

## User-Centered Interaction Design

INFO 424  
Lecture 9  
October 26, 2006  
Polle Zellweger

## Overview

- ◆ What is Interaction Design?
- ◆ Discuss Interaction Design process
  - ❖ Walk through steps
  - ❖ Relate to work for INFO 424 final project

Fall 2006 #9

INFO 424

2

## What is Interaction Design?

- It is a process:
  - a goal-directed problem solving activity informed by intended use, target domain, materials, cost, and feasibility
  - a creative activity
  - a decision-making activity to balance trade-offs
- It is a representation:
  - a plan for development
  - a set of alternatives and successive elaborations

Fall 2006 #9

INFO 424

3

## Four basic activities

There are four basic activities in Interaction Design:

1. Identifying needs and establishing requirements
2. Developing alternative designs
3. Building interactive versions of the designs
4. Evaluating designs

Fall 2006 #9

INFO 424

4

## Three key characteristics

Three key characteristics permeate these four activities:

1. Focus on users early in the design and evaluation of the artefact
2. Identify, document and agree on specific usability and user experience goals
3. Iteration is inevitable. Designers never get it right the first time

Fall 2006 #9

INFO 424

5

## User-centered design

- ◆ A good way to avoid flawed requirements
- ◆ Real users and their goals, not just technology, should be the driving force behind the development of a product or system
- ◆ Thus, a well-designed system should make the most of human skill and judgment, should be directly relevant to the work at hand, and should support rather than constrain the user

Fall 2006 #9

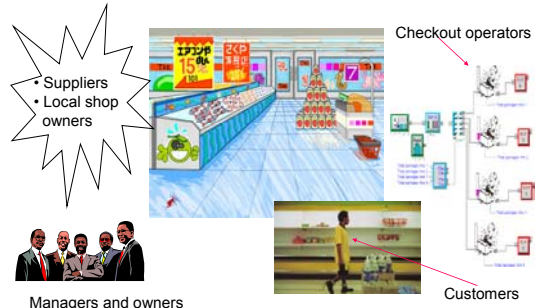
INFO 424

6

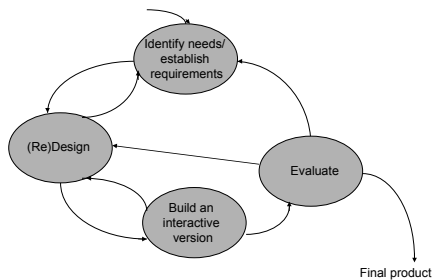
### Who are the users/stakeholders?

- ◆ Not as obvious as you think:
  - ❖ those who interact directly with the product
  - ❖ those who manage direct users
  - ❖ those who receive output from the product
  - ❖ those who make the purchasing decision
  - ❖ those who use competitor's products
- ◆ Three categories of user (Eason, 1987):
  - ❖ **primary**: frequent hands-on \*\* IV project
  - ❖ **secondary**: occasional or via someone else \*\* IV project
  - ❖ **tertiary**: affected by its introduction, or will influence its purchase

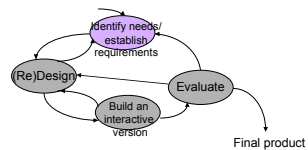
### Who are the stakeholders?



### A simple user-centered interaction design model



### Identify needs / Establish requirements



### Identify needs / establish requirements

- ◆ Understand as much as possible about the users, their work, and the context of that work
- ◆ **requirement**: a statement about an intended product that specifies what it should do or how it should perform
- ◆ Make requirements as specific, unambiguous, and clear as possible

### Importance of requirements

Requirements definition: the stage where failure occurs most commonly

Getting requirements right is crucial

### Types of requirements

- ◆ Functional
  - ❖ what the system should do (i.e., tasks)
- ◆ User
  - ❖ user's abilities, skills, interests
  - ❖ novice/expert, casual/frequent
- ◆ Environmental (context of use)
  - ❖ circumstances in which the system will operate
  - ❖ physical, social, organizational, technical
- ◆ Data
  - ❖ type, volatility, size/amount, persistence, accuracy, values
- ◆ Usability
  - ❖ learnability, efficiency, memorability, errors, satisfaction

Fall 2006 #9 INFO 424 13


### Data gathering

- ◆ These techniques can be used for requirements and/or evaluation
- ◆ Questionnaires
- ◆ Interviews
- ◆ Focus groups or workshops
- ◆ Naturalistic observation
- ◆ Studying documentation

Fall 2006 #9 INFO 424 14

### Different kinds of requirements

- ◆ Users: who are they?
  - ❖ characteristics: ability, background, attitude to computers
  - ❖ system use: novice, expert, casual, frequent
    - novice: step-by-step (prompted), constrained, clear information
    - expert: flexibility, access/power
    - frequent: short cuts
    - casual/infrequent: clear instructions, e.g. menu paths



Fall 2006 #9 INFO 424 15

### Usability Criteria

- ◆ **Learnability:** How easy is it for users to accomplish basic tasks the first time they encounter the design?
- ◆ **Efficiency:** Once users have learned the design, how quickly can they perform tasks?
- ◆ **Memorability:** When users return to the design after a period of not using it, how easily can they re-establish proficiency?
- ◆ **Errors:** How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- ◆ **Satisfaction:** How pleasant is it to use the design?

Fall 2006 #9 INFO 424 16

### Task descriptions

- ◆ Scenarios
  - ❖ an informal narrative story
  - ❖ simple, 'natural', personal, not generalisable
- ◆ Use cases
  - ❖ assume interaction with a system
  - ❖ assume detailed understanding of the interaction
- ◆ Essential use cases
  - ❖ abstract away from the details
  - ❖ does not have the same assumptions as use cases

Fall 2006 #9 INFO 424 17

### Scenarios

- ◆ Describe human activities or tasks in a story that allows exploration and discussion of contexts, needs, and requirements
- ◆ Concrete stories that concentrate on realistic and specific activities

Fall 2006 #9 INFO 424 18

### Scenario for shared calendar

“A professor is trying to schedule a meeting of the faculty search committee. She selects the search committee to enter the names of meeting participants, and she also enters some constraints such as the length of the meeting and roughly when it needs to take place. The system checks the individuals’ calendars and the central departmental calendar for matching times. Because of the size of the committee, the system also checks for an available meeting room.

The system presents the professor with a series of dates on which everyone is free at the same time and a meeting room is available. She selects one of those dates, and the system adds the new meeting into the member’s calendars in a special color as an externally-scheduled meeting. The system also emails each member an alert to request confirmation for the meeting and adds the meeting to the calendar for the meeting room.”

Fall 2006 #9

INFO 424

19

### Use case for shared calendar

1. The user chooses the option to arrange a meeting.
2. The system prompts the user for the names of attendees.
3. The user types in a list of names.
4. The system checks that the list is valid.
5. The system prompts the user for meeting constraints.
6. The user types in meeting constraints.
7. The system searches the calendars for a date that satisfies the constraints.
8. The system displays a list of potential dates.
9. The user chooses one of the dates.
10. The system writes the meeting into the calendar.
11. The system emails all the meeting participants informing them of the appointment.

Fall 2006 #9

INFO 424

20

### Example essential use case for shared calendar

USER INTENTION	SYSTEM RESPONSIBILITY
arrangeMeeting arrange a meeting	request meeting attendees & constraints
identify meeting attendees & constraints	search calendars for suitable dates suggest potential dates
choose preferred date	book meeting

Fall 2006 #9

INFO 424

21

### IV Project: Identify needs/establish requirements

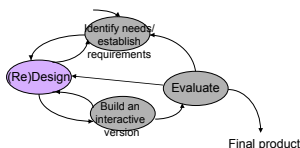
- ◆ User
  - ❖ identify and describe the target users
- ◆ Functional
  - ❖ identify questions that target users may ask about the data
  - ❖ describe the specific tasks that your system will support
- ◆ Context of use
  - ❖ find, collect or create a dataset to support these tasks
  - ❖ describe the data and its schema in detail (types, ranges of values)
- ◆ Scenarios
  - ❖ document and provide insight into requirements
  - ❖ may demonstrate some usability requirements

Fall 2006 #9

INFO 424

22

### (Re)Design



### Develop alternative designs

- ◆ Conceptual design
  - ❖ what should product do, behave & look like
- ◆ Physical design
  - ❖ details, including colors, sounds, images, menus, icons, etc.

Fall 2006 #9

INFO 424

24

*Where do alternatives come from?*

- ◆ Humans stick to what they know works
- ◆ But considering alternatives is important to 'break out of the box'
- ◆ Designers are trained to consider alternatives, software people generally are not
- ◆ How do you generate alternatives?
  - 'Flair and creativity': research and synthesis
  - Seek inspiration: look at similar products or look at very different products

Fall 2006 #9

INFO 424

25

Using scenarios in conceptual design

- ◆ Express proposed or imagined situations
- ◆ Used throughout design in various ways
  - ❖ scripts for user evaluation of prototypes
  - ❖ concrete examples of tasks
  - ❖ as a means of co-operation across professional boundaries
- ◆ Plus and minus scenarios to explore extreme cases

Fall 2006 #9

INFO 424

26

*Using prototypes in conceptual design*

- Allow evaluation of emerging ideas
- Low-fidelity prototypes used early on, high-fidelity prototypes used later

Fall 2006 #9

INFO 424

27

IV Project: (Re)Design

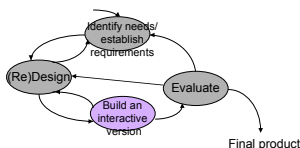
- ◆ Consider alternative visualization designs
  - ❖ view examples from Spence, Viz4All, other class Resources
  - ❖ try different encoding schemes for data attributes
    - representation: position, color, texture, connectivity,... (Spence Chapter 3)
    - presentation: space and time constraints (Spence Chapter 4)
    - interaction modes (Spence Chapter 5)
- ◆ Proposal version
  - ❖ create a paper prototype; iterate based on user feedback
- ◆ Final version
  - ❖ use feedback to design a final prototype

Fall 2006 #9

INFO 424

28

Build an interactive version



Fall 2006 #9

INFO 424

30

### *What is a prototype?*

In interaction design it can be (among other things):  
 a series of screen sketches  
 a storyboard, i.e. a cartoon-like series of scenes  
 a Powerpoint slide show  
 a video simulating the use of a system  
 a lump of wood (e.g. PalmPilot)  
 a cardboard mock-up  
 a piece of software with limited functionality  
 written in the target language or in another language

Fall 2006 #9

INFO 424

31

### *Why prototype?*

- Evaluation and feedback are central to interaction design
- Stakeholders can see, hold, interact with a prototype more easily than a document or a drawing
- Team members can communicate effectively
- You can test out ideas for yourself
- It encourages reflection: very important aspect of design
- Prototypes answer questions, and support designers in choosing between alternatives

Fall 2006 #9

INFO 424

32

### *What to prototype?*

- Technical issues
- Work flow, task design
- Screen layouts and information display
- Difficult, controversial, critical areas

Fall 2006 #9

INFO 424

33

### *Low-fidelity Prototyping*

- Uses a medium which is unlike the final medium, e.g. paper, cardboard
- Is quick, cheap and easily changed
- Examples:
  - sketches of screens, task sequences, etc
  - 'Post-it' notes
  - storyboards
  - 'Wizard-of-Oz'

Fall 2006 #9

INFO 424

34

### *Storyboards*

- Often used with scenarios, bringing more detail, and a chance to role play
- It is a series of sketches showing how a user might progress through a task using the device
- Used early in design

Fall 2006 #9

INFO 424

35

### *Sketching*

- Sketching is important to low-fidelity prototyping
- Don't be inhibited about drawing ability. Practice simple symbols



Fall 2006 #9

INFO 424

36

### Paper prototyping

- ◆ Separate page for each screen
- ◆ Post-its, transparent overlays can help with interactive areas
- ◆ How-to video in lab tomorrow

Fall 2006 #9 INFO 424 37

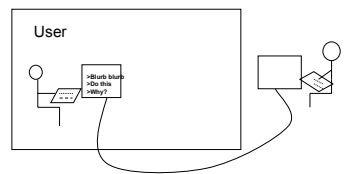
### Advantages of paper prototypes

- ◆ The simplest, easiest method
- ◆ All team members can participate
- ◆ Design and testing tool
  - ❖ start usability testing very early in a project
- ◆ Communicates design to others
- ◆ Facilitates regular testing

Fall 2006 #9 INFO 424 38

### *'Wizard-of-Oz' prototyping*

- The user thinks they are interacting with a computer, but a developer is responding to output rather than the system.
- Usually done early in design to understand users' expectations
- What is 'wrong' with this approach?



Fall 2006 #9 INFO 424 39

### *High-fidelity prototyping*

- Uses materials that you would expect to be in the final product.
- Prototype looks more like the final system than a low-fidelity version.
- For a high-fidelity software prototype common environments include Macromedia Director, Visual Basic, and Smalltalk.
- Danger that users think they have a full system.....see compromises

Fall 2006 #9 INFO 424 40

### *Compromises in prototyping*

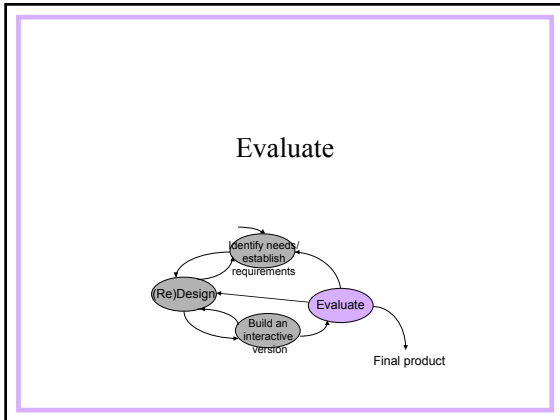
- All prototypes involve compromises
- For software-based prototyping maybe there is a slow response? sketchy icons? limited functionality?
- Two common types of compromise
  - 'horizontal': provide a wide range of functions, but with little detail
  - 'vertical': provide a lot of detail for only a few functions
- Compromises in prototypes mustn't be ignored. Product needs engineering:
  - usability (of course), reliability, robustness, maintainability, integrity, portability, efficiency, ...

Fall 2006 #9 INFO 424 41

### IV Project: Build an interactive version

- ◆ Proposal version
  - ❖ build and test paper prototype(s)
- ◆ Final version -- build one of the following:
  - ❖ a more polished paper prototype using, for example, Illustrator or Photoshop
  - ❖ a Flash-based interactive prototype, possibly connected via HTTP to a dataset
  - ❖ a prototype based on an existing infovis tool like Tableau or Treemap (with extensions of your own design to support your particular domain and tasks)
  - ❖ a custom application using an infovis toolkit like prefuse

Fall 2006 #9 INFO 424 42



### How do you choose among alternatives?

- ◆ Evaluation with users or with peers, e.g. prototypes
- ◆ Technical feasibility: some not possible
- ◆ Quality thresholds: Usability goals lead to usability criteria set early on and check regularly
  - safety: how safe?
  - utility: which functions are superfluous?
  - effectiveness: appropriate support? task coverage, information available
  - efficiency: performance measurements

Fall 2006 #9 INFO 424 44

### Usability Criteria

- ◆ **Learnability:** How easy is it for users to accomplish basic tasks the first time they encounter the design?
- ◆ **Efficiency:** Once users have learned the design, how quickly can they perform tasks?
- ◆ **Memorability:** When users return to the design after a period of not using it, how easily can they re-establish proficiency?
- ◆ **Errors:** How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- ◆ **Satisfaction:** How pleasant is it to use the design?

Fall 2006 #9 INFO 424 45

### Evaluation

- ◆ **Usability**
- +
- ◆ **Utility:** Does it do what the users need?

Fall 2006 #9 INFO 424 46

### Two main types of evaluation

- ◆ **Formative evaluation** is done at different stages of development to check that the product meets users' needs.
- ◆ **Summative evaluation** assesses the quality of a finished product.

Our focus is on formative evaluation

Fall 2006 #9 INFO 424 47

### What to evaluate

Iterative design & evaluation is a continuous process that examines:

- ◆ Early ideas for conceptual model
- ◆ Early prototypes of the new system
- ◆ Later, more complete prototypes

Designers need to check that they understand users' requirements.

Fall 2006 #9 INFO 424 48

### Bruce Tognazzini tells you why you need to evaluate

“Iterative design, with its repeating cycle of design and testing, is the only validated methodology in existence that will consistently produce successful results. If you don’t have user-testing as an integral part of your design process you are going to throw buckets of money down the drain.”

See AskTog.com for topical discussion about design and evaluation.

### When to evaluate

- ◆ Throughout design
- ◆ From the first descriptions, sketches etc. of users needs through to the final product
- ◆ Design proceeds through iterative cycles of ‘design-test-redesign’
- ◆ Evaluation is a key ingredient for a successful design.

### Key points

- ◆ Evaluation & design are closely integrated in user-centered design
- ◆ Some of the same techniques are used in evaluation & requirements but they are used differently (e.g., interviews & questionnaires)
- ◆ Triangulation involves using a combination of techniques to gain different perspectives
- ◆ Dealing with constraints is an important skill for evaluators to develop

### Testing & modeling users



### Experiments, user testing & usability testing

- ◆ Experiments test hypotheses to discover new knowledge by investigating the relationship between two or more things – i.e., variables.
- ◆ User testing is applied experimentation in which developers check that the system being developed is usable by the intended user population for their tasks.
- ◆ Usability testing uses a combination of techniques, including user testing & user satisfaction questionnaires.

### User testing is not research

#### User testing

- ◆ Aim: improve products
- ◆ Few participants
- ◆ Results inform design
- ◆ Not perfectly replicable
- ◆ Controlled conditions
- ◆ Procedure planned
- ◆ Results reported to developers

#### Research experiments

- ◆ Aim: discover knowledge
- ◆ Many participants
- ◆ Results validated statistically
- ◆ Replicable
- ◆ Strongly controlled conditions
- ◆ Experimental design
- ◆ Scientific paper reports results to community

### User testing

- ◆ Goals & questions focus on how well users perform tasks with the product
- ◆ Comparison of products or prototypes common
- ◆ Major part of usability testing
- ◆ Focus is on time to complete task & number & type of errors
- ◆ Often informed by video & interaction logging
- ◆ User satisfaction questionnaires provide data about users' opinions

Fall 2006 #9

INFO 424

55

### Testing conditions

- ◆ Usability lab or other controlled space
- ◆ Major emphasis on
  - selecting representative users
  - developing representative tasks
- ◆ 5-10 users typically selected
- ◆ Tasks usually last no more than 30 minutes
- ◆ The test conditions should be the same for every participant
- ◆ Informed consent form explains ethical issues

Fall 2006 #9

INFO 424

56

### Sample user testing data (Wilson & Wixon, '97)

- Time to complete a task
- Time to complete a task after a specified time away from the product
- Number and type of errors per task
- Number of errors per unit of time
- Number of navigations to online help or manuals
- Number of users making a particular error
- Number of users completing task successfully

Fall 2006 #9

INFO 424

57

### Usability engineering orientation

- Current level of performance
- Minimum acceptable level of performance
- Target level of performance

Fall 2006 #9

INFO 424

58

### How many participants is enough for user testing?

- ◆ The number is largely a practical issue
- ◆ Depends on:
  - schedule for testing
  - availability of participants
  - cost of running tests
- ◆ Typical 5-10 participants
- ◆ Some experts argue that testing should continue until no new insights are gained

Fall 2006 #9

INFO 424

59

### Argument for testing with only 5 users



- ◆ Testing 0 users provides no insight!
- ◆ Testing 5 users finds ~80% of usability problems
- ◆ Use additional budget to iterate & run more tests [Nielsen 2000]

Fall 2006 #9

INFO 424

60

### Comparing Products: Sample within-subjects evaluation plan

User A	User B
Prototype 1	Prototype 2
Task set 1	Task set 2
Complete questionnaire for Task 1	Complete questionnaire for Task 2
Prototype 2	Prototype 1
Task set 2	Task set 1
Complete questionnaire for Task 2	Complete questionnaire for Task 1

### Human Subject Ethics

- ◆ Guidelines
  - ❖ Acknowledge that that system is being tested, not the participant (remind repeatedly)
  - ❖ Tell the participant that she is free to leave at any time
  - ❖ Reveal who is watching & what is being recorded
  - ❖ Do not report results such that a participant is identified
  - ❖ Avoid telling the participant that he is making mistakes or doing things wrong
  - ❖ Acknowledge participants efforts but in a neutral fashion
- ◆ Bottom line: Treat people with great respect
- ◆ **Please have your participants fill in the consent form (include these in your submission)**

### Being a good moderator

- ◆ Spend almost all your time listening, observing carefully, and planning what to say (or not say) next
- ◆ ‘Encourage’ participants in a neutral fashion
- ◆ When people become quiet say
  - ❖ “Can you keep talking”

### IV Project: Evaluate

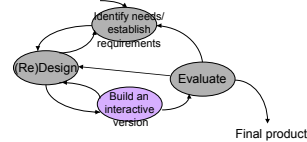
- ◆ Proposal version
  - ❖ evaluate paper prototype(s) with 2-3 target users
  - ❖ test usability and utility for selected tasks
  - ❖ improve paper prototype between users as needed
- ◆ Final version
  - ❖ evaluate final prototype with 3-5 participants from outside the class
  - ❖ test usability and utility for selected tasks
  - ❖ ideally, compare against a simple Tableau visualization of your data

### Usability Evaluation

- ◆ A very quick detour BUT important for your projects
- ◆ There’s much more to learn about usability
- ◆ Remember:
  1. Plan as carefully as you can
  2. Do a task analysis
  3. Create a usability test plan
  4. Run a pilot study
  5. Don’t worry if it is not perfect – learn from each evaluation and try again

### Summary: Interaction Design

#### ◆ Interaction design process



- ◆ Related to work for INFO 424 final project

### Sources Used

- ◆ Preece, J. et al. *Interaction Design: beyond human-computer interaction*, 2002
- ◆ Preece, J. et al. <http://www.ID-Book.com>
  
- ◆ Jakob Nielsen Alertbox website
- ◆ User Interface Engineering website
- ◆ Dave Hendry Info 424 Autumn 2005