Introduction to the Course

Topics

- Motivation
- Objectives of course
- Product and process
- Interaction design vs. software design

The Need for Good User Interfaces

- What is age of youngest effective user of a computer?
- Costs of hardware & software vs. “personware”
- To users, the interface is the system
- Communication vs. computation
How Can You Know if you Have Good Usability?

- Cannot measure usability directly; must measure indicators
  - Speed of user task performance
  - User error rate
  - Subjective user satisfaction
  - Ease of learning
  - Retention over time
  - Usability “in the large”: Ease of use, plus usefulness
  - Usability engineering

Objectives of this course

Course is designed to help you develop more usable interaction designs for graphical user interfaces (GUIs) and Web applications by:
- Understanding and applying interaction design guidelines
- Using an iterative, evaluation-centered usability engineering life cycle

Objectives of this course

- Participating in systems analysis, including user, needs, task, and functional analyses
- Doing conceptual and detailed design
- Establishing usability specifications
- Building rapid prototypes
- Performing formative usability evaluation
- Iteratively refining the interaction design
- Knowing how to get started with these new ideas
People who develop UIs don’t intentionally make them lousy!

Evolution of a good GUI or Web design requires:
- Product – application or web site: content, human factors of an interaction design
  - “what” – general GUI guidelines are largely applicable to web
- Process – usability engineering: techniques and tools for developing an interaction design
  - “how” – ENSURES usability, same process for GUI and Web
- Significant cause of poor usability in product is the lack of understanding of proper development process

Usability is Not User-Friendliness
- We want good usability, user-centered design, not “user-friendly”
Interaction Design is not Software Design

- Developing a GUI or Web user interface involves:
  - Interaction component: how a user interface works, its "look and feel" and behavior in response to what a user hears, sees, and does
  - Interface software component: code that instantiates the interaction component

Development of the user interface

Development of user interface software component

UI software requirements

Problems, constraints

Premise: Describing interaction from user’s view should result in more usable design than describing it from software or programmer view

Inherent conflict of interest!

“One head, two hats” – emphasizes different roles

Popular Misconceptions About Usability Engineering

- Usability engineer is a building inspector (or UI police)
  - Fear and dread, but no respect
- Usability engineer is a “priest in a parachute”
  - Drop into project, bless it, and leave quickly
- The “Peanut Butter Theory” of usability
  - It can be spread on, after the design is done