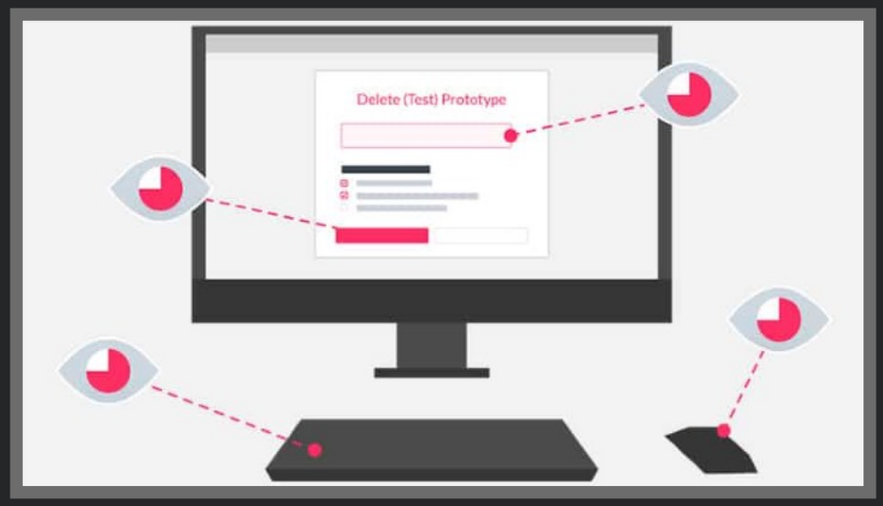
SWE 432 -Web Application Development

Fall 2022

George Mason University

Dr. Kevin Moran

Week 11: User Centered Design



Administrivia

- *HW Assignment 3* - Grades and comments will be posted by Thursday.
- *HW Assignment 4 -* Out now, Due in a little over two weeks (November 17th) *at midnight*.
  - Extra Credit Opportunity!

Class Overview

- ***Part 1:*** In-Class Activity on React + CSS
  - Working with Functional Components
- ***Part 2:*** Introduction to User-centered design
  - Quick Lecture
  - Heuristic Evaluation Activity

Functional React & CSS Exercises

- *React Exercise:*
  - *Instructions:* Implement conditional rendering so that the Message is displayed when a user presses the button.
- *CSS Exercise:*
  - *Instructions:*
    - Center a component inside it's container
    - Use a display grid to create layout with multiple rows and columns
    - Override one of the Bootstrap selectors
  - https://replit.com/@kmoran/swe-432-react-example#src/App.jsx

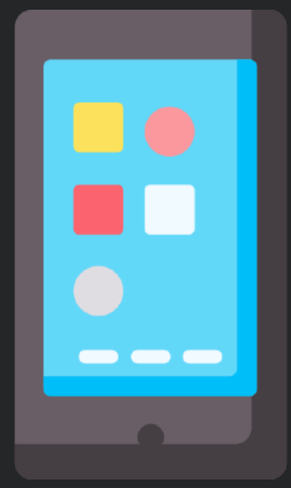
Please Turn-in Your In-Class Activity for a Quiz Grade

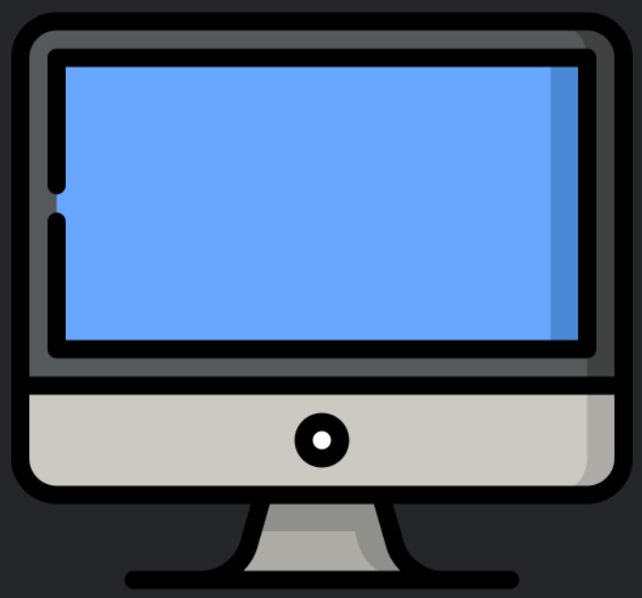
`https://bit.ly/3SDGckh`

Please provide the URL of your Replit Repo

User-Centered Design

Web Apps are Ubiquitous





“Good Design” is incredibly important

"Good Design" is incredibly important

… and is centered on ***usability***

What is Usability?

Ease of Use

Productivity

Learnability

Efficiency

Retainability

User Satisfaction

Effectiveness

Usable or Unusable?

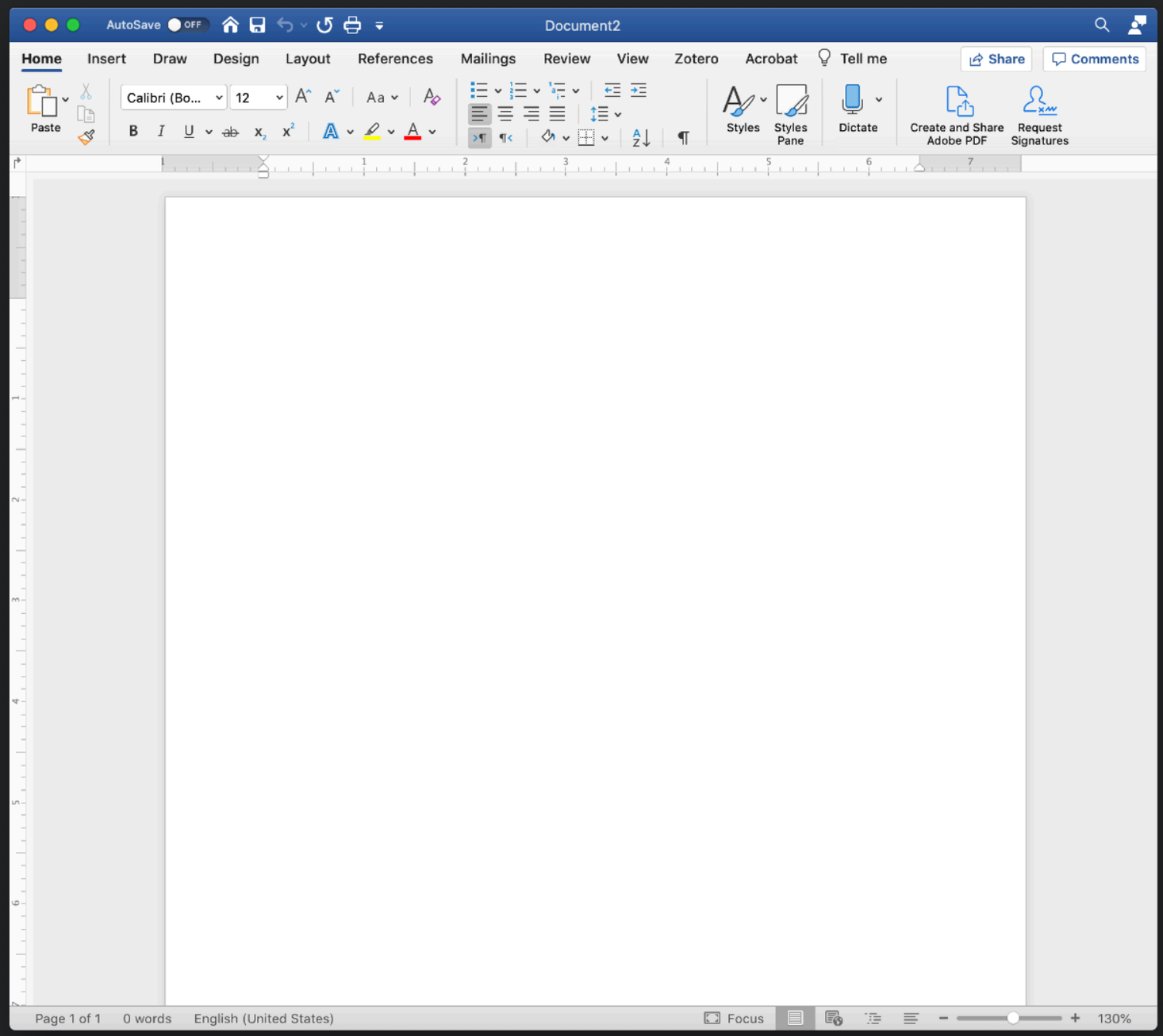
A Teapot

Usable or Unusable?

A Door

Usable or Unusable?

A Word Processor



Usability

- A property of the relationship between
  - humans with goal-driven tasks
    - an artifact
- The speed and success with which the goals can be accomplished (task ***performance***)

Needfinding

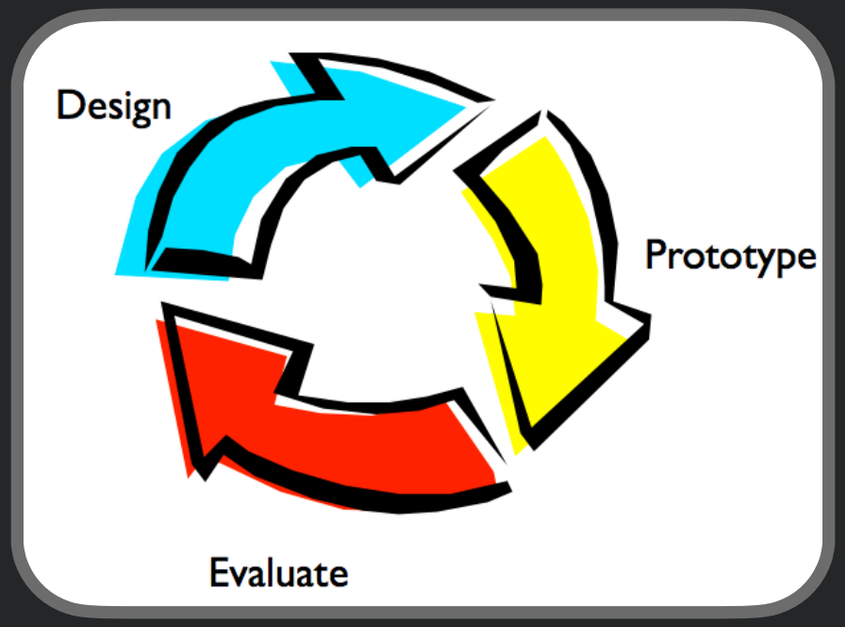
- Given an existing artifact and humans doing a set of tasks, determine ***goals*** and identify ***usability issues*** that decrease task performance

User-Centered Design

- Given humans with goals and tasks, design an ***artifact*** that helps to accomplish these tasks

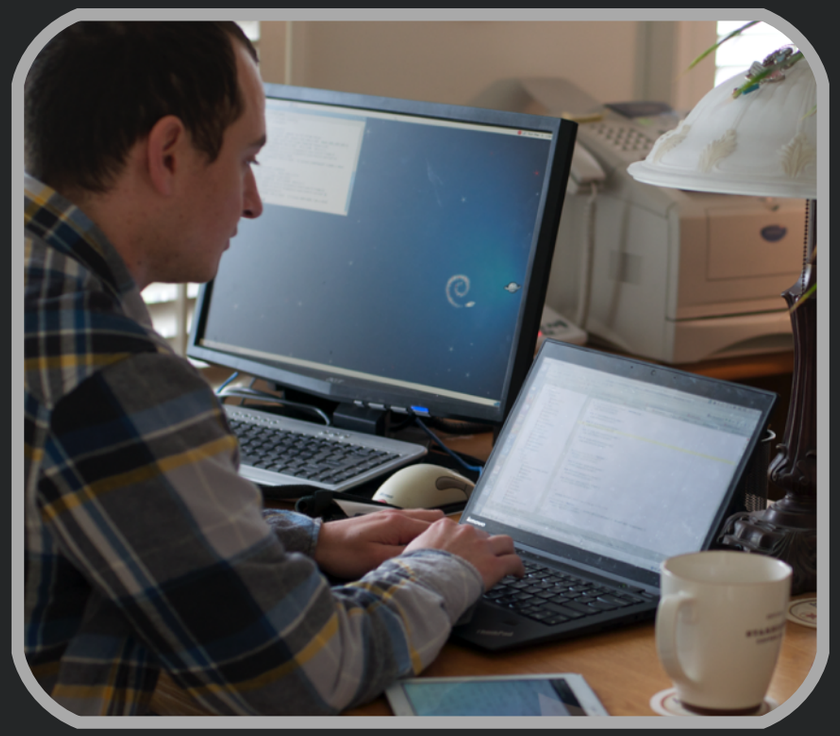
Iterative User-Centered Design

- Given humans with goals and tasks, ***re***design an ***existing*** artifact that helps to accomplish these tasks ***faster*** and more ***successfully***



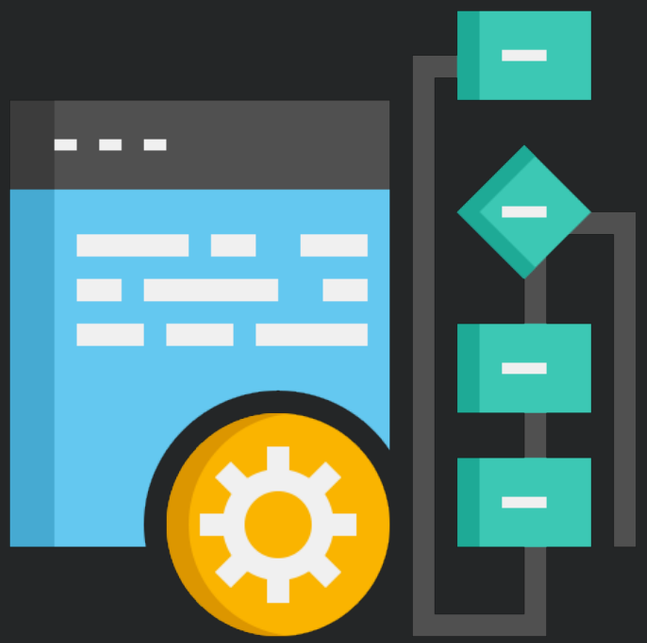
Empirical: Usability **Study**

- Given humans with goals and tasks an artifact, ***observe humans*** to identify usability issues that decrease task performance



- ***"Ground Truth"***

Analytical: Usability **Principles**



- Given humans with goals and tasks and an artifact, ***<u>assess for conformance to UI principles</u>*** to identify usability issues that decrease task performance

- ***Approximation of "ground truth"***

Why Study Usability?

*"The results show that in today's applications, an average of **48% of the code** is devoted to the user interface portion."*

"The average time spent on the user interface portion is 45% during the design phase, 50% during the implementation phase, and 37% during the maintenance phase."

– Myers & Rosson, CHI'92

Why Study Usability?

Life-Threatening Errors

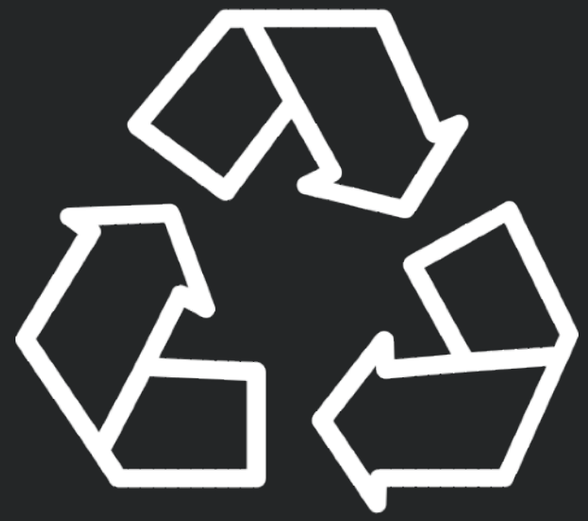
- 1995 American Airlines jet crashed into canyon wall, killing all aboard
- On approach to Rozo airport in Colombia
- Pilot skipped some of the approach procedures
- Pilot typed in “R” and system completed full name of airport to Romeo
- Guidance system executed turn at low altitude to head for Romeo airport
- 9 seconds later plane struck canyon wall
- Is the pilot to blame?
- *http://en.wikipedia.org/wiki/American_Airlines_Flight_965*

Iterative Model of User-Centered Design

Observation

(Re)Define the Problem

Understand User Needs



Idea Generation

Brainstorm
what to build

Prototype/ Implementation

Build

Test

Evaluate what
you have built

Heuristic Evaluation

Heuristic Evaluation (Analytical)

- "***Discount*** usability engineering methods" - Jakob Nielsen

- Involves a small team of evaluators to evaluate an interface based on recognized usability principles

- Heuristics – "rules of thumb"

Adapted from slides by Bonnie John and Jennifer Mankoff

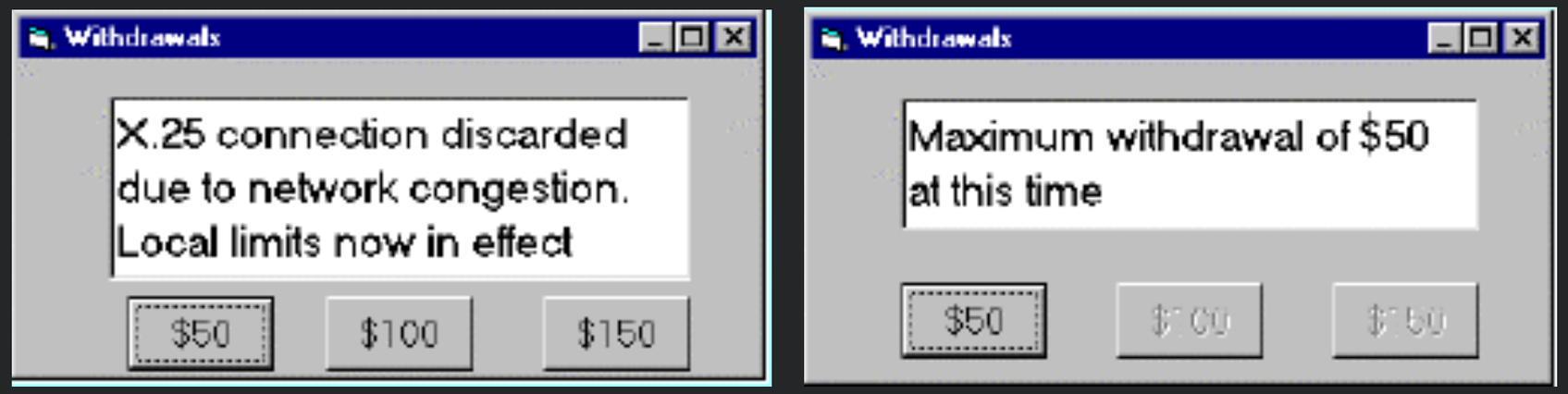
Heuristic Evaluation

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition vs. recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

H1: Visibility of System Status

- ***What input has been received*** - Does the interface above say what the search input was?
- ***What processing it is currently doing*** - Does it say what it is currently doing?
- ***What the results of processing are*** - Does it give descriptive results?
- Feedback allows user to monitor progress towards solution of their task, allows the closure of tasks and reduces user anxiety (*Lavery et al*)

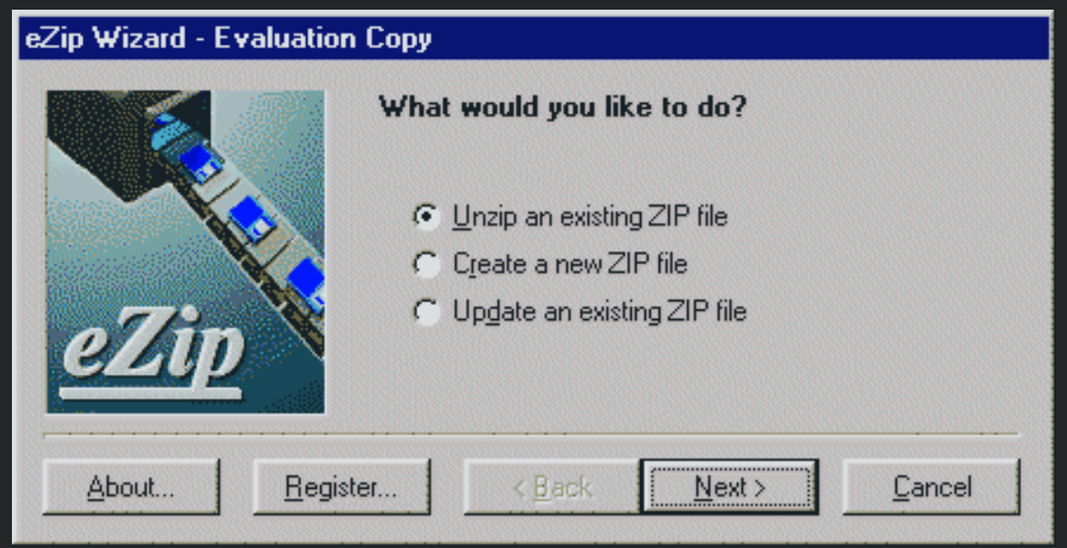
H2: Match Between System & Real World



- Speak the users' language
- Follow real world conventions

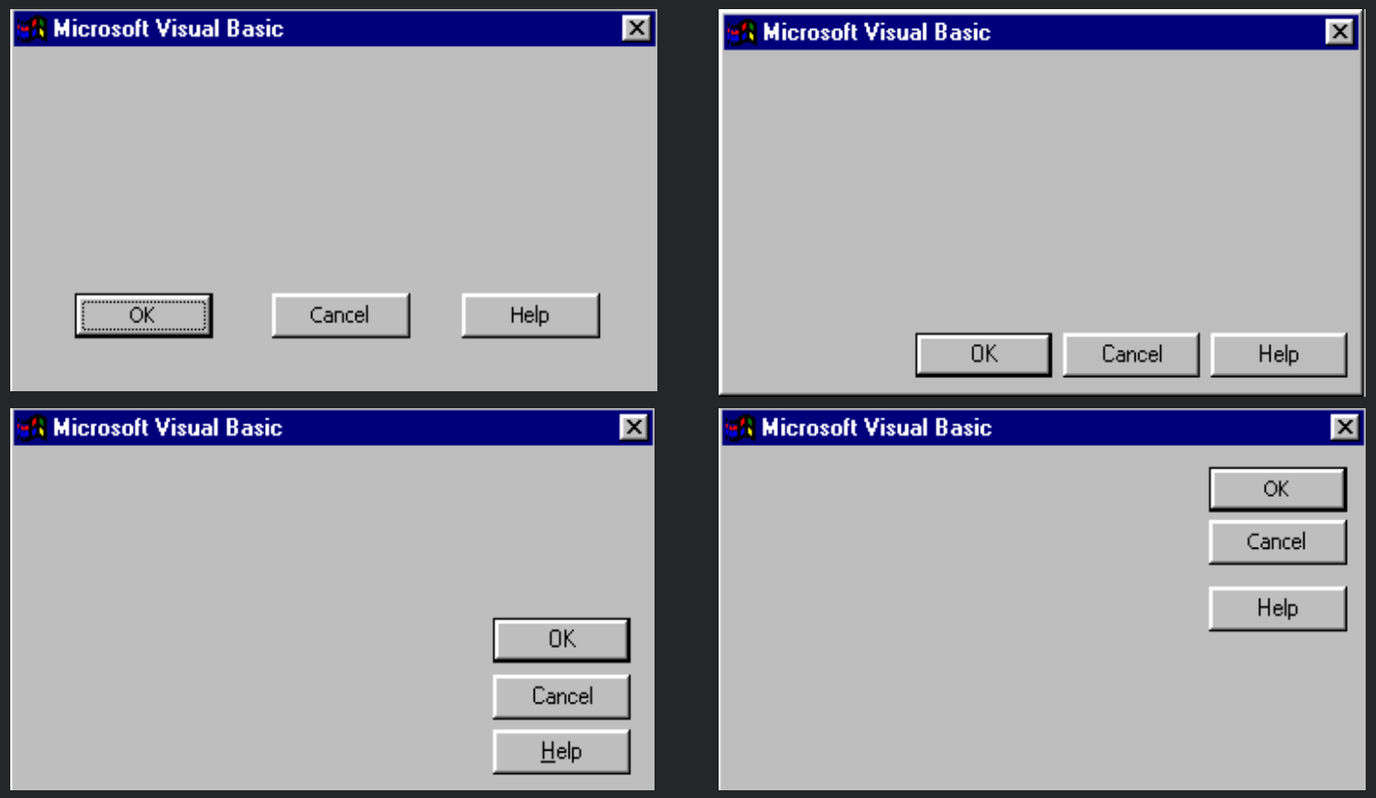
H2: Match Between System & Real World

H3: User Control & Freedom



- “Exits” for mistaken choices, undo, redo
- Don’t force down fixed paths

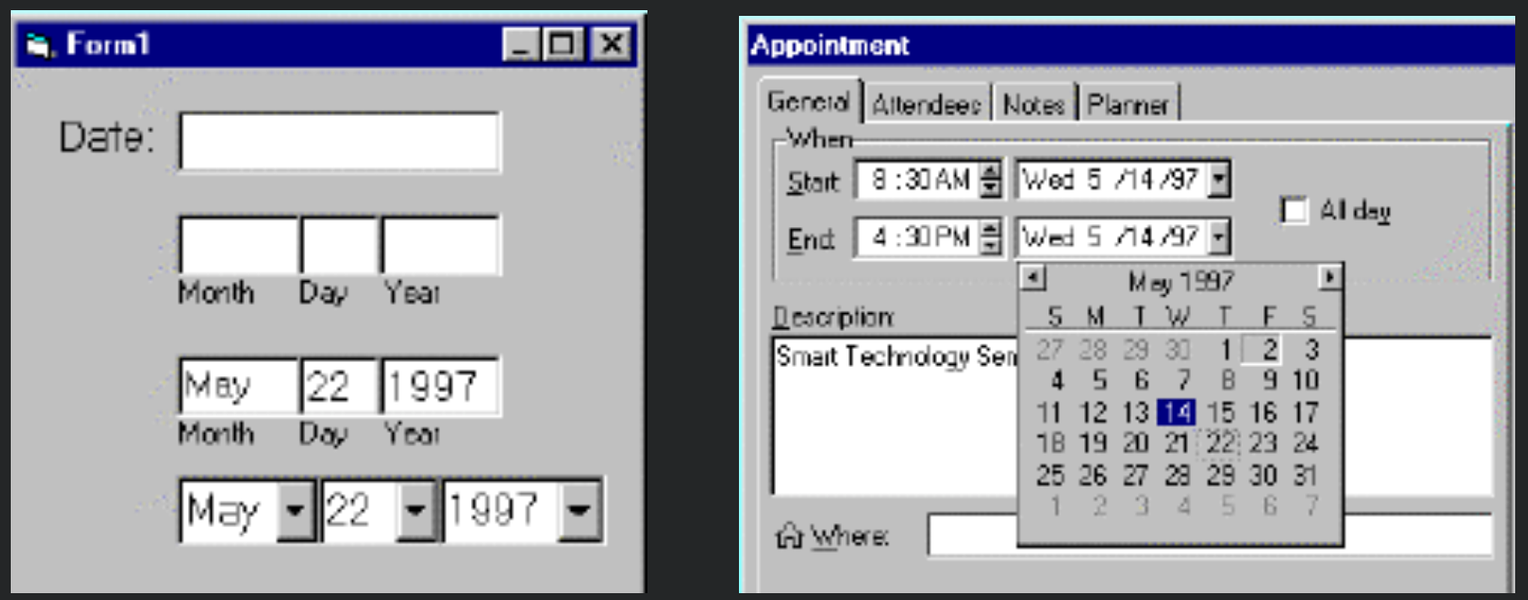
H4: Consistency & Standards



- Same words, situations, actions, should mean the same thing in ***similar*** situations; same things look the same, be located in the same place.
- Different things should be different

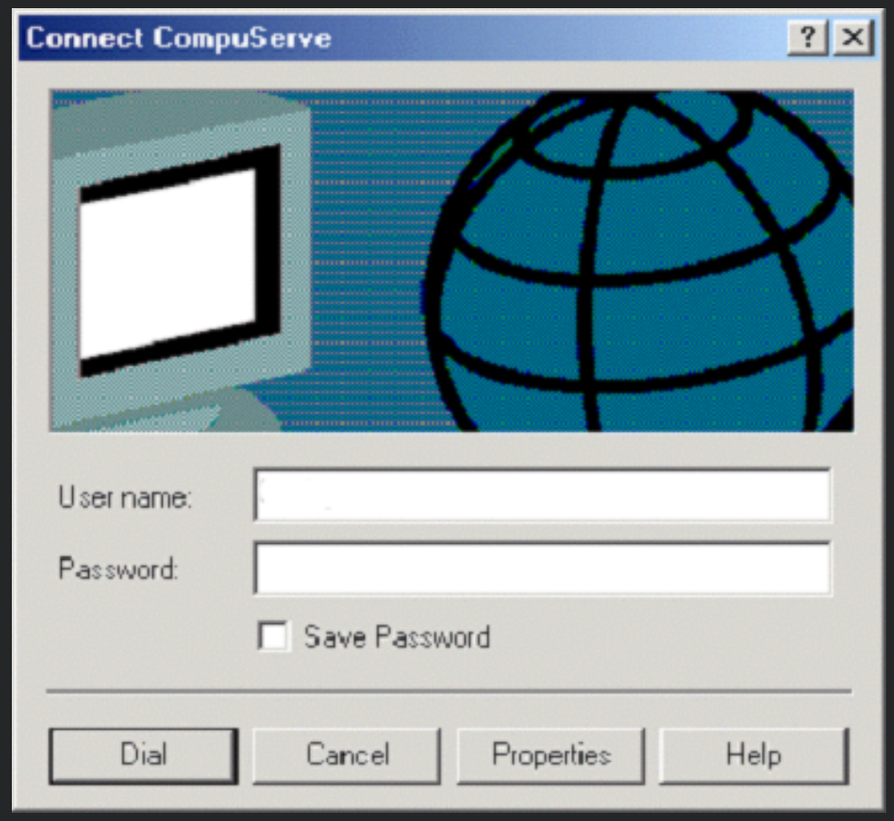
H4: Consistency & Standards

H5: Error Prevention



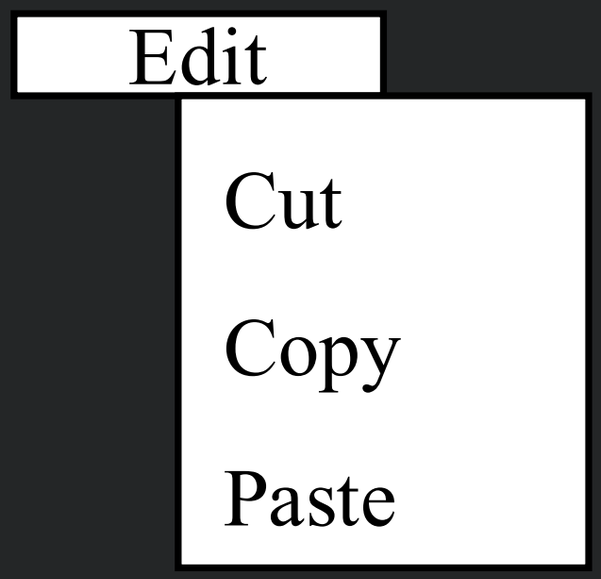
- Careful design which prevents a problem from occurring in the first place

H6: Recognition Not Recall



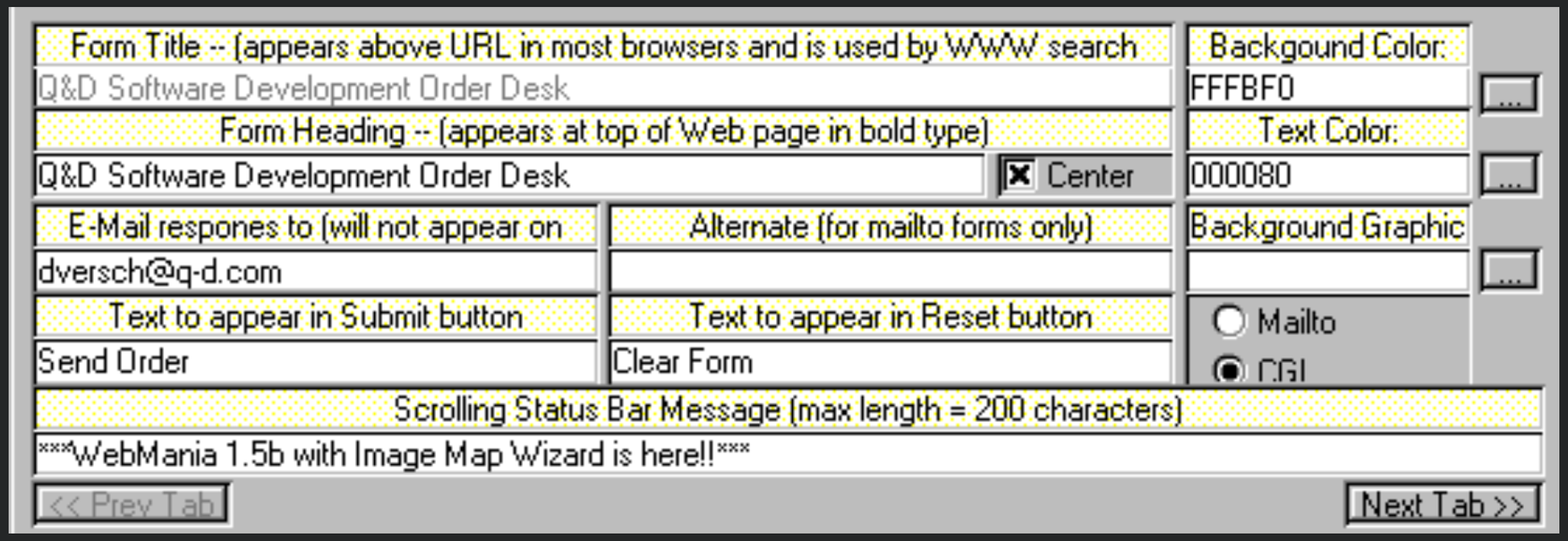
- Make objects, actions and options visible or easily retrievable

H7: Flexibility & Efficiency of Use



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

H8: Aesthetic & Minimalist Design



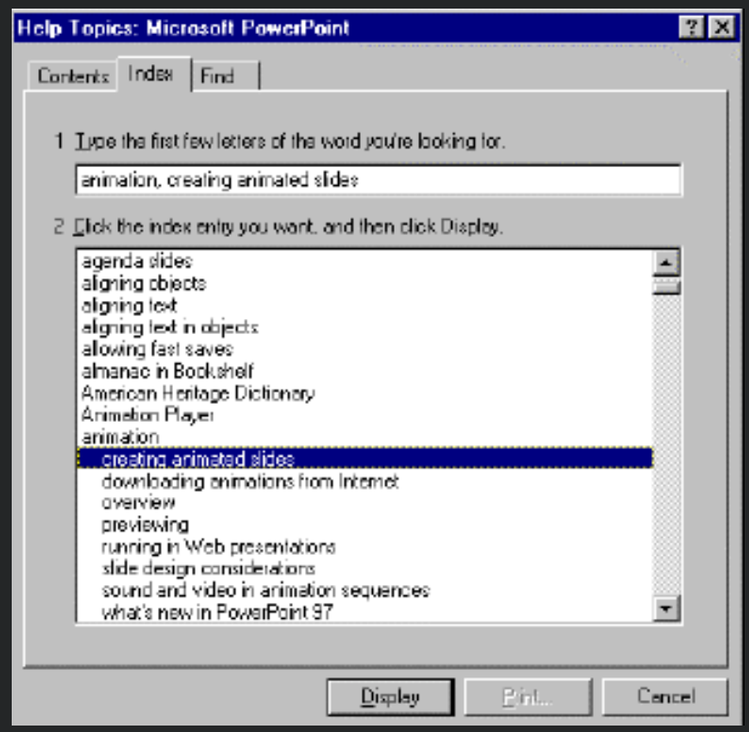
- Interfaces should not contain irrelevant or rarely needed information

H9: RDR from Errors

Help Users *Recognize*, *Diagnose*, and *Recover* from Errors

- Error messages in language user will understand
- Precisely indicate the problem
- Constructively suggest a solution

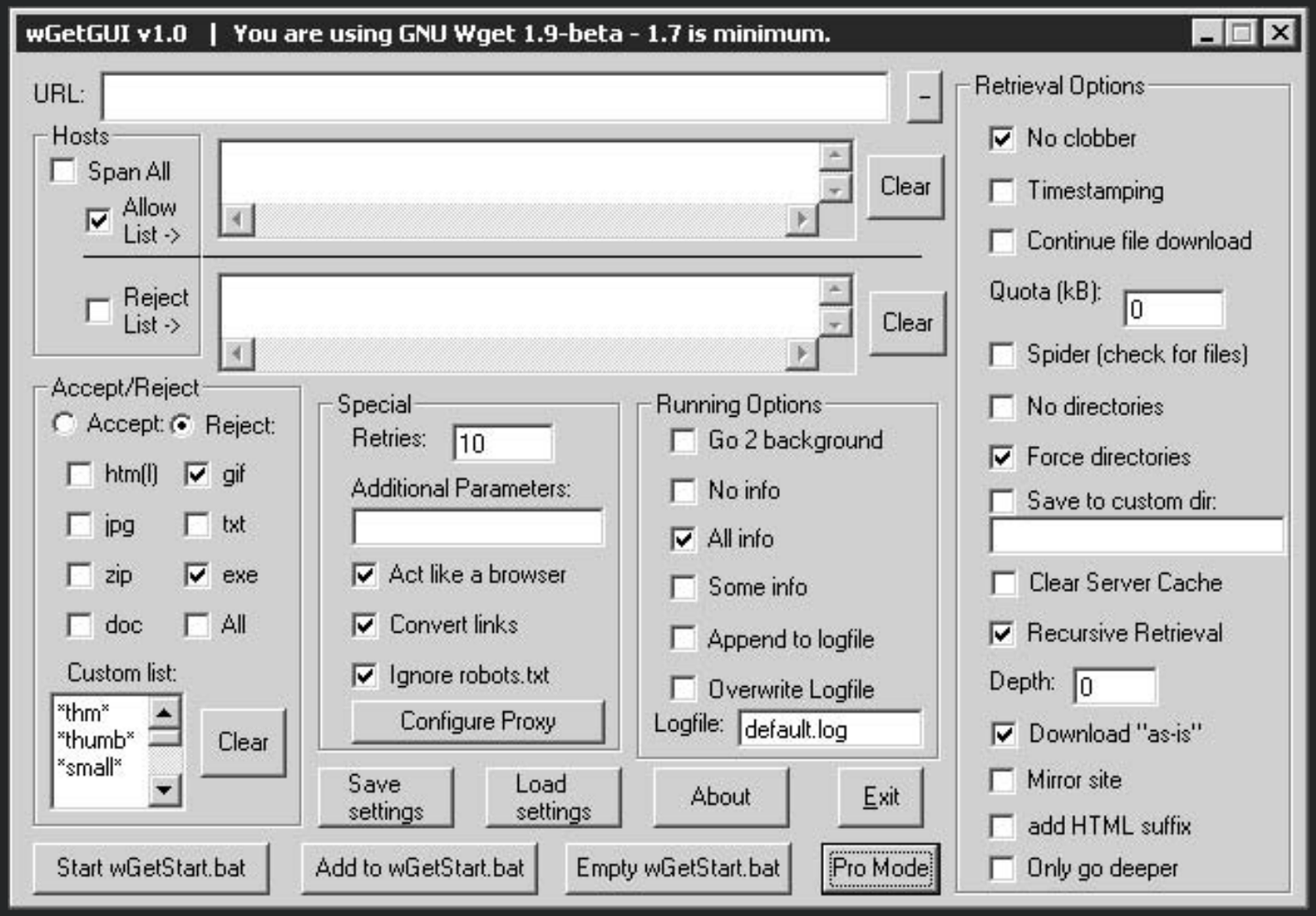
H10: Help & Documentation



- Easy to search
- Focused on the user's task
- List concrete steps to carry out
- Always available

Example

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition vs. recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation



Advantages of Heuristic Evaluation

- “Discount usability engineering” - Intimidation low
- Don’t need to identify tasks, activities
- Can identify some fairly obvious fixes
- Can expose problems user testing doesn’t expose
- Provides a language for justifying usability recommendations

Disadvantages of Heuristic Evaluation

- Un-validated
- Do not employ real users
- Can be error prone
- Better to use usability experts
- Problems unconnected with tasks
- Heuristics may be hard to apply to new technology

Using Heuristic Evaluation

- Can be used informally to identify issues in a website
- Can be used as a more formal usability inspection method
- Evaluators each first separately identify issues
- Issues then combined from each evaluator

Ways to Use Heuristic Evaluation

- Early in design process to catch major issues
- When time or resources are not available for empirical usability evaluation

In-Class Activity

- Form a group with 2-4 students
- Together select an application or website (e.g., Word, Twitter)
- Work individually to identify at least 1 usability issue
- For each issue, identify the heuristic, identify the functionality in the application, and summarize how the heuristic is violated in a few sentences
- Use Online Google Document shared on Ed

Acknowledgements

- Slides adapted from Dr. Thomas Latoza's SWE 432 course