Administrivia

- Project idea due today.
- These will be used to put together project teams.
The Xerox Star group (1975-81)

- The Star design team developed a new methodology for system design:
  - Task analysis
  - Wide range of users
  - Usage scenarios
  - Decomposition of design:
    * display and control interface
    * User’s conceptual model
  - Many prototyping cycles
  - Desktop metaphor, direct manipulation, WYSIWYG

1/28/2003
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Norman: Human-centered design

People here care a lot about features.

People here want reliability, convenience “no fuss or bother.”
Norman: User Experience is “atomic”

You can’t separately build the aspects of user experience:

i.e.

* Design the features of the product
* Bring in usability experts for usability analysis
* Graphic and industrial design for appearance
* Technical writers to explain the product

Doesn’t work!

Marketing and product development often separated from the design groups. - also Bad
Integrated Design of User Experience

- You need an interdisciplinary team.
- You need to involve social scientists to really explore the users’ needs.
- Those needs should drive the design process - technical commitments come at the end.
Cooper Interaction design emphasizes “egoless design“:

- You design for a customer, not yourself.
- Although good UI designs are visually pleasing, they are not works of art.
- Design is about expressing the customers goals and needs, not the designer’s.
“know thy user”

First step in good design is to identify the user community. It seems obvious but its hard anyway.

Some techniques: photographs & video in context (IDEO). Some types of picture to try for:

- Rituals
- Sacred places
- Relationships
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals, desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Personae

1. Personae are *concrete* representations of the user group as individuals.

2. Things to strive for in a good persona:
   - Attributes (age, gender, occupation)
   - Likes, dislikes
   - Values and desires (or life’s goals)

3. A good persona is generative (of ideas) - a good fictional character.

* Concrete representation is the opposite of abstract representation - it widens the designer’s perspective while abstraction narrows it.
You know it’s a good persona if the design team is passionate about what the persona would or would not like.
Social Sciences

- Social and behavioral scientists are “domain experts” on user behavior:
  - anthropology - ethnography
  - psychology
  - sociology
- Design should start by observing the customer
Values-based design

Interval Research explored “values-based design”, a systematic study of customer lifestyles to discover market segments.
Values-based design

From their analysis of many homes, they concluded that:
* Households are displays.
* Households are sanctuaries.
* Family life is the household priority.
* Women are the household communicators.

Aside: Our scientific understanding of home life is woefully inadequate.
It is seen as an excellent emerging market for technology, which will be deployed regardless.
Know yourself: values

Q: What values drive your own interest in engineering, computer science, cog sci. or...?
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Choose representative tasks and analyze them

Choose real tasks that users articulated during interviews.

Crisp task analysis is important: hierarchical task descriptions make design easier.

Beware of abstraction (loss of information).

Information flow from contextual inquiry to task analysis should be “Fat”:

* keep interview transcripts, photos, narrative descriptions
Borrow from previous designs

- Good design is usually evolutionary rather than revolutionary
  - Borrow from other designs
  - Use design patterns if they exist
  - Use search tools and design databases

- Users rely on common conventions to learn a new interface:
  - File and edit menus
  - Left click/right click etc.
  - Turn a steering wheel CW steers the car to the right - best to follow this!
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Rough out the design

- Put things on paper to negotiate them with other designers.
- Focus on high-level issues (what features are needed and why).
- Keep the task analysis and user profiles in mind when discussing features.
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Think about the design

- This is the phase to do engineering analysis if appropriate.
- For usability, automated systems are not very powerful, and there are few (GOMS, EPIC).
- Heuristic evaluation is a systematic method for human evaluation of an interface.
- Another method is “cognitive walkthroughs” explained later in Lewis and Rieman.
- More elaborate techniques include:
  * scenario development
  * role-playing
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Prototype the design

- Prototypes let you simulate a lot of detail of an interface.
- Informal (paper or digital sketch) interfaces keep designs more fluid - more changes happen.
- They allow presentations to the user.
- The “Wizard of Oz” method has the designer simulate the behavior as well as the appearance of the system.
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Test the prototype

- Scenarios and role-playing are no substitute for user testing.
- Test with users with similar backgrounds to your target users.
- Doing the design will give you a large set of expectations about what users will do with the design.
- Testing will reinforce or contradict your expectations. You learn from that process.
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- **Iterate**
- Build a production version (and ship it!)
- Track use
- Evolve the design
Iterate!

- Testing will expose problems with various severity
- You can then attack those problems in order of severity - and work on features in order of value
- Beware of interactions between design elements - fixing one may break another
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Build It!

- Some prototyping tools (IDEs or UIMS) allow you to move prototype code to production code - most do not, and this method is not recommended.
- When you move from prototype to production code, remember that commitments you make will be hard to undo - check everything first!
- Remember that UI code is typically half of all code for interactive systems. Allow enough time for development.
Build and Release

- Early releases (alpha and beta) allow yet more testing. Make sure you have good mechanisms in place to get developer/early user feedback.

- The time from "fully-working" code to "industrial-strength" code can be 6 months or more.
  * Program defensively, anticipate and deal with errors inside and outside your system.
  * Test at appropriate scale
  * Introduce stress on the system (other apps, lots of users). Stress on testers would be a good idea - but hard to implement!
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Track Use

- Remember the Zoomer!
- Interview real users, log their complaints and praise.
- Talk to maintenance and support staff.
- Put in logging and bug reporting software.
  * Be very careful about privacy.
Toolbelt Design

- There is a trend in design to build suites of interoperable tools that the customer can adapt (something like MS office + VBasic).
- Toolbelt design allows user evolution of the basic features of the design.
- New generations of the system can move user ideas into the core system.

- In other words, users can become your best designers.
Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design
Evolve the Design

- Real user feedback should help you figure out what needs to change.
- It's often a challenge to accept what you find, and act on it: the company's strategy and assumptions may have been wrong (Zoomer again).
- But remember that most truly innovative products in IT were 2\(^{nd}\) or 3\(^{rd}\) attempts:
  * Palm Pilot
  * Apple Mac
  * Windows 3.1
Evolve the Design

- Users as designers:
- e.g. email is used by people for
  - Calendaring/Reminders
  - As a list of contacts
  - As a to-do list
- New versions of a design can take these into account.
Summary

- Human-centered design starts with the user.
- Time spent in the early phases pays the most dividends.