CSE440: Introduction to HCI
Methods for Design, Prototyping and Evaluating User Interaction

Lecture 12:
Heuristic Evaluation

Nigini Oliveira
Abhinav Yadav
Liang He
Angel Vuong
Jeremy Viny
What we will do today

Inspection-based methods

Heuristic evaluation in practice

Working on the final design
What we will do today

What is a heuristic?
What we will do today
What we will do today

Heuristics Evaluation
Inspection-Based Methods

We have cut prototyping to its minimum
- Sketches, storyboards, paper prototypes
- Rapid exploration of potential ideas

But we need evaluation to guide improvement
- Evaluation can become relatively slow and expensive
- Study participants can be scarce
- May waste participants on fairly obvious problems
Inspection-Based Methods

Simulate study participants

Instead of actual study participants, use inspection to quickly and cheaply identify likely problems

Inspection methods are rational, not empirical
Heuristic Evaluation

Developed by Jakob Nielsen
Helps find usability problems in a design
Small set of evaluators examine interface
  three to five evaluators
  independently check compliance with principles
different evaluators will find different problems
evaluators only communicate afterwards
Can perform on working interfaces or sketches
Why Multiple Evaluators?

Every evaluator doesn’t find every problem
Good evaluators find both easy & hard ones
Results of Using Heuristic Evaluation

Discount: benefit-cost ratio of 48

cost was $10,500 for benefit of $500,000

how might we calculate this value?

in-house → productivity; open market → sales

Single evaluator achieves poor results

only finds 35% of usability problems

5 evaluators find ~ 75% of usability problems

why not more evaluators?

Nielsen, 1994
Number of Evaluators?

- Nielsen, 1994
Decreasing Returns

Nielsen, 1994
Nielsen’s 10 Heuristics

Too few unhelpful, too many overwhelming
  “Be Good” versus thousands of detailed rules

Nielsen seeks to create a small set
  Collects 249 usability problems
  Collects 101 usability heuristics
  Rates how well each heuristics explains each problem
  Factor analysis to identify key heuristics

Nielsen, 1994
Nielsen’s 10 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 recognize, diagnose, and recover from errors
10. Help and documentation

Nielsen, 1994
1. Visibility

Visibility of system status

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

Gmail Progress Indicator

Loading user@example.com...
Heuristics

[Image of mobile app screens demonstrating navigation and ride-hailing features]

https://uxgorilla.com/nielsens-heuristics/
Heuristics

Visibility of system status

*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.
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, mental model, metaphor, mapping, and sequencing
2. Real World Match
2. Real World Match

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 in Control

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 in Control

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.

Not just for navigation exits, but for getting out of any situation or state.
Heuristics
Heuristics

User control & freedom

provide “exits” for mistaken choices, undo, redo
don’t force down fixed paths
Heuristics

How do you want to enter your 1095-A?

- Upload the PDF
  - Choose a file to upload
  - Browse...

- Type it in

How do I get my 1095-A PDF?
Heuristics

User Account Control

Do you want to allow this app from an unknown publisher to make changes to your device?

cmd.exe

Publisher: Unknown
File origin: Hard drive on this computer

Show more details

[ ] Yes [ ] No
Heuristics

User control & freedom
provide “exits” for mistaken choices, undo, redo
don’t force down fixed paths

Wizards
must respond to question before going to next
good for beginners, infrequent tasks
not for common tasks
4. Consistency

Consistency and standards

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

Consistency and standards

    Users should not have to wonder whether 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.
# Heuristics
Heuristics

Consistency & Standards
Heuristics

External Consistency
5. Error Prevention

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.
5. Error Prevention

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.

Try to commit errors and see how they are handled. Could they have been prevented?
5. Error Prevention
Heuristics

Your password must have:
- 8 or more characters
- Upper & lowercase letters
- At least one number

Strength: strong

Avoid passwords that are easy to guess or used with other websites.
6. Recognition not Recall

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.
6. Recognition not Recall

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.

People should never carry a memory load
6. Recognition not Recall

Addresses visibility of features & information
where to find things

Visibility addresses system status & feedback
what is going on
6. Recognition not Recall

Problems with affordances may go here

- hidden affordance: remember where to act
- false affordance: remember it is a fake
6. Recognition not Recall

What are you interested in?
Pick whatever catches your eye...you can always fine-tune things later

- Technology
- Men's Apparel
- Travel
- Art
- Gadgets
- Recipes
- Design
- Photography
- Funny Pictures
- Tattoos
- Luxury Cars
- Woodworking
7. Flexibility and Efficiency

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.
7. Flexibility and Efficiency

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.

Concerns anywhere users have repetitive actions that must be done manually. Also concerns allowing multiple ways to do things.
7. Flexibility and Efficiency
7. Flexibility and Efficiency
7. Flexibility and Efficiency

Flexibility and Efficiency of Use

accelerators for experts (e.g., keyboard shortcuts) allow tailoring of frequent actions (e.g., macros)
8. Aesthetic Design

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.
8. Aesthetic Design

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.

Not just about “ugliness”. About clutter, overload of visual field, visual noise, distracting animations, and so on.
Heuristics
Heuristics

Aesthetic & Minimalist design

no irrelevant information in dialogues
Heuristics
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 constructively suggest a solution.
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 constructively suggest a solution.

Error prevention is about preventing errors before they occur. This is about after they occur.
9. Error Recovery

Help recognize, diagnose, & recover from errors

- error messages in plain language
- precisely indicate the problem
- constructively suggest a solution
9. Error Recovery

Help recognize, diagnose, & recover from errors
9. Error Recovery

Overheard in a Lander lounge:
Boi: ACCHHOOOOOOOO!
Girl: “FOR THE FIFTEENTH TIME, GOD BLESS YOU AND GOD BLESS AMERICA!!!”

Leave Page?

You haven't finished your comment yet. Do you want to leave without finishing?

[Stay on This Page] [Leave This Page]
9. Error Recovery

- Sorry, we couldn't find an account with that username. Can we help you recover your username?

- Username: freshsparkss

- Password

- Log In

- Stay logged in

- Create an account  •  Trouble logging in?

- Sorry, that password isn't right. We can help you recover your password.

- Username: freshsparks

- Password

- Log In

- Stay logged in

- Create an account  •  Trouble logging in?
10. Help

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.
10. Help

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.

This does not mean that the user must be able to ask for help on every single item.
10. Help

Slackbot 1:28 AM
how to change settings?

Slackbot 1:28 AM
I searched for that on our Help Center. Perhaps these articles will help:
- Change a member’s email address
- Guide to single sign-on settings
- Customize message and file retention policies

Abhinav Yadav 1:28 AM
how can I disable snooze?

Slackbot 1:28 AM
I searched for that on our Help Center. Perhaps these articles will help:
- Pause notifications with Do Not Disturb
- Guide to Slack notifications
- Remove single sign-on

Abhinav Yadav 1:29 AM
what is love?

Slackbot 1:29 AM
I'm sorry, I don't understand! Sometimes I have an easier time with a few simple keywords.

Or you can head to our wonderful Help Center for more assistance!
Heuristic Evaluation Process

Evaluators go through interface several times
  - inspect various dialogue elements
  - compare with list of usability principles

Usability principles
  - Nielsen’s “heuristics”
  - supplementary list of category-specific heuristics (competitive analysis or testing existing products)

Use violations to redesign/fix problems
Examples

Can’t copy info from one window to another
violates “Minimize memory load” (H6)
fix: allow copying

Typography uses different fonts in 3 dialog boxes
violates “Consistency and standards” (H4)
  slows users down
  probably wouldn’t be found by usability testing
fix: pick a single format for entire interface
Phases of Heuristic Evaluation

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

2) Evaluation
individuals evaluate interface & make lists of problems

3) Severity rating
determine how severe each problem is

4) Aggregation
group meets & aggregates problems (w/ ratings)

5) Debriefing
discuss the outcome with design team
How to Perform 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
  be specific & list each problem separately
Example Heuristic Violation

1. [H4 Consistency]

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.
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
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
Severity Rating

0 - Do not agree this is a problem.

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 will not be able to accomplish their goals. Users may quit using system all together.
Example Heuristic Violation

1. [H4 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.
Debriefing

Conduct with evaluators, observers, and development team members
Discuss general characteristics of interface
Suggest potential improvements to address major usability problems
Development team rates how hard to fix
Make it a brainstorming session
Fixability Scores

1 - Nearly impossible to fix. Requires massive re-engineering or use of new technology. Solution not known or understood at all.

2 - Difficult to fix. Redesign and re-engineering required. Significant code changes. Solution identifiable but details not fully understood.

3 - Easy to fix. Minimal redesign and straightforward code changes. Solution known and understood.

4 - Trivial to fix. Textual changes and cosmetic changes. Minor code tweaking.
Example Heuristic Violation

1. [H4 Consistency] [Severity 3] [Fix 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.

Fix: Change second screen to "Save".
Ask us something!
Finalizing the Design
Select one design and two tasks:

- Why this design and these tasks?
  (Designs can be combined if necessary)
- What makes the design better suited to the people for whom you are targeting your design?
- Why are these tasks more compelling than the others?
Working Time