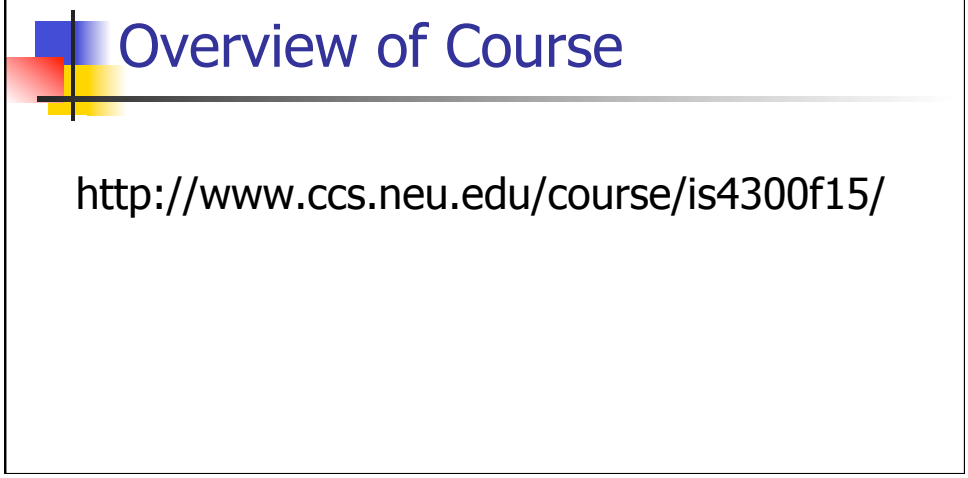
Human-Computer Interaction
IS 4300

Overview for Today

- Brief review
- HCI Development Process
- Critical Analysis of UIs
  - Heuristic Evaluation
  - Cognitive Walk-through Evaluation
- Homework I2



Course Website

IS4300– Human-Computer Interaction

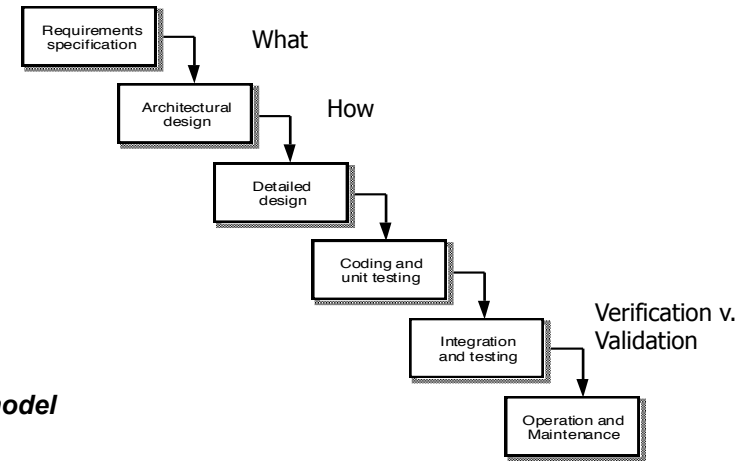
[Syllabus] [Schedule] [Homework] [Projects] [Resources] [Directory]

Schedule

Date	Topics & Readings	Assignments	
		Due	Start
9/9	Overview of HCI and course. Getting started on projects.		I1,P1
9/14	HCI development process (Benyon Ch 1 & 3). Critical Analysis of UIs (Benyon 4.5 & 10.2). Team project brainstorming.	I1	I2
9/16	Humans (Benyon Ch 21 & 25). Team project brainstorming.		
9/21	Doing observational studies (Benyon Ch 2 & 7; Fetterman; Example 1; Example 2).	I2	I3

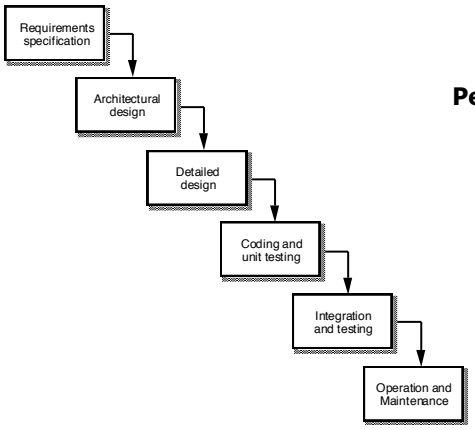
User-Centered Design

The Software Lifecycle



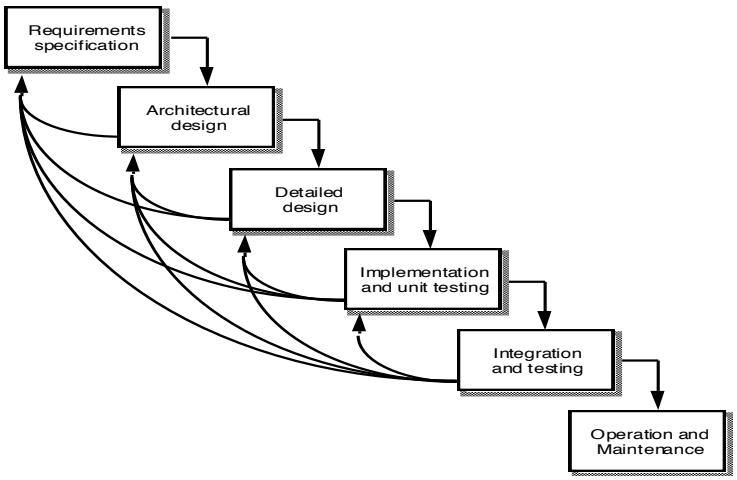
The waterfall model

Why doesn't this work for UIs?



People are insanely complicated.

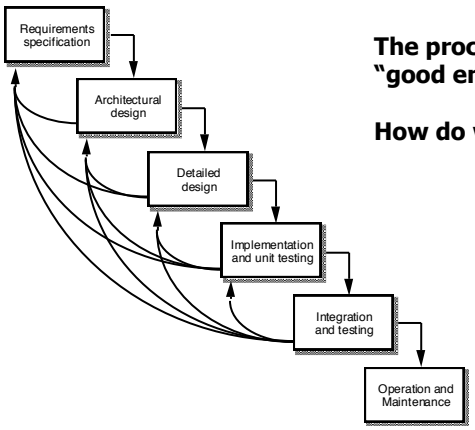
Lifecycle for UIs



User-Centered Design

- Try lots of stuff. See how it plays with users.
  - Involve representative users in all stages of the development process.
  - Minimize the cost of and commitment to prototypes.
  - Users often can't tell you which alternative is "better" – have to test and measure.

Usability Engineering



The process by which we achieve "good enough" usability.

How do we know when we're there?

Usability Engineering

- Must define usability attributes (multi-dimensional)
- Must define specific measures for each
- Must define "good enough" (goal) levels for each
  - If appropriate, current & ideal levels for each
- Example attributes (measures?)
  - Learnability
  - Efficiency
  - Memorability
  - Low error rate
  - Subjectively pleasing

User-Centered Design

- Putting people first; designing interactive systems to support people and for people to enjoy
- Thinking about what people want to do rather than what the technology can do
- Involving users in the design process
  - Participatory design

The process of … design

- Understanding (2 weeks)
  - Requirements
  - Stakeholders
- Design (2 weeks)
  - Scenario-based Design
  - Envisionment
- Evaluation (2 weeks)
  - Inspection
  - User evaluations

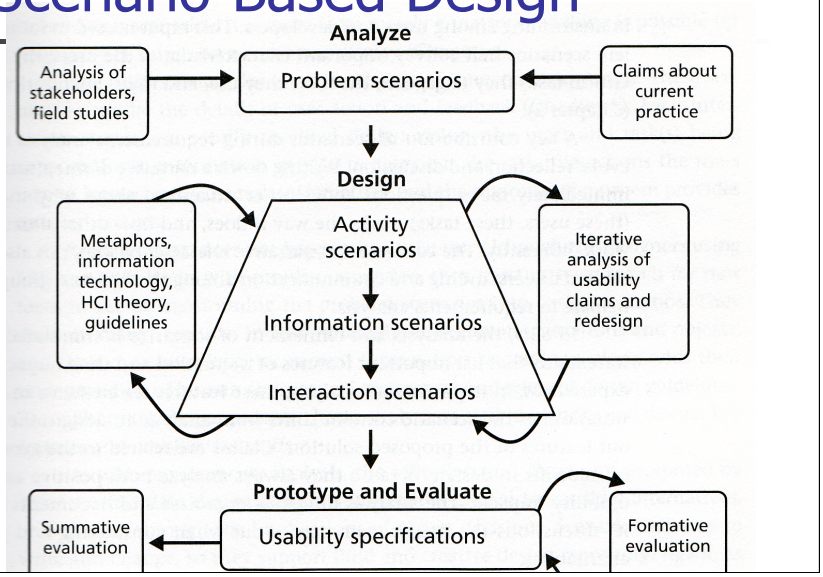
First Step: Requirements & Specifications

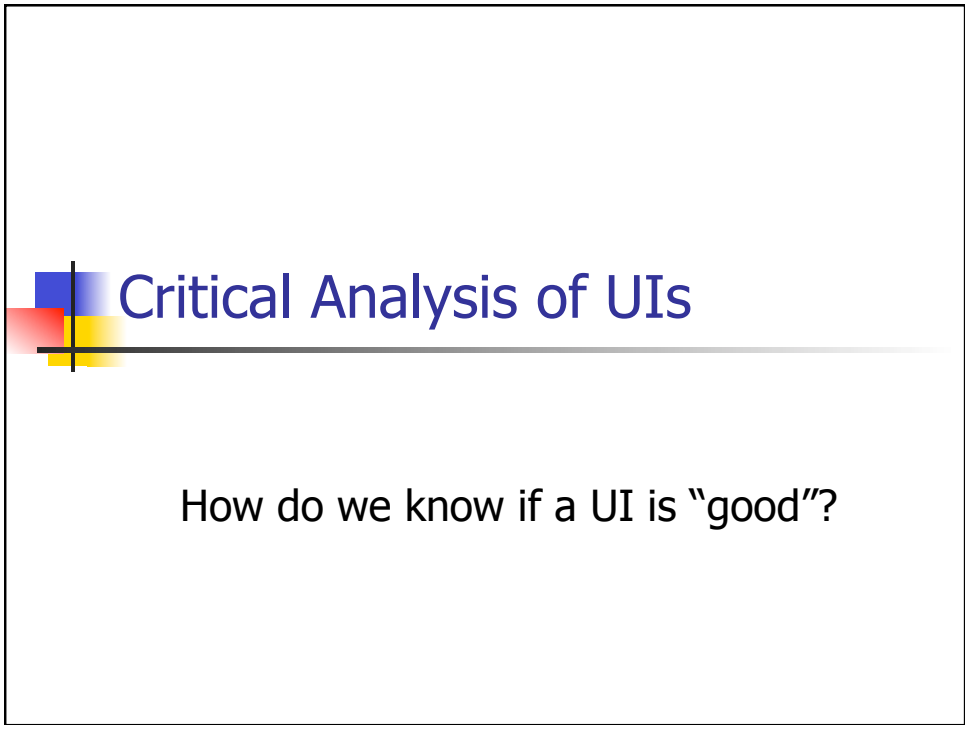
- Know your user
- Knowledge about people in general
- Very, very specific knowledge about users and work environment
- Ethnography in 1 week

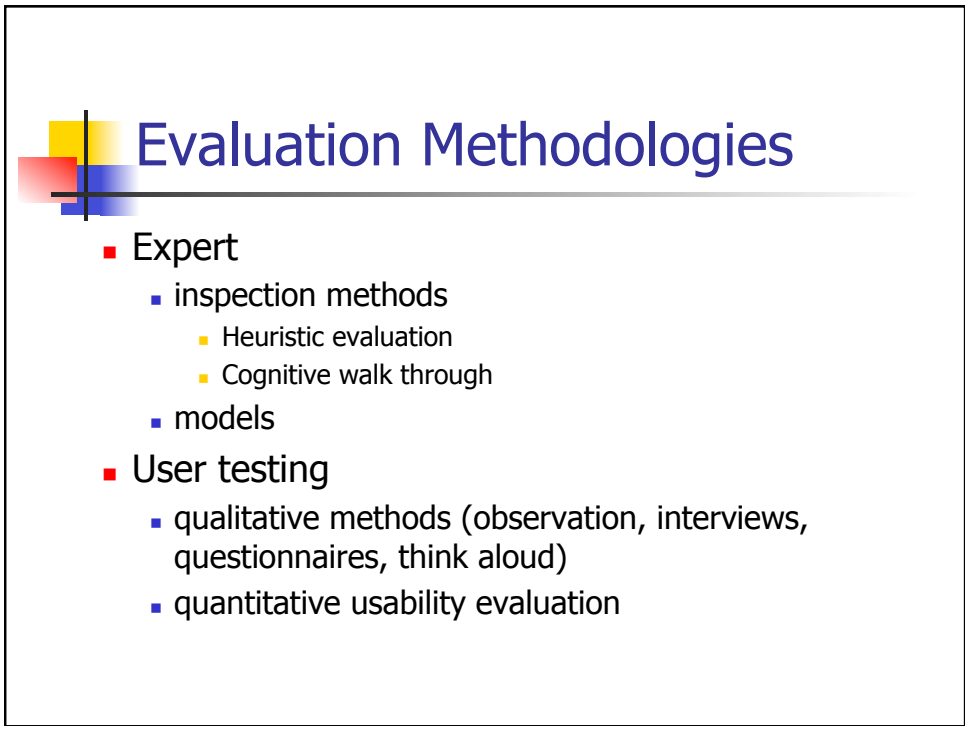
The process of … design

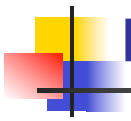
- Personas
- Scenarios
  - Stories - real-world experiences of people.

Rosson & Caroll Scenario-Based Design



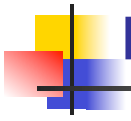

Critical Analysis of UIs
How do we know if a UI is "good"?


Evaluation Methodologies
Expert
inspection methods
Heuristic evaluation
Cognitive walk through
models
User testing
qualitative methods (observation, interviews, questionnaires, think aloud)
quantitative usability evaluation



Design Heuristics

- "Rules of Thumb" for improving usability
- aka "Design principles"



Heuristics / Design Principles

- There are many "checklists" available
  - Nielsen's 10 design heuristics
  - Tognazzi's First Principles of Interaction Design
  - Gerhardt-Powals' cognitive engineering principles
  - etc

Nielsen's Heuristics
1. Feedback

- Clearly tell user effects of input actions
- Keep user informed of system state
  - Cursor change
  - Selection highlight
  - Status bar
- Feedback re: processing - response time
  - < 0.1 s: seems instantaneous
  - 0.1-1 s: user notices, but no feedback needed
  - 1-10 s: display busy cursor or other feedback
  - > 10 s: display progress bar

Feedback

Feedback

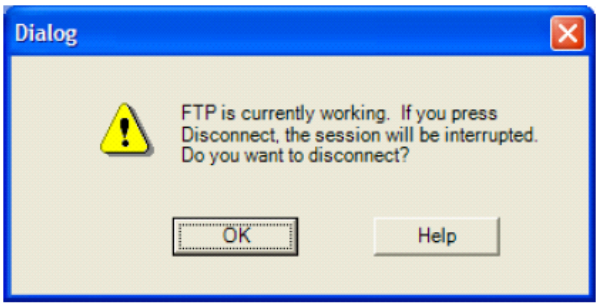
- Air France Flight 447, 1 June 2009, Airbus A330-200
- Stalled, crashed, killed 216 passengers and 12 aircrew
- Final report:
  - Initial cause: icing of airspeed sensors
  - Many feedback problems:
    - Inconsistency between the airspeed measurements
    - Incomprehension of the situation when the autopilot disconnection occurred,
    - The lack of a clear display in the cockpit of the airspeed inconsistencies identified by the computers
    - A failure to identify the aural stall warning
    - The appearance at the beginning of the event of transient warnings that could be considered as spurious
    - The absence of any visual information to confirm the approach-to-stall after the loss of the limit speeds

Nielsen's Heuristics
2. Speak the User's Language

- Use common words, not techie jargon
  - But use domain-specific terms where appropriate
- Don't put limits on user defined names
- Allow aliases/synonyms in command languages
- Use good metaphors

Nielsen's Heuristics
3. Clearly Marked Exits / Navigation / Freedom

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



Nielsen's Heuristics
4. Consistency

- 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 (aka "familiarity")
  - Metaphorical

Nielsen's Heuristics
5. Prevent Errors

- Selection is less error-prone than typing
- Disable illegal commands
- Description Error
  - different things/commands should look and act different
- Mode Error
  - Eliminate modes
  - Visibility of mode
  - Spring-loaded or temporary modes

Nielsen's Heuristics
6. Minimize User Memory Load

- Use menus, not command languages
- Use combo boxes, not textboxes
- Use generic commands where possible (Open, Save, Copy Paste)
- All needed information should be visible

Nielsen's Heuristics
7. Shortcuts / Flexibility / Accelerators

- Provide easily-learned shortcuts for frequent operations to improve efficiency
  - Keyboard accelerators
  - Command abbreviations
  - Styles
  - Bookmarks
  - History

Nielsen's Heuristics
8. Simple design

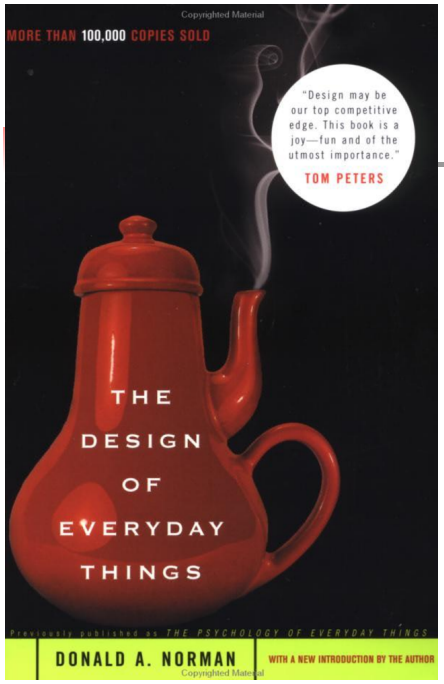
- "Less is More" / KISS
  - Omit extraneous info, graphics, features

Nielsen's Heuristics
9. Good Error Messages

- 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
10. Help and Documentation

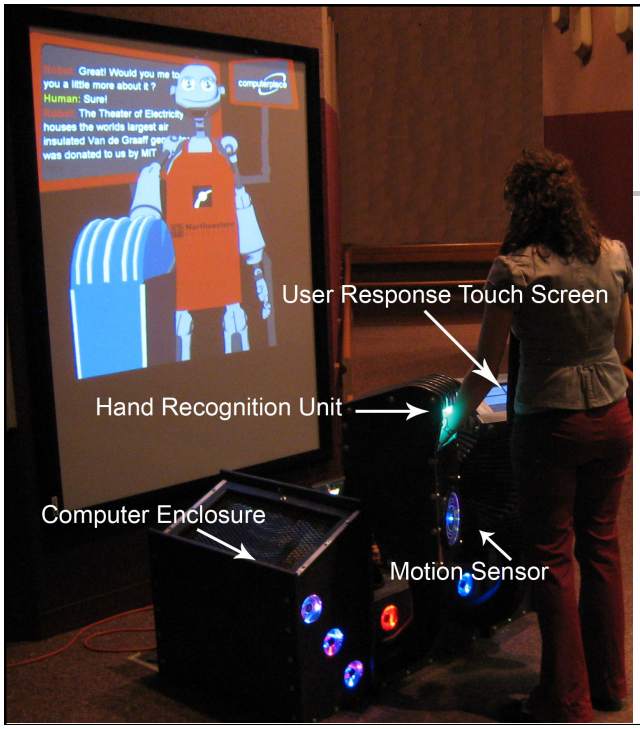
- Model
  1. Searching
  2. Understanding
  3. Applying
- Important features
  - Index
  - Overview map
  - Help visible while user is applying
  - Describe confirmatory feedback



A few additional rules/principles…

Affordances

- The fundamental properties of a thing that determine just how it could possibly be used.
  - Examples?
    - A chair affords sitting
    - Knobs are for turning.
    - Slots are for inserting things into.



- How to get visitors to put their hand in the box?

Visibility

- *aka "Obviousness"*
- The correct parts must be visible.
- They must convey the correct message.
- Impacts learnability.

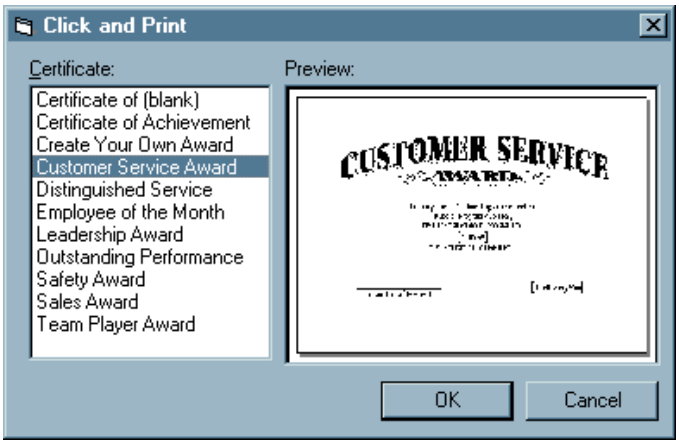
- How different from affordance?
- Examples?

Some (bad) examples…

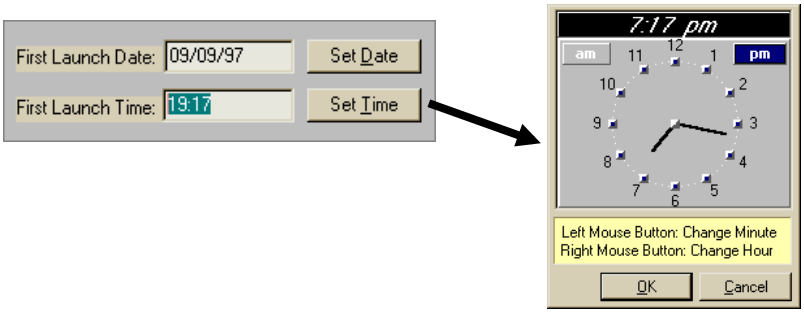
What's wrong with these?

Example 1

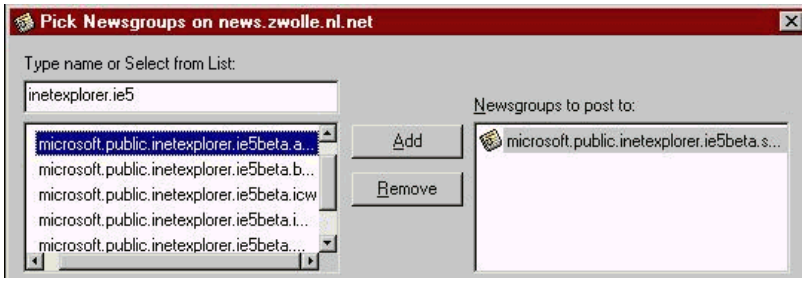
Example 1 - redesign



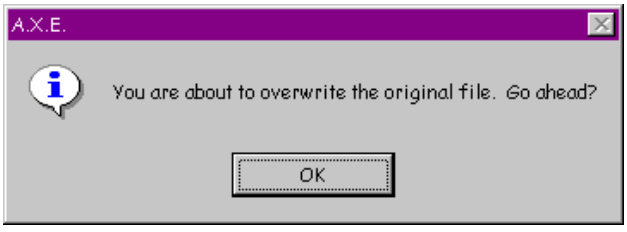
Example 2

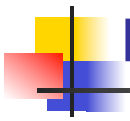


Example 3



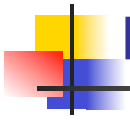
Example 4





Exercises in Inspection

- Break into groups
- One person must have a laptop or smart phone
- Pick an uncommon web site (preferred) or app
- Do a heuristic evaluation



Exercise 1

- "Heuristic evaluation"
- Critique a UI using Heuristics discussed
  1. Feedback
  2. Speak the User's Language
  3. Clearly Marked Exits
  4. Consistency
  5. Prevent Errors
  6. Minimize User Memory Load
  7. Flexibility / Shortcuts
  8. Simple Design
  9. Good Error Messages
  10. Help and Documentation
  11. Use Appropriate Affordances
  12. Visibility / Obviousness

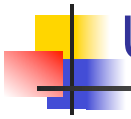
Inspection methods

- Cognitive walkthrough
  - Walk through each step in the task and evaluate:
    1. Given current user goal, is the choice of action obvious?
    2. Will users see that the action is available?
    3. Once users have found the action, will they know it is the one they need?
    4. After the action is taken, will users understand the feedback they get?

Exercise 2

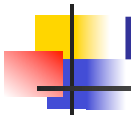
- Cognitive walk-through
  - A more methodical approach to heuristic evaluation

1. Define a task (as end goal, not how-to)
2. For each step (UI action)
   - Is the next action obvious?
   - Is the effect of the action taken obvious?



Individual Homework #2 UI Critique

- Find 2 good & 2 bad examples of UI design
- Use Nielsen's Heuristics
  - Make explicit reference to them!
- Include visuals (screen shots)
- Make suggestions for improvement



Project Review

- Must have a substantial UI
- UI must be interactive
- Creative, original, non-obvious is better
- Ideas: research papers & past CHI, UIST, IUI
- Each project should have 3-5 members
- Ideally complementary skills

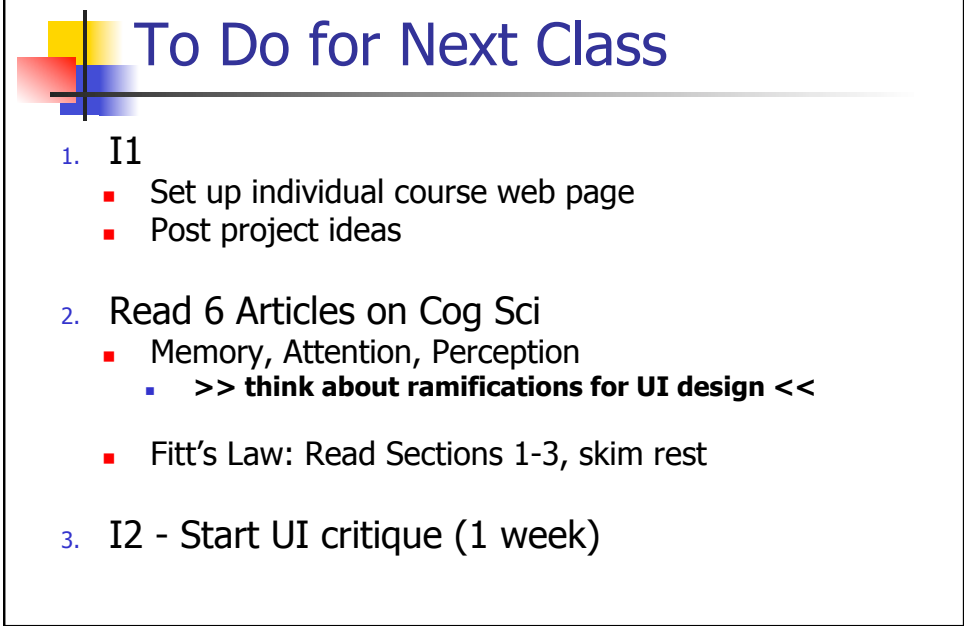
Projects

- Next class
  - You each present your top idea to the class (30 seconds)
- By 9/23
  - Email me a brief description and list of team members.
  - I'll reply with OK, or suggestions for change.
- 9/28 – Project proposal due

Plug for study

- Volunteers Needed to Participate in a Study to Evaluate an Automated System to Promote Wellness
- It takes one hour and pays $15

https://www.surveymonkey.com/s/AgentStudy


To Do for Next Class
1. I1
Set up individual course web page
Post project ideas
2. Read 6 Articles on Cog Sci
Memory, Attention, Perception
>> think about ramifications for UI design <<
Fitt's Law: Read Sections 1-3, skim rest
3. I2 - Start UI critique (1 week)