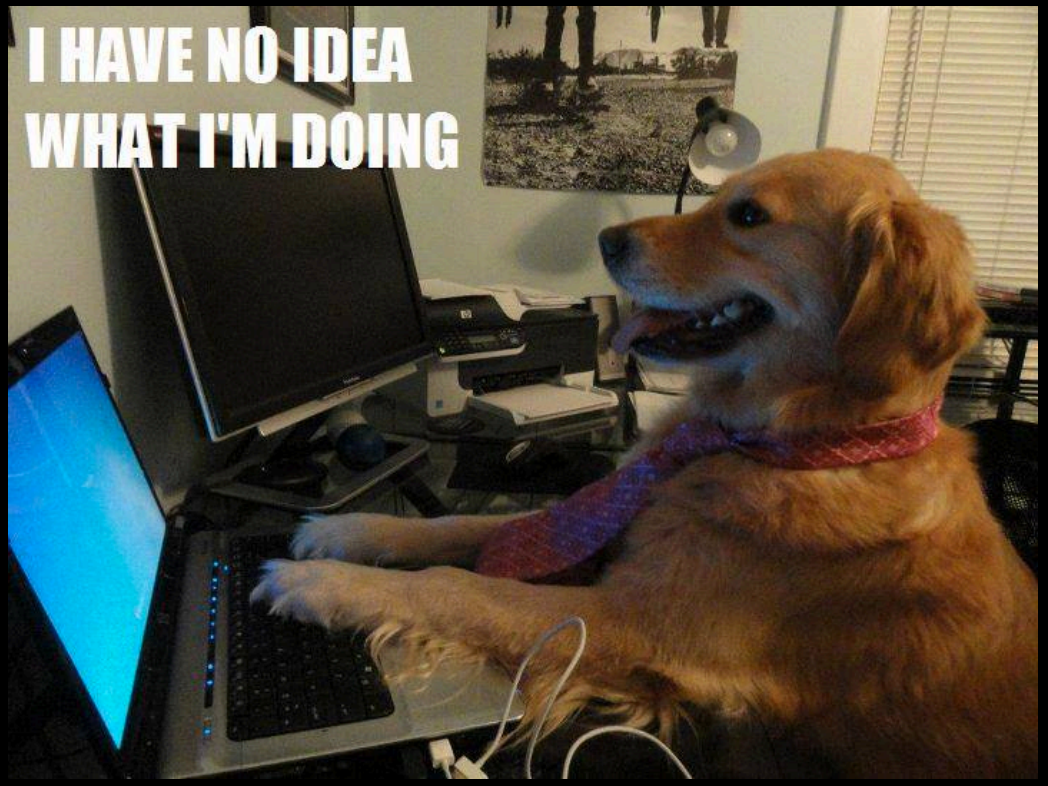


Interactive prototypes & analyzing study data

CSCI 3002 Fall 2018

Today

- Finishing up paper prototype testing
- Making interactive prototypes
- Analyzing findings

Documenting our prototypes

What to do with paper prototypes?

- Keeping them on paper is great for testing
  - Easy to make, alter, etc.
- But paper is not very good for keeping a record of your prototype
  - Still need a human “computer” to make it work
- Alternatives?

Documenting our prototype

- Video
  - we'll do this later, still some set up required
- Write code
  - works but we lose the interactivity

- Create a "mid-fidelity" prototype
  - Convert our sketches into something interactive and shareable

Goals of mid-fi prototyping

- Refine and standardize the UI design and components
- Test out on a realistic device at the correct size
- Integrate gestures and other interactions as appropriate

- (We could focus more on visual design here, but there are other things we need to do here also)

Mid-fi prototyping tools

- PowerPoint (with clickable links)
- HTML and CSS
- Many programs
  - Sketch, Axure, Balsamiq, Marvel, POP

- We'll use an app called figma.com
  - Free plan for students, easy to share, easy to deploy to mobile devices

Figma demo

Questions about mid-fi prototyping?

Analyzing data

Data

- So we’ve run a usability test
- Now what?

Outputs from a user test

- Test user comments
- Video and screen recording
- Observations and notes

- We may want to consider feature suggestions, design suggestions, etc.
- But right now we are especially looking for usability bugs / areas of improvement

Documenting usability problems

- We want to document what happened in our test in a way that we can act on it
- What do we want to know?

Things we want to know

- What is the actual problem?
- Why did it happen? (if we know)
- Contextual factors
- Effects on use of the system
- Severity

Writing up usability issues

- Can use a document called the **Usability Aspect Report**

UAR #:	Problem/Good:	Rated by:
Name:		
Relevant heuristic:		
Steps to reproduce:		
Detailed explanation:		
Possible solution:		
Severity (low, medium, high, critical):	See also:	

<- We'll talk about this next week

Using UARs

- Can keep them in a bug tracking system
- Goal is to represent the problem in a way that's understandable to someone who wasn't there, and actionable

- But, be careful about overfitting
  - Some feedback will contradict. Possible to introduce regression errors.

Making sense of messy data

Dealing with data

- We may have lots of data after a user test
  - Suggestions, usability issues/UARs
- We could just report everything, but that's too much
  - "Participant 1 did X and Y, Participant 2 did Y…"
- We want to be able to generalize from our findings
  - "The major issues with the web site are the confusing login form (6 people had trouble with this), the add friend button (4 people)…"

Qualitative analysis

- Data
  - Field notes
  - Recorded audio (transcribe!)
  - Participant-completed documents (questionnaires etc.)
  - Photographs or video

- **Goal: identify themes and relationships**

- Be sure to remove identifying information; use aliases

Affinity diagramming

1. Enumerate all of the ideas that you have; write each on a Post-It; **one note per idea**
2. Put post-its on the wall
3. Arrange into logical groups
4. Label the groups

Similar to “card sorting”


Walker sucks
Break Minor and Steven Blue from Madison were locked out on March 1st 2011
The password is Money
Support Union Rights!
Honey I'm Home
Public Employees ARE Citizens & Voters
Pull up the moat! The voters are approaching!!!
Strongly support the Unions!
I AM ASHAMED!
Democracy not Democrazy
LET US IN!
I WAS HERE to witness the attempt to kill democracy
This is the 1st time I have been locked out of the Capitol. I live in WI! Let us in!
HEY!!! THE COURT SAID WE CAN COME IN!!
Katie - 28 Newton WI Please let me in!
TAX CORP.S & the Rich
WE ARE WINNING!
THE PEOPLE WHO ARE GOING TO RECALL YOUR ASS
This isn't right. Why are you doing this to the people of WI?
Jim Johnston 53226
Cathy Johnston 53226
SADDENED By your lack of respect to the taxpayers
WE WANT SOLIDARITY
entering the capitol
Atty. Gen. 8.
abridging the
of the press; or the right of the
http://wi.opencapitol.us

Let’s try it

- Pair up with 3-4 neighbors
- I’ll pass around Post-It notes, or use your own paper
- Write at least one issue with Facebook
  (and put your IdentiKey in the lower right hand corner so we know who participated)
- Create notes and lay them out (don’t worry about grouping yet)

Forming clusters

- Spend 5 minutes working with your group to cluster issues

What did you find?

- Hard to concisely define categories
- Categories first?
- Balance positive & negative
- Difficult to re-categorize

Card sorting

- People
- Degrees “Academics”
- Resources
- Administration

- Faculty
- Courses
- Research
- Degree requirements
- Resources
- About us
- Application
- Contact
- Funding
- Prospective students
- Events
- Facilities

Card sorting

- Open – no categories provided
- Closed - you give the categories

What kinds of things are we looking for when affinity diagramming?

- Themes
  - Common problems
  - Problem areas or “pain points”
- User goals
- Card sorting: identifying categories for an application, web site, etc.

Heuristic evaluation

What’s a heuristic?

- Guideline
- Measure
- Hyposthesis about something

Heuristics

- How do you identify a good roommate?
  - Pays rent on time
  - Neat
  - Respects privacy
  - Cleans
  - Doesn’t make you look bad
  - Honest

What's a heuristic?

- "experience-based techniques for problem solving, learning, and discovery that find a solution which is not guaranteed to be optimal, but good enough for a given set of goals." (Wikipedia)

- Examples: (http://c2.com/cgi/wiki?HeuristicRule)
  - Deciding to eat at restaurant B rather than restaurant A only because B has more cars in its parking lot
  - When the level in the fuel tank drops to 1/2 tank or less, always filling up the fuel tank at the very next re-filling station.

Heuristic evaluation

- Use our own (expert) judgment to identify usability problems
- Based on common sense usability rules
- Identify what the problem is and **why** it's a problem
- Then, identify steps to fix it

HE vs. user testing

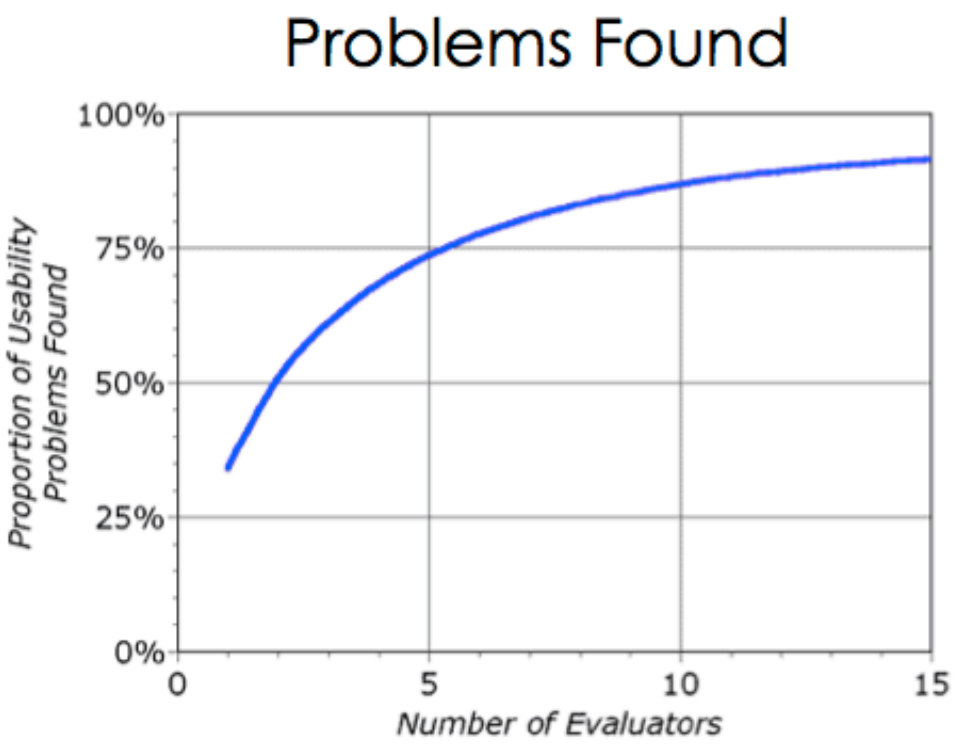
- When we can, we want to test with real users
- HE is a “discount” usability technique
- When it’s useful:
  - When real users are unavailable
  - Very early in the design
  - As a sanity check (but not a replacement for user testing)
  - To back up / explain problems you see in testing

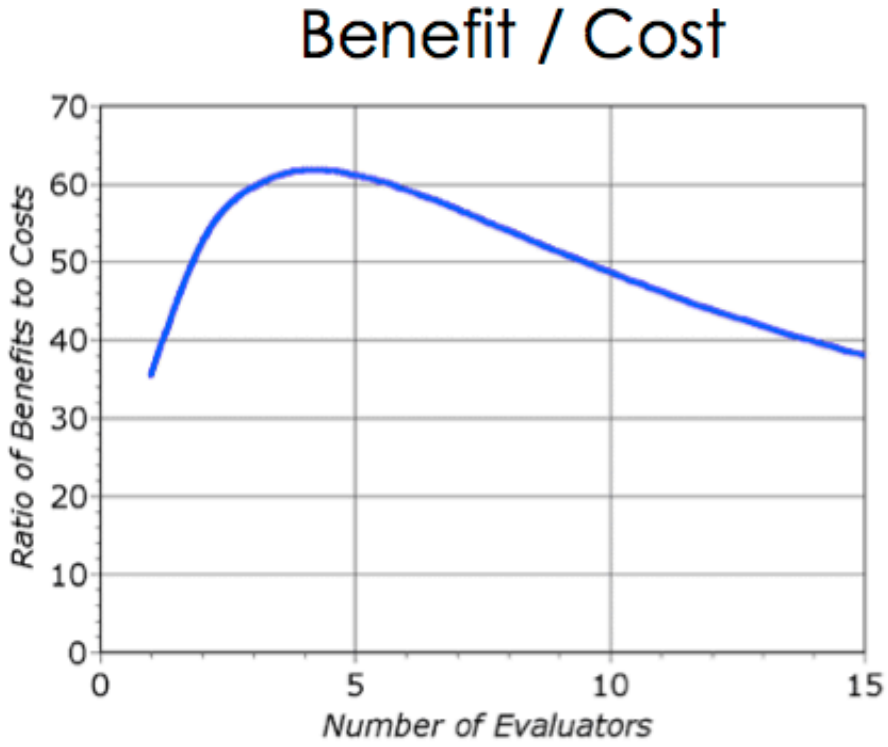
Why HE is good

- Inexpensive
  - Doesn’t “spend” users
- Fast
  - 1-2 days (instead of 1 week)
- Good
  - Proven effective: the more careful you are, the better it gets
- Easy to use
  - Relatively easy to learn, can be taught

How many evaluators are needed?

- Nielsen recommends at least 3, 5 even better





Phases of Heuristic Evaluation

0) Pre-evaluation training (optional)

   Give evaluators needed domain knowledge & information on the scenario & heuristics

1) Evaluate the interface to find usability problems
   - Screen by screen or heuristic by heuristic
   - Use it like a checklist

2) Record the problem

3) Aggregate problems

4) Assign severity rating

Which heuristics to use?

- Many possible heuristic sets
- Some standard sets (e.g. Nielsen's usability heuristics)
- You might create your own heuristics, e.g. for specific applications

- We'll focus on Nielsen's, which cover a range of **general** usability issues

Nielsen's usability heuristics

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 rather than 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

http://www.nngroup.com/articles/ten-usability-heuristics/

Nielsen's Severity Ratings

1. **Usability Blemish.** Mild annoyance or cosmetic problem. Easily avoidable.

2. **Minor usability problem.** Annoying ,misleading, unclear, confusing. Can be avoided or easily learned. May occur only once.

3. **Major usability problem.** Prevents users from completing tasks. Highly confusing or unclear.
   Difficult to avoid. Likely to occur more than once.

4. **Critical usability problem.** Users won't be able to accomplish their goals, and may quit using system.

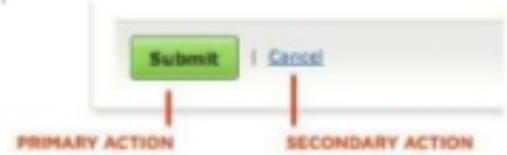
Examples

5. Error prevention (PREVENTION)

Even better than good error messages is a careful design, which prevents a problem from occurring in the first place

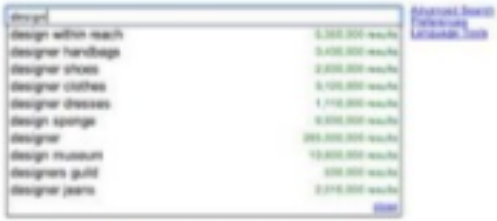
5.0 Yammer

Disables the update button after it is clicked, so the person cannot update the post twice by accident



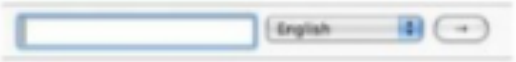
5.1 Example from "Web form Design:Filling in the Blanks" by Luke W.

Make the primary action prominent with a larger click area. Cancel and secondary actions are just shown as links



5.2 Google Auto Recommend

The auto recommend feature cuts down on mis-spellings



5.2 Wikpedia

Auto focus on input prevents a common source of frustration, typing only to realize nothing is displayed because the field did not have focus

- http://www.slideshare.net/sacsprasath/ten-usability-heuristics-with-example

Record the problem

- No standard format (but can use Usability Aspect Reports or UARs)

- Each usability problem (or feature) should be documented, and include
  - Descriptive title
  - Evaluator name
  - Location (or steps to reproduce)
  - A picture
  - Description of problem
  - Relevant heuristic
  - Severity (e.g., low, medium, high, critical)
  - Proposed solution

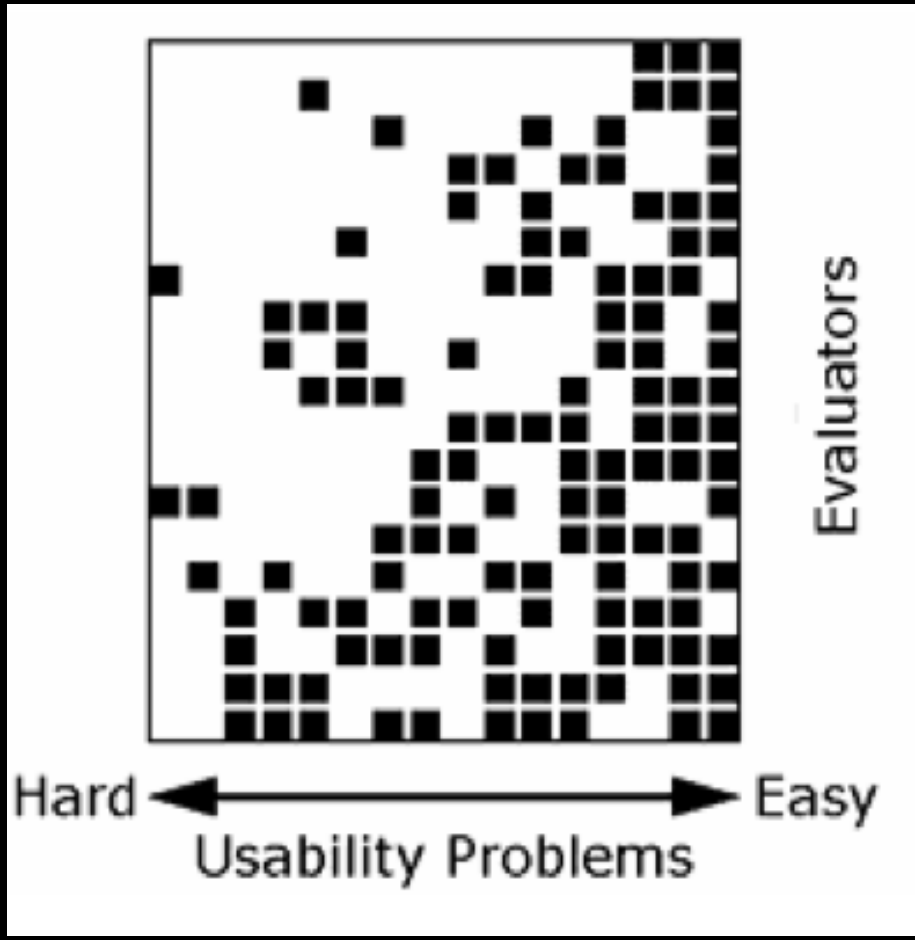
Usability Aspect Report Template
From Shaun Kane, based on UAR Template from Brad A. Myers and Bonnie John
http://www.cs.cmu.edu/~bam/uicourse/UARTemplate.doc

Complete this form *for each* problem or good aspect that you observe.

UAR #:	**Problem/Good:**	**Rated by:**
Name:		
Relevant heuristic:		
Steps to reproduce:		
Detailed explanation:		
Possible solution:		
Severity (low, medium, high, critical):	**See also:**	

Keep looking for problems!

- Usually takes a few hours
- A shorter time may not find important problems
- For very large interfaces, it is good to break heuristic evaluation into several sessions



HE vs. User Testing

- User tests may be more effective at revealing when a system's conceptual model is confusing
- User tests may be less effective at finding obscure problems
- User tests may be less effective at finding all instances of a problem
- User tests are also much more expensive
- **Advice: use HE first, to find the obvious problems, then user test.**

HE Activity

- Pair up with a partner
- Choose an app or a web site, or piece of technology in this class
- Go through heuristics and find positive/negative examples

Visibility of system status

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

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.

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.

Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention

Even better than good error messages is a careful design which 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.

Recognition rather than recall

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.

Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

HE examples

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 rather than 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

- Guidelines for usable interfaces
- Designing buttons, targets, etc.