

# Usability Principles

based on Chapter 1 of “GUI Bloopers 2.0” by Jeff Johnson

with contributions from

**Daqing Hou**

EE408 Fall 2008

# Johnson's 9 First Principles

- Basic Principle 1: Focus on the users and their tasks, not on the technology.
- Basic Principle 2: Consider function first, presentation later.
- Basic Principle 3: Conform to the users' view of the task.
- Basic Principle 4: Design for the common case.
- Basic Principle 5: Don't complicate the users' task.
- Basic Principle 6: Facilitate learning.
- Basic Principle 7: Deliver information, not just data.
- Basic Principle 8: Design for responsiveness.
- Basic Principle 9: Try it out on users, then fix it.

**1. Focus on *both* users/tasks and technology.**

**2. Presentation follows function.**

**Principles 1, 2, and 9 are about processes. Principles 3-8 are design how-to's.**

# Focus on users/tasks *and technology!!!*

- users and user analysis

1. For whom is this software designed for? Who are the intended users? Who are the intended customers?

- *decide the users*

2. What are the skills and knowledge of the intended users? Are they motivated to learn? How do they learn? Are there different classes of users, with different skills, knowledge, and motivation?

- *build user profiles (or personas)*

3. What are the intended users' preferred ways of working? How will software fit in? How will it change them?

- *collaborate with users to learn about their characteristics*

# Focus on users/tasks

- tasks
  1. What is the software for? What activity is it intended to support? What problems will it help users solve? What value will it provide?
  2. What are the steps of each task?
  3. What is the input to each task? output? where is input from and where does output go?
  4. What tools are used to do each task?
  5. What tasks does the person do that are relevant to the application's target task area? Who does each task?
  6. How are tasks related? What kinds of communication are needed to do the tasks?
  7. What problems do the intended users have now? What do they like and dislike about the way they work now? What kinds of mistakes do they make? How damaging are the mistakes?
  8. Which tasks are common, and which ones are rare?
  9. Which tasks are most important, and which ones are least important?

# Focus on users/tasks

- task analysis
  - *study the organization*  
*strategic goals; perception of market opportunities and niches*  
*mature versus emerging market*  
*employee expertise*  
*past history, assets, processes, and infrastructure*
  - *investigate the intended tasks*  
*via interviews and observations*
  - *collaborate with the users to learn about their tasks*  
*especially important when working for poorly understood tasks.*  
*e.g. developing software for biologists, and scientists and*  
*engineers in general.*

# Presentation Follows Function

- Software system embodies certain concepts and relationships between concepts. Designers should fully define the concepts and relationships before designing how to present the concepts to users.
- build *conceptual model* for target domain
  - *objects and concepts visible to users*
  - concepts from task domain versus new concepts*
  - *object relationships (has-a versus is-a)*
  - *actions on objects*
  - *lexicon*

# Presentation Follows Function

- build user interface based on *conceptual model*.  
Benefits of doing this include
  - *task focus*
  - *consistency*
  - *importance*
  - *lexicon*
  - *scenarios*
  - *kick-start development*
  - *focus team and process*

# Try it out on users, then fix it!

- Test results can surprise even experienced designers.
- Schedule time to fix problems found by tests.

# Hou's Simplified View on 6 Usability Principles (3-8)

- reduction of complexity by hiding details
- maximally utilizing users' prior knowledge and skills
- respecting human cognitive limits and habits
- When multiple design goals conflict, remember that *designing is about trading-off*.

# Human cognitive limits and habits

- People have a very small short term memory.
  - (7+/-2, Miller 1956)
- People can multitask well-understood tasks, but need concentration when problem-solving.
  - focus users on tasks; don't make them think about UI.
- People need to spend time to learn new things; new things/tasks are hard to do, easy to forget, and frustrating to recall.
  - e.g., arbitrary password rules.
  - user names cannot be longer than 16 characters.
- Information forage theory; people scan, not read text words by words.
  - use headings, short catchy labels, bulleted list, graphics and charts.
  - highlight main points.

# Human cognitive limits and habits - *cont.*

- People perceive changes easier if the changes are made obvious and easy to perceive. Remember that
  - the screen belongs to the user
  - preserving display inertia helps users work efficiently
  - responsive UI is important
- People can be conservative and afraid of making mistakes.
  - encourage learning by exploration by making it easier for user to recover from a mistake.
- People can be confused by ambiguous texts or graphics.

# Maximally utilizing users' prior knowledge and skills

- Consistency
  - internal versus external consistency (Unix commands, “copy src dest”, MS office versus iWork)
  - this is the goal of keeping consistency.
- Adopt common UI designs and behavior
  - use double-click rather than a single click to open an item
  - use dialog rather than tabbed pane
  - use common menu layout, accelerator keys, etc.
- Adopt users' and task domain terminology
  - recall “type mismatch” confused some users!
- Support and preserve users' skills or working habits
  - **GUIs that fully support keyboard operations (story)**

# Reduction of complexity by hiding details

- abstraction is about hiding unnecessary details
  - “I will have the usual.”
  - sensible defaults
  - canned solutions, or templates
  - wizards or procedure guides
- progressive disclosure
- generic commands
- support common tasks very well. What are common tasks? (pp 35, Table 1.1)
  - needed by many or few: visible
  - frequently or rarely used: less or more clicks

*Designing is about  
trading-off, in context.*

**Alas, this ends it.**