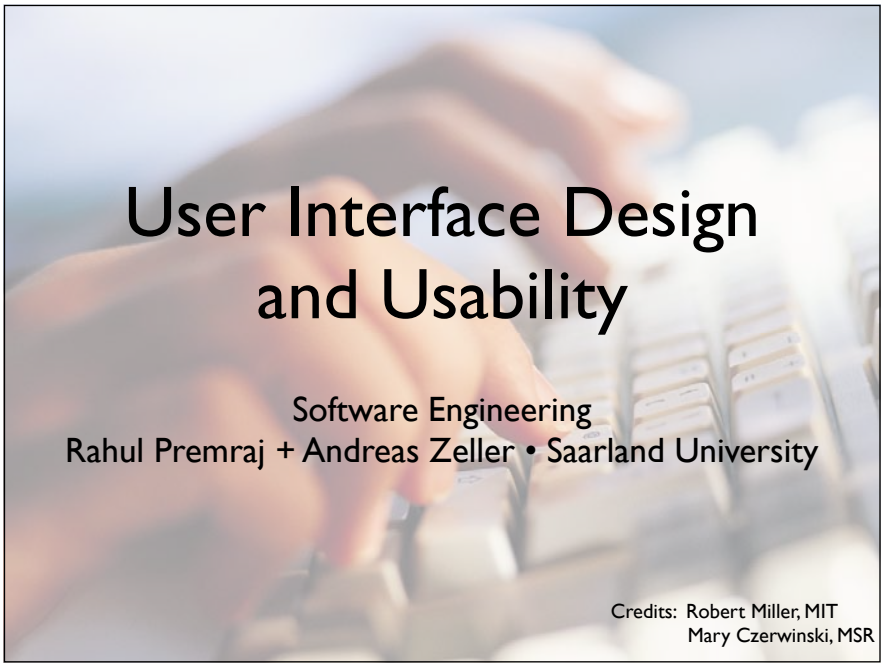


User Interface Design and Usability

Software Engineering
Rahul Premraj + Andreas Zeller • Saarland University

Credits: Robert Miller, MIT
Mary Czerwinski, MSR

What we expect

- A set of *requirements*
  contract style • 5–10 pages
- A set of *use cases*
  Pressman style • 20–40 pages
- A GUI design
  covering most of the use cases
- Architectural models and data models
  covering most of the use cases
- An executable *prototype*
  covering 5–95% of the use cases (negotiable)

All numbers are negotiable depending on project

What we expect

- A set of *requirements*
  contract style • 5–10 pages
- A set of *use cases*
  Pressman style • 20–40 pages

} Dec 3 (draft)
Dec 10 (final)

- A GUI design
  covering most of the use cases

December

- Architectural models and data models
  covering most of the use cases

January

- An executable *prototype*
  covering 5–95% of the use cases (negotiable)

February

What we expect

- A set of *requirements*
  contract style • 5–10 pages
- A set of *use cases*
  Pressman style • 20–40 pages

Dec 3 (draft)
Dec 10 (final)

- A GUI design
  covering most of the use cases — December
- Architectural models and data models
  covering most of the use cases — January
- An executable *prototype*
  covering 5–95% of the use cases (negotiable) — February

User Interface Design and Usability

Software Engineering
Rahul Premraj + Andreas Zeller • Saarland University

Credits: Robert Miller, MIT
Mary Czerwinski, MSR

What is good design?

http://www.ingenfeld.de/

Don't go to the right?

http://www.baddesigns.com/examples.html

7

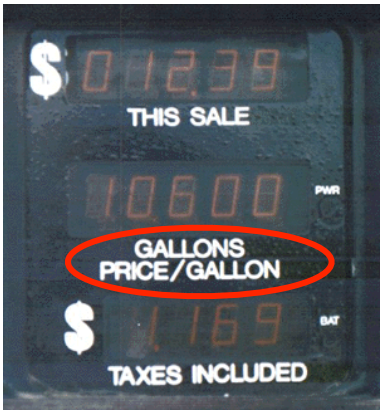
Check the link for examples of bad designs.

What do these symbols mean?

8

Real example from a (expensive) car (as in the picture, no idea which model)---the icons on the buttons placed on the car's dashboard are unclear. I have highlighted the vague ones in red.

How much is the gas?



9

It is not obvious which label belongs to which field.

Interface

Examples of “cool” interfaces.

10

Interface

Some non-apple “cool” interfaces.

11

Interface

definition

interface

n. Computer Science

- The point of interaction or communication between a computer and any other entity, such as a printer or human operator.
- The layout of an application's graphic or textual controls in conjunction with the way the application responds to user activity: an interface whose icons were hard to remember.

12

What is Design?

13

What is Design?

Design is not just what it looks like and feels like.

Design is how it works.

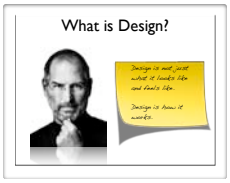
14

What is Design?

Super cool chopstick - the front end doesn't touch the table.

15

What is Design?



Apple isn't perfect. Some examples of problems with Apple products – faulty CDs, discolored handrests, smoking connectors, and exploding batteries.

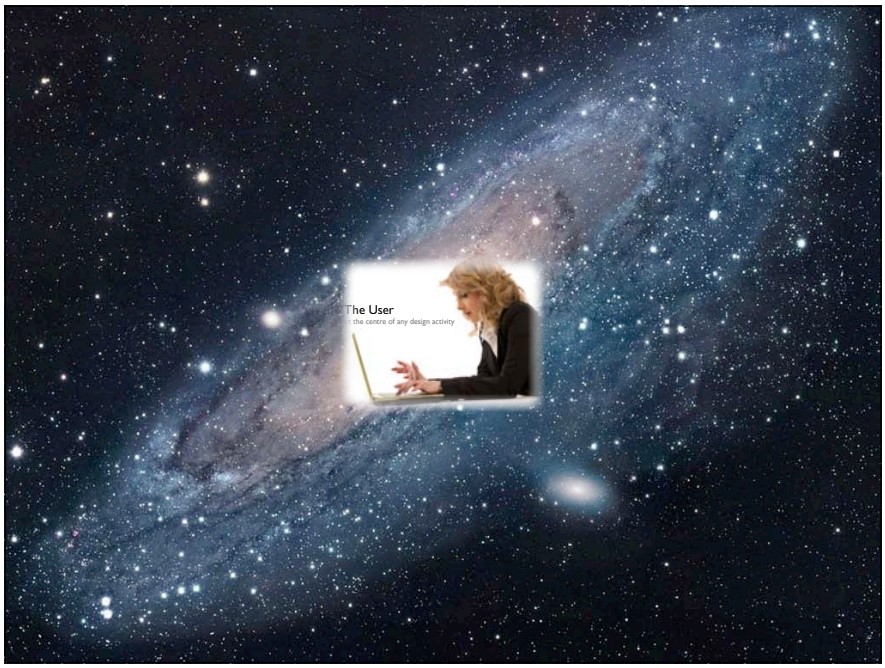
16

What is Design?

2007
Balenciaga Collection

It is easy to overdo design and make the product utterly useless.

17



User is centric to design. Every decision should be made keeping the user in mind.

18

User-Centric Design

- Cost saving!
- Competitive market - user expectations.
- Political demands
- Is Help always helpful?

Credits: Mary Czerwinski

Human Capabilities

- Memory
- Attention
- Visual Perception
- Learning
- Color
- Language + Communication
- Ergonomics

Memory

- Associations are built by repetition.
- Scaffold model (more likely to remember items that have many associations).
- Recognition is easier than recall.
- Working memory has small capacity.
- Long-term memory has large capacity.

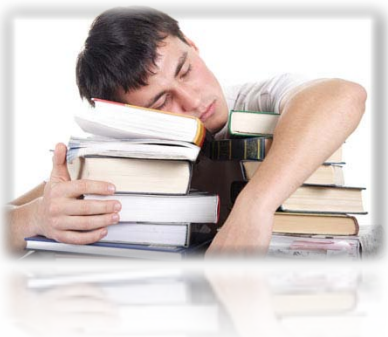
Attention

- Attention is a resource – gets divided amongst tasks.
- Automatic well-learnt processes not need much attention.
- Important to get (for you as a designer).

Visual Perception

- We excel at pattern recognition.
- We automatically try to organize visual displays and look for cues.
- Motion, grouping, contrast, color can make different parts of a display more or less salient.

Learning



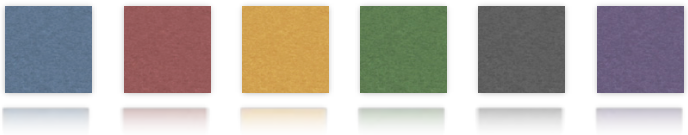
- Learning is improved by organization.
- Consistency and mnemonics improve learning.
- Targeted feedback facilitates learning.
- Learning occurs across people and organizations.

Learning

- Incrementally presented information accelerates learning.
- Some users like to explore systems to learn; others will not.
- Workers focus on accomplishing tasks, not learning software.

25

Color



- Red-green color blindness (protanopia & deuteranopia)
  - 8% of males
  - 0.4% of females
- Blue-yellow color blindness (tritanopia)
  - Far more rare
- Guideline: don't depend solely on color distinctions
  - use redundant signals: brightness, location, shape

Protanopia = absence of red receptors

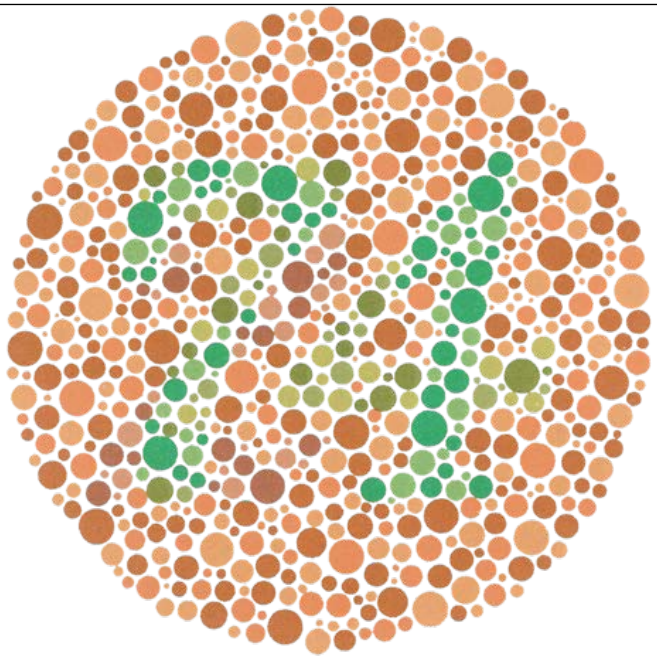
Deuteranopia = absence of green receptors

Tritanopia = absence of blue receptors

26

Traffic lights are readable even for color-blind people (due to location of lights). Also notice the blueish tint in the "green" light.

27



Example of an Ishihara color test plate. [Note 1] The numeral "74" should be clearly visible to viewers with normal color vision. Viewers with dichromacy or anomalous trichromacy may read it as "21", and viewers with achromatopsia may not see numbers. [Wikipedia[

28

Language + Communication

syntax, semantics, pragmatics; conversational interaction, specialized languages

29

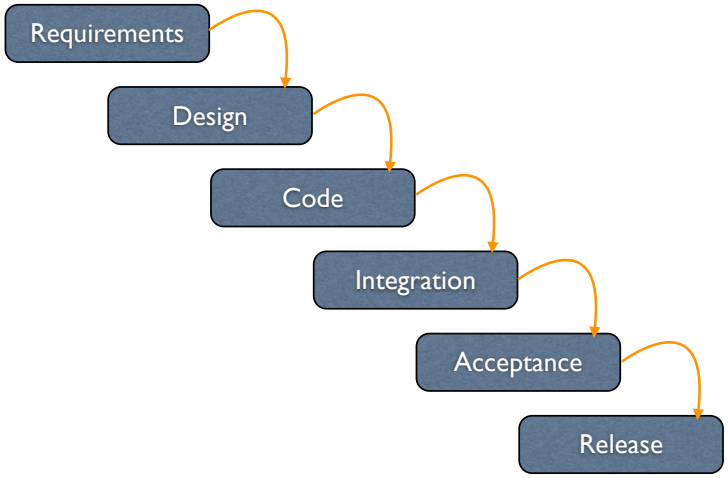
Ergonomics

arrangement of displays and controls; cognitive and sensory limits; effects of display technology; fatigue and health; furniture and lighting; design for stressful and hazardous environments; design for the disabled...

30

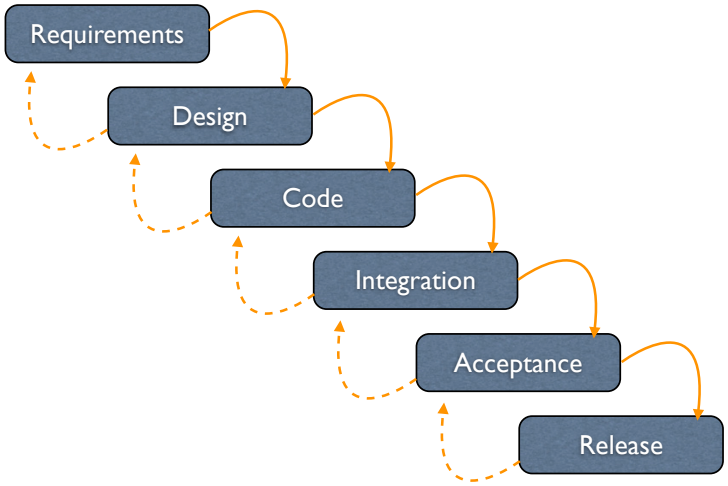
Where does UCD fit into the development process?

Traditional Waterfall Model



Traditional Waterfall Model

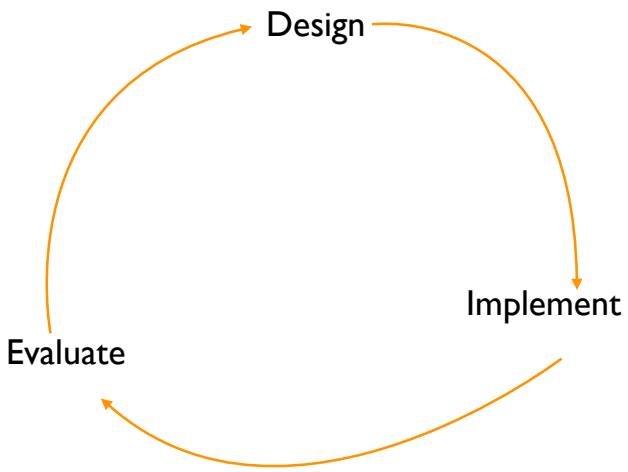
with Feedback



Waterfall Model Poor for UI Design

- UI design is risky.
  - So we are likely to get it wrong.
- Users are not involved in validation until acceptance testing.
  - So we won't find out until the end.
- UI flaws often cause changes in requirements and design.
  - So we have to throw away carefully written and tested code.

Iterative Design

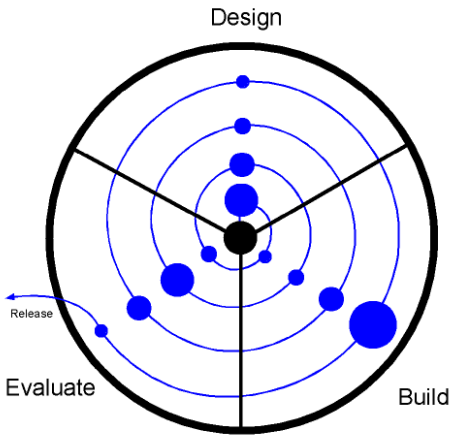


Iterative design is the current best-practice process for developing user interfaces. It's a specialization of the spiral model described by Boehm for general software engineering.

Why NOT Iterative Design?

- Every iteration corresponds to a release
  - Evaluation (complaints) feeds back into next version's design
- Using your paying customers to evaluate your usability
  - They won't like it
  - They won't buy version 2

Spiral Model



each iteration has a cost or fidelity or accuracy

Spiral Model Iterations

- Early iterations use cheap prototypes (paper prototyping).
- Later iterations have richer implementations.
- More iterations generally means better UI.
- Only mature iterations are seen by the world.

Paper Prototyping

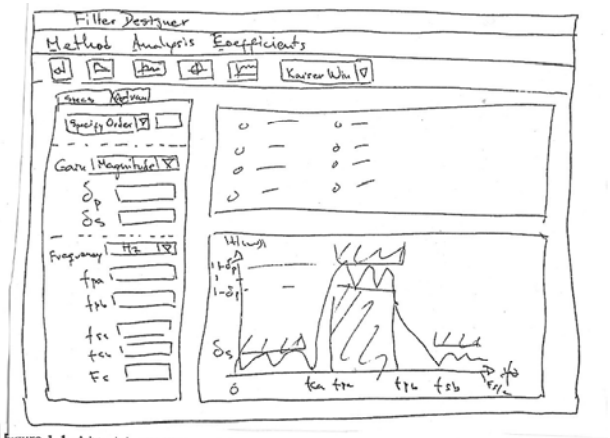
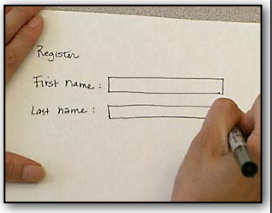
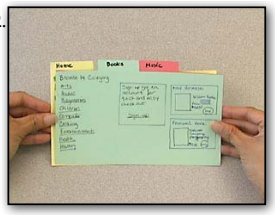


Figure 1.1 A hand-drawn paper prototype of a screen from an application used to design filters for scientific data.

Paper Prototyping

Credits: Nielsen Norman Group

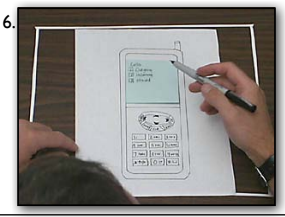
1. 

2. 

3. 

4. 

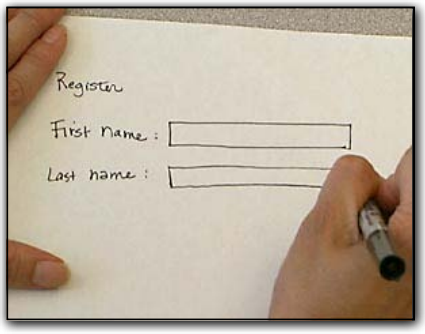
5. 

6. 

7. 

Paper Prototyping

Credits: Nielsen Norman Group



Benefits

- Fast way to mock up an interface - no coding required.
- Finds a variety of problems with the interface.
- Allows an interface to be refined based on user feedback before implementation begins.
- A multidisciplinary team can participate.
- Encourages creativity from the product team and users alike.

Credits: Paper Prototyping

Disadvantage

- Doesn't produce any code.
- Does not find all classes of problems with an interface.
- Can affect the way users interact with the interface.
- Users might think it is unprofessional. (ch. 13)
- Has stronger benefits in some situations than in others.

Credits: Paper Prototyping

For 2nd pt., imagine you need to test how to draw a curved line on Adobe Photoshop. Paper prototyping is not the best way!

UI Analysis & Design

- Iterative Design using a Spiral Model.
- Early focus on users and tasks.
  - User analysis: who the users are.
  - Task analysis: what they need to to?
  - Involve users as evaluators, consultants and sometimes designers.
- Constant Evaluation

Based on Rob Miller: "UI Design and Implementation – User-Centered Design"

Know Your User

- Novice
- Knowledgeable, intermittent user
- Knowledgeable, frequent user
- Age, gender, ethnicity
- Physical abilities
- Domain experience
- Application experience
- Work environment
- Communication patterns

Know Your User

- Techniques
  - Questionnaires
  - Interviews
  - Observations
- Obstacles
  - Artificial barriers between developers and users.
  - Some users are expensive to talk to.

Example: Self-Service Grocery Checkout

- Who are the users?
  - Grocery shoppers
  - Wide age range
  - Possibly no computer experience
  - No training
  - Knowledge of products, but not management
  - Shoppers help each other.
  - Mostly women with small children.
  - Store assistants to help users.

Let's look at an example. Suppose we've been charged with designing a system that will
allow grocery shopper to ring up and pay for their purchases themselves.

Task Analysis

- Identify the individual tasks to be solved.
- Each task is a goal.
- Start with the big goal and then, decompose hierarchically.
  - Overall goal: Shoppers want to purchase groceries.
  - Tasks:
    - Register groceries into the system.
    - Pay

The next step is figuring out what tasks are involved in the problem. A task should be
expressed as a goal: whatneeds to be done, not how.

Essential Parts of Task Analysis

1. *What must be done?*
   - Goal
2. *What must be done before to make it possible?*
   - Preconditions
     - Tasks on which this task depends
     - Information that must be known to the user
3. *What steps are involved in doing the task?*
   - Subtasks (may be decomposed recursively)

Once you've identified a list of tasks, fill in the details on each one. Every task in a task analysis should have at least these parts.

Example: Self-Service Grocery Checkout

- Goal
  - Enter groceries into register
- Preconditions
  - All groceries that you want are in the cart
- Subtasks
  - Enter pre-packaged items
  - Bag loose items, weigh and register them.

Dangers of Task Analysis

- Duplicating a bad existing procedure in software.
- Example: Flipping through a book
- Failing to capture good aspects of existing procedure
- Ask users *why* they do what they do, not just what they do

Suppose we did a task analysis by observing users interacting with paper manuals. We'd see a lot of page flipping: "Find page N" might be an important subtask. We might naively conclude from this that an online manual should provide really good mechanisms for paging & scrolling, and that we should pour development effort into making those mechanisms as fast as possible. But page flipping is an artifact of physical books! It would pay off much more to have fast and effective searching and hyperlinkingin an online manual. That's why it's important to focus on why users do what they do, not just

Improve Task Analysis

- Questions to ask
  - Why do you do this? (goal)
  - How do you do it? (subtasks)
- Look for weaknesses in current situation
  - Goal failures, wasted time, user irritation
- Contextual inquiry
- Participatory design

Observe users doing real work, Challenge assumptions and probe surprises

User Design Principles

Usability Principles

Jacob Nielson

Nielson's 10 Principles Of UI Design

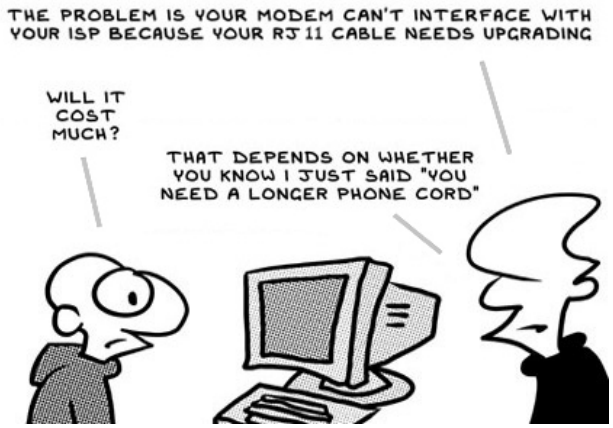
- Shneiderman's 8 Golden Rules
- Tog's 16 Principles

Nielson's Principles

1. Match the real world
2. Consistency and Standards
3. Help and Documentation
4. User Control and Freedom
5. Visibility of System Status
6. Flexibility and Efficiency
7. Error Prevention
8. Recognition, not Recall
9. Error Reporting, Diagnosis, Recovery
10. Aesthetic and Minimalist Design

Match the Real World

1



Match the Real World

- Examples
  - Desktop
  - Trashcan
- Dangers of metaphors
  - Often hard for designers to find
  - Deceptive
  - Constraining
  - Breaking the metaphor
- Use of a metaphor doesn't excuse other bad design decisions

Direct Manipulation

- User interacts with visual representation of data objects
  - Continuous visual representation
  - Physical actions or labeled button presses
  - Rapid, incremental, reversible, immediately visible effects
- Examples
  - Files and folders on a desktop
  - Scrollbar
  - Dragging to resize a rectangle
  - Selecting text
- Visual representation and physical interaction are important

Affordances

of direct manipulation

- Perceived and actual properties of a thing that determine how the thing could be used
  - *Chair* is for sitting
  - *Knob* is for turning
  - *Button* is for pushing
  - *Listbox* is for selection
  - *Scrollbar* is for continuous scrolling or panning
- Perceived vs. actual

Natural Mapping

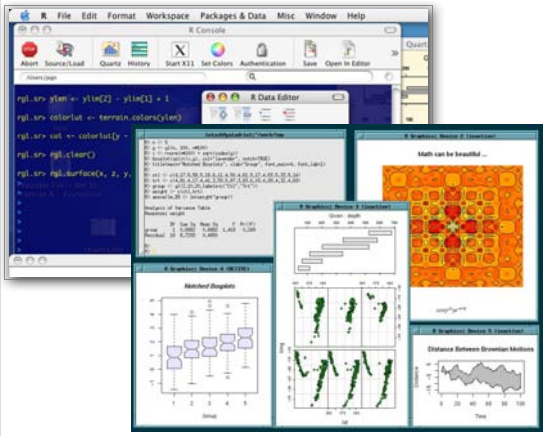
- Physical arrangement of controls should match arrangement of function
- Best mapping is direct, but natural mappings don't have to be direct
  - Light switches
  - Stove burners
  - Turn signals
  - Audio mixer

Feedback / Responsiveness

- Actions should have immediate, visible effects
  - Push buttons
  - Scrollbars
  - Drag & drop
- Kinds of feedback
  - Visual
  - Audio
  - Haptic (conveyed by sense of touch)

Consistency and Standards

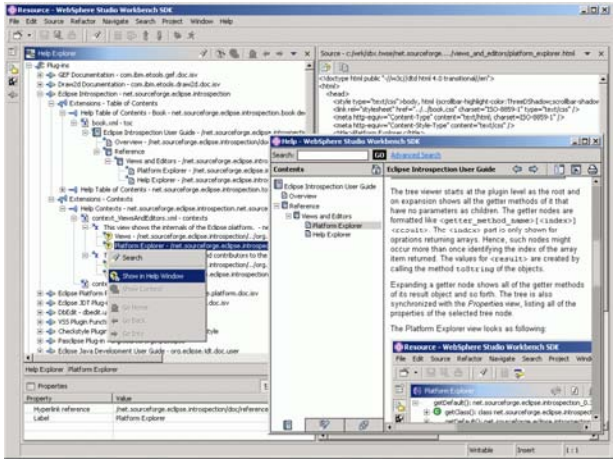
2



Mac, Windows, Gnome, and KDE guidelines

Help and Documentation

3



Help should be (a) searchable, (b) context-sensitive, (c) task senstitive, (d) concrete, (e) short

User Control and Freedom

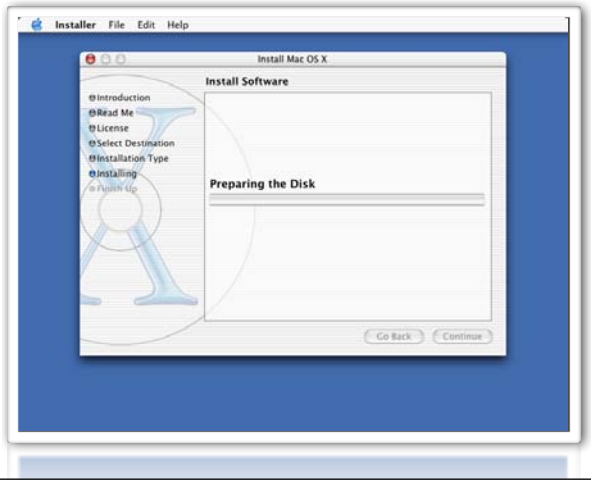
4

Provide Undo
Long operations should be allowed to be paused/suspended
~~all dialogs should have a cancel button~~

64

Visibility of System Status

5

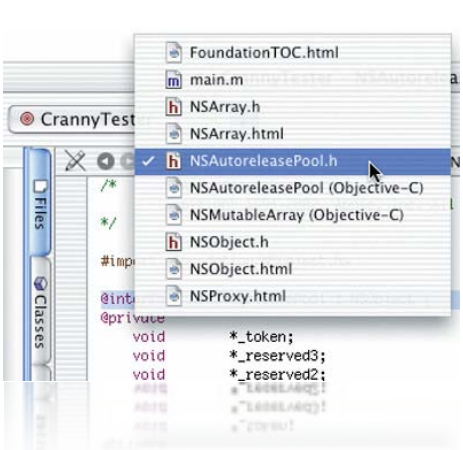


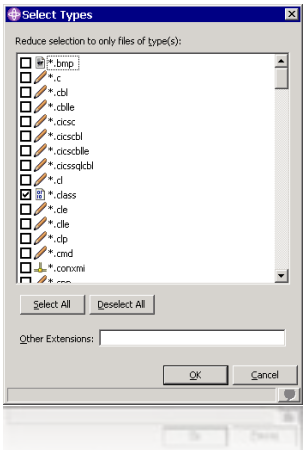
change cursor to indicate action
use highlights to show selected objects
use status bar to show progress

65

Flexibility and Efficiency

6



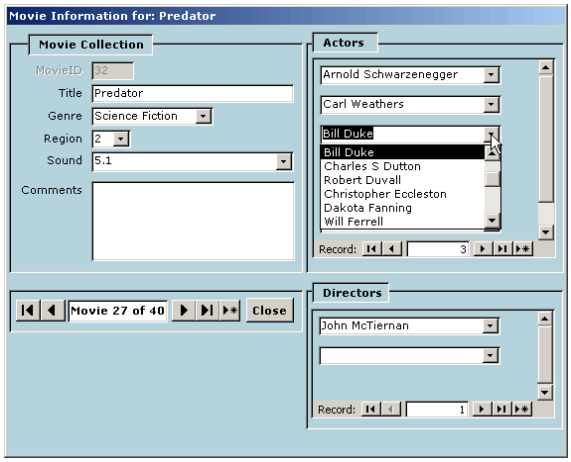


Recently-used history is one very useful kind of shortcut, like this recently-used files menu

66

Error Prevention

7



Murphy's Law - "if something can go wrong, it will"

One way to prevent errors is to allow users to select rather type. Misspellings then become impossible.

Recognition, not Recall

8

use menus, not command languages
use combo boxes, not textboxes
use generic commands
all needed information must be visible

Error Reporting, Diagnosis, Recovery

9

A good error message should (1) be precise; (2) speak the user's language, avoiding technical terms and details unless explicitly requested; (3) give constructive help; and (4) be polite

Aesthetic and Minimalist Design

10

70

71

Principle 1

Focus on the user and their tasks, not the technology.

72

Some other important principles

Principle 2

Consider function first, presentation later.

Principle 3

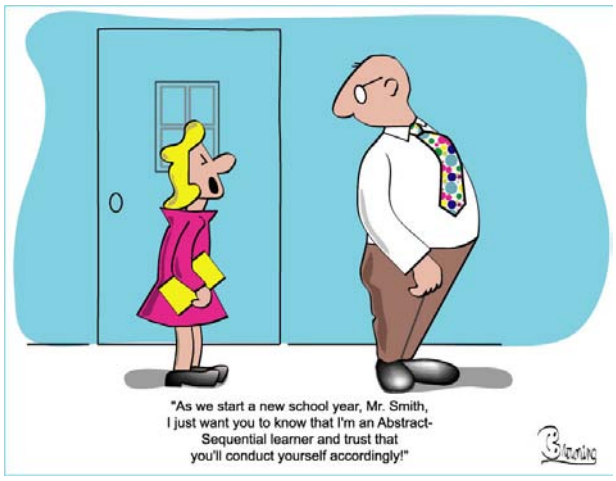
Conform to users' view of the task.

Principle 4

Don't complicate the users' task.

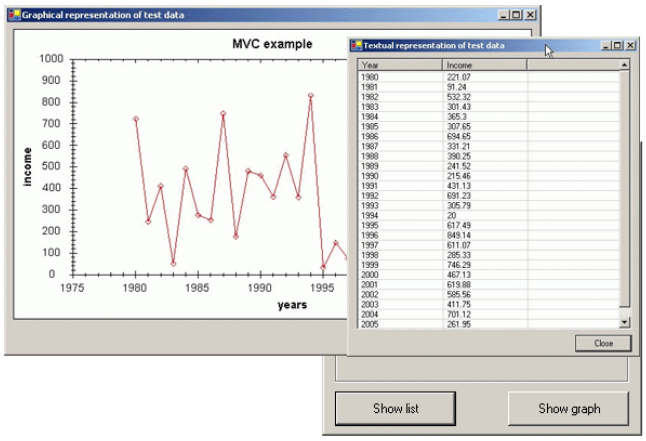
Principle 5

Promote Learning



Principle 6

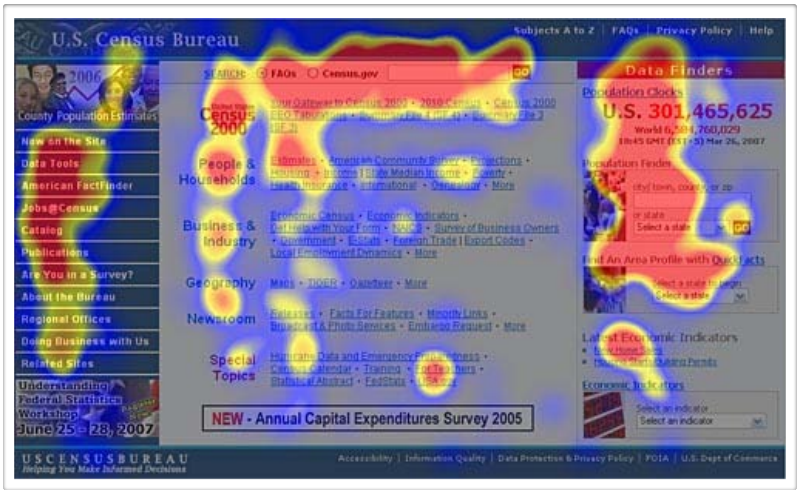
Deliver information, not just data.



Principle 7

Try it out on users, then fix it!

Usability Tests



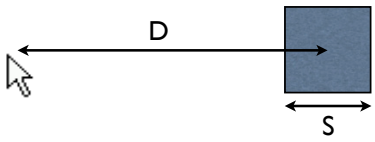
The following heatmap from one of our eyetracking studies shows how users looked at this homepage. Their task was to find the current population of the United States.

79

Fitt's Law

- Time T to move your hand to a target of size S at distance D away is:

  T = a + b log (2D/S)



- Depends only on index of difficulty log(2D/S)

http://en.wikipedia.org/wiki/Fitts'_law
No formulae, just understand the law and its implications.

80

Implications of Fitt's Law

- Targets at screen edge are easy to hit
  - Mac menubar beats Windows menubar
  - Unclickable margins are foolish
- Hierarchical menus are hard to hit
  - Gimp/GTK: instantly closes menu
  - Windows: .5 s timeout destroys causality
  - Mac does it right: triangular zone

81

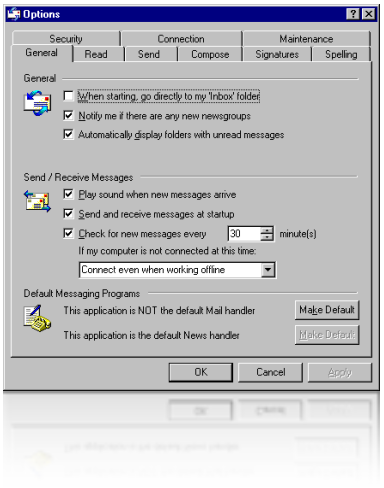
All these aspects

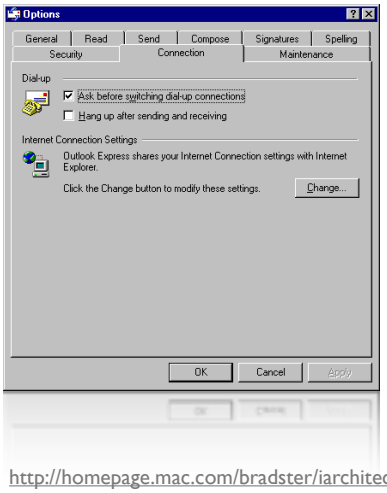
- Functionality
- Performance
- Cost
- Security
- Usability
- Size
- Reliability
- Standards

Design decisions involve tradeoffs among different attributes.

Usability is only one aspect of s/w development

GUI Hall of Shame

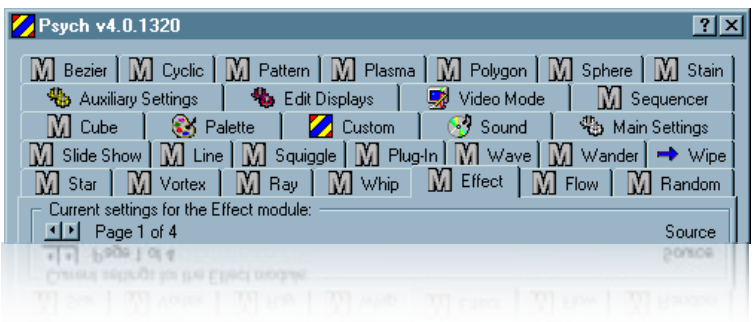




http://homepage.mac.com/bradster/iarchitect/

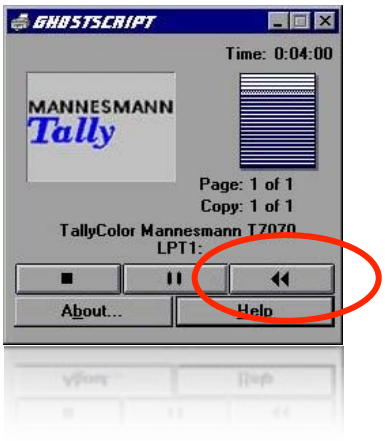
This and the following are poor examples of GUI design. In this slide, there is basically so many options, full of text, non-descriptive icons.

Tabs



Too many tabs???

Rewind



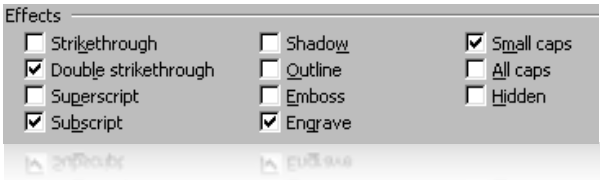
This seems to be a print dialog. Only the designers know what does the "rewind" button mean.

Help

In Microsoft Assistant Killed in Denver, it was reported that Microsoft program managers demonstrated a technique to kill the assistant to a crowd attending a development conference.

MS super letter writing assistant!

Options



On MS-word, there are so many possible effects on the same text. Note that options such as Strikethrough and ~~Doublestrikethough~~ can be opted together for the same text. Similarly subscript and supersubscript.

Puzzle

88

503 Polite People

"503 polite people say hello first"

89

Type "Mismatch"

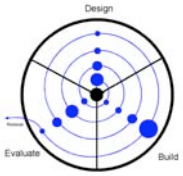
The poor secretary, confronten with this message, simply typed "mismatch" – without success :-(

90

Interface

Summary

Spiral Model



Usability Principles

Jacob Nielson

Nielson's 10 Principles Of UI Design

- Shneiderman's 8 Golden Rules
- Tog's 16 Principles