Introduction to Usability Engineering

CS 352
Summer 2011
Instructor: Chris Bogart
Logistics: CS352

Syllabus: http://classes.engr.oregonstate.edu/eecs/summer2011/cs352/
What is Usability Engineering?

Usability Engineering is the process of **Methodically** designing systems which are

– Useful
– Usable

Which includes

– Determining **what is useful**
– Determining **what is usable**
– **Evaluating** these two factors **empirically**
Usability by Other Names

- Usability Engineering
- Human-computer interaction
- Ergonomics
- Interaction Design
- User-interface design

Many Fields Involved

- Informatics/information systems/library science
- Psychology/cognitive science
- Ergonomics
- Industrial engineering/design
- Art
- Social sciences
Why bother with usability engineering?

- Most software is supposed to help people be productive.
- Build better software.
- Help people like it enough to buy it (keep your job!).
Two kinds of usability flaws

- Gulf of execution
  - "How do I [...]?"
- Gulf of evaluation
  - "What happened then?"
Isn’t this stuff obvious?
Goals of Usability Engineering

(See ch. 1):

- effectiveness at task
- utility
- safety
  - eg, privacy, losing work, mistakes that endanger
- learnability
- memorability
- efficiency
  - differences among last 3
Design Principles:

• Visibility of user’s options/actions.
• Feedback
• Constraints
  – (making certain errors impossible)
    e.g., menus vs. typing to prevent syntax errors.
• (Internal) consistency
• Affordance
  – makes clear what I can do with
HW #1: Design Principles

- **HW #1**
- Pairs for this assignment.
CS 352, Summer 2011
HW #1: Design principles
Due Thursday 6/23/2011
2-person group assignment

Your goal is to apply the design principles of visibility, feedback, constraints, consistency, and affordance from Section 1.6.3 to an existing piece of software.

- Select any software that already existed by Tuesday, 6/21/2011. Preferably something you have helped to write (e.g., for a class or work), but commercial or open source software is OK, too. It can have a very simple (e.g., just textual) or very complex user interface. Include one or more screenshots or sketches to refer to.

- For each design principle in Section 1.6.3, provide a concrete example of how the software's UI supports that principle. Do this by referring to specific things on that picture. Be sure to justify why you are saying that the example supports that principle (3-5 sentences per principle). If nothing about your UI at all supports a design principle, you can say so, but you might have to write a lot more to justify that.

- For each design principle, provide a concrete example of how the software violates that principle (with screenshot, justification, etc., as above). If nothing about your UI at all violates a design principle, you can say so, but you might have to write a lot more to justify that.

Turn in hard-copy at the beginning of class on the due-date.
How to do interaction design/usability engineering

• Process activities (See ch. 1)
  – Identifying needs/requirements
    • of the user experience.
  – Developing many alternative design ideas
    • that meet the requirements.
  – Building interactive versions of the designs
    • to communicate/assess.
  – Evaluating
    • throughout the process.
When to do interaction design

• At beginning of software project:
  – to help establish needs/requirements correctly in the first place

• During design/implementation:
  – to continuously evaluate/monitor

• During testing.
  – to evaluate.
Don’t forget to

• Read 1.4-1.6 and skim the rest of Ch1.
• Start on HW1 (due on Thursday 6/23)
Historical Perspective

How the history of computing is tied to the history of computer usability
Eniac (1943) - Gen 0

A general view of the ENIAC, the world's first all electronic numerical integrator and computer.

From IBM Archives.
Debugging

Grace Hopper 1945

(Photograph courtesy of the United States Naval Historical Center)
Batch Processing – Gen 1

- Computer performed one task at a time
- No “interaction” once computation started
- Switches, wires, punch cards and tapes for I/O
- Very limited, highly trained group of operators
Command Line (Mid 1960s) – Gen 2

- Computers hit “big business”
  - More varied tasks; text processing, editing, email etc
  - Need for interactivity
  - Used by secretaries, salesmen, accountants, CS students etc
  - Reduced training

Need for HCI
The Glass Teletype: late 60s

- 24 x 80 characters
- Up to 19,200 bps (Wow - was big stuff!)

Source: http://www.columbia.edu/acis/history/vt100.html
Generation 4 – A computer in every home

Pushing beyond Computing in Business

Need to do more with less

– Need to rethink usability
  Little or no training for users
  More diverse populations
  More diverse uses
WIMP / GUI

Windows, Icons, Menus, Pointers
Graphical User Interface

WIMP interface emulates existing work practices
Direct manipulation
Desktop metaphor

Why was this such an innovation?

What were the innovations making this possible?
Ivan Sutherland

- **SketchPad** - 1963 PhD thesis at MIT
  - Hierarchy - pictures & subpictures
  - Master picture with instances (ie, OOP)
  - Constraints
  - Icons
  - Copying
  - Light pen input device
  - Recursive operations
Douglas Engelbart

• The Problem (early ‘50s)

  “...The world is getting more complex, and problems are getting more urgent. These must be dealt with collectively. However, human abilities to deal collectively with complex / urgent problems are not increasing as fast as these problems.

  If you could do something to improve human capability to deal with these problems, then you'd really contribute something basic.”

  ...Doug Engelbart
The First Mouse (1964)
Xerox Star - 1981

• First commercial PC designed for “business professionals”
  – desktop metaphor, pointing, WYSIWYG, high degree of consistency and simplicity

• First system engineered for usability
  – Paper prototyping and analysis
  – Usability testing and iterative refinement
Xerox Star Desktop
Lessons form Xerox Star?

• Usability matters, usability sells
  – Star flopped, but Mac succeeded
    • Cost $15,000
    • Lacked spreadsheet, standard business software

• Usability can be “engineered”
  – Birth of HCI as a design discipline
Evolution from Xerox Star?
Evolution from Xerox Star?
Evolution from Xerox Star?

- 1981: Mac OS 5.0
- 1985
- 1987: Windows 2.0
Evolution from Xerox Star?

1981

1985

1987

1992

Mac OS 7

Windows 3.0
Evolution from Xerox Star?

1981
1985
1987
1992
1998
Evolution from Xerox Star?
Evolution from Xerox Star?
Where do we go next?

User Productivity

Batch

Command Line

WIMP (Windows)

1940s – 1950s 1960s – 1970s 1980s - Present

Time

?
Examples of new paradigms

• Mobile computing
• Wearable computing
• Tangible computing
• Ubiquitous computing
  – and many more….