Institute for Software Research, International

Usability 1

Methods: Decide: What to Design

Usability attributes

Usability through prototyping

Classes of prototyping tools

Today

Back to the users

Usability attributes

Usability through prototyping

Classes of prototyping tools

Sequence and Structure

- Flow model <-> sequence model
- Storyboards <-> User environment design
- Use case model <-> ?
  - object model
  - prototype

Interactions with Users

- Use case modeling
- 7. Don't involve subject matter experts in creating, reviewing, or verifying use cases.
  - (They'll only raise questions!)
- 8. If you involve users at all in use-case definition, just "do it."
  - (Why bother to prepare for any time with the users? It just creates a bunch of paperwork, and they keep changing their minds all the time, anyway.)
- 9. Write your first and only use-case draft in excruciating detail.
  - (Why bother iterating with end users when they don't even know what they want, and they only want you to show them meaty stuff, anyway?)
- 10. Don't validate or verify your use cases.
  - (That will only cause you to make revisions and do more rework, and it will give you change control problems during requirements gathering. So forget about it!)

From Ellen Gottesdiener
### Today
- Back to the users
- Usability attributes
- Usability through prototyping
- Classes of prototyping tools

### Usability
- “The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component”
- Usability and utility -- from user’s perspective
  - Utility: does it do the right things?
  - Usability: does it do these things right?

### Is Usability Important?
- Example: intranet usability, e.g.,
  - Research a company policy
  - Find a form
  - Find information about a department or person
  - Enter an expense report
- Study performed user tests
  - Employees in 14 companies
  - Sixteen common tasks
- Results
  - Low usability intranets: $3,000/employee/year
  - Average usability: $2,000/employee/year
  - High usability: $1,600/employee/year

### What Is Usability?
- Usability Attributes
  - Learnability
  - Efficiency
  - Memorability
  - Errors
  - Satisfaction
Learnability

- Often the most important characteristic
  - How much investment in learning is user willing to make?
  - If too hard to learn, other characteristics don’t matter
- Can readily be measured
  - Time to reach specified level of proficiency, e.g.,
    - perform a bank transaction successfully,
    - Create, save, and print a document
- For business professionals, most highly rated characteristics:
  - Easy-to-understand error messages
  - Possible to do useful work before learning it all
  - Availability of undo
  - Confirming question before execute risky command

Systems Designed for Novice and Expert Users

Novice/Expert Tradeoffs

- Different interaction styles, e.g., menus versus function keys
- More capability for experts, but may make “simple” tasks hard
  - E.g., MS Word versus FrameMaker
- May be possible to “ride” both curves, e.g.,
  - Judicious use of “accelerators”
  - Multiple interaction styles, such as “wizard” and manual configuration

Efficiency of Use

- Expert’s performance at asymptote
- Most users plateau after learning “enough”
  - Often reflects suboptimal investment
- Measurement
  - Define “experienced” users
    - Self-report
    - Some number of hours, weeks, months of use
    - Observe for some number of hours, or until curve flattens
  - Measure time to complete typical set of tasks
Memorability

- Most important for casual, occasional users, e.g.,
  - Utility programs
  - Tax programs
  - Kerberos, KClient, etc.
- Measurement
  - Time for typical tasks with users who have been away from system for specified amount of time
  - This attribute measured less often than others

Few and Noncatastrophic Errors

- Action that does not accomplish user goal
- Vary widely in effects
  - Trivial
  - Annoying
  - Create faulty product or destroy work
- Measurement
  - Number of errors while performing standard task
  - Should be no catastrophic errors

Subjective Satisfaction 1

- How "pleasant" it is to use the system
- Extremely important for discretionary software
- Also important for adoption generally
- Measurement
  - Interview -- rich but nonquantitative data
  - Likert scales, e.g.,
    "It was very easy to learn how to use this system."
    "This system was very pleasant to use."
    "Using this system was a very frustrating experience."
  - Each statement is followed by 5 or 7 point agreement scale:

Subjective Satisfaction 2

- Measurement (ctd.)
  - Semantic differential rating scales, e.g.,
    "Please place a check in a position that best describes X"
    - Pleasing
    - Irritating
    - Simple
    - Complicated
    - Fast to use
    - Slow to use
  - Subjects tend to give slightly inflated ratings, e.g., 3.5, not 3.0, as average on 5-point scale
  - Scales need to be pilot tested
### Today

- Learning from codified experience, grasping uniqueness
- Usability attributes
- Usability through prototyping
- Classes of prototyping tools

### Prototyping

- Desirable properties
  - Cheap to construct
  - Appear rough
  - Malleable enough to allow exploration of alternatives
- Purposes
  - Evaluate alternative approaches, e.g.,
    - GUI
    - Command line
  - Tune interface ideas, e.g.,
    - Clusters of functions
    - Choose interaction mechanism, e.g., button or menu
  - Identify omissions

### Prototype Interviews

- Purpose is to have the user simulate performing specific work with the prototype
- Information gained triggers a mini-iteration of the whole modeling-interpreting-design cycle

### Prototype Fidelity

<table>
<thead>
<tr>
<th>Low Fidelity</th>
<th>High Fidelity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper</td>
<td>Interface Builders (plus components &amp; scripting language)</td>
</tr>
<tr>
<td>Façade Tools</td>
<td></td>
</tr>
</tbody>
</table>
Low Fidelity Prototypes

- Advantages
  - Very cheap to construct
  - Can change during prototype interview
  - Can iterate quickly
- Disadvantages
  - Capturing user behavior is awkward
  - Cannot execute

Medium Fidelity Prototypes

- Create screen appearance, use canned data
  - E.g., Powerpoint animation
- Advantage
  - Appears to execute
  - Can capture “video” of entire interaction
- Disadvantages
  - Nothing is reusable
  - May create too “finished” of an appearance
  - Difficult to edit on the fly
  - May oversell application

High Fidelity Prototypes

- Advantages
  - Easily edited and run
  - May be able to use the code
- Disadvantages
  - Often need to make design commitments too early
    - E.g., select actual components
  - Involves significant effort to make executable
    - Write scripts
  - Difficult to edit with user

Hybrid Prototyping System

- Denim
  - http://guir.berkeley.edu/projects/denim/