Micro-survey top topics we’ll review today:
Metaphors, sketching, grouping and other design principles

THIS WEEK:
Wednesday:
• Evaluation
DUE SUNDAY: Project Part 2

NEXT WEEK:
Monday:
Finish evaluation; in-class work
Wednesday:
Part 3 checkpoint 1 due TODAY
Due SATURDAY: Peer Eval for Part 2
UI metaphors

• folder for storing data
• desktop to see different apps
• trash can for deleting data – bad use: to eject a USB stick
• traffic light – to signify when required info is missing, also window control
• more is up, down is less e.g., arrows to affect volume
• drag and drop to supply info to apps
• wizard to handle app setup – can be more difficult to use though; linear metaphor whereas often doesn’t need to be
Sketching

• No “standards” – sketching an interface though means it shouldn’t need extra notes; if you have notes you can make sure they are separated from the interface design.

• Sketching is a type of brainstorming – you’re just getting ideas down graphically.

• Evaluate both low-fidelity and high-fidelity designs to test things that cannot be included in a sketch, but will be noticeable in the final product.

Grouping, other design principles

• If you have ideas for multiple ways to group things, create multiple designs and test them with users.

• Reducing – critically study everything in your design and decide if users absolutely need it at hand; take away what they don’t.
Design and Evaluation

Learning objectives:
1. Understand the iterative nature of design.
2. Understand types of evaluations and when they should be used.
3. Understand why to develop an evaluation plan and what goes into it.
4. Understand different types of user studies and their major components.

See the Evaluation Guide for details on evaluation paradigms and techniques. You are responsible for this material and are expected to include it as applicable in your plans.
Design Practice Tips

• Iterate and don’t get attached
• Ignore implementation details
• Make sure you consider errors

What is the iteration cycle?
Iteration/Testing

• Iteration Cycle:

Note that evaluation is collecting data, which increases understanding of user needs, which enables you to design...

• Good design absolutely requires iteration
• You must be prepared to throw out designs
• Constantly evaluate
Ignore Implementation Details

A well designed, interactive system that addresses clearly identified user needs is more important than your ability to implement a system.
Errors

• Users will make errors

• System can make errors in interpreting input

• You must design for errors

• You must design for people to accurately correct the input.
Handling Errors

• When designing systems which must interpret “natural” input or potentially ambiguous input, you must assume the system will get it wrong more often than not

• Even if system is accurate 90% of the time, user will perceive system getting it wrong more often than not

• Always include ways to provide backward and/or forward error correction

• Provide rich system feedback so user knows current state of system
Evaluation

Definition: “Process of systematically collecting data that informs us about what it is like for a particular user or group of users to use a product for a particular task in a certain type of environment” (Preece et al. 2002)

• In short, how well is system accomplishing what we intend it to accomplish?
• How well is it meeting the goals and needs of users?
Project Part 3

• Goals:
  – Evaluate your designs and perform evaluations of others' designs.
  – Refine your design to develop a final, high fidelity prototype.

• Evaluations:
  – You will need to develop and follow an evaluation plan for your designs using multiple techniques/methods (at least 3 different techniques) to assess them
  – Evaluate your design with actual users and with 2 other project groups, who will act as expert evaluators. (We will assign evaluators.)
Example Evaluation Goals

1. To assess extent and accessibility of systems functionality
   – Does system do enough? Can users access functions?

2. To assess users’ experience of interaction
   – Do they like it? Do they understand it?

3. To identify specific problems with system
   – Is something done wrong? Can aspects be improved?

4. To understand real world context
   – How do users use technology? Can design be improved, can work be automated, can we help a potential user group?

5. To compare designs
   – Best/better/worse *Essential* features

6. To engineer toward a target
   – Is design good enough?

7. To check conformance to a standard
   – E.g., Microsoft design guidelines, Mac interface guidelines
Types of Evaluation

• Formative
  – Evaluate new system as it is being designed and developed
  – Assumption is system is incomplete and that feedback can be quickly folded back into design process

• Summative
  – Evaluation performed on a completed system
  – Provides information regarding how well system meets expectations and goals previously decided upon
  – Used to assess competing, completed products

• In both cases, you need to be aware of biasing and control for it
Biases in the Real World

• Hawthorne effect/John Henry effect

• Experimenter effect/Observer-expectancy effect

• Placebo effect

• Novelty effect
Biases in the Real World

- Hawthorne effect/John Henry effect
  - Motivational effect of interest being shown (H)
  - Realization that you are in a control group makes you work harder (JH)

- Experimenter effect/Observer-expectancy effect

- Placebo effect

- Novelty effect
Biases in the Real World

• Hawthorne effect/John Henry effect

• Experimenter effect/Observer-expectancy effect
  – A researcher’s bias influences what they see: if you expect to see something, maybe something in that expectation leads you to see it
  – Solved via double-blind studies

• Placebo effect

• Novelty effect
Biases in the Real World

• Hawthorne effect/John Henry effect

• Experimenter effect/Observer-expectancy effect

• Placebo effect
  – Subject expectancy
  – If you think the new GUI, etc., has some benefit, then it may

• Novelty effect
Biases in the Real World

• Hawthorne effect/John Henry effect

• Experimenter effect/Observer-expectancy effect

• Placebo effect

• Novelty effect
  – Performance improves when technology is instituted because people have increased interest in new technology
How do you control for biases?

• Cannot fully – it’s more an awareness issue

• Approach any test data with some skepticism

• Assume subjects are trying to be helpful, so any errors they bring up must be pretty serious

• Aggressively seek contradictory data
Critical Evaluation Concepts

• *Reliability* of a study/test/measure: Can you repeat it and get same results every time?

• *Validity* of a study/test/measure: Is what you claim you’re measuring actually what you are measuring?

• *Ecological validity*: How close testing environment is to actual environment
  – Experimental lab setup: Low ecological validity
  – Actual workplace: High ecological validity
Evaluation Process

1. Develop an evaluation plan
2. Recruit subjects
3. Gain consent
4. Perform evaluation
5. Interpret results
6. Act on results
Why have them?

To ensure critical evaluation concepts – reliability, validity, and ecological validity of the evaluation findings.
Developing an Evaluation Plan

1. Identify *specific goals* of system

2. Decide what *questions* should be answered

3. Identify specific *metrics* and *measures* to assess success of reaching goals and of answering questions

4. Choose one or more *evaluation techniques*

5. Decide *where* to run evaluations

6. Decide *who* will participate in evaluations
1. Identify **specific** system goals

• What is important for your system to accomplish?
  – Faster task completion, fewer errors, greater throughput, higher satisfaction...

• How do you develop, select, and prioritize goals for evaluation?
  – Data collected from earlier field studies helps set the agenda for the evaluation plan and its goals
    • Cultural models, what users value most
    • Time, effort observed with particular tasks
  – Let the *real-world data* tell you what is most important in your evaluation
2. Decide **questions** to be answered

- What do you want to learn from an evaluation?
- Questions can be quantitative or qualitative
- Examples
  - Are people faster? (quantitative)
  - More satisfied? (qualitative)
  - How well does it blend with existing work practices?
  - How does it mesh with culture of practice?
2. Decide **questions** to be answered

- What do you want to learn from an evaluation?
- Questions can be quantitative or qualitative
- Examples
  - Are people faster? (quantitative)
  - More satisfied? (qualitative)
  - How well does it blend with existing work practices?
  - How does it mesh with culture of practice?

Figure out questions answer to determine how well your system meets your goals. These go into your evaluation plan.
3. Identify specific **metrics** and **measures**

- Identify specific *metrics* to **measure success of reaching goals**
  - Speed
  - Throughput
  - Error rate
  - Time to learn
  - Satisfaction

- But you cannot always *directly* measure some goals’ success...
3. Identify specific **metrics** and **measures**

- Identify specific *metrics* to **measure success of reaching goals**
  - Speed
  - Throughput
  - Error rate
  - Time to learn
  - Satisfaction

- But you cannot always *directly* measure some goals’ success...

  **Figure out metrics to measure to answer your questions. These go into your evaluation plan.**
4. Choose **evaluation paradigms and techniques**

Always use multiple paradigms and techniques – why?

– Multiple techniques will reveal different ways the same system affects users
5. Decide **where** to run evaluations

- Controlled environment

- Actual work environment

- What about a simulated work environment?
5. Decide **where** to run evaluations

- **Controlled environment**
  - We can focus on very specific questions
  - Easier to collect and later analyze quantitative data
  - But lack the reality and authenticity of the actual workplace

- **Actual work environment**

- **What about a simulated work environment?**
5. Decide **where** to run evaluations

- Controlled environment

- Actual work environment
  - View use of system *in situ*
  - A truer test of worth of system to support real-world work practices
  - But harder to collect quantitative data. How could you do this?

- What about a simulated work environment?
5. Decide **where** to run evaluations

- Controlled environment

- Actual work environment

- What about a simulated work environment?
  - What elements of the authentic work location would need to be replicated? Could they be replicated?
6. Decide **who** will participate

- Who will evaluate the designs?
- There is no one right answer...
- Experts lend experience to evaluation
- Actual end users provide authentic test
- Guy down the hall can provide a quick “reality check” on system’s design

- But *always* test with actual end users if you have access
Micro-survey

Learning objectives:
1. Understand the iterative nature of design.
2. Understand types of evaluations and when they should be used.
3. Understand why to develop an evaluation plan and what goes into it

Pre-work for Monday – Think Aloud (see Progress page)
Evaluation paradigms and techniques

For our use, a **paradigm** is a set of concepts or **patterns** that **include research methods** and standards for what creates legitimate contributions to a field.

We’ll talk about **4 paradigms**, and some of their associated **techniques**.
Evaluation paradigms

These are ways to approach evaluation, not actual methods or techniques to perform evaluation.

A. “Quick and dirty” evaluations
B. Usability testing
C. Field studies / naturalistic studies
D. Analytical evaluations
Paradigm A. Quick and Dirty

• What it is:
  – Informal feedback: low cost, fast, qualitative data

• Typical “metrics”:
  – Does the user understand system?

• How to do it:
  – Woz, scenarios, storyboards, sketches, ...
Paradigm A. Quick and Dirty

• Keep in mind:
  – *Any* feedback is better than no feedback
  – In general, the less familiar the person is with project, the better their feedback will be

• Pros and Cons:
  – Positives: fast, cheap, you can do it at ANY time
  – Negatives: strictly qualitative data that may not be generalizable
Technique 1: Think Aloud

• User tells you what they’re thinking as they use a prototype to accomplish a specific task

• PILOT THE TASK!
  – Decide your question(s): What do you want to learn?
  – Create a *specific* task for subject to perform

• Ask user to say whatever comes to mind as they use the prototype to accomplish the task
Paradigm B. Usability Testing

• What it is:
  – Carefully measured performance on prepared tasks typical for a system’s intended uses

• Typical “metrics”:
  – Number of errors (error rate), time to complete task, number of tasks performed
  – Satisfaction, cognitive load (NASA TLX), learning time

• How to do it:
  – Plan, script, and pilot usability experiments
  – Analyze data
Paradigm B. Usability Testing

• Keep in mind:
  – Choice of subject significantly affects the validity of the results

• Pros and Cons:
  – Positives: Quantitative data, focus on specific questions
  – Negatives: Significant start-up costs, including functional system for subjects to use
Usability Testing Results

“Results from our tests indicate it took the user 30 seconds, on average, to input a new entry into the database.”

– If we’re trying to improve an interface, what does a result like this tell us?
Usability Testing Results

“We added shortcut keys, switched to a multiple-document GUI, ported it to the Mac, eliminated the need for a mouse to be used, and added type-ahead capabilities to our application. Our users are 50% faster because of the type-ahead feature.”

— What can you say about this claim?
Usability Testing Results

“To test our interface, subjects first performed 10 tasks using the current interface. Subjects then performed the same 10 tasks using our new interface. Results indicate that our new interface improved task performance by 70%.”

— How much faith can we have in these results?
Determining Cause and Effect

• To determine cause-effect relationships, we must carefully observe differences between when suspected cause is, and is not, present

• Making tightly controlled *comparisons* is only way we can know how one particular change affects user

• Experimental process provides framework for establishing cause-effect relationships
Technique 2: Experimental Process

Usability testing paradigm technique/method.

1. Formulate a hypothesis
2. Identify independent, dependent variables
3. Design a controlled experiment
4. Check for:
   – Validity
   – Reliability
   – Confounds
5. Pilot the study
6. Select representative subjects
7. Randomly assign to conditions
8. Analyze results
Main Experimental Study Designs

• Between-subjects
  – Each subject exposed to only one condition
  – Positives: no learning effects
  – Negatives: need more people

• Within-subjects
  – Every subject experiences every condition
  – Positives: fewer subjects than between-subjects
  – Negatives: must control for learning effects
Experiment Design Checks

• Validity
  – Are we measuring what we say we are measuring?

• Reliability
  – If we run the experiment several times, do we get the same results every time?

• Confounds
  – Are there variables we didn’t control for which may be influencing the results we’re obtaining?

• Situation
  – If we test new features after subjects do the control condition (no new features); what’s the issue?
Running a Study

• The more subjects the better

• For large effects, fewer subjects are needed to notice that the effect is statistically significant

• Interpreting the results: consult your nearest, friendly statistics expert...
Paradigm C. Field / Naturalistic Studies

• What it is:
  – Interviews and Observations

• Typical “metrics”:
  – Does the system mesh with desired work practices?
  – What parts of the system do not seem to work well?
  – How do users actually adopt and use the system?

• How to do it:
  – Like Contextual Inquiry
Paradigm C. Field / Naturalistic Studies

• Keep in mind:
  – Choosing the right re-designed work to observe is critical

• Pros and Cons:
  – Positives: “true,” real-world test of system
  – Negatives: qualitative data, requires a fully functional system
Technique 3: Observations

• Identify specific re-designed work you want to observe.

• Observe users as they work through the re-designed work.
Paradigm D. Analytical Evaluation

• What it is:
  – Evaluation using theoretical models

• Typical “metrics”:  
  – Depends on the technique/method used

• How to do it:
  – Techniques/methods prescribe a set of questions to ask of the system
Paradigm D. Analytical Evaluation

• Keep in mind:
  – Need a description of the interface, task, and actions required to perform the task

• Pros and Cons:
  – Positives: No functioning system, no subjects
  – Negatives: Real-world users may have different mental models than experts who developed the techniques/methods
Technique 4: Heuristic Evaluation

• Structured critique of system using heuristics

• Performed by expert evaluators on design specs

• Evaluators assess according to heuristics and note violations and severity
Nielsen’s Usability Heuristics

1. Visibility of system status
2. Match between system and real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

http://www.nngroup.com/articles/ten-usability-heuristics/
Technique 5: Cognitive Walkthrough

• Experts analyze design for system learnability

• You must develop a detailed, written list of actions needed to complete a detailed representative task

• Evaluators answer questions for each step of action sequence
Checks in a Cognitive Walkthrough

1. Is the effect of the action the same as the user’s goal at that point?

2. Will users see that the action is available?

3. Once users have found the correct action, will they know it is the one they need?

4. After the action is taken, will users understand the feedback they get?