

Lecture 11 Evaluation - 1

Xiaojun Bi

Sony Brook University

xiaojun@cs.stonybrook.edu

10/17/2019

What is Evaluation?

- Formative
 - Are we building the right thing?
 - What should be different in the next iteration?
- Summative
 - Does it work?
 - Is it better than existing solutions?
 - Can this teach us something about how people or the world work?
 - Have we learned anything general about system design?
 - Can we make others believe that we have an important insight?

Qualitative vs. Quantitative Evaluation

Qualitative Studies

Qualitative:

e.g., contextual Inquiry: try to understand user's tasks and conceptual model

Qualitative methods help us

- Understand what is going on
- Look for problems
- Roughly evaluate usability of interface

Quantitative Studies

Quantitative

- Use to reliably measure some aspect of interface
- Compare two or more designs on a measurable aspect

Approaches

- Controlled experiments

Examples of measures

- Time to complete a task
- Average number of errors on a task
- Users' ratings of an interface
 - Ease of use, elegance, performance, robustness, speed,...

Quantitative Testing is Costly

Its very expensive – you need to schedule (and normally pay) many subjects.

It takes many hours of the evaluation team's time.

Comparison

Qualitative studies

- Faster, less expensive → esp. useful in early stages of design cycle
- In real-world design quant. study not always necessary

Quantitative studies

- Reliable, repeatable result → scientific method
- Best studies produce generalizable results

Topics

- Discount Usability Engineering
- Heuristic Evaluation
- The Process of Heuristic Evaluation
- Pros and Cons of Heuristic Evaluation

Discount Usability Engineering

Cheap

- No special labs or equipment needed
- The more careful you are, the better it gets

Fast

- On order of 1 day to apply
- Standard usability testing may take a week

Easy to use

- Can be taught in 2-4 hours

Heuristic Evaluation

Usability Heuristics

“Rules of thumb” describing features of usable systems

- Can be used as design principles
- Can be used to evaluate a design

Example: *Minimize users' memory load*

Pros and cons

- Easy and inexpensive
 - Performed by experts
 - No users required
 - Catch many design flaws
- More difficult than it seems
 - Not a simple checklist

Heuristic Evaluation



Developed by Jakob Nielsen (1994)

Can be performed on working UI or sketches

Small set (3-5) of evaluators (experts) examine UI

- Check compliance with usability heuristics
- Different evaluators will find different problems
- Evaluators only communicate afterwards to aggregate findings
- Use violations to redesign/fix problems

Heuristics

H2-1: Visibility of system status

H2-2: Match system and real world

H2-3: User control and freedom

H2-4: Consistency and standards

H2-5: Error prevention

H2-6: Recognition rather than recall

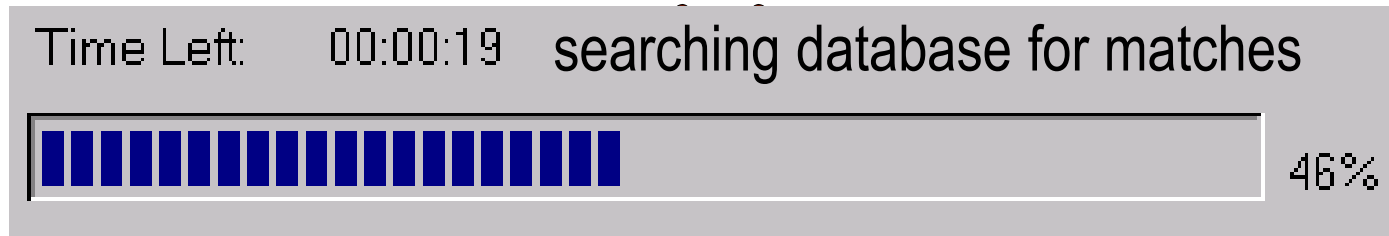
H2-7: Flexibility and efficiency of use

H2-8: Aesthetic and minimalist design

H2-9: Help users recognize, diagnose and recover from errors

H2-10: Help and documentation

Heuristic: Visibility (Feedback)



H2-1: Visibility of system status

- Keep users informed about what is going on
- Example: pay attention to response time
 - 0.1 sec: no special indicators needed
 - 1.0 sec: user tends to lose track of data
 - 10 sec: max. duration if user to stay focused on action
- Short delays: Hourglass
- Long delays: Use percent-done progress bars
 - Overestimate usually better

Heuristic: Visibility (Feedback)

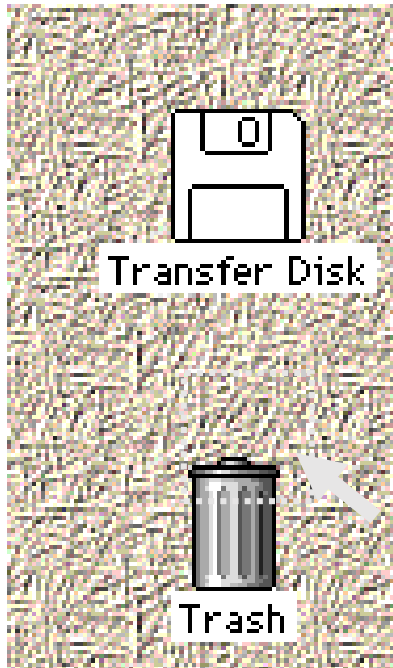
Users should always be aware of what is going on

- So that they can make informed decision
- Provide redundant information



Feedback: Toolbar, cursor, ink

Heuristics: Match System & World



H2-2: Match between system & real world

- Speak the users' language
- Follow real world conventions
- Pay attention to metaphors

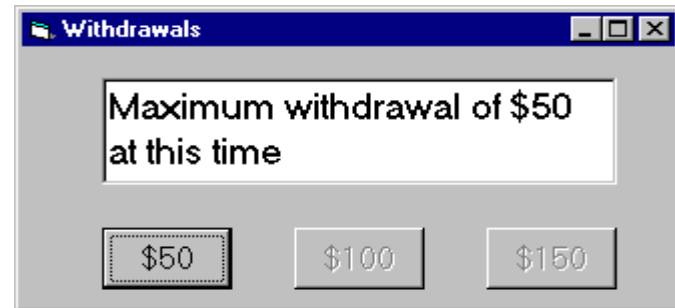
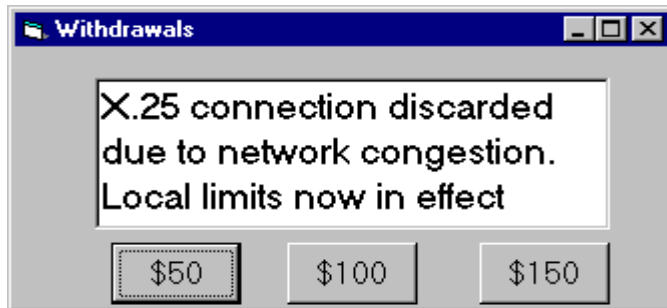
Bad example: Mac desktop

- Dragging disk to trash
- Should delete it, **not** eject it

Heuristics: Match System & World

Speak the users' language (H2-2)

- Withdrawing money at ATM



- Use meaningful mnemonics, icons and abbreviations



Heuristics: Control & Freedom

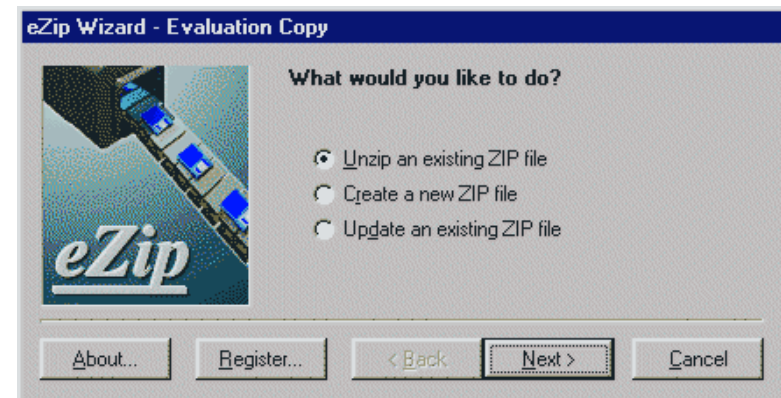


H2-3: User control & freedom

- “Exits” for mistaken choices, undo, redo

Wizards

- Must respond to Q before going to next
- Good
 - For infrequent task (e.g. Internet Config)
 - Beginners (2 versions in WinZip)
- Not good
 - For common tasks



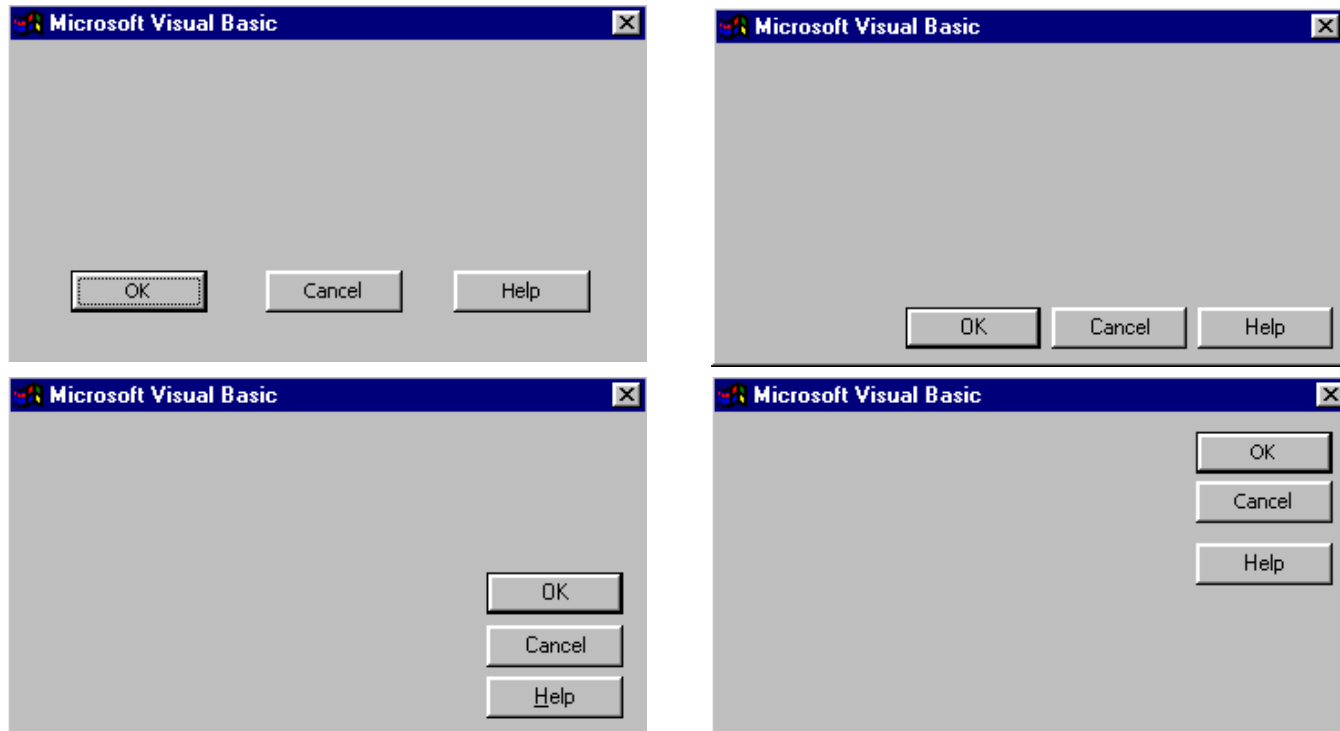
Heuristics: Control & Freedom

- Mark exits: Users don't like to be trapped!



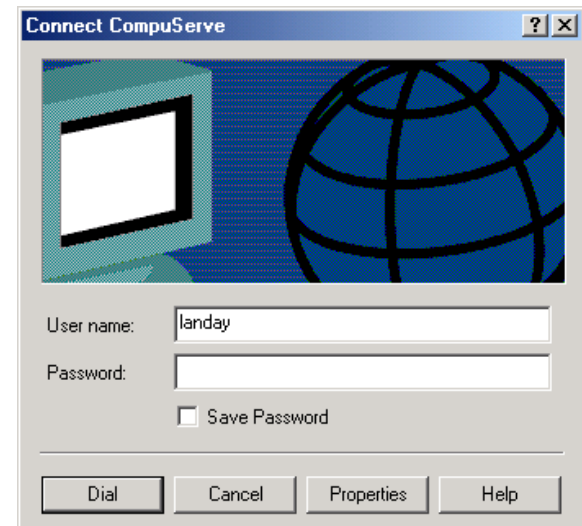
- Strategies
 - Cancel button (or Esc key) for dialog
 - Make the cancel button responsive!
 - Universal undo

Heuristics: Consistency



H2-4: Consistency and standards

Heuristics: Errors and Memory



H2-5: Error prevention

H2-6: Recognition rather than recall

- Make objects, actions, options, & directions visible or easily retrievable

MS Web Publishing Wizard

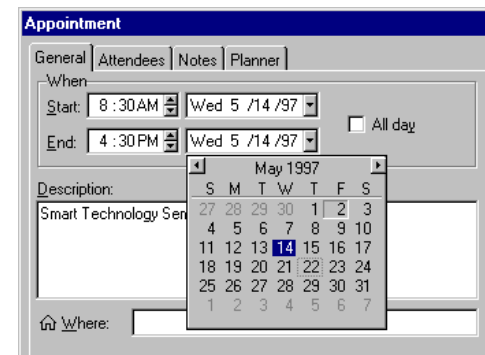
- Before dialing, asks for id & password
- When connecting, asks again for id & pw

Heuristic: Errors and Memory

- Promote recognition over recall
 - Recognition is easier than recall



- Describe expected input clearly
 - Don't allow for incorrect input

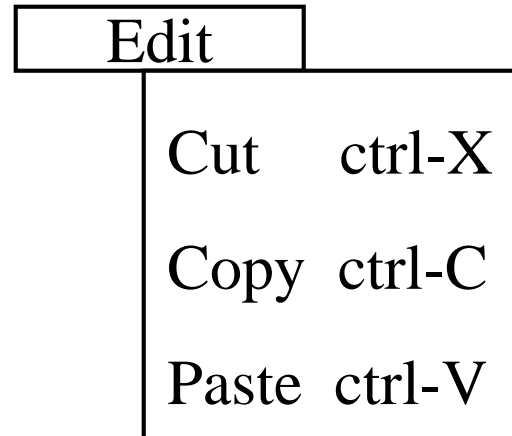


Preventing Errors

Error types

- Mistakes
 - Conscious decision with unforeseen consequences
- Slips
 - Automatic behaviors kicking in
 - Drive to the store, end-up in the office
 - Press enter one time too many...
 - Mode errors
 - Forget the mode the application is in
 - Loss of activation
 - Forget what your goals were

Heuristics: Flexibility



H2-7: Flexibility and efficiency of use

- Accelerators for experts (e.g., gestures, shortcuts)
- Allow users to tailor frequent actions (e.g., macros)

Heuristics: Aesthetics

Form Title -- (appears above URL in most browsers and is used by 'w/w/w' search engines)		Background Color:
Q&D Software Development Order Desk		FFFBF0
Form Heading -- (appears at top of Web page in bold type)		Text Color:
Q&D Software Development Order Desk <input checked="" type="checkbox"/> Center		000080
E-Mail responses to (will not appear on Web page)	Alternate (for mailto forms only)	Background Graphic
dversch@q-d.com		
Text to appear in Submit button	Text to appear in Reset button	<input type="radio"/> Mailto
Send Order	Clear Form	<input checked="" type="radio"/> CGI
Scrolling Status Bar Message (max length = 200 characters)		
****WebMania 1.5b with Image Map Wizard is here!!****		
<< Prev Tab		Next Tab >>

H2-8: Aesthetic and minimalist design

- No irrelevant information in dialogues

Heuristic: Aesthetics

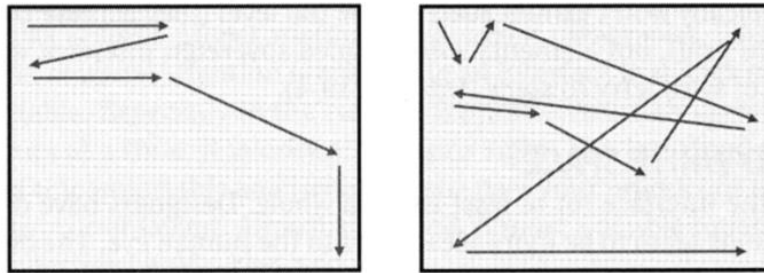


From Cooper's "The inmates are running the asylum"

Heuristic: Aesthetics

Simple and natural dialog

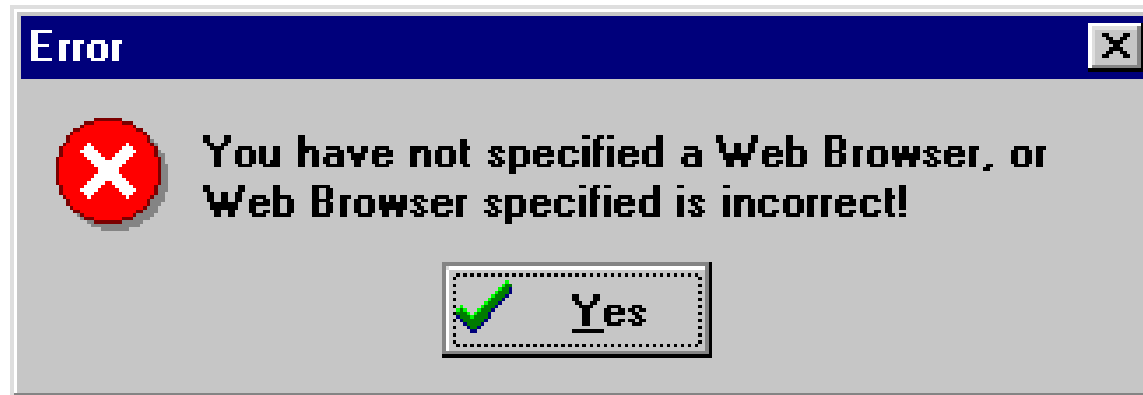
Present information in natural order



From Cooper's "About face 2.0"

- Occam's razor
 - Remove or hide irrelevant or rarely needed information
 - They compete with important information on screen
- Use windows frugally
 - Avoid complex window management

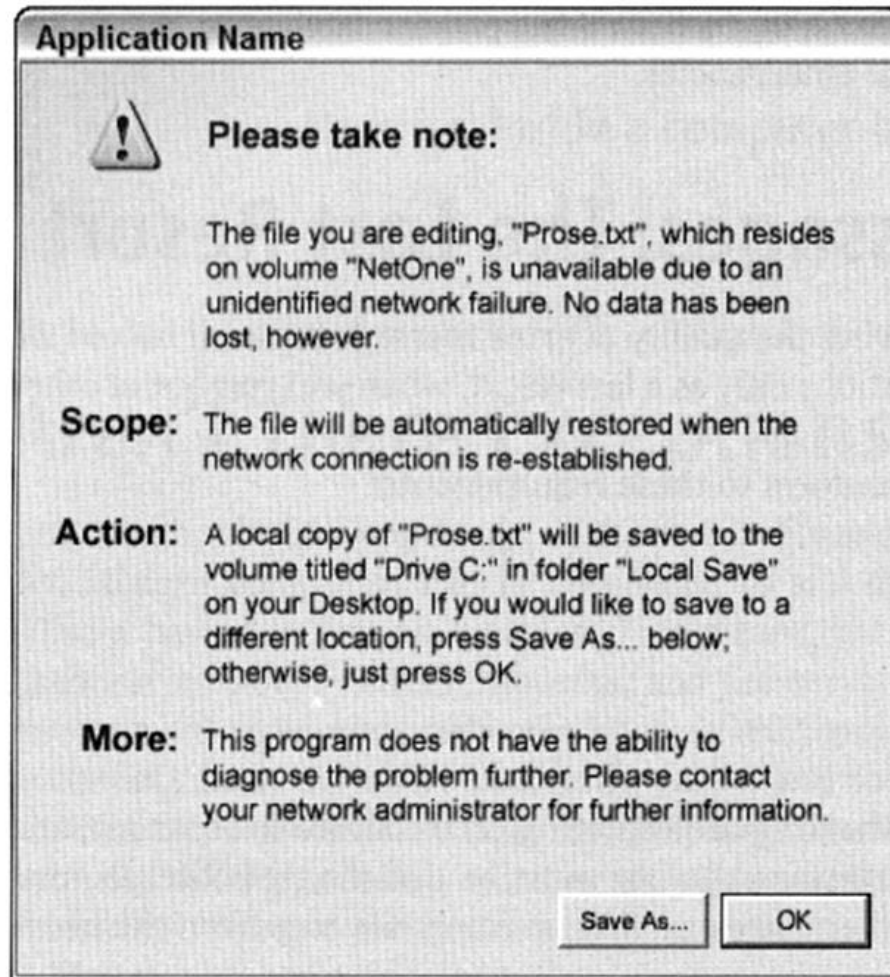
Heuristics: Help Users



H2-9: Help users recognize, diagnose, and recover from errors

- Error messages in plain language
- Precisely indicate the problem
- Constructively suggest a solution

Good Error Messages

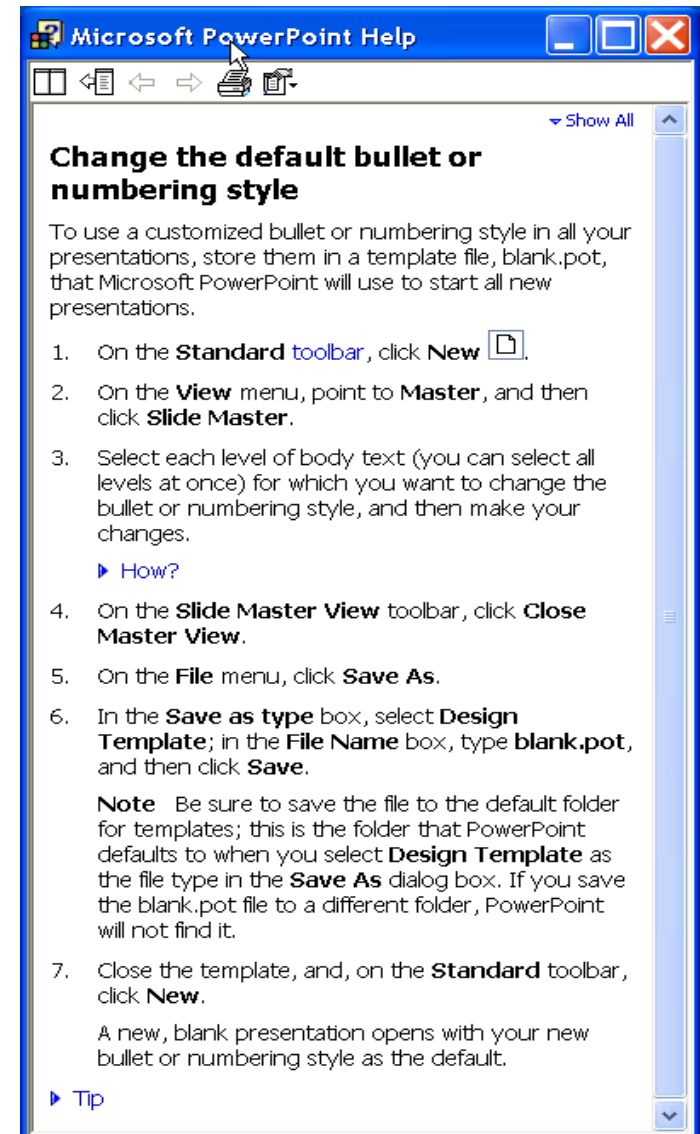


From Cooper's "About Face 2.0"

Heuristics: Docs

H2-10: Help and documentation

- Easy to search
- Focused on the user's task
- List concrete steps to carry out
- Not too long



The Process of Heuristic Evaluation

Phases of Heuristic Eval. (1-2)

1) Pre-evaluation training

- Provide the evaluator with domain knowledge if needed

2) Evaluation

- Individuals evaluate interface then aggregate results
 - Compare interface elements with heuristics
- Work in 2 passes
 - First pass: get a feel for flow and scope
 - Second pass: focus on specific elements
- Each evaluator produces list of problems
 - Explain why with reference to heuristic or other information
 - Be specific and list each problem separately

Phases of Heuristic Eval. (3-4)

3) Severity rating

- Establishes a ranking between problems
 - Cosmetic, minor, major and catastrophic
- First rate individually, then as a group

4) Debriefing

- Discuss outcome with design team
- Suggest potential solutions
- Assess how hard things are to fix

Examples

Can't copy info from one window to another

- Violates “Minimize the users’ memory load” (H2-3)
- Fix: allow copying

Typography uses mix of upper/lower case formats and fonts

- Violates “Consistency and standards” (H2-4)
 - Slows users down
 - Fix: pick a single format for entire interface
-
- Probably wouldn't be found by user testing

Severity Rating

Used to allocate resources to fix problems

Estimates of need for more usability efforts

Combination of

- Frequency
- Impact
- Persistence (one time or repeating)

Should be calculated after all evaluations are in

Should be done independently by all judges

Levels of Severity

- 0 - don't agree that this is a usability problem
- 1 - cosmetic problem
- 2 - minor usability problem
- 3 - major usability problem; important to fix
- 4 - usability catastrophe; imperative to fix

Severity Ratings Example

1. [H2-4 Consistency] [Severity 3][Fix 0]

The interface used the string "Save" on the first screen for saving the user's file, but used the string "Write file" on the second screen. Users may be confused by this different terminology for the same function.

Pros and Cons of Heuristic Evaluation

HE vs. User Testing

HE is much faster

- 1-2 hours each evaluator vs. days-weeks

HE doesn't require interpreting user's actions

User testing is far more accurate (by def.)

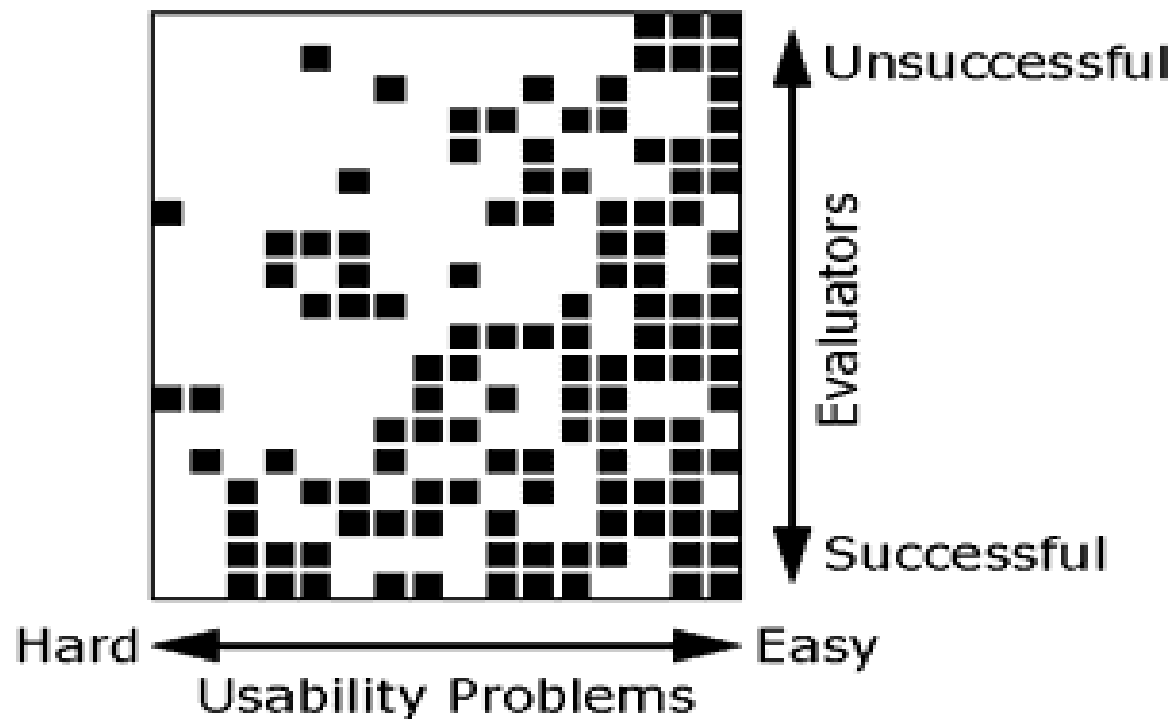
- Takes into account actual users and tasks
- HE may miss problems & find “false positives”

Good to **alternate** between HE & user-based testing

- Find different problems
- Don't waste participants

Why Multiple Evaluators?

- Every evaluator doesn't find every problem
- Good evaluators find both easy & hard ones



Number of Evaluators

Single evaluator achieves poor results

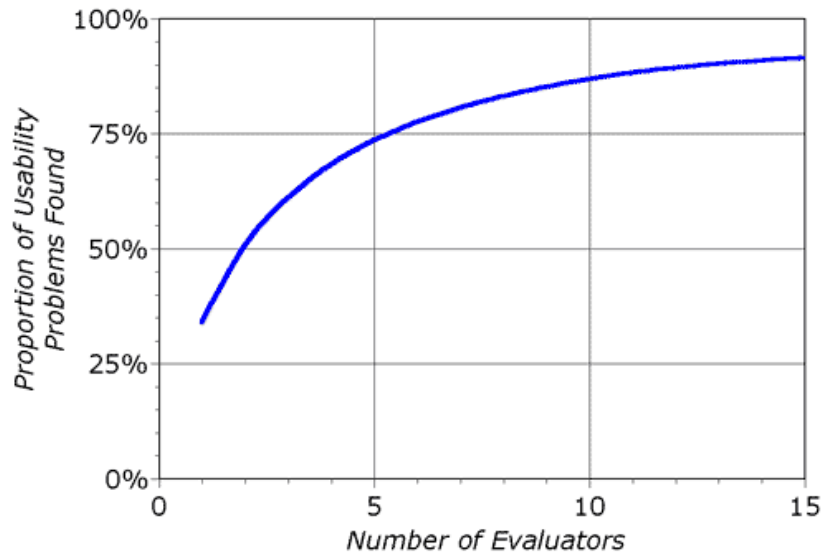
- Only finds 35% of usability problems
- 5 evaluators find ~ 75% of usability problems
- Why not more evaluators???? 10? 20?
 - Adding evaluators costs more
 - Many evaluators won't find many more problems

But always depends on market for product:

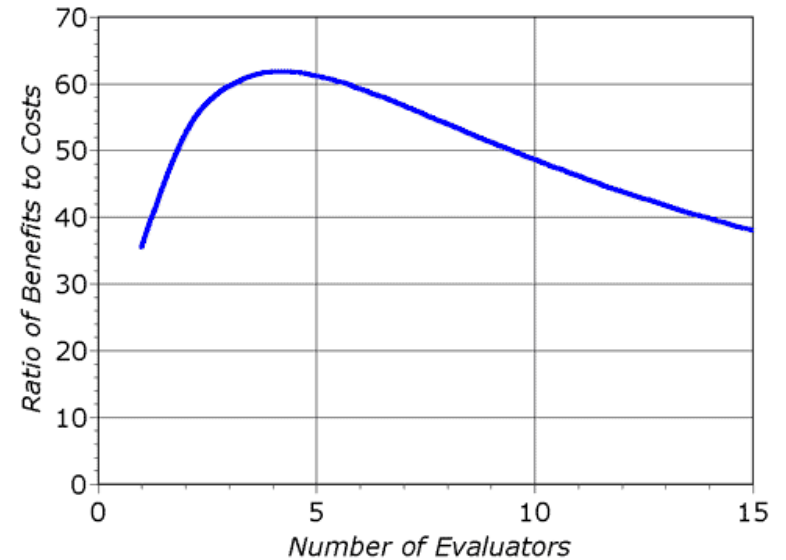
- popular products → high support cost for small bugs

Decreasing Returns

Problems Found



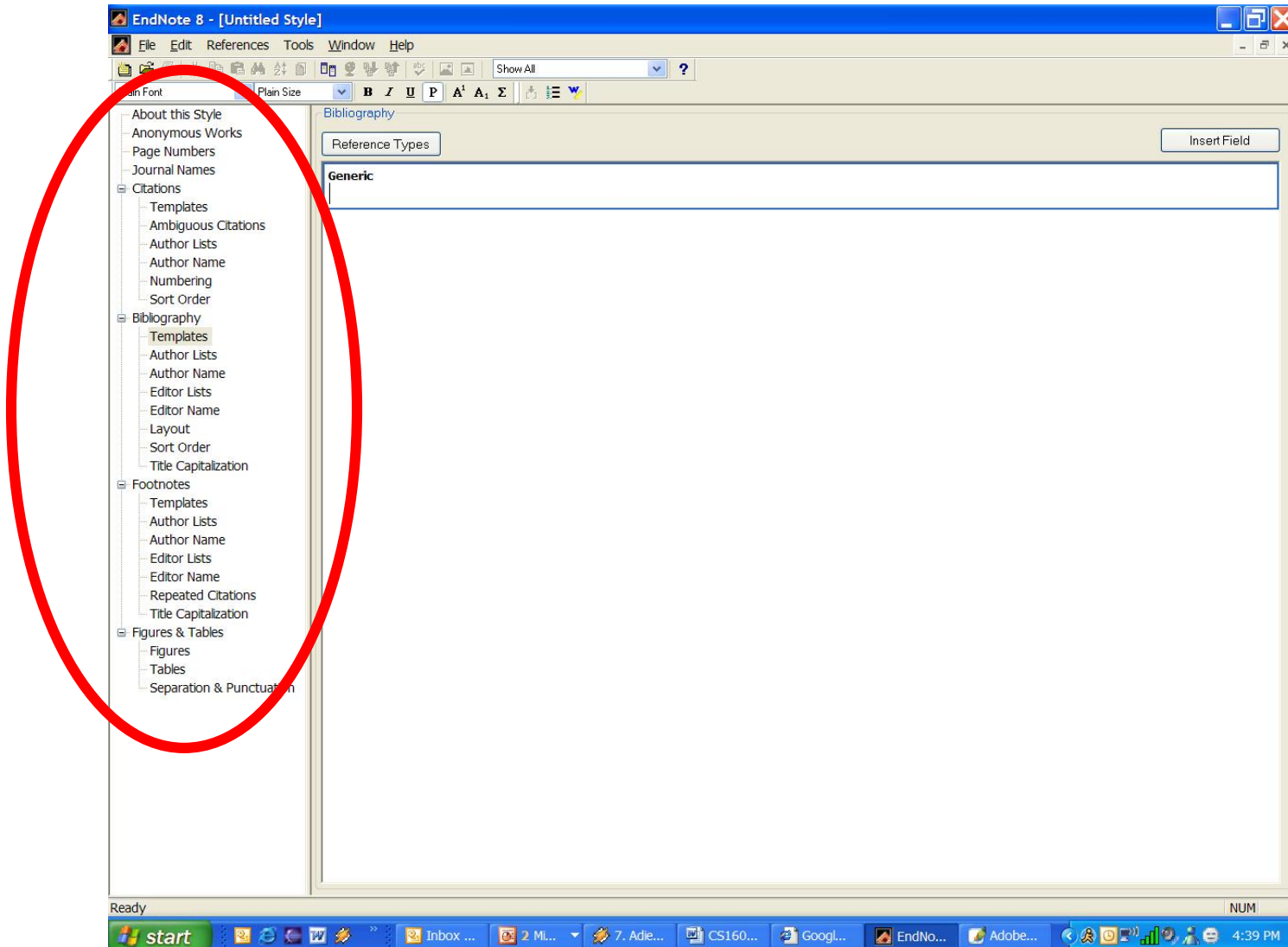
Benefits / Cost



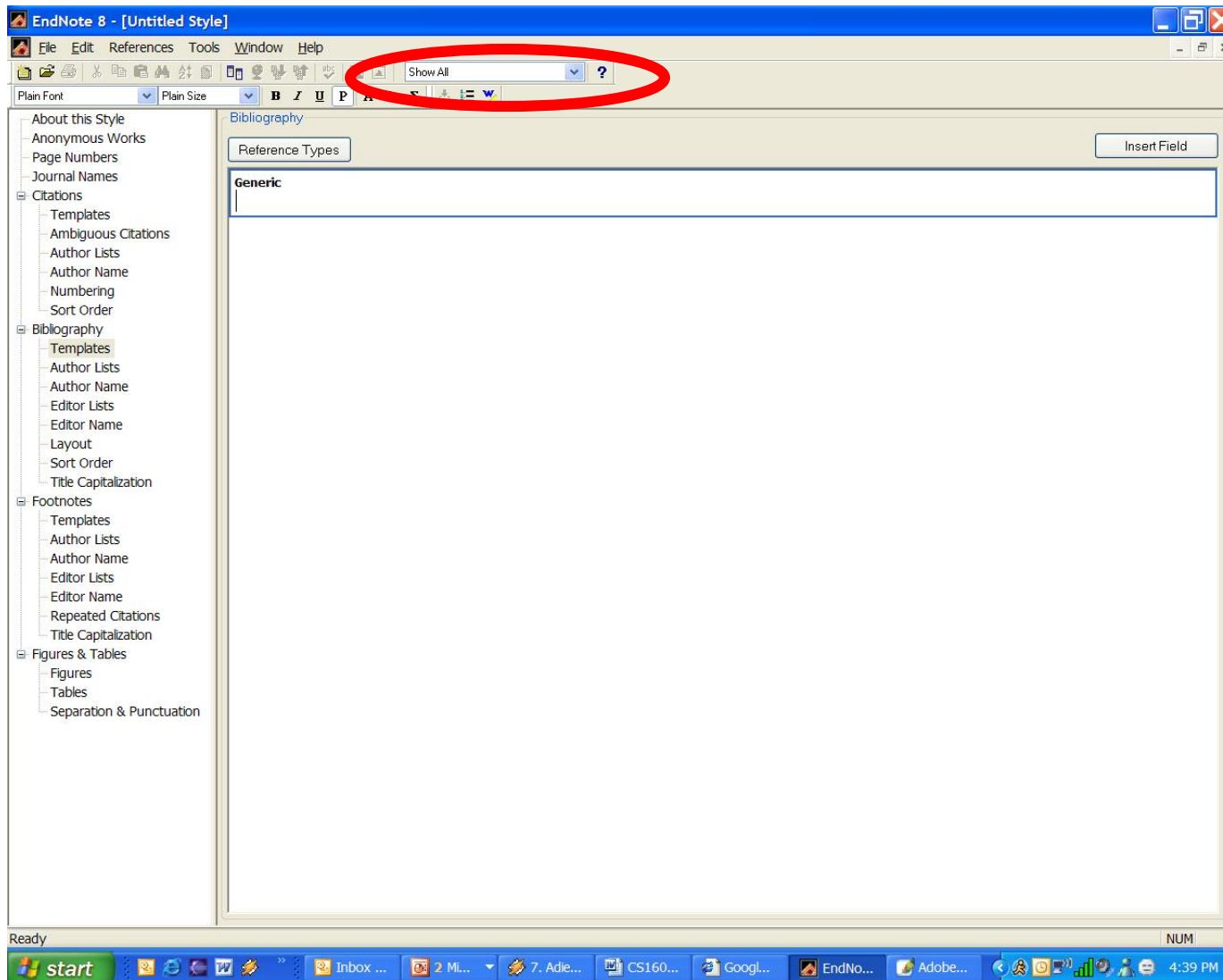
Caveat: Graphs are for a specific example

Summary

- Heuristic evaluation is a discount method
- Have evaluators go through the UI twice
 - Ask them to see if it complies with heuristics
 - Note where it doesn't and say why
- Have evaluators independently rate severity
- Combine the findings from 3 to 5 evaluators
- Discuss problems with design team
- Cheaper alternative to user testing
 - Finds different problems, so good to alternate

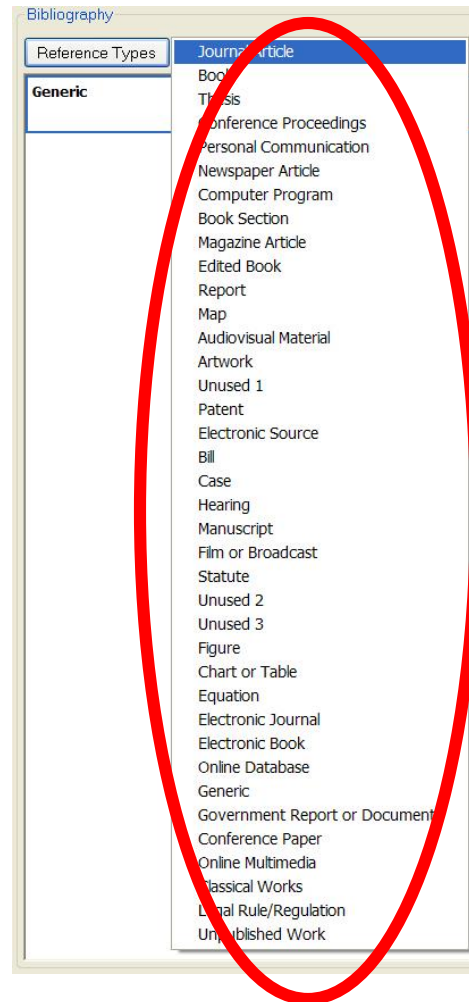


H2-8: Aesthetic and minimalist design



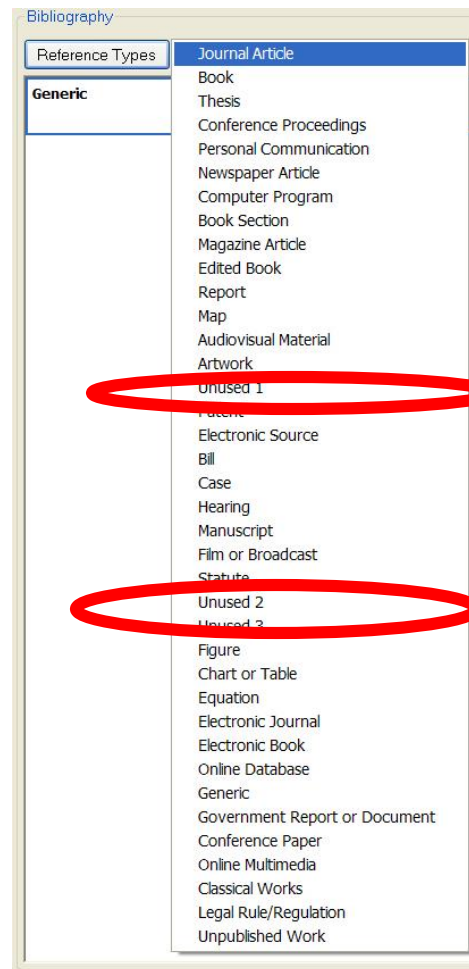
H2-2: Match system and real world

Screen Shot #2

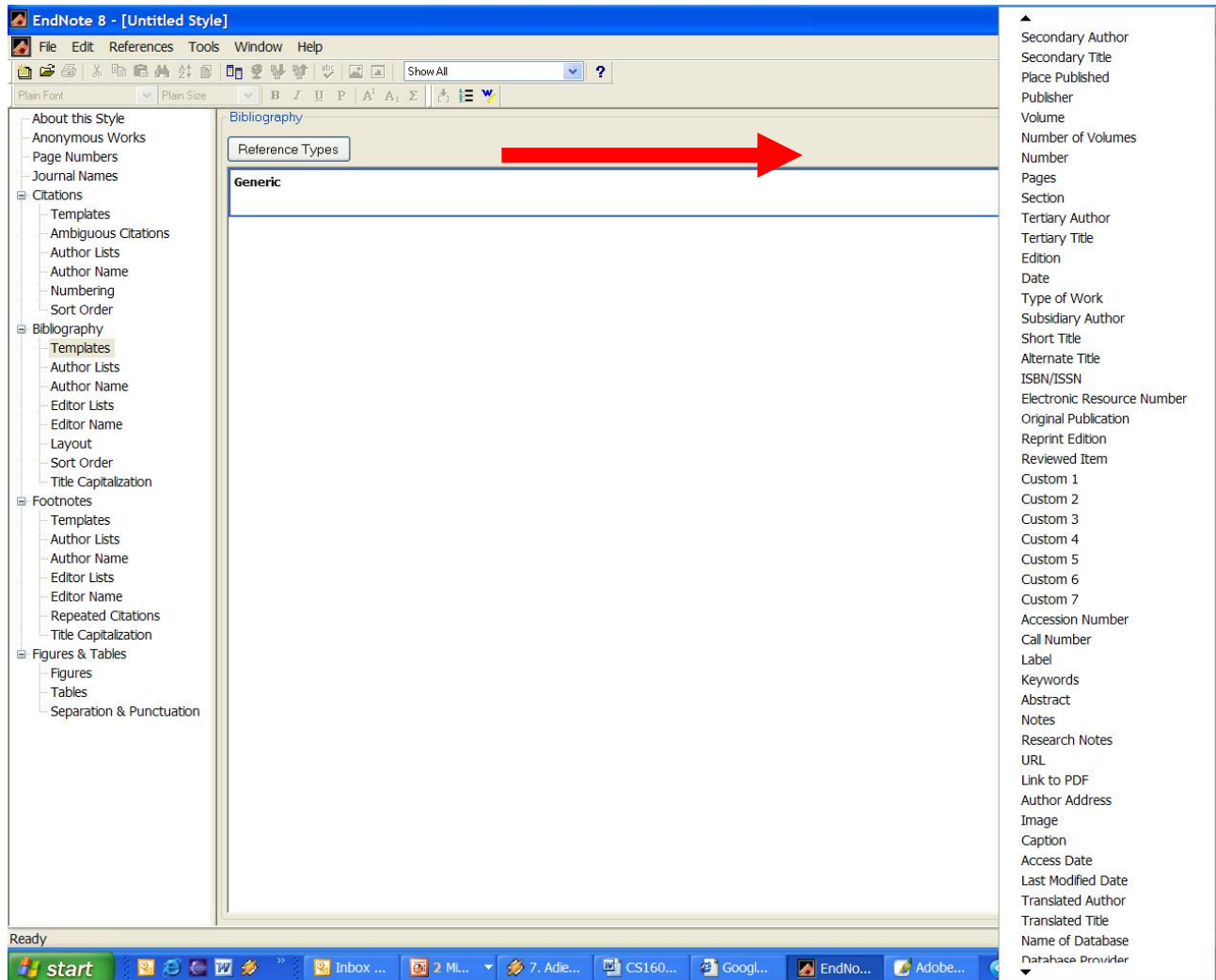


H2-8: Aesthetic and minimalist design

Screen Shot #2

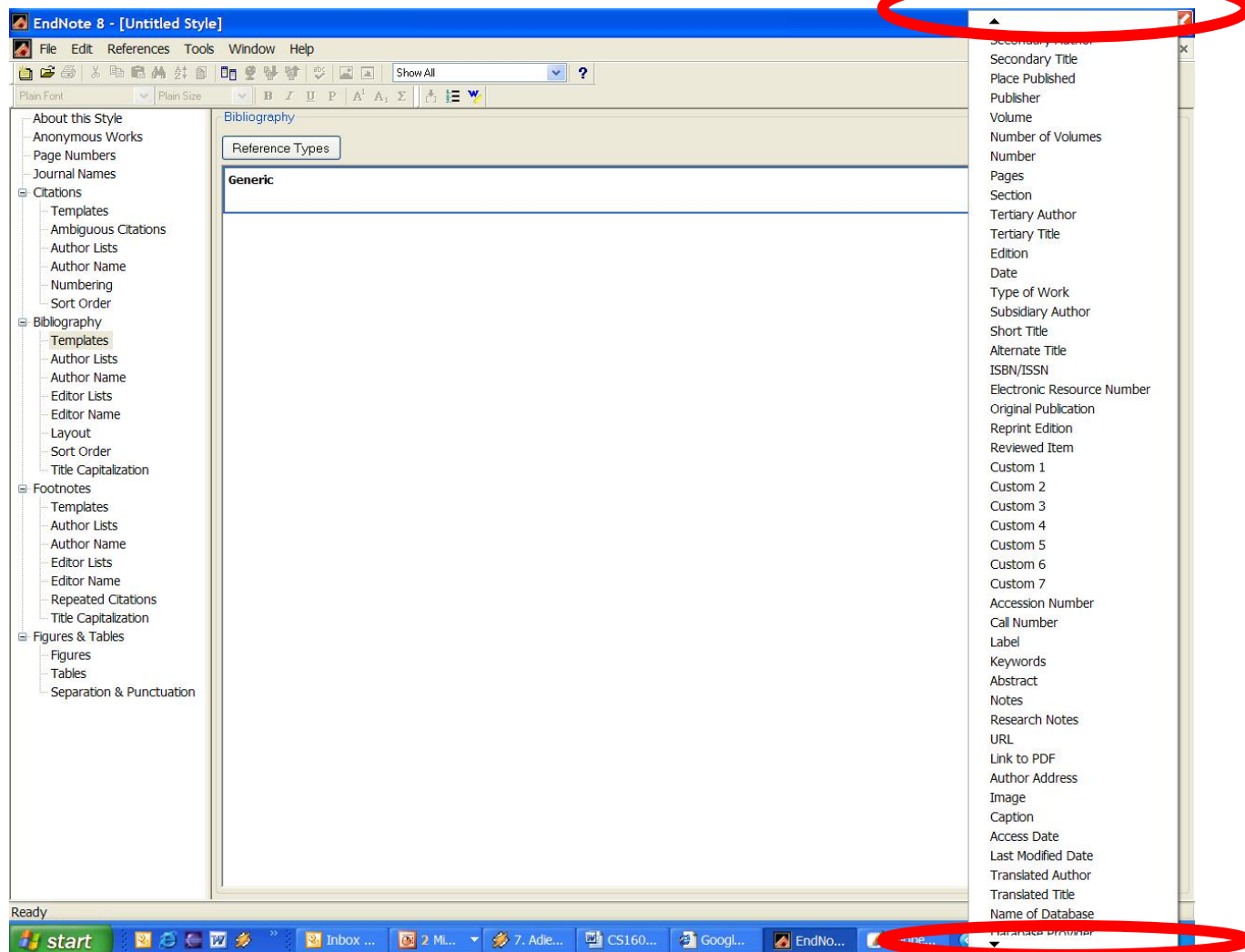


H2-4: Consistency and standards

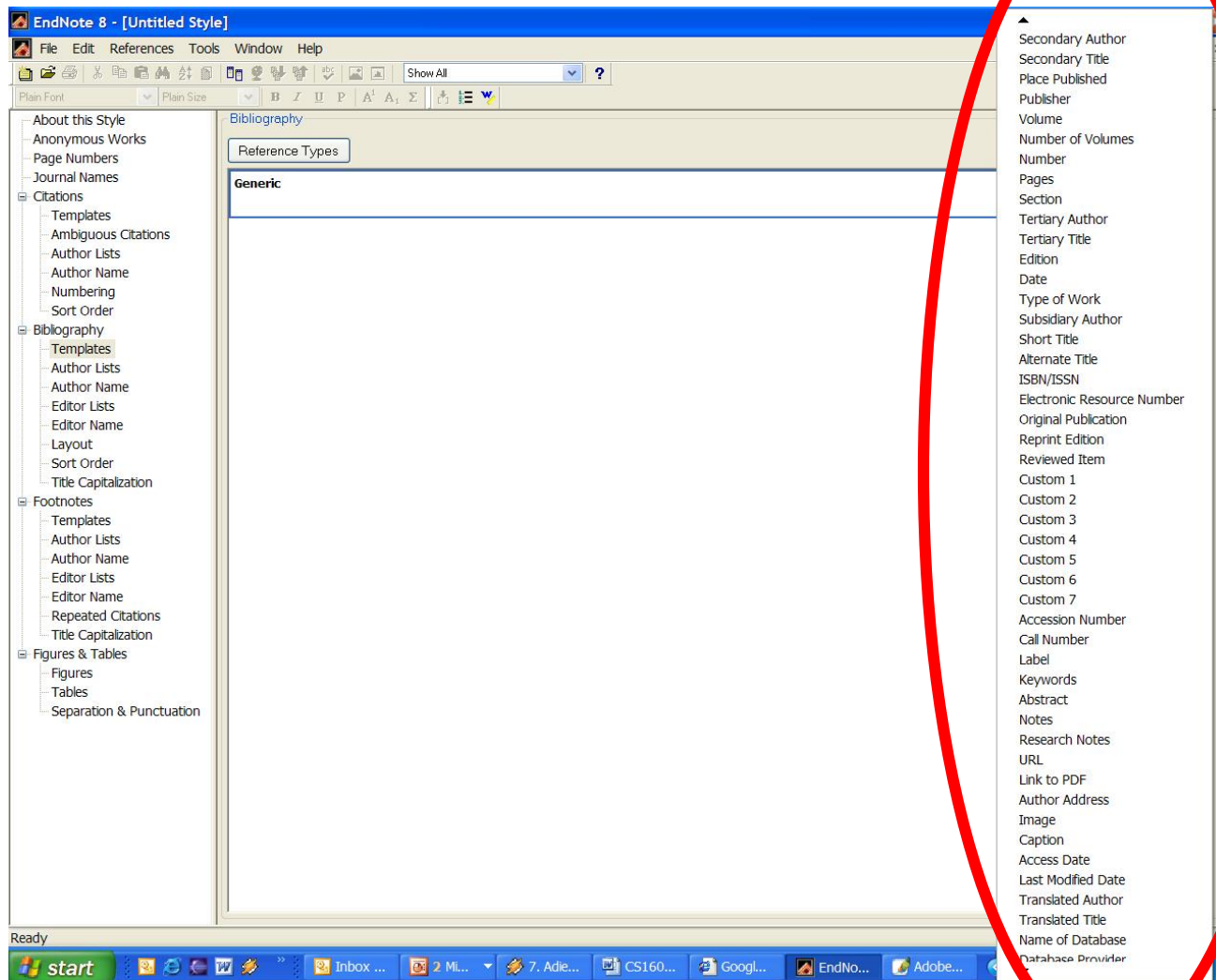


H2-4: Consistency and standards

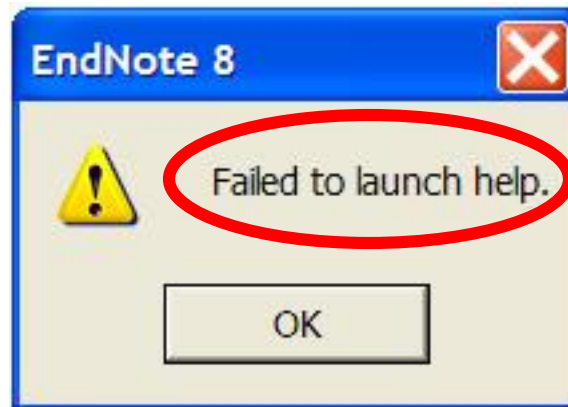
H2-2: Match system and real world



H2-7: Flexibility and efficiency of use



H2-8: Aesthetic and minimalist design



H2-9: Help users recognize, diagnose and recover from errors