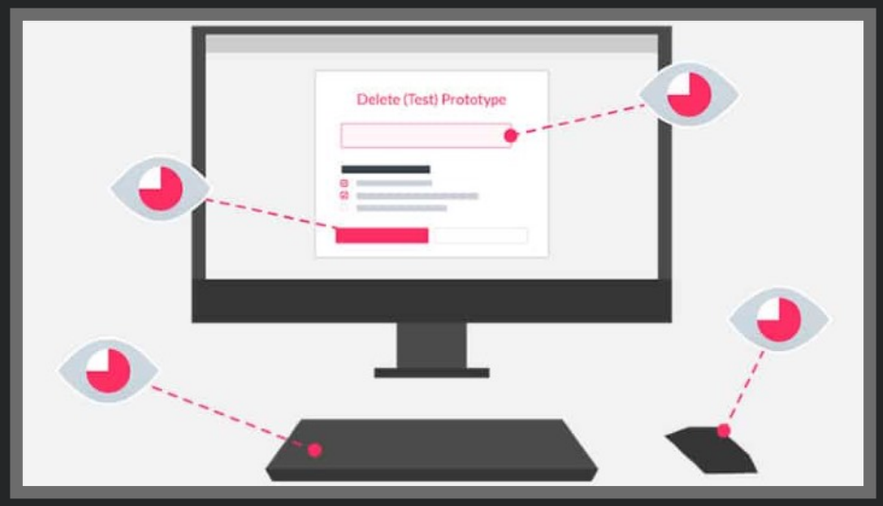
SWE 632 - Design & Development of User Interfaces

Fall 2020

George Mason University

Dr. Kevin Moran

Week 1: Course Overview & Heuristic Evaluation



Welcome to SWE 632!

- **<u>Initial Logistics:</u>**
  - Welcome to the Lecture!
  - This Lecture is being recorded
  - During Lectures please ***<u>keep your microphone muted</u>***, you can unmute in breakout sessions
  - You can keep your video on (let's see how bandwidth is)
  - Feel free to ask questions in the chat! David and I will monitor and respond

Introductions

Instructor: Kevin Moran

Education: Ph.D. from William & Mary - 2018

Research Interests: Software Engineering , UI Analysis, Machine Learning

Office Hours: Thursdays, 4:00pm-5:00pm, or by appointment

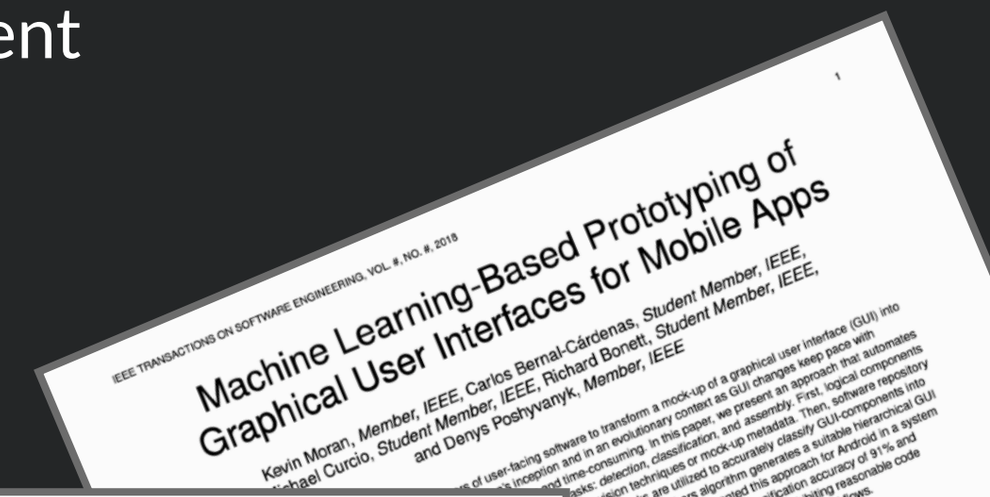
Introductions

Instructor: Kevin Moran

Education: Ph.D. from William & Mary - 2018

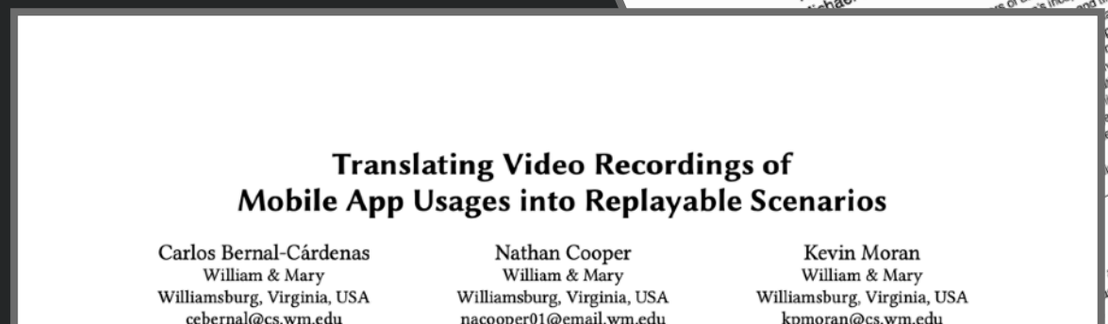
Research Interests: Software Engineering , UI Analysis, Machine Learning

Office Hours: Thursdays, 4:00pm-5:00pm, or by appointment


IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. #, NO. #, 2018

Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps

Kevin Moran, *Member, IEEE*, Carlos Bernal-Cárdenas, *Student Member, IEEE*, Michael Curcio, *Student Member, IEEE*, Richard Bonett, *Student Member, IEEE*, and Denys Poshyvanyk, *Member, IEEE*


Translating Video Recordings of Mobile App Usages into Replayable Scenarios

Carlos Bernal-Cárdenas, William & Mary, Williamsburg, Virginia, USA, cebernal@cs.wm.edu

Nathan Cooper, William & Mary, Williamsburg, Virginia, USA, nacooper01@email.wm.edu

Kevin Moran, William & Mary, Williamsburg, Virginia, USA, kpmoran@cs.wm.edu

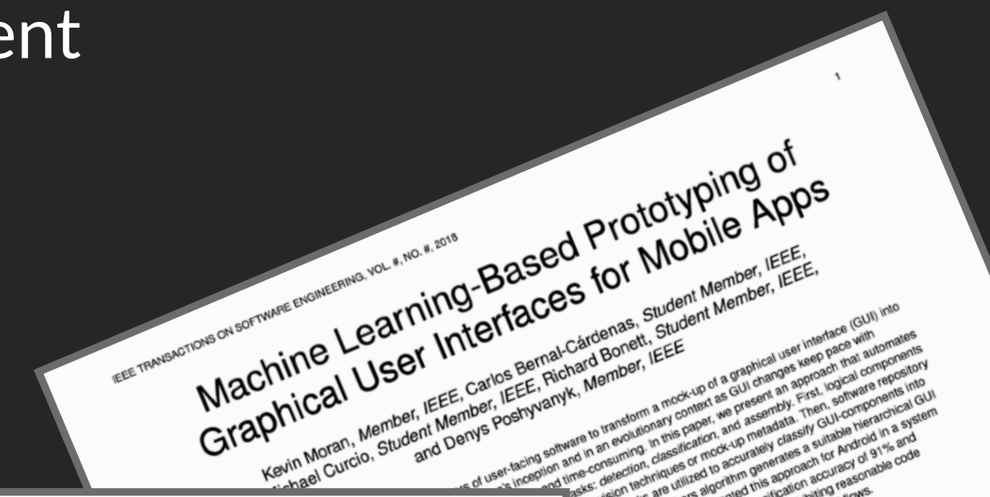
Introductions

Instructor: Kevin Moran

Education: Ph.D. from William & Mary - 2018

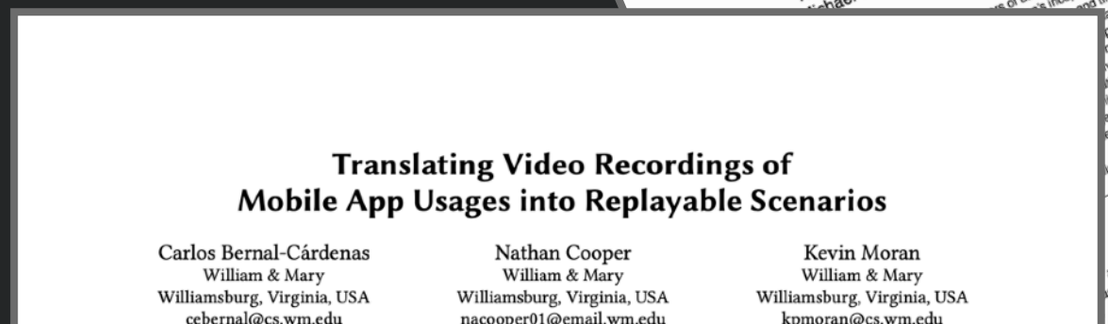
Research Interests: Software Engineering , UI Analysis, Machine Learning

Office Hours: Thursdays, 4:00pm-5:00pm, or by appointment


IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. #, NO. #, 2018

Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps

Kevin Moran, *Member, IEEE*, Carlos Bernal-Cárdenas, *Student Member, IEEE*, Michael Curcio, *Student Member, IEEE*, Richard Bonett, *Student Member, IEEE*, and Denys Poshyvanyk, *Member, IEEE*


Translating Video Recordings of Mobile App Usages into Replayable Scenarios

Carlos Bernal-Cárdenas, William & Mary, Williamsburg, Virginia, USA, cebernal@cs.wm.edu

Nathan Cooper, William & Mary, Williamsburg, Virginia, USA, nacooper01@email.wm.edu

Kevin Moran, William & Mary, Williamsburg, Virginia, USA, kpmoran@cs.wm.edu

Introductions

Teaching Assistant: David Samudio

Education: Current Ph.D. Student at GMU

Research Interests: Creating human-centered support tools for developers

Office Hours: TBA

Today's Agenda

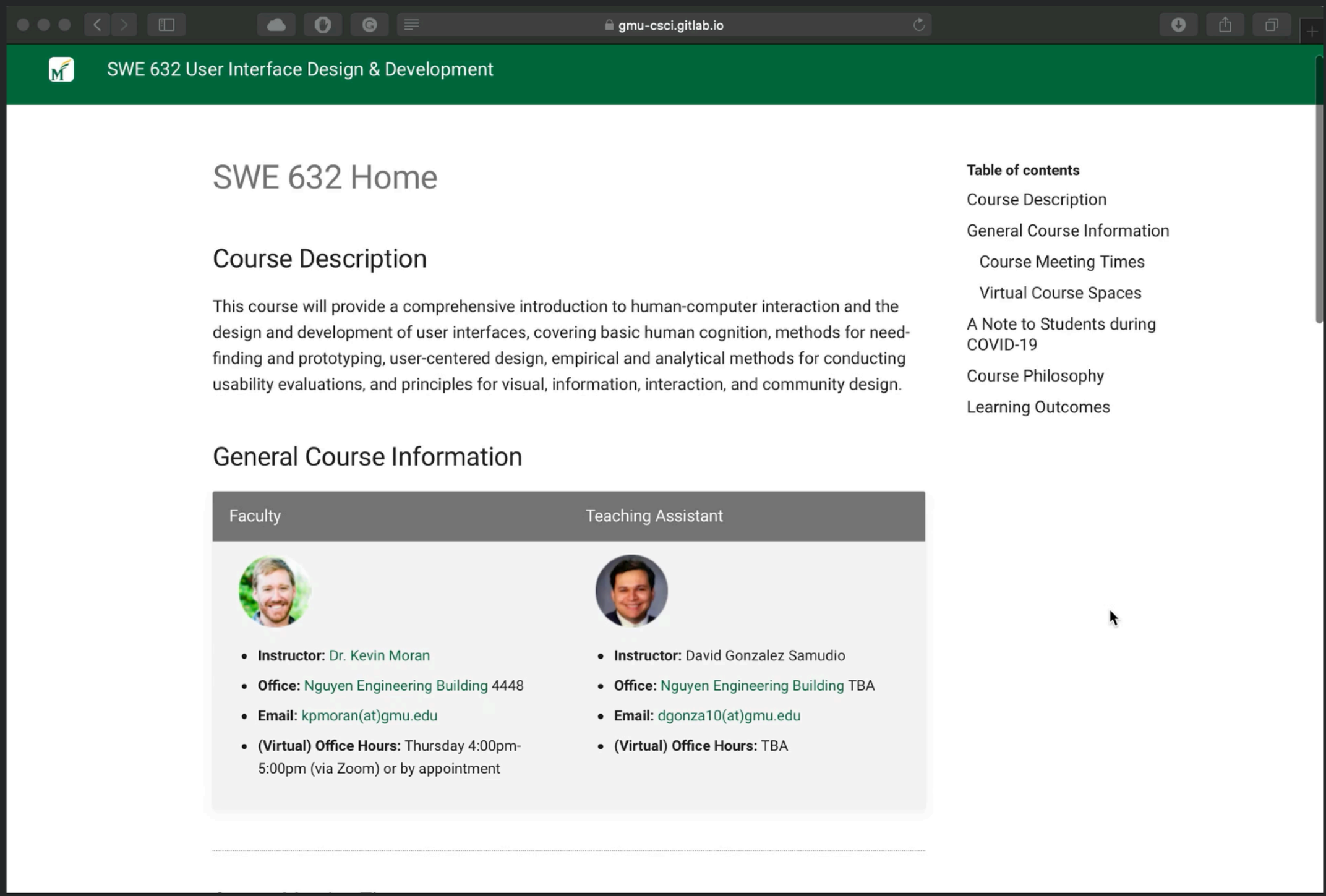
1. Provide an overview of the ***Course Logistics*** - (15-20 mins)
2. Discuss the ***Philosophy*** & ***Purpose*** of the Course -(20 mins)
3. Discuss ***Heuristic Evaluations*** of User Interfaces - (20 mins)
4. ***Group Activity*** applying Heuristic Evaluations - (20 mins)
5. Google Firebase ***Tech Talk*** - (15mins)

Course Logistics

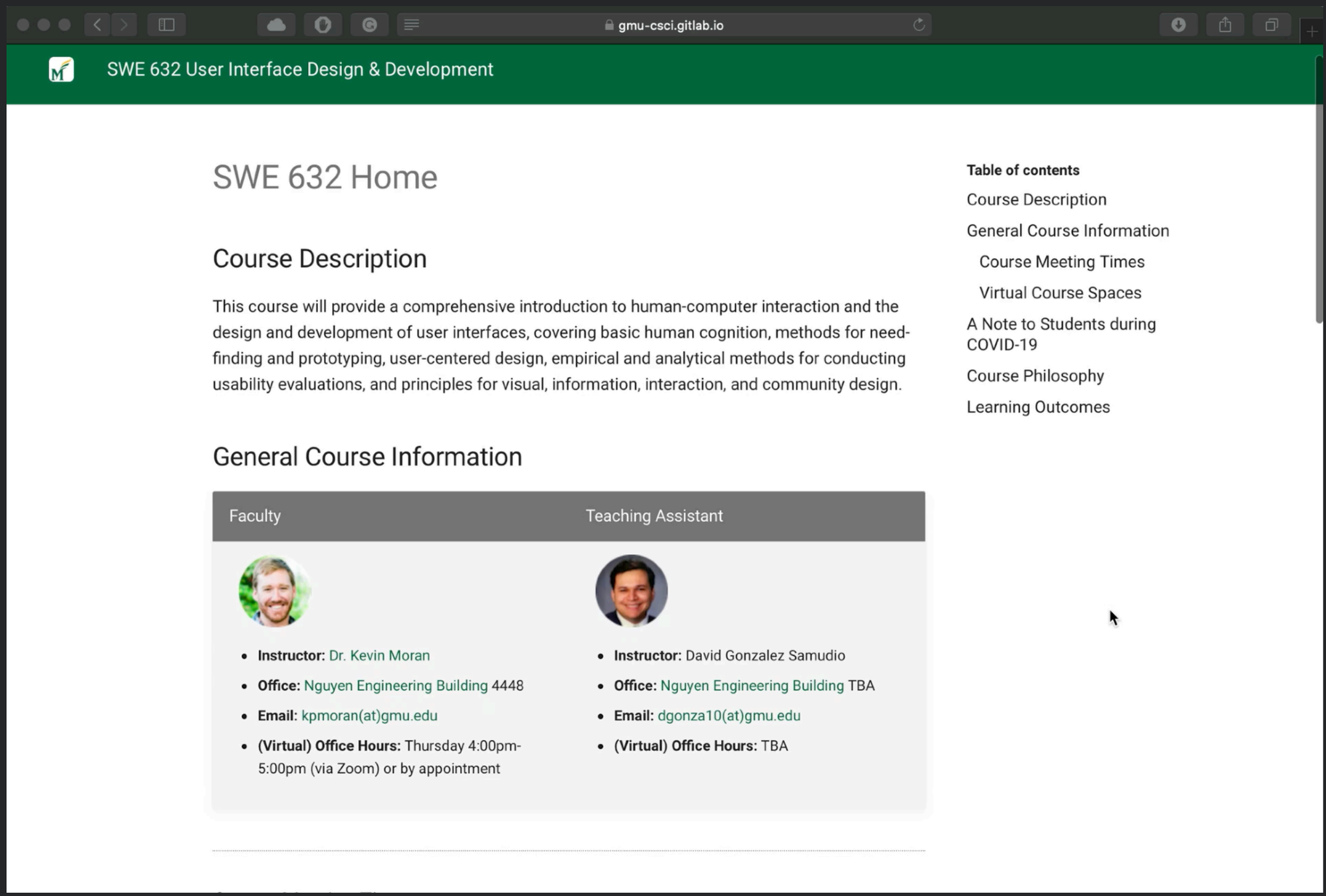
Course Resources

- Course Website: Syllabus, Schedule, Assignments, Lecture slides/recordings
- Piazza: Announcements, Discussions
- Blackboard (MyMason): Grades
- Zoom: Class Meetings & Office Hours

CourseWebsite



CourseWebsite



Grading Breakdown

- Participant & Discussion - (10%)
- Tech Talk - (10%)
- Project Checkpoints & Presentation - (40%)
- Mid-Term Exam - (20%)
- Final Exam - (20%)

Participation & Discussion

- Weekly discussion questions assigned after class
- You must post your response by *Friday @ Midnight*
- You can either create your own response, or reply to another students
  - If you respond to another student, you must advance the conversation
- We expect these response to be *1 or 2 paragraphs*

In-Class Activities

- Work together in small groups to gain experience trying out methods and concepts with examples
- No grades (*pandemic*), but very important, as you will learn a lot from your classmates during these exercises

Tech Talks

- 15 minute overview of a front-end web technology
- Groups of 3 (collaborate w/ Zoom, Slack, etc…)
- Use piazza to find a partner, reserve topics
    - Dedicated thread will be posted after class
- Only 1 group can cover a technology
- *Signup by start of class next Tues. (Sept 1st)*

Course Project

- Build a (*really simple*) web app
- Use usability concepts and methods to identify usability issues
- Iteratively improve your app to address usability issues
- There will be 8 “Project Checkpoints” throughout the semester - culminating in a project presentation

Project Checkpoint 0

- Due next Tues. before class (Sept 1)
- Form a group of 1, 2, or 3
- Pick an app to build
- Describe what you propose to build in ~1 page

Policy on Code Reuse

- Can borrow code from online sources as much or as little as you'd like
- *You must document instances of code that you reuse*

Late Policy - Project Checkpoints

- HWs will often involve *peer evaluations*
- Can submit up to:
  - 24 hours late, lose 10%
  - 48 hours late, lose 20%
- HW submissions more than 48 hrs late will receive a 0
- ***These are difficult times, if you have unforeseen problems, please contact me & David before the deadline!***

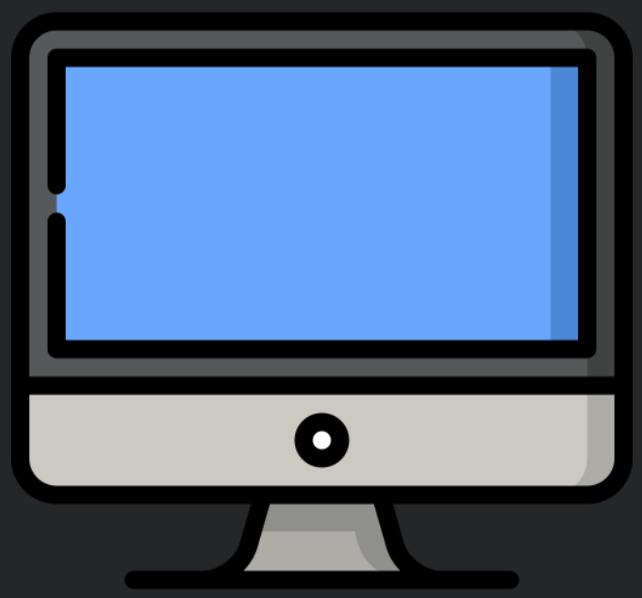
Exams

- Midterm & Comprehensive Final Exam
- Includes both in class lectures and material from assigned readings
- Synthesis-style, short essay questions
- Open Everything - *but must cite sources*
- Exams will be released on Piazza, you will have 3 hours to complete

Course Philosophy & Purpose

Software is Ubiquitous

Software is Ubiquitous



“Good Design” is incredibly important

"Good Design" is incredibly important

… and is centered on ***usability***

What is Usability?

What is Usability?

Ease of Use

Productivity

Learnability

Efficiency

Retainability

User Satisfaction

Effectiveness

Usable or Unusable?

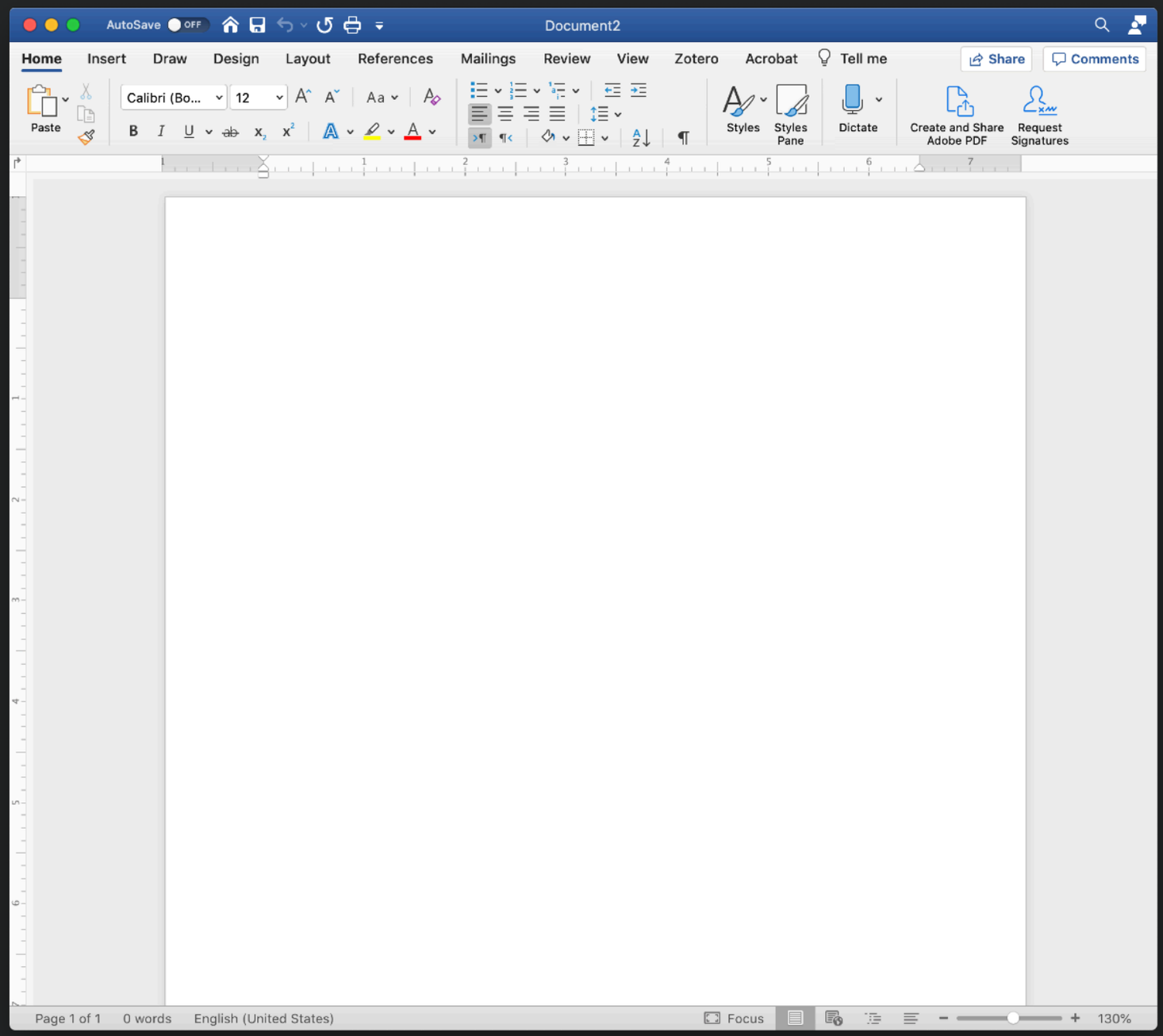
A Teapot

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

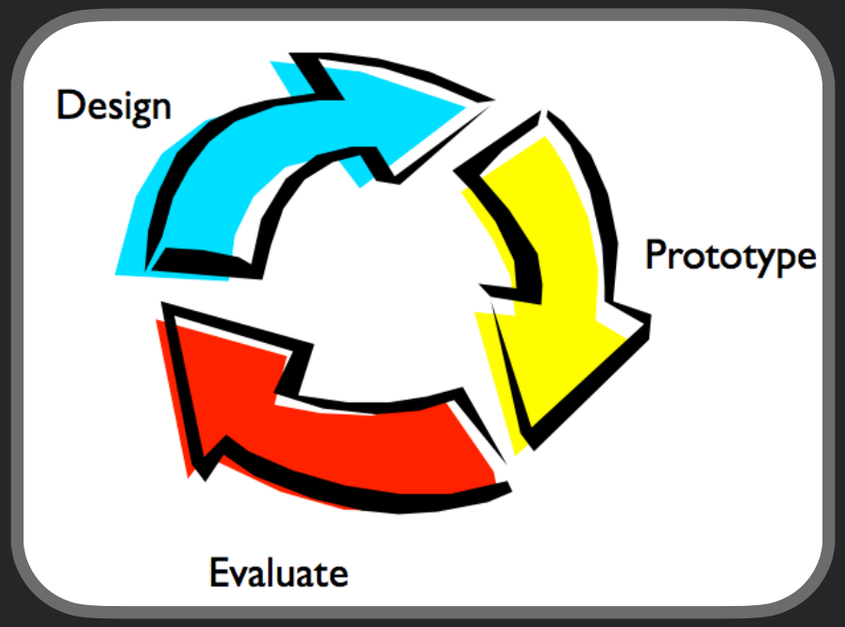
- Given an existing artifact and humans doing a set of tasks, determine ***goals*** and identify ***usability issues*** that decrease task performance

User-Centered Design

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

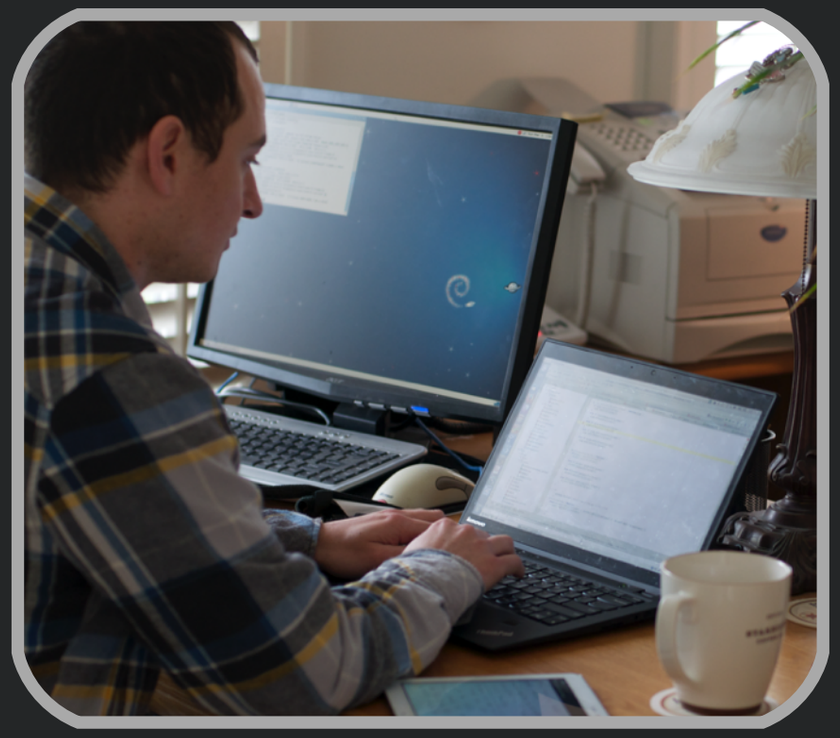
Iterative User-Centered Design

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



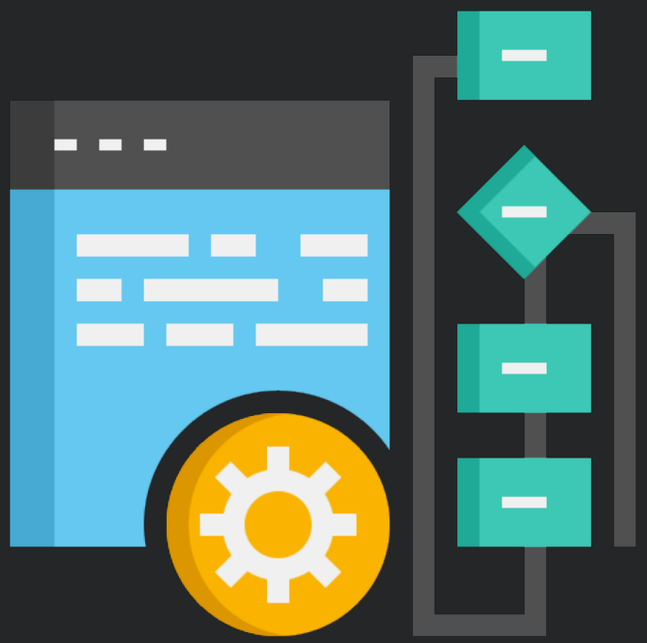
Empirical: Usability **Study**

- 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"***

What Usability is NOT

- Not “dummy proofing”
- Not being “user friendly”
- Not just “usability testing”
- Not just making software pretty

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?

Adapted from Maneesh Agrawala & Bjoern Hartmann

Why Study Usability?

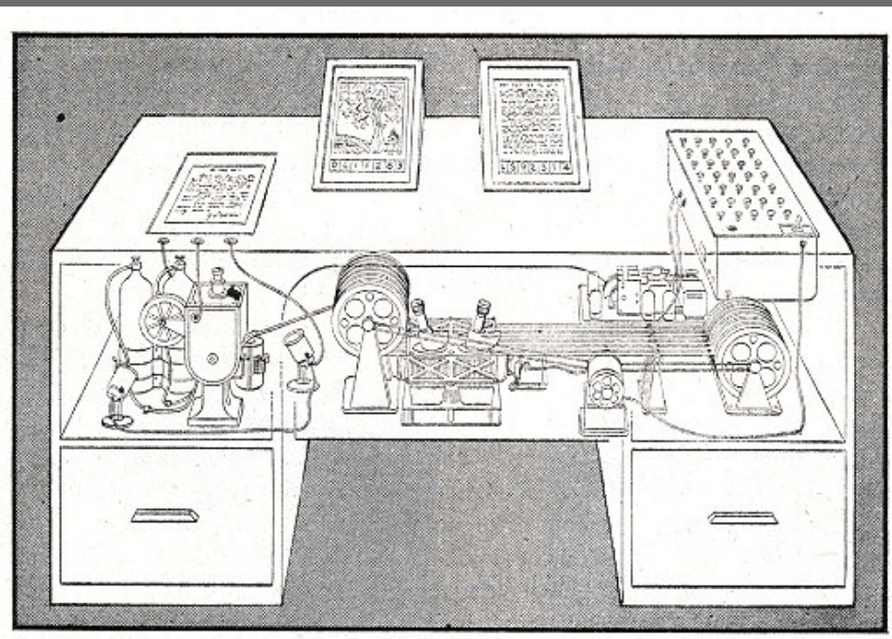
Life-Threatening Errors

- 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*

The Promise of Tech

- Vannevar Bush, The Atlantic, July 1945

- Described the Memex and predicted hypertext, personal computers, the Internet, the WWW, speech recognition, online encyclopedias



MEMEX in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference.

Human-Computer Interaction

"A discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them."

- ACM SIGCHI Curriculum Development Group Report, 1992

This Course

- Comprehensive introduction to usability and human-computer interaction (HCI)
- Basic cognition, user-centered design, usability evaluations, principles for UI design

Heuristic Evaluation

Heuristic Evaluation

- "***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

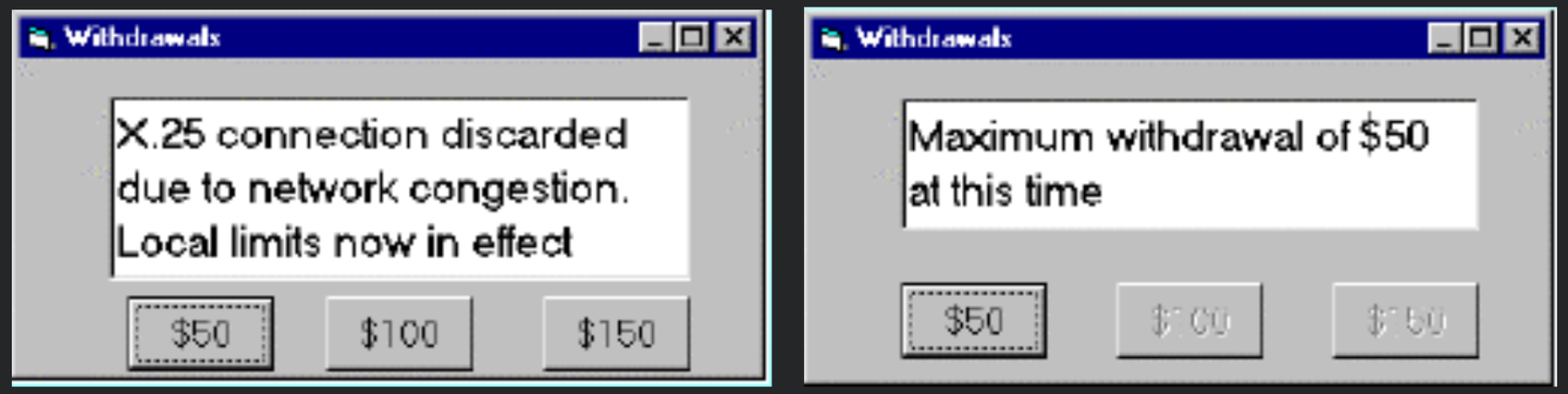
Heuristic Evaluation

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*)

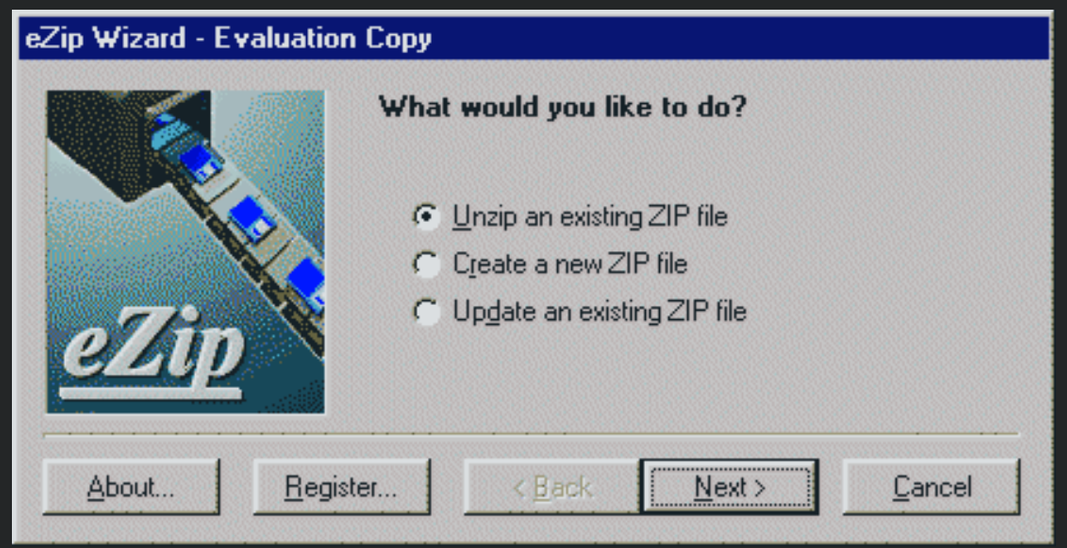
H2: Match Between System & Real World



- Speak the users' language
- Follow real world conventions

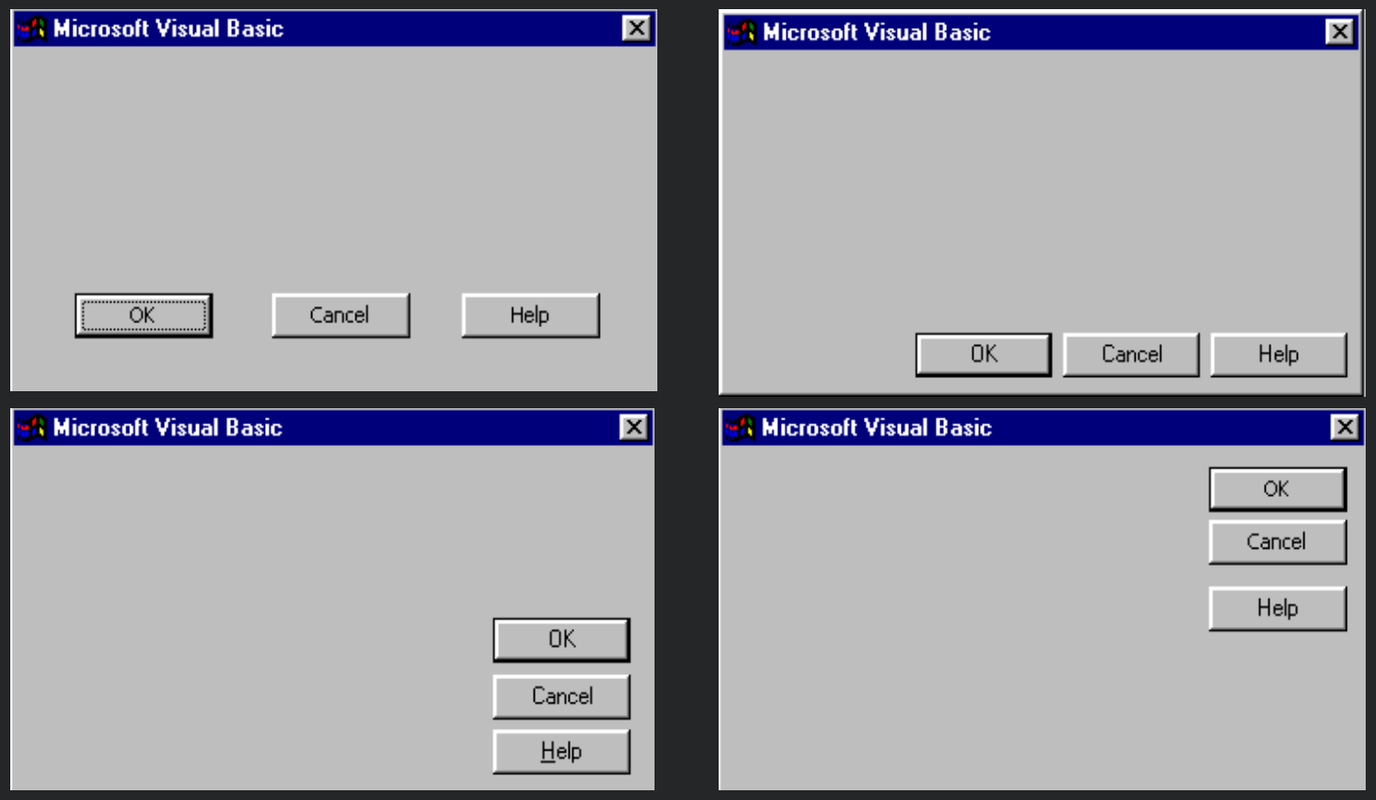
H2: Match Between System & Real World

H3: User Control & Freedom



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

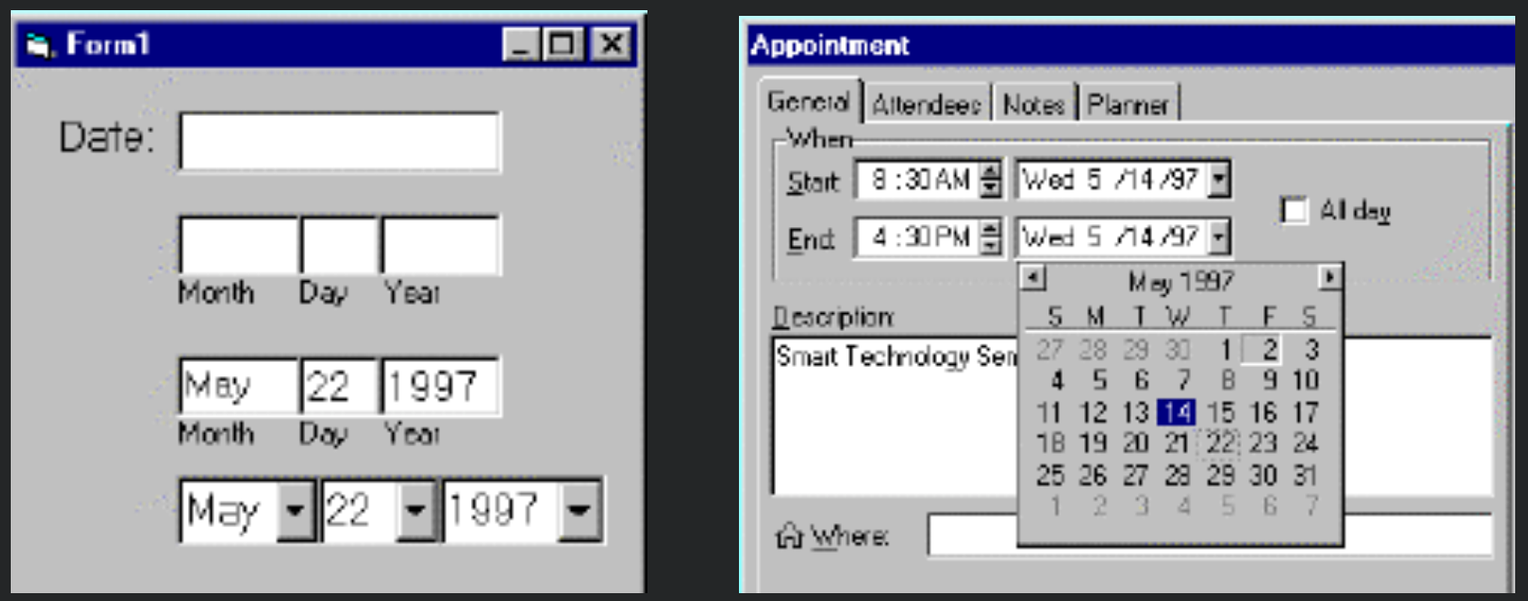
H4: Consistency & Standards



- 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

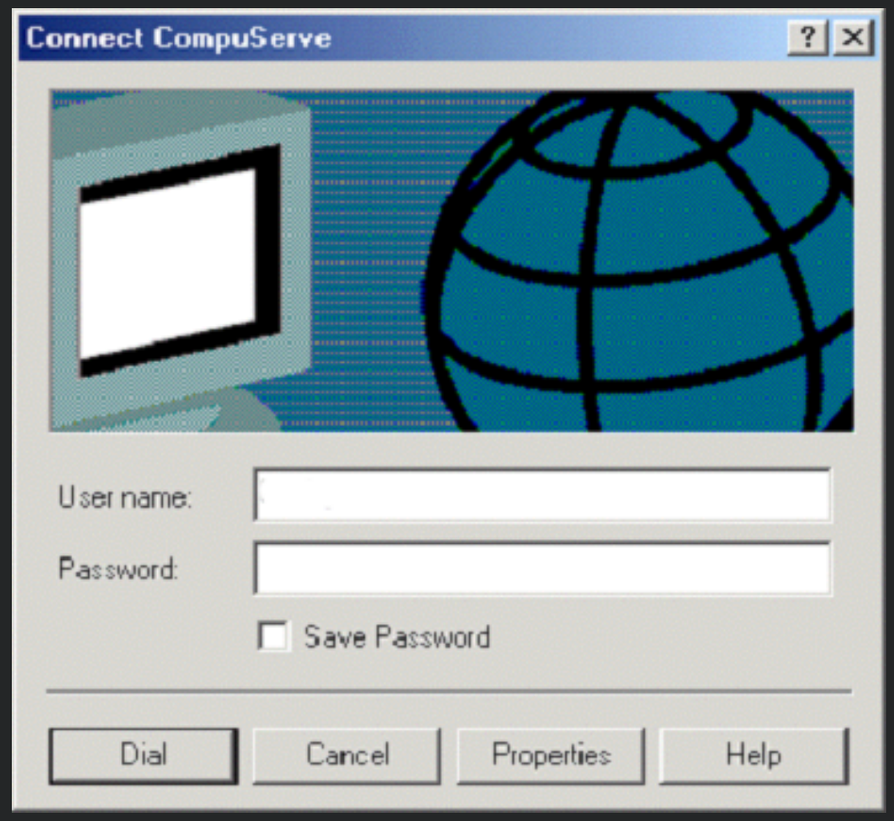
H4: Consistency & Standards

H5: Error Prevention



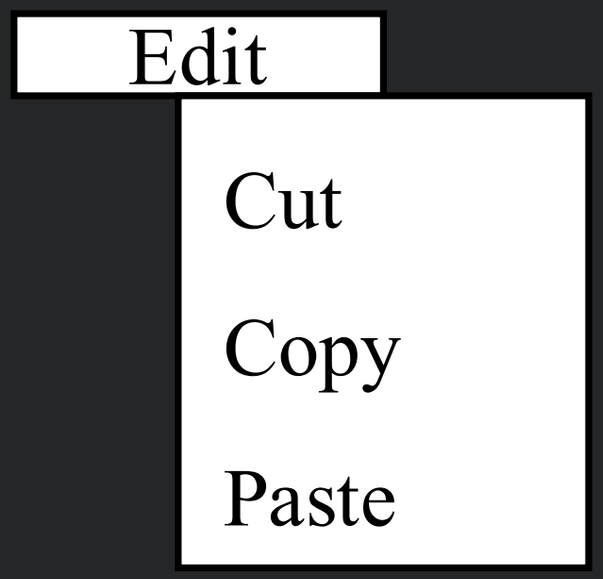
- Careful design which prevents a problem from occurring in the first place

H6: Recognition Not Recall



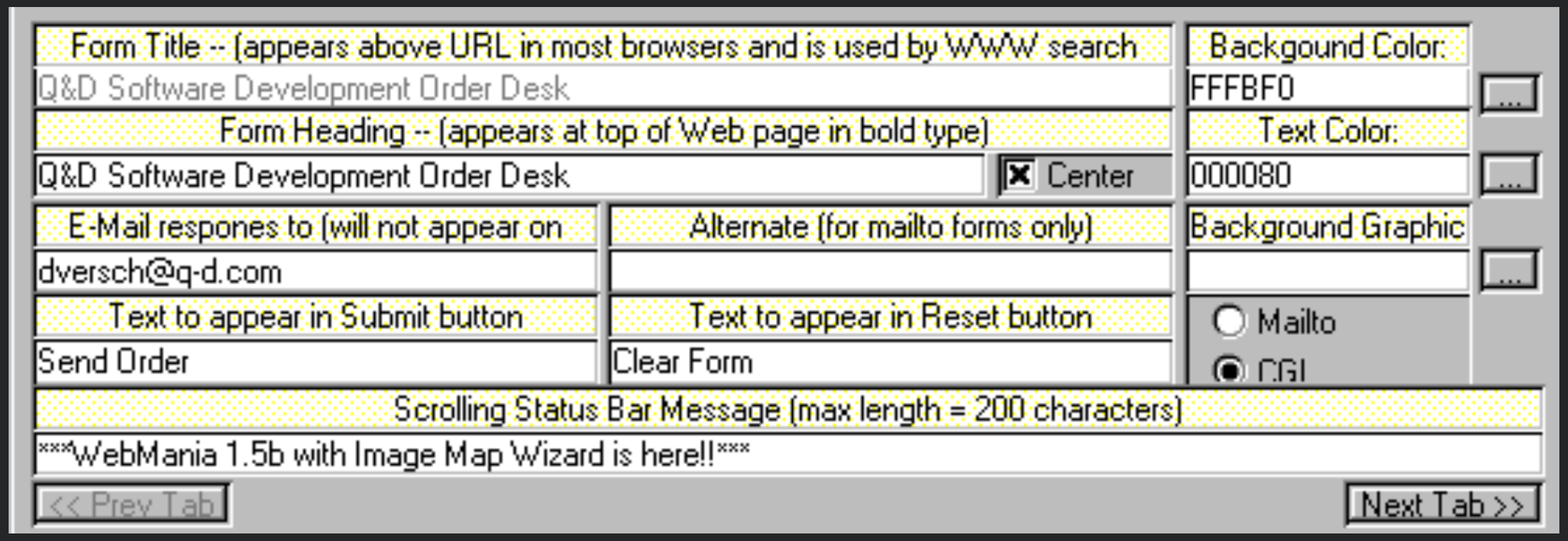
- Make objects, actions and options visible or easily retrievable

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



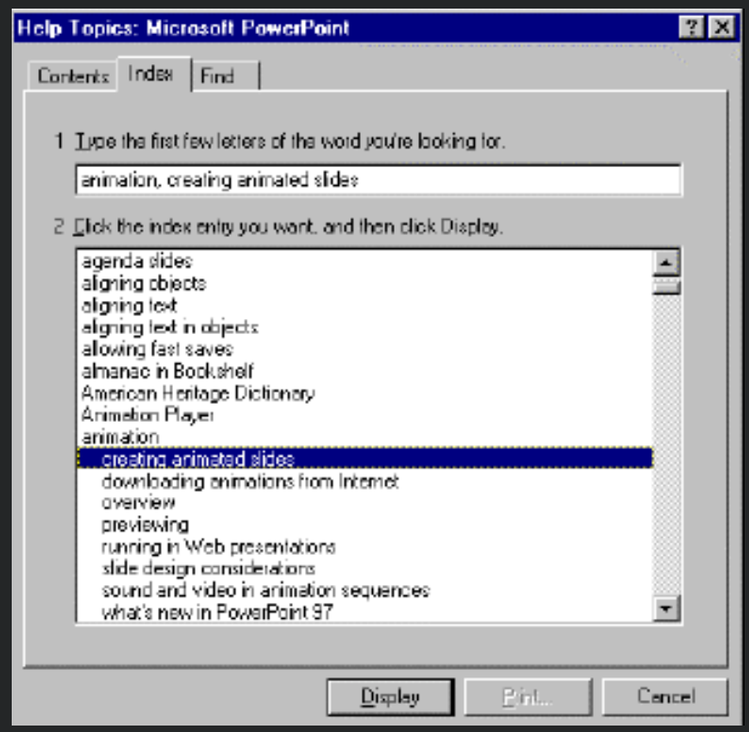
- Interfaces should not contain irrelevant or rarely needed information

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

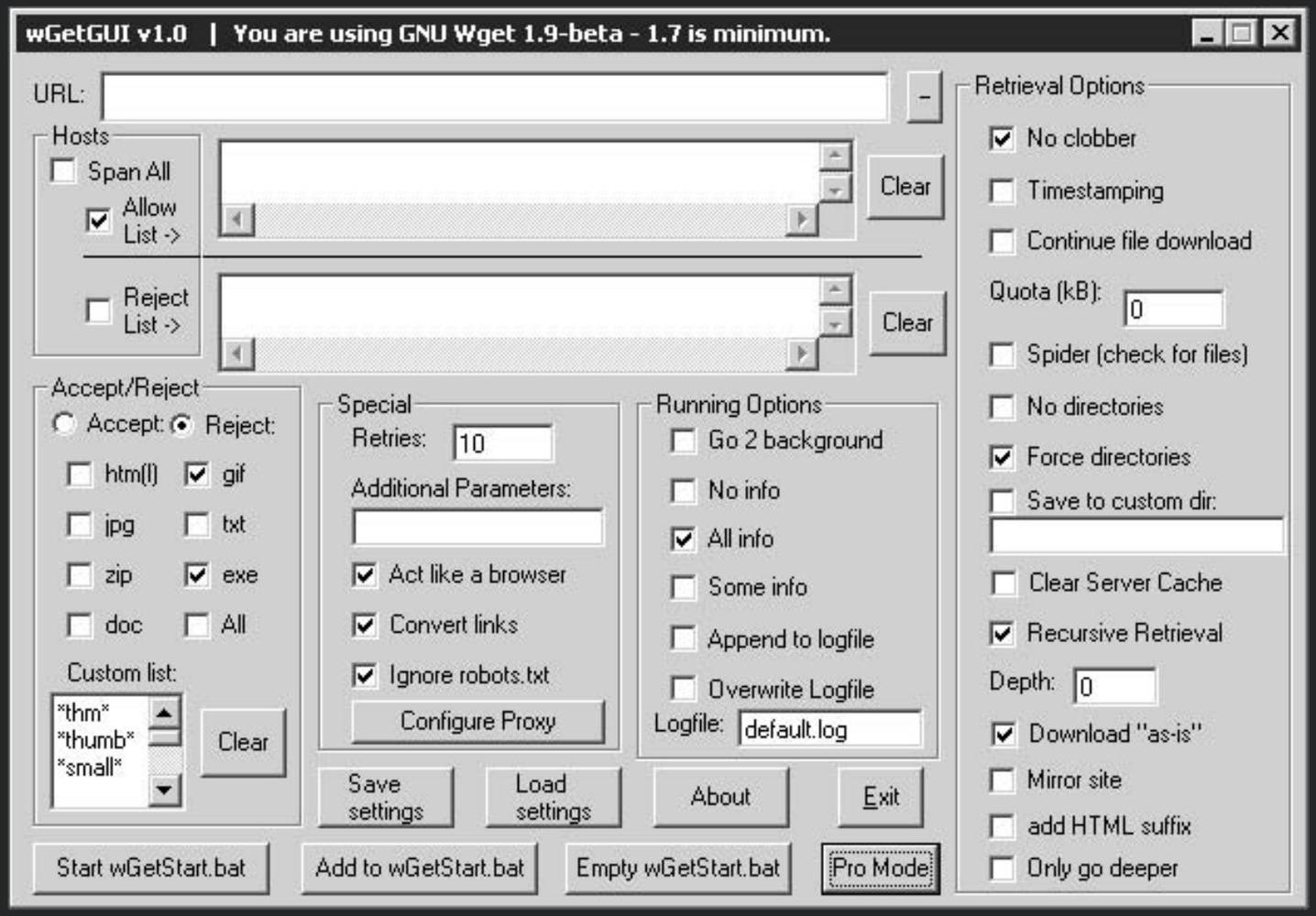
H10: Help & Documentation



- 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

- Breakout Rooms with 3-4 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 Word Documents shared in Zoom chat

Acknowledgements

- Slides adapted from Dr. Thomas Latoza's SWE 632 course