HCI and Design

SPRING 2016

Topics for today

- Heuristic Evaluation
  - 10 usability heuristics
  - How to do heuristic evaluation
- Project planning and proposals

Usability Testing

Formal usability testing in a lab

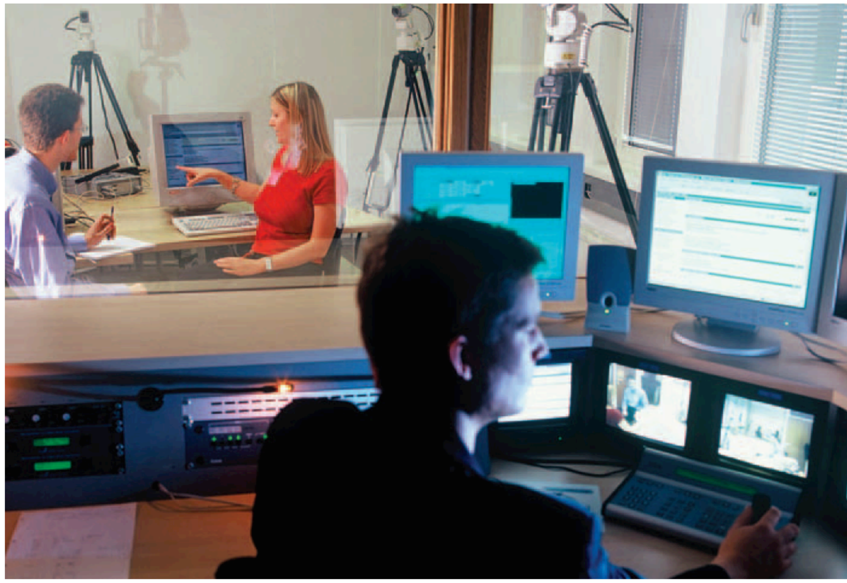
Performed with participants representative of the target population

Pros:

Reveals more problems than other methods
Finds global problems
Finds unique problems

Cons:

Time-intensive / Expensive
Does not detect local/minor problems well
Target population may be difficult to define



Discount Usability Methods

Heuristic evaluation: Helps find usability problems in a design

- Term coined by Jakob Nielsen
- "Guerilla HCI" (1994) http://www.useit.com/papers/guerrilla_hci.html

Reaction to recommended usability engineering methods (expensive, intimidating)

10 principles

Small set (3-5) of evaluators examine interface

- independently check compliance with principles
- different evaluators will find different problems
- evaluators only communicate afterwards

Can be done on working interfaces or sketches

Nielsen’s 10 heuristics (1994)

Visibility of system status

Match between system and the real world

User control and freedom

Consistency and standards

Error prevention

Recognition rather than recall

Flexibility and efficiency of use

Aesthetic and minimalist design

Help recognize, diagnose, and recover from errors

Help and documentation

1. Visibility

Visibility of system status

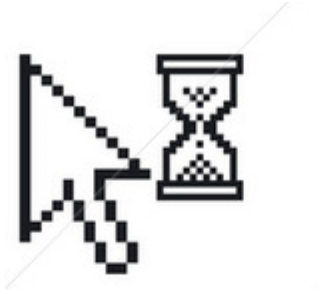
- The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

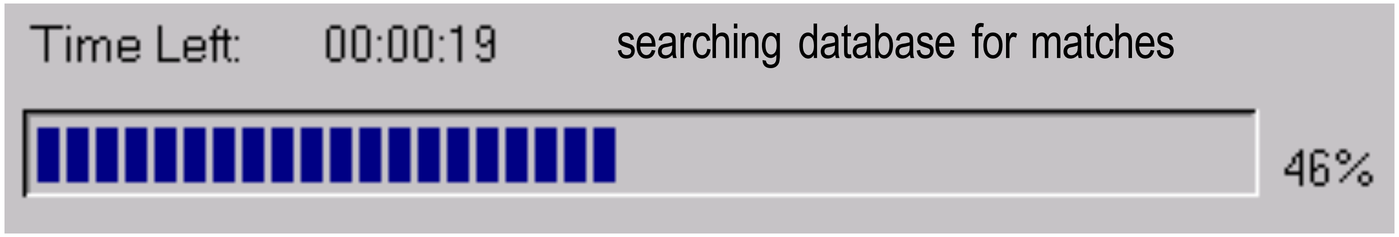
Refers to both visibility of system status and use of feedback.

- Anytime wondering what state the system is in, or the result of some action, this is a visibility violation.

1. Visibility

- Pay attention to response time
  - 0.1 sec: no special indicators needed
  - 1.0 sec: user tends to lose track of data
  - 10 sec: maximum duration if user to stay focused on action
  - longer delays absolutely require percent-done progress bars

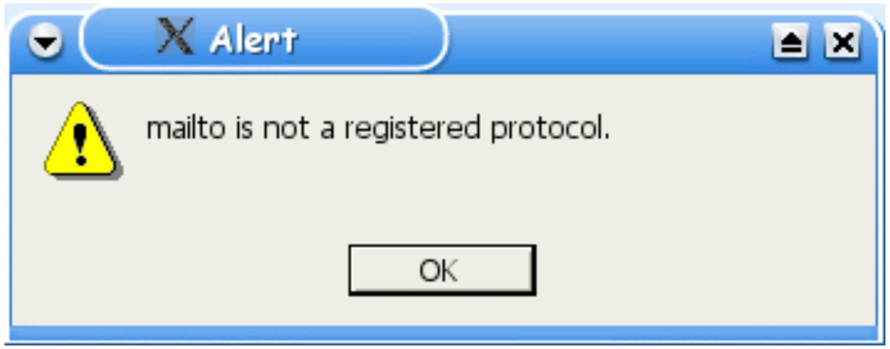




2. Real world match

- Match between system and the real world
- The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- Refers to word and language choice, conceptual model, metaphor, mapping, and sequencing.

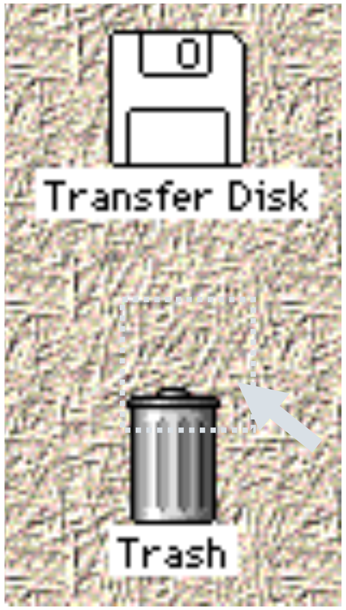
2. Real world match



mailto is not a registered protocol.

"Mailto", "protocol"?

Match system to real world
- Speak the user's language



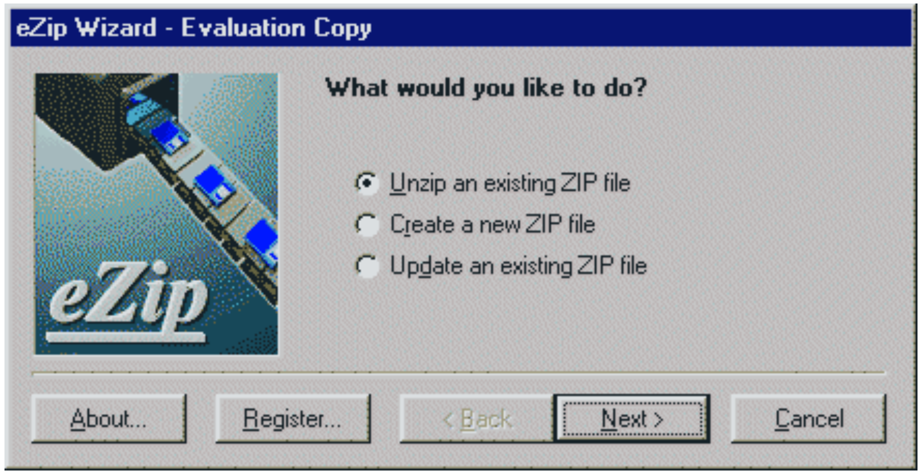
Mac desktop
- Dragging disk to trash should delete, not eject it

Match system to real world
- Speak the user's language
- Follow conventions

3. User control and freedom

- Users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialogue.
- ***Support undo and redo.***

3. User control and freedom


eZip Wizard - Evaluation Copy

What would you like to do?

(●) Unzip an existing ZIP file
( ) Create a new ZIP file
( ) Update an existing ZIP file

Don’t force users down fixed paths

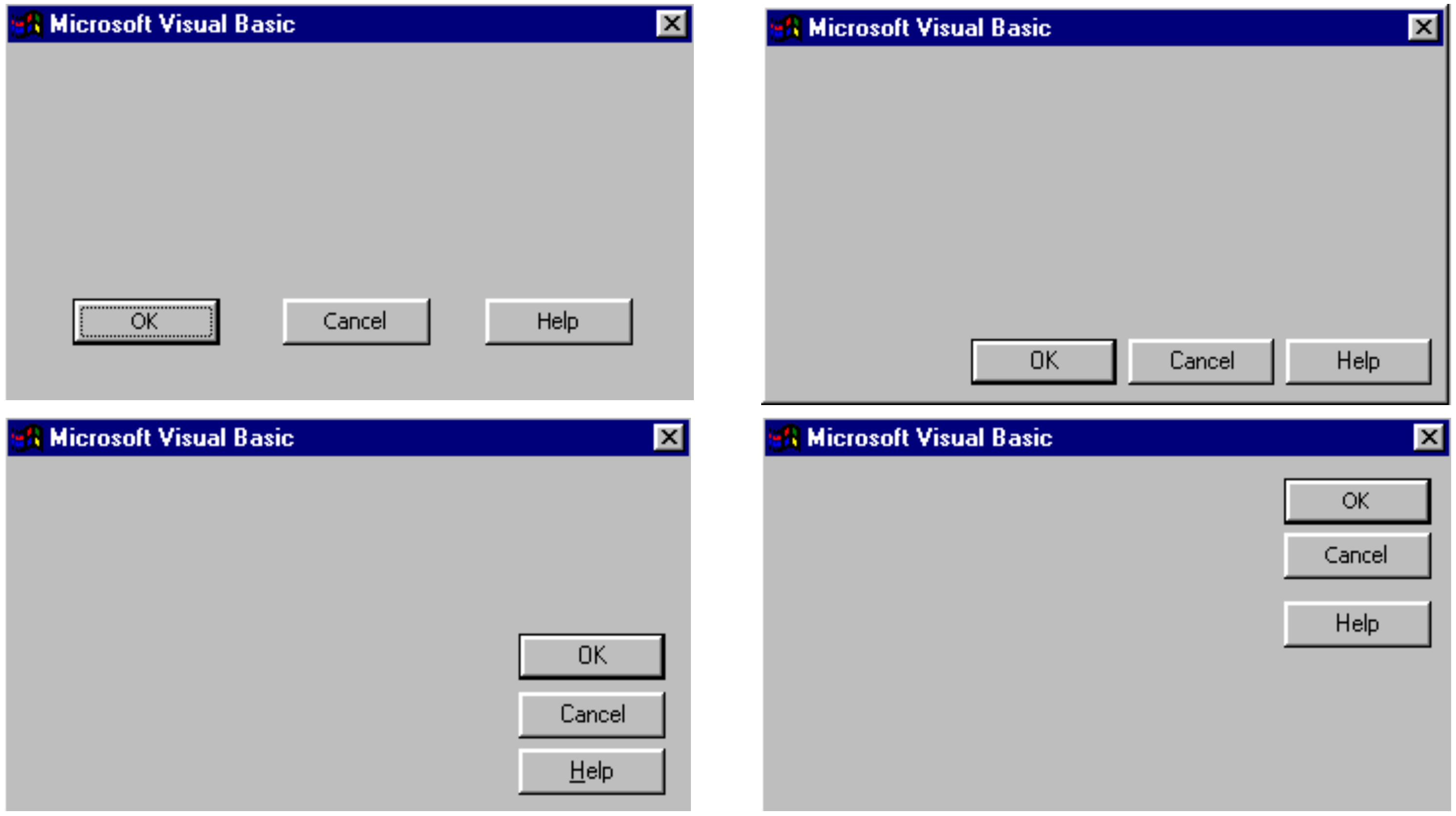
Wizards

- must respond to question before going to next
- good for beginners, infrequent tasks
- not for common tasks

4. Consistency and Standards

- Users should not have to wonder if different words, situations, or actions mean the same thing.
- Follow platform conventions.
- Internal consistency is consistency throughout the same product.
- External consistency is consistency with other products in its class.

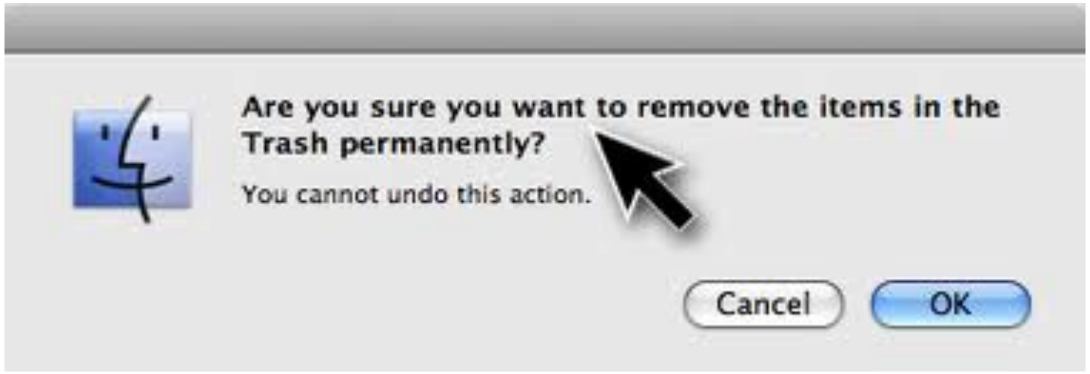
4. Consistency and Standards



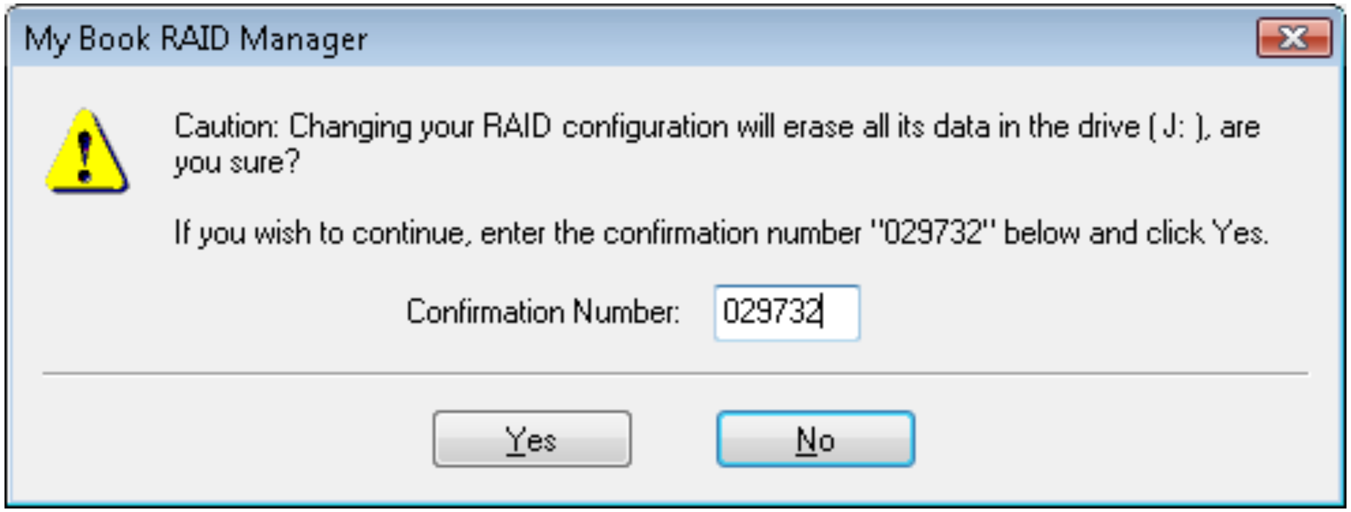
5. Error prevention

- Even better than good error messages is a careful design that prevents a problem from occurring in the first place.
- Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
- Try to commit errors and see how they are handled. Could they have been prevented?

5. Error prevention


Are you sure you want to remove the items in the Trash permanently?

You cannot undo this action.


My Book RAID Manager

Caution: Changing your RAID configuration will erase all its data in the drive (J:), are you sure?

If you wish to continue, enter the confirmation number "029732" below and click Yes.

Confirmation Number: 029732

6. Recognition not recall

- People should never carry a memory load

  Obvious exception: passwords (but can be lessened…)

- Minimize the user's memory load by making objects, actions, and options visible.

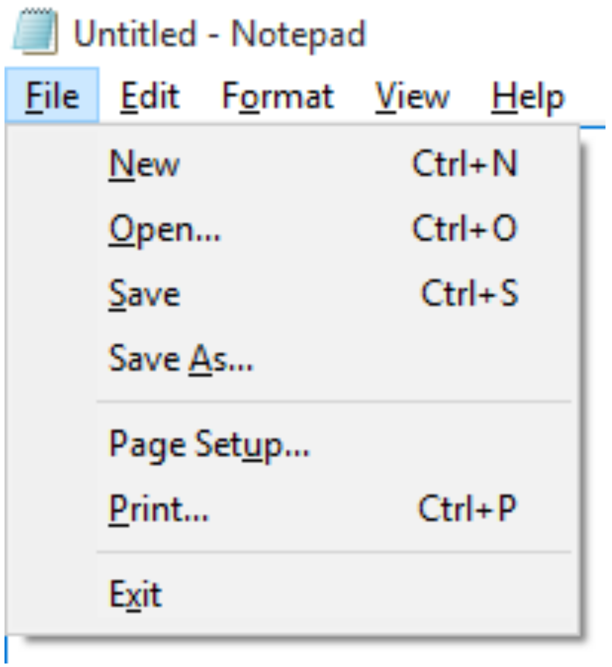
- The user should not have to remember information from one part of the dialogue to another.

- Instructions for use of the system should be visible or easily retrievable whenever appropriate.

- Problems with affordances often go here

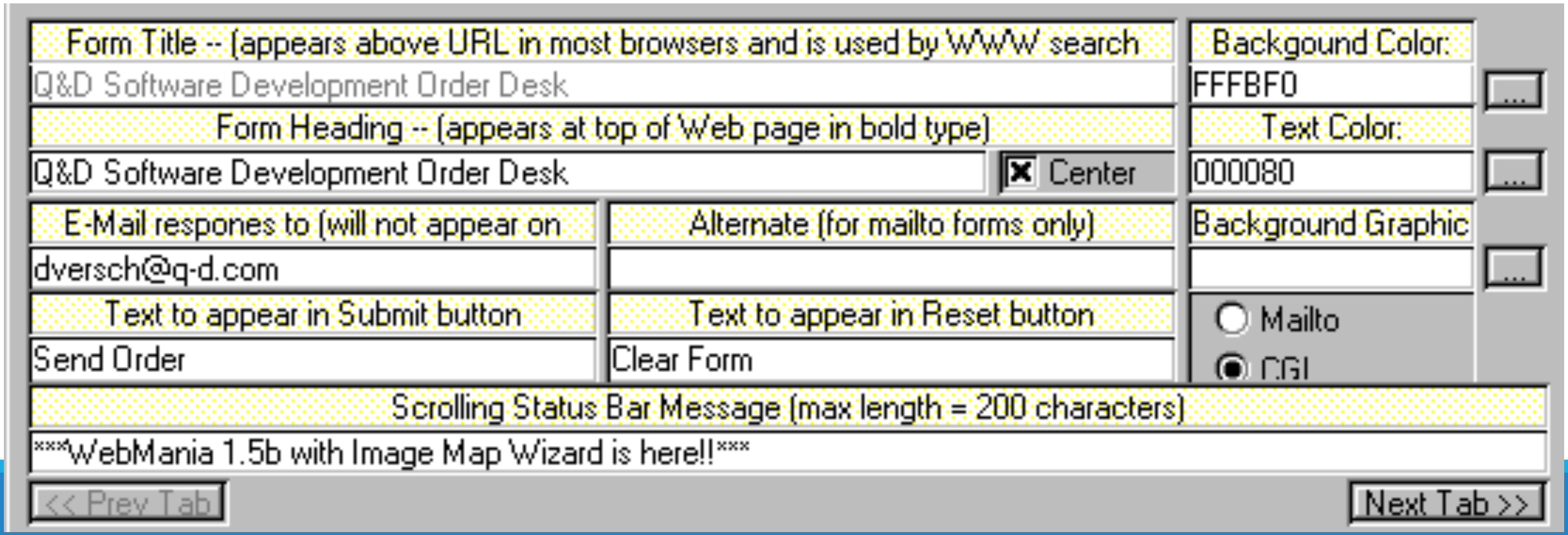
7. Flexibility and efficiency

- Cater to inexperienced and experienced users.
- Accelerators (e.g., keyboard shortcuts) may speed up interaction for expert users
- Allow users to tailor frequent actions (e.g., macros).
- Concerns anywhere users have repetitive actions that must be done manually.
- Also allows multiple ways to do things.



8. Aesthetic and minimalist design

- Dialogues should not contain information which is irrelevant or rarely needed.
- Every extra unit of information competes with relevant units of information and diminishes their relative visibility.
- Not just about “ugliness” but about clutter, overload of visual field, visual noise, distracting animations, etc.



Form Title -- (appears above URL in most browsers and is used by WWW search
Q&D Software Development Order Desk
Backgound Color:
FFFBF0
Form Heading -- (appears at top of Web page in bold type)
Q&D Software Development Order Desk
[✓ Center]
Text Color:
000080
E-Mail respones to (will not appear on
dversch@q-d.com
Alternate (for mailto forms only)
Background Graphic
Text to appear in Submit button
Send Order
Text to appear in Reset button
Clear Form
Mailto [✓ CGI]
Scrolling Status Bar Message (max length = 200 characters)
WebMania 1.5b with Image Map Wizard is here!!

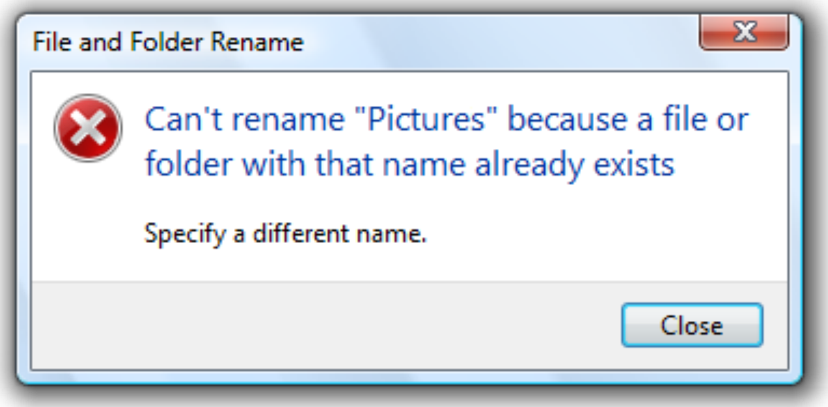
9. Error recovery

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes)

Precisely indicate the problem, and suggest a solution.

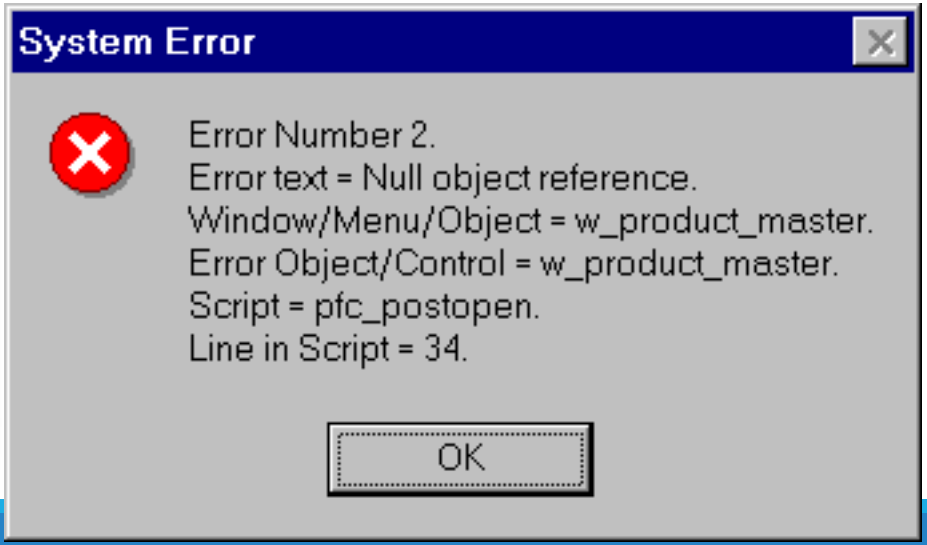
Error prevention (#5) is about preventing errors before they occur. This is about after they occur.



File and Folder Rename

Can't rename "Pictures" because a file or folder with that name already exists

Specify a different name.

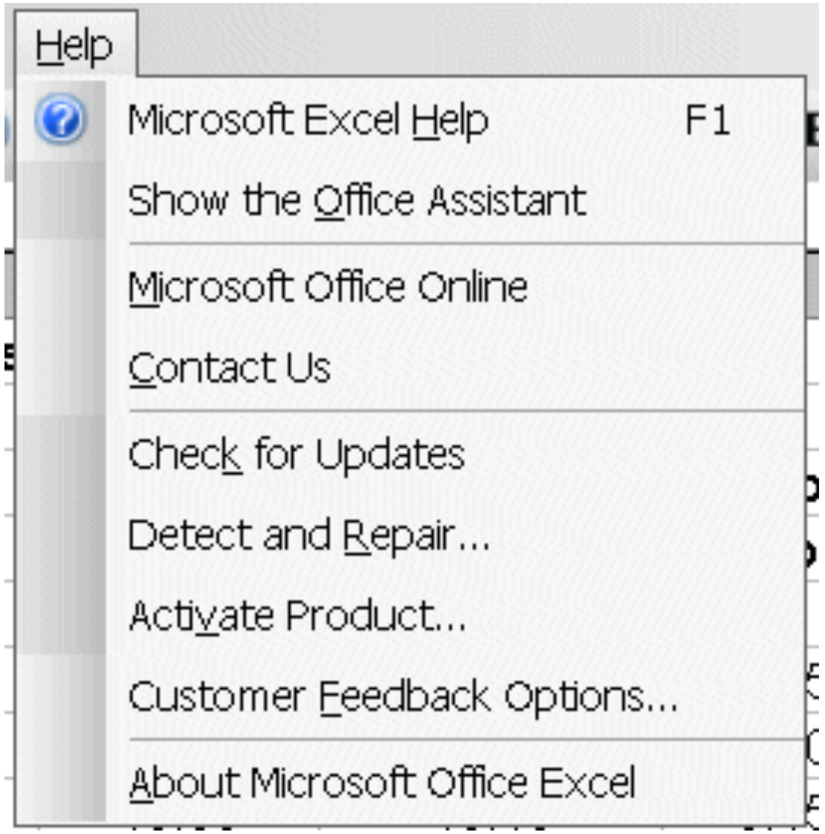


System Error

Error Number 2.
Error text = Null object reference.
Window/Menu/Object = w_product_master.
Error Object/Control = w_product_master.
Script = pfc_postopen.
Line in Script = 34.

10. Help and documentation

- Even though it is better if the system can be used without documentation, it is also often necessary to provide help and documentation.
- Any information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.
- This does not mean that the user must be able to ask for help on every single item.



10 heuristics

Visibility of system status

Match between system and the real world

User control and freedom

Consistency and standards

Error prevention

Recognition rather than recall

Flexibility and efficiency of use

Aesthetic and minimalist design

Help recognize, diagnose, and recover from errors

Help and documentation

How to Perform Heuristic Evaluation

Evaluators go through interface several times

- inspect various dialogue elements
- compare with list of usability principles
- consider other principles/results that come to mind

Use violations to redesign/fix problems

Phases of Heuristic Evaluation

1) Pre-evaluation training
   - give expert evaluators needed domain knowledge & information on the scenario

2) Evaluation
   - individuals evaluate interface and make lists of problems

3) Severity rating
   - determine how severe each problem is

4) Aggregation
   - group meets and aggregates problems (w/ severity ratings)

5) Debriefing
   - discuss the outcome with design team

How to Perform Heuristic Evaluation

At least two passes for each evaluator

- first to get feel for flow and scope of system
- second to focus on specific elements

If system is walk-up-and-use or evaluators are domain experts, no assistance needed

- otherwise might supply evaluators with scenarios

Each evaluator produces list of problems

- explain why with reference to heuristic or other information
- be specific & list each problem separately

How to Perform Heuristic Evaluation

Why separate listings for each violation?
- risk of repeating problematic aspect
- may not be possible to fix all problems

Where problems may be found
- single location in interface
- two or more locations that need to be compared
- problem with overall structure of interface
- something that is missing
  - common problem with paper prototypes
    - sometimes features are implied by design documents and just haven't been "implemented" – relax on those

Severity Rating

Used to allocate resources to fix problems

Combination of frequency, impact, persistence (one time or repeating)

Should be calculated after all evaluations are in

Should be done independently by all evaluators

Rating: 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

[Consistency] [Severity 3]

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.

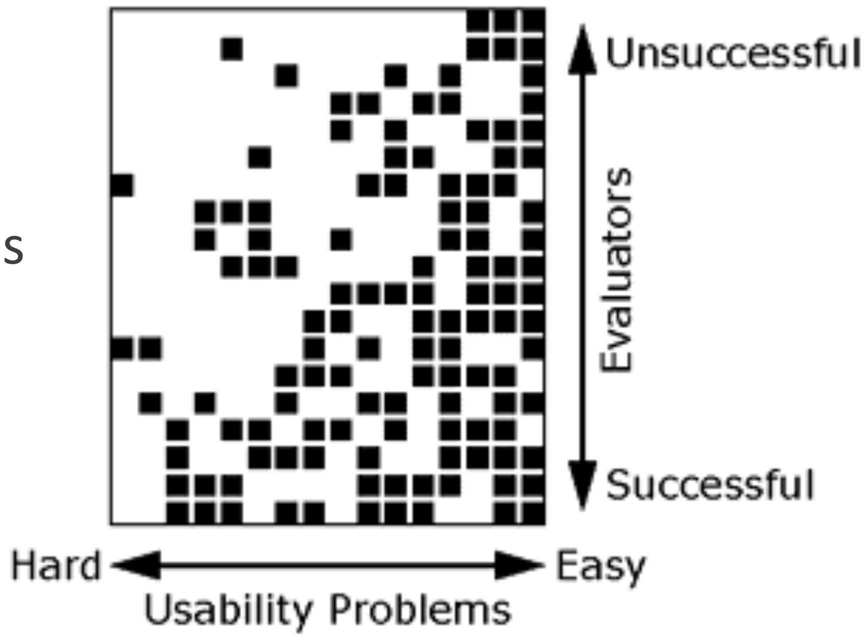
Severity 3 - major usability problem; important to fix

Why Multiple Evaluators?

Every evaluator doesn't find every problem

Good evaluators find both easy and hard problems

Single evaluator achieves poor results
- only finds 35% of usability problems
- 5 evaluators find ~ 75% of usability problems
- why not more???? 10? 20?
  - adding evaluators costs more and won't find more problems



Debriefing

Conduct with evaluators, observers, and team members

Discuss general characteristics of interface

Suggest improvements to address major usability problems

Development team rates how hard to fix

Make it a brainstorming session

Heuristic Evaluation vs. Usability Testing

Heuristic evaluation is much faster

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

Heuristic evaluation doesn’t require interpreting user’s actions

User testing is more accurate (by definition)

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

Good to alternate between heuristic evaluation & user testing

- Find different problems
- Don’t waste participants

Heuristic Evaluation Summary

- Evaluators examine interface against a set of guidelines
- Normally performed by a team of three to five experts
- Rate individually, form consensus
- Can be performed on early designs, mockups, prototypes, products
- Much cheaper than usability experiments
- Reasonably effective
  - 1 expert 40% of errors, 2: 50%, 3: 60%, 5: 80%

Next time…

- No class on Tuesday (Feb break)
- Work on your project proposals (due Feb 18th)
- Check slack for updates on next class