



CS5340
HUMAN-COMPUTER INTERACTION

March 14, 2013

www.hcibook.com/e3

TODAY'S CLASS

- T3: extra credit opportunity & feedback
- T7
- Evaluation techniques (part 1)
- Paper Presentations

T3 EXTRA CREDIT

- **Optional**
- Due by March 21, 6pm, no exceptions.
 - Blackboard
- You may resubmit T3
 - Address feedback provided to you
 - Focusing on areas where you received low marks

T3 EXTRA CREDIT

- What to submit
 - Revised T3 submission
 - **You must provide** a concise, neatly organized, bullet-point overview of
 - What was lacking in the previous submission
 - How you have updated it to address this problem
 - **You must use this format:**
 - Issue #1: We did not describe stakeholder characteristics.
 - Fix #1: We updated the secondary and tertiary stakeholder descriptions to include a discussion of X, Y and Z.
 - **We will not grade if you do not follow this format**

T3 FEEDBACK

- Relate to T2/I2
 - Discuss *specifically* what you learned from T2/I2 that led you to state your requirements, etc.
 - Must provide evidence, not “Based on our T2/I2, we decided on requirement X”
 - Be more specific – what about your data led you to this conclusion

T3 FEEDBACK

- Stakeholder Analysis
 - Don’t just name the group, *what are their characteristics? What distinguishes them? What about them is important to know as you design?*
- Task Analysis
 - Decomposition
 - Plans
 - Effort: multiple levels of hierarchy, not just a simplistic set of nodes

T3 FEEDBACK

- Non-functional requirements
 - Not just generic, “should work on multiple platforms”
 - Be more specific about why that requirement is important, given
 - your target problem & populations
 - the kind of system you are hoping to design
- Storyboards
 - Neatness
 - Show context of use
 - what’s happening when people are using system, where are they, what events lead them to use it, etc.

T3 FEEDBACK

- Interaction Metaphors
 - A way of comparing 2 things
 - Real-world concept that you are leveraging to help users understand how to engage with your system
 - Desktop (WIMP OS)
 - Mailbox (email)
 - Chat “room” (space where you gather to talk)
 - “Mystery” for discovering reasons for tiredness

T7

Due 3/28, by 6pm

T7

MEDIUM-FIDELITY PROTOTYPE

- Move from paper to software prototype
- High fidelity in
 - look
- Medium fidelity in
 - feel
 - Breadth
- Low fidelity in
 - Depth

T7

MEDIUM-FIDELITY PROTOTYPE

- High fidelity in look
 - explore the graphic design of your final implementation
 - colors, fonts, alignment, icons, and white space
 - Lay out screens as you want them to appear in your final implementation.
 - Won't be identical to your final implementation
 - Will iterate one more time

T7

MEDIUM-FIDELITY PROTOTYPE

- Medium fidelity in feel
 - Main focus is on “look”...
 - But must be responsive
 - Buttons clickable, etc.
 - **Because classmates will be using it!**
 - Advanced interactions
 - Ok if not high fidelity
 - You can simulate these interactions with animation
 - or at least with a popup that describes in English what would happen

T7

MEDIUM-FIDELITY PROTOTYPE

- Medium fidelity in breadth.
 - Your prototype should include a set of features
 - correspond to the 3 functional requirements you chose to focus on in T4 (or a revised set of 3 functional requirements)
 - include every major screen or dialog you expect to have in your final implementation

T7

MEDIUM-FIDELITY PROTOTYPE

- Low fidelity in depth
 - Don't implement any back end
 - Where system responses are needed, make them canned (i.e., always the same) or random
 - Write minimal code!
- **The focus for this prototype should be on the front-end.**

T7

MEDIUM-FIDELITY PROTOTYPE

- Prototype will be evaluated in class: 3/28
 - Must make it easy for other people in the class to be able to run your prototype
 - If prototype requires special hardware, bring whatever is required to run your software to this class session
 - Can require a particular web browser/platform to ensure the correct appearance/operation
 - BUT browser/platform must be commonly available

T7

MEDIUM-FIDELITY PROTOTYPE

- Post
 - To Blackboard (PDF) **AND** to your team webpage (in an obvious location)
 - A link to your prototype
 - must remain frozen and accessible at this location
 - Startup instructions
 - Installing/running whatever software is required to run your prototype (this **MUST** be easy)
 - Specify the platform/browser requirements
 - Give any special instructions for starting it up

T7

MEDIUM-FIDELITY PROTOTYPE

- Post
 - System Overview
 - neatly organized, concise
 - Describing the purpose of your application (i.e., the problem(s) it solves or helps with) and background information about the domain (1 page max)

T7

MEDIUM-FIDELITY PROTOTYPE

- Post
 - Team Credits
 - A couple sentences describing each team member's contribution to the interface
 - Software Credits
 - **ALL** code, graphics, etc. that your prototype utilizes that you did not create yourselves
 - (i.e., stuff you got from the web or other sources).

T7 MEDIUM-FIDELITY PROTOTYPE

- Post
 - In your **blackboard report ONLY**:
 - 1 page (max!) bullet point description of how you updated your design based on T6 feedback
 - Bad: “We changed our design to include an option for text messaging friends.”
 - Good: “Users assumed that there would be some way for them to communicate with friends through our interface. As such, we changed our design to include an option for text messaging friends.”

T7 MEDIUM-FIDELITY PROTOTYPE

- Post
 - In your **blackboard report ONLY**:
 - 1 page (max!) bullet point description of the usability AND graphic design principles that you followed in creating this UI
 - Bad: “We focused on making our system learnable, so that new users can accomplish tasks quickly.
 - Good: “We focused on making our system learnable by addressing the specific principle of synthesizability. We did this with feature A, B, and C, by providing YYY animation that shows what happens when users do ZZZ”

T7

MEDIUM-FIDELITY PROTOTYPE

- Post
 - In your **blackboard report ONLY**:
 - 1 page (max!) bullet point description of your functional requirements, and what features you've designed to address them
 - Bad: "For functional requirement XXX we designed feature YYY".
 - Good: "For functional requirement XXX we designed feature YYY. This feature addresses our functional requirement by blah blah blah."

T7

MEDIUM-FIDELITY PROTOTYPE

- Successful assignments will
 - be aesthetically pleasing
 - Graphic design principles
 - Uncluttered
 - Neatly organized (grouping, whitespace, etc.)
 - ...
 - focus on designing a smaller set of features well
 - Vs tons of features that are not well-designed
 - Show you are thinking about usability principles

- A good time to begin “Introduction” for T10 (report)
 - *Remainder of T10 instructions still being updated*

EVALUATION

EVALUATION

- Test
 - Functionality
 - Usability
 - Experience
 - Impact
 - e.g., behavior/attitude/physiological/organizational/social change
- Important in all stages in the design life cycle

EVALUATION

- Factors to consider when choosing a method?
 - Stage (design v. implementation)
 - Laboratory v. Field
 - Evaluator subjectivity v. objectivity
 - Same results for any evaluator?
 - Quantitative v. qualitative data collection
 - Quantitative v. qualitative data analysis
 - Frequency, magnitude, duration, etc.
 - Quality (the nature, attributes of something), characteristics of an experience, attitude, etc.
 - Low v. high level information
 - Intrusiveness

GOALS OF EVALUATION

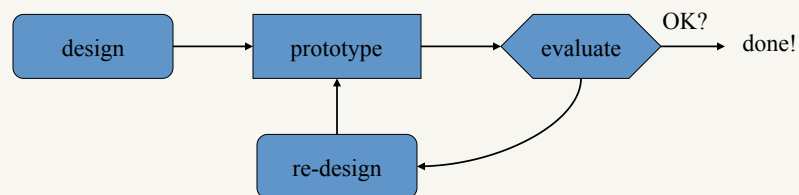
- Assess effect of system interaction on
 - User
 - Groups
 - Environments
- identify specific problems, shortcomings & positives
- identify nature of system interaction
 - How do people use it?
 - Short-term
 - Long-term (adoption)

EVALUATION

- Formative
 - Start in a good place ...
 - Observations
 - Interviews
 - Task Analysis
 - Stakeholder Analysis
 - Surveys
 - Iteration!
- Summative
 - End of process
 - If this is where you start evaluating, you are too late

ITERATIVE PROTOTYPING

- You never get it right first time
- If at first you don't succeed ...



EXPERT EVALUATIONS

- **Cognitive Walkthrough**
- **Heuristic Evaluation**
- Models (Ch 12, e.g. GOMS)
- Previous work
- Used at what stage(s) of the software lifecycle?

COGNITIVE WALKTHROUGH

- Detailed review of likely user interactions with system
- Start with
 1. Prototype or detailed system specification
 2. (representative) task descriptions
 3. Actions needed to complete tasks
 4. Description of users (the knowledge, experience etc. that evaluators can assume)

COGNITIVE WALKTHROUGH

- For each action, step through and “try to tell a believable story” about:
 - Do effects of actions match goals?
 - Will users **see** action is available?
 - Will users **know** action is one they need?
 - If action taken, will users understand feedback they get?

USABILITY HEURISTICS

- General principles, rules of thumb
 - Many to choose from
 - Nielsen's 10 principles
 - Shneiderman's 8 golden rules
 - Norman's rules from Design of Everyday Things
 - Mac, Windows, Gnome, KDE, Java guidelines
- Help designers choose design alternatives
- Help evaluators find problems in interfaces ("heuristic evaluation")

SOME WE'VE DISCUSSED

- User-centered design
 - Know your users
 - Understand their tasks
- Fitts' Law
 - Size and proximity of controls should relate to their importance
 - Tiny controls are hard to hit
- Memory
 - Use chunking to simplify information presentation
 - Minimize working memory
 - Rely more on recognition than recall

SOME WE'VE DISCUSSED

- Color guidelines
 - Don't depend solely on color distinctions (color blindness)
- Principles of direct manipulation
 - Affordances
 - Feedback

HEURISTIC EVALUATION

- "Systematic inspection of a user interface for usability" Nielsen'93
 - By experts
- Discount technique
 - Cheap (done w/HCI team)
 - Can use early
 - Flexible (throughout design process)
- ? evaluators find 75% of problems
 - 5
 - A single evaluator misses most problems!

HEURISTIC EVALUATION

- Use Neilson's ten heuristics
 - Note violations
 - Written
 - Vocalize + Observer
- Heuristic evaluator explicitly documents usability issues
 - Vs. researcher interpreting user actions to *infer* usability issues

HEURISTIC EVALUATION

- Distinct from traditional user testing
 - Observer can answer evaluators' questions about
 - Domain
 - Helps them better assess usability
 - how interface works
 - But after they have tried to understand it themselves & commented on usability issue
- Goal: expert opinion of how users might receive the system

HEURISTIC EVALUATION

- Use Neilson's ten heuristics
 - Assess severity of problems
 - 0: This is not a usability problem
 - 1: Cosmetic problem (only fix if extra time)
 - 2: Minor problem (low-priority fix)
 - 3: Major problem (important to fix, high-priority)
 - 4: Catastrophe (must fix)

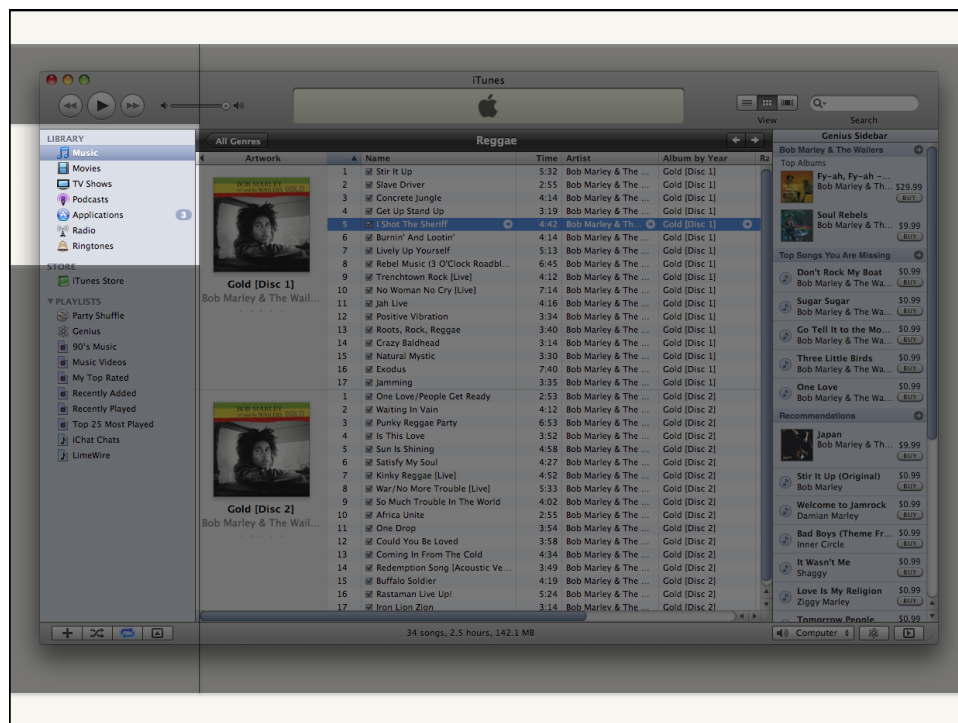
HEURISTIC EVALUATION

- Use Neilson's ten heuristics
 - Answer the following questions:
 - How common is problem
 - Does this issue happen in multiple aspects of the design?
 - Will problem persist
 - Will users keep running into this issue?
 - How easy for user to overcome
 - Is it a barrier to them doing what they need to do?
 - How seriously will problem be perceived?
 - A small annoyance or major disturbance?

NIELSEN'S HEURISTICS

1. MATCH THE REAL WORLD

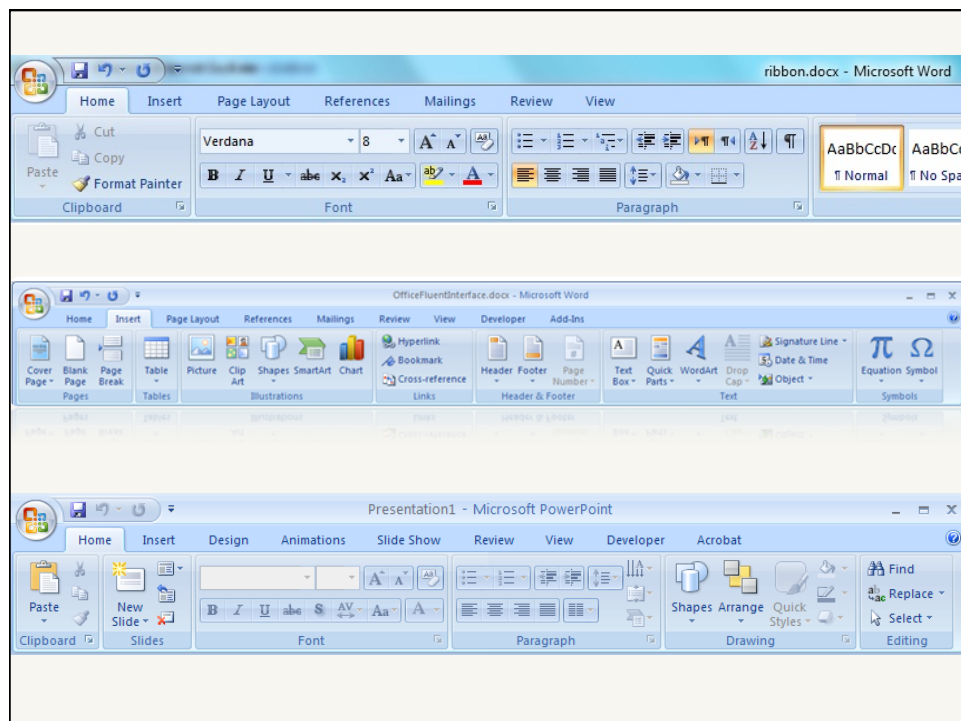
- “Speak the User’s Language”
 - Words, concepts etc. that are familiar to them
- Use common words, not techie jargon
 - domain-specific terms where appropriate
- Aliases/ synonyms in command languages
- Metaphors are useful but may mislead



NIELSEN'S HEURISTICS

2. CONSISTENCY & STANDARDS

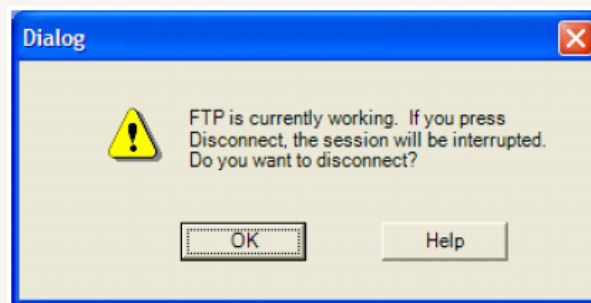
- Principle of Least Surprise
 - Similar things should look and act similar
 - Different things should look different
- Other properties
 - Size, location, color, wording, ordering, ...
- Follow platform standards
- Kinds of Consistency
 - Internal
 - External
 - Metaphorical



NIELSEN'S HEURISTICS

3. USER CONTROL & FREEDOM

- User preemptive
- Provide undo
- Long operations should be cancelable
- All dialogs should have a cancel button



NIELSEN'S HEURISTICS

4. VISIBILITY OF SYSTEM STATUS

- Keep user informed of system state
 - Cursor change
 - Selection highlight
 - Status bar
- Response time
 - < 0.1 s: seems instantaneous
 - 0.1-1 s: user notices, but no feedback needed
 - 1-5 s: display busy cursor
 - > 1-5 s: display progress bar

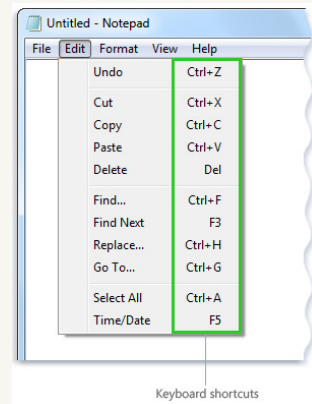


designingwebinterfaces.com/6-tips-for-a-great-flex-ux-part-5

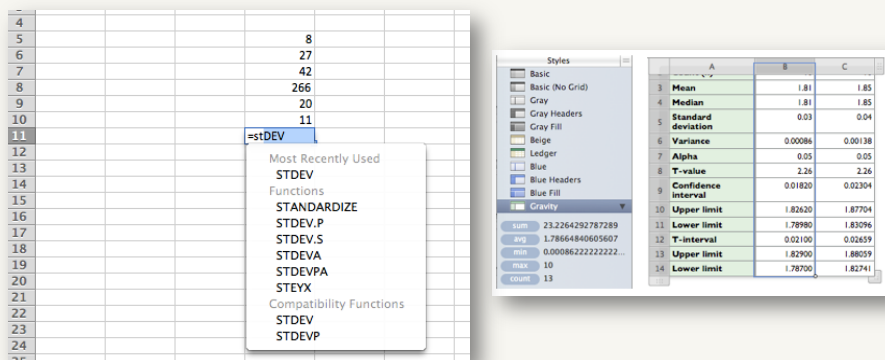
NIELSEN'S HEURISTICS

5. FLEXIBILITY & EFFICIENCY

- Provide easily-learned shortcuts for frequent operations
 - Keyboard accelerators
 - Command abbreviations
 - Bookmarks
 - History
- Accelerators to speed up interactions
 - Helps novices + experts

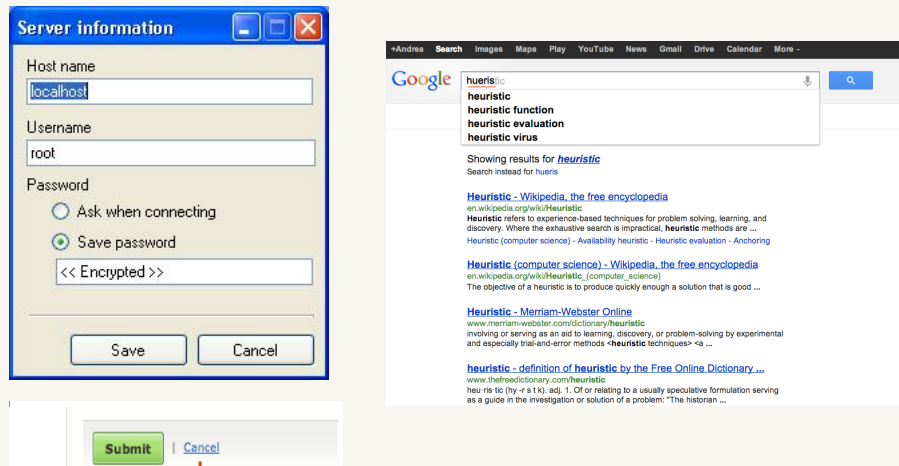


windows.microsoft.com/en-us/windows7/using-your-keyboard



NIELSEN'S HEURISTICS

6. ERROR PREVENTION

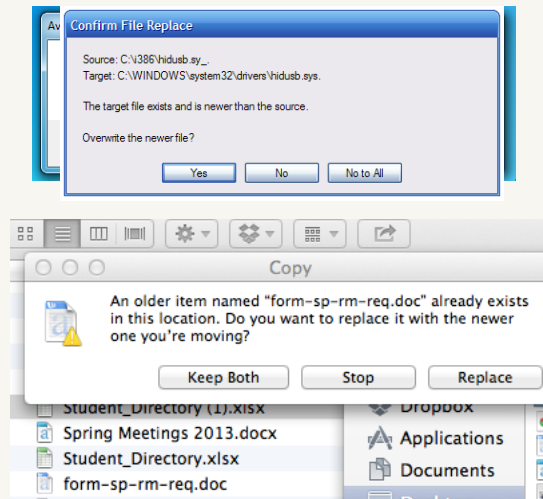


designingwebinterfaces.com/6-tips-for-a-great-flex-ux-part-5

NIELSEN'S HEURISTICS

6. ERROR PREVENTION

- Selection is less error-prone than typing
- Disable illegal commands
- Description Error
 - when two actions are too similar
 - e.g., case sensitivity
 - different things should look and act different
- Mode Error
 - Limit use of modes
 - Visibility of mode
 - Spring-loaded or temporary modes

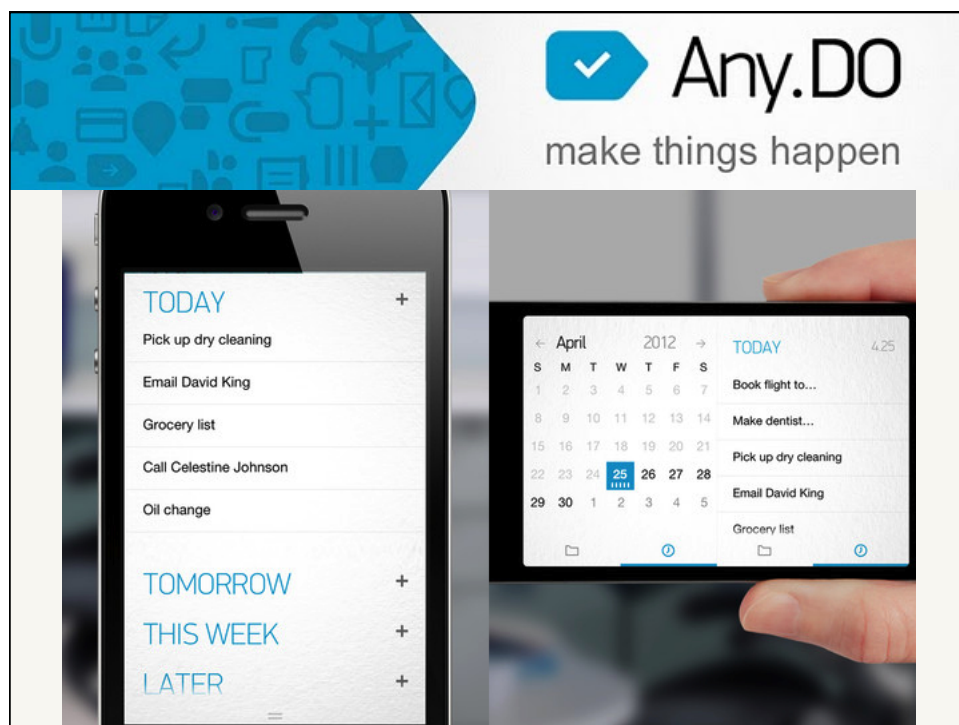
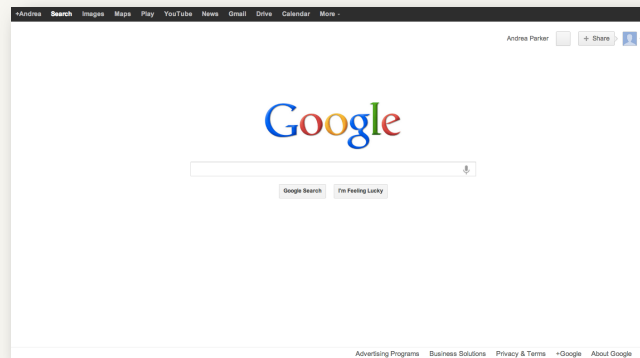


- Be precise; restate user's input
 - Not "Cannot open file", but "Cannot open file named paper.doc"
- Give constructive help
 - why error occurred and how to fix it
- Be polite and non-blaming
 - Not "fatal error", not "illegal"
- Hide technical details (stack trace) until requested

NIELSEN'S HEURISTICS

8. AESTHETIC AND MINIMALIST DESIGN

- Less is More" / KISS
- Omit extraneous info, graphics, features
 - Why particularly important for an interface like Google?



FOR NEXT WEEK

- I3 due by 6pm 3/21
- T3 extra credit (optional) due by 6pm 3/21
- No reading - use the extra time to work on T8
- Next week's class
 - Attendance taken
 - Guest lecture
 - Project time

PAPER PRESENTATIONS

- Sambasivan et al., *Ubicomp4D: infrastructure and interaction for international development--the case of urban indian slums*, Ubicomp'09 **Rohit Wadke**
- Froelich et al., *The Design and Evaluation of Prototype Eco-Feedback Displays for Fixture-Level Water Usage Data*, CHI'12 **Paras Chauhan**
- Kuznetsov et al., *Red Balloon, Green Balloon, Sensors in the Sky*, Ubicomp'11 **Jingtong Liu**