transfer of learning from one application to another

Consistency

- The same information should be presented in the same location on all screens and all dialog boxes and it should be formatted in the same way to facilitate recognition
- if users know the same command or action will always have the same effect they will feel more confident in using the system

Feedback

- The system should continuously inform the user about what it is doing and how it is interpreting the user's input
  - shouldn't wait until an error occurs
  - provide positive feedback and partial feedback
  - system feedback should not be expressed in abstract and general terms but should restate and rephrase the user's input to indicate what is being done with it
  - different types of feedback need different degrees of persistence

Feedback (Continued)

- Response times
  - feedback especially important where the system has long response times
  - Guidelines:
    - 0.1 second - limit for having the user feel that the system is reacting instantaneously. No special feedback necessary
    - 1.0 second - limit for the user's flow of thought to stay uninterrupted. The user will notice the delay but normally, no special feedback is necessary.
    - 10 seconds - limit for keeping the user's attention focussed on dialogue. For longer delays the user will want to perform other tasks while waiting. Computer should give feedback indicating when it will be done. Especially important when response times are highly variable.

Feedback (Continued)

- Response times (continued)
  - progress indicators
    - reassure the user that the system has not crashed but is working on the problem
    - indicate approximately how long the user can be expected to wait
    - provide something for the user to look at, making the wait less painful
    - for fast operations, an indicator is probably overkill, providing distraction
- System failure
  - informative feedback should be given when the system fails
  - "NO" feedback is the worst possible feedback

Clearly marked exits

- System should offer the user an easy way out of as many situations as possible
  - undo facility
  - high degree of responsiveness
  - must be visible to the user!

Shortcuts

- Should be possible for the experienced user to perform frequently used operations fast, using shortcuts
  - function keys, command keys, control sequences, double-clicking, right mouse-button clicking, gestures, etc.
  - type-ahead and click-ahead
  - jump directly to the desired location
  - reuse interaction history
    - 35% of all commands were identical to one of the five previous commands
    - 74% of the commands had been issues at least once before
  - system provided defaults

Good error messages

- Critical because ….
  - represent situations where the user is in trouble and potentially will be unable to use the system
- Four simple rules:
  - phrased in clear language and avoid obscure codes
  - be precise rather than vague or general
  - should constructively help the user solve the problem
  - should be polite and not intimidate the user or put the blame on the user
- Should also provide good error recovery

Good error messages (Continued)

- Multiple-level messages
  - use shorter messages that will be faster to read
  - give easy access to a more elaborate message

Prevent errors

- Try to design your system to avoid the error completely
  - instead of spelling things out, choose from a list
  - confirmation before executing actions that may have serious consequences
  - However, be careful of “capture error”
- Avoid modes
  - modes partition the user actions so that not all actions are available at all times which can be frustrating
  - the same user interaction will be interpreted differently depending on the current mode which can lead to errors
  - if you cannot avoid modes, prevent errors but explicitly recognizing the modes in the interface design
- Avoid having too-similar commands - “description error”
  - the descriptions of the two situations are almost identical and therefore likely to be confused.

Help and documentation

- Most users do not read manuals
- if users do want to read the manual, then they are probably in some kind of panic and will need immediate help

- on-line documentation has the potential for getting precise information faster than paper-based documentation
- on-line documentation adds an extra set of features to a system therefore complicating the interface

Help and documentation (Continued)

- Quality of writing is important
- 3 stages of interacting with manuals
  - searching
  - understanding the information
  - applying what you learned
- Searching
  - two main search tools are the index and the overview map of the structure of the information space

Help and documentation

- Understand
  - information should be written without jargon and should be written in a way that corresponds to the tasks the user wants to do and the sequence of actions necessary
- Apply
  - keep online help systems visible in a separate window while the user returns to the main application
  - let the user copy examples directly from the help system to have them executed
  - should be stated in step-by-step procedures
  - easy to check that they are understood the instructions and carried them out correctly

Usability guidelines:

- simple and natural dialogue
- speak the users' language
- minimize the users' memory load
- consistency
- feedback
- clearly marked exits
- shortcuts
- good error messages
- prevent errors
- help and documentation

Wednesday, June 23rd

- Guest Lecture from John Dill
- Midterm Review

- Readings:
  - Article 14: Designing for Error by Clayton Lewis and Donald A. Norman
  - Article 15: A Field Study of Exploratory Learning Strategies by John Rieman

CMPT 363

Assignment #3

** Due Dates: Part I due on June 30th and Part II due on July 7th **

Part I. (Due June 30th)

Design Decision (6%)

Form a project group of 4 people. Get together as a group and discuss your individual designs proposed in Assignment #2. When discussing your representative scenarios, you may find it useful to perform informal cognitive walkthroughs for each design. Based on this evaluation choose a design direction for your group. Write a one-page report documenting your proposed design direction along with some of the problems uncovered in the original designs. Make sure you justify your design choices.

Horizontal Prototype (6%)

Create a **horizontal prototype** of your proposed system using your choice of software (i.e. MetaCard, VB, HTML). The horizontal prototype should support three of your group's representative tasks (carried forward from Assignment #2). Place the prototype in the class directory along with a text file describing your three representative tasks. Please put your files in /gfs1/CMPT/363.ldumont/assn3. (This is on the SFU Campus Computer Network - CSIL), and please name your files as follows: LoginName_363assn3, where LoginName is the user id of one person on your team. (The file extensions will differentiate the prototype from the text file.)

Part II. Heuristic Evaluation (Due July 7th) (6%)

You will perform a heuristic evaluation on another group's prototype. The heuristic evaluations are to be conducted INDIVIDUALLY (i.e. each member of your group will submit their own heuristic evaluation of another group's prototype). Submit a one-page report of your individual evaluation for marking, and also submit a copy of your evaluation to the other team.

General comments about the assignment:

- The assignment is to be done in groups of 4.
- In the report, please include the names and login ids of all group members.
- All portions of this assignment will be due by the first break of the lecture.
- A good example of a cognitive walkthrough and a heuristic evaluation can be found at: ftp://ftp/cs/colorado.edu/pub/cs/distribs/clewis/HCI-Design-Book/chap-4.v-1
- A strong emphasis will be put on your ability to present your ideas clearly.
- Poorly written reports will be penalized.