CSE 440:
Introduction to HCI

14: Inspection

May 9, 2024

Jesse J. Martinez | Avery Mack | Simona Liao

Objectives

Be able to:

Describe why we use inspection-based methods

Given the full text of Nielsen's heuristics, be able to:

- explain what each of them means
- apply them to identify usability failures in an interface

Describe an effective heuristic evaluation process

Explain why the typical recommendation for heuristic evaluation is 3 to 5 independent evaluators

Project Status

Looking Forward

- 3b: Heuristic Evaluation (1 in class today, iterate, another)
  - 3b EXP: Do a *thorough* Heuristic Eval of your own prototype
  - EXP due Tuesday (5/14) @ 8pm
- 3c: Usability Testing Check-In (includes findings from 3b)
  - Usability Test in Section next week on Friday (5/17)

Exam

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

- Can become relatively slow and expensive
- Study participants can be scarce
- Can waste participants on obvious problems

Inspection-Based Methods

Simulate study participants

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

Inspection methods are rational, not empirical

- Usage and usability data is by definition more accurate, but inspection data is much easier to obtain

Heuristic Evaluation is the most well-known method

Heuristic Evaluation

Developed by Jakob Nielsen

- Helps find usability problems in a design
- Not a method for “coming up with” 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

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 heuristics explain problems
- Factor analysis to identify key heuristics

Nielsen, 1994

Nielsen’s 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

Nielsen, 1994

1. Visibility

Visibility of system status

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

Refers to both visibility of system status
and providing appropriate feedback

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

2. Real World Match

Match between system and the real world

The system should speak a person’s language,
with words, phrases and concepts familiar to the person,
rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

Refers to language choice, mental model, metaphors, mapping, and sequencing.

3. Control and Freedom

User control and freedom

People 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.

4. Consistency

Consistency and standards

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

Reminder:

Internal consistency is consistency throughout a design.

External consistency is consistency with other designs.

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 people 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?

6. Recognition not Recall

Recognition rather than recall

Minimize a person's memory load by
making objects, actions, and options visible.
A person 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 and information

- where to find things

Visibility addresses system status and feedback

- what is going on

Problems with affordances may go here

- hidden affordance: remember where to act
- false affordance: remember it is a fake

7. Flexibility and Efficiency

Flexibility and efficiency of use

> Accelerators, while unseen by novices,
> may often speed up the interaction for experts
> such that the system can cater to both
> inexperienced and experienced use.
> Allow people to tailor frequent actions.

Concerns anywhere people have repetitive actions that must be done manually. Also concerns allowing multiple ways to do things.

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.

9. Error Recovery

Help people 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.

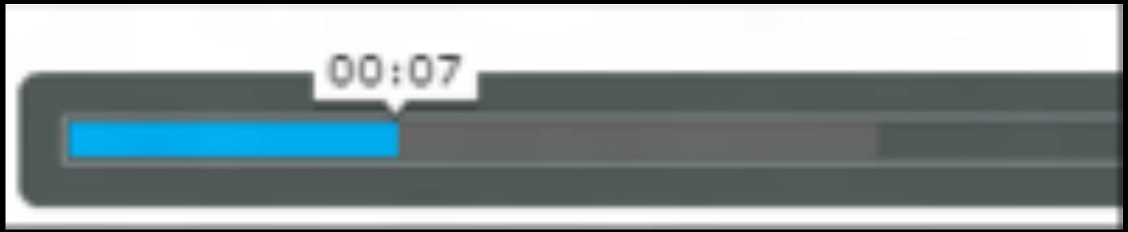
10. Help

Help and documentation

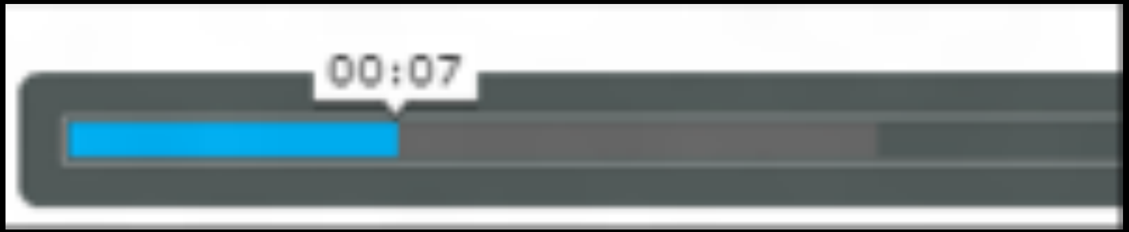
> Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search,
> focused on a person's task, list concrete steps to be carried out, and not be too large.

This does not mean that a person
must be able to ask for help on every single item.

Heuristics



Heuristics



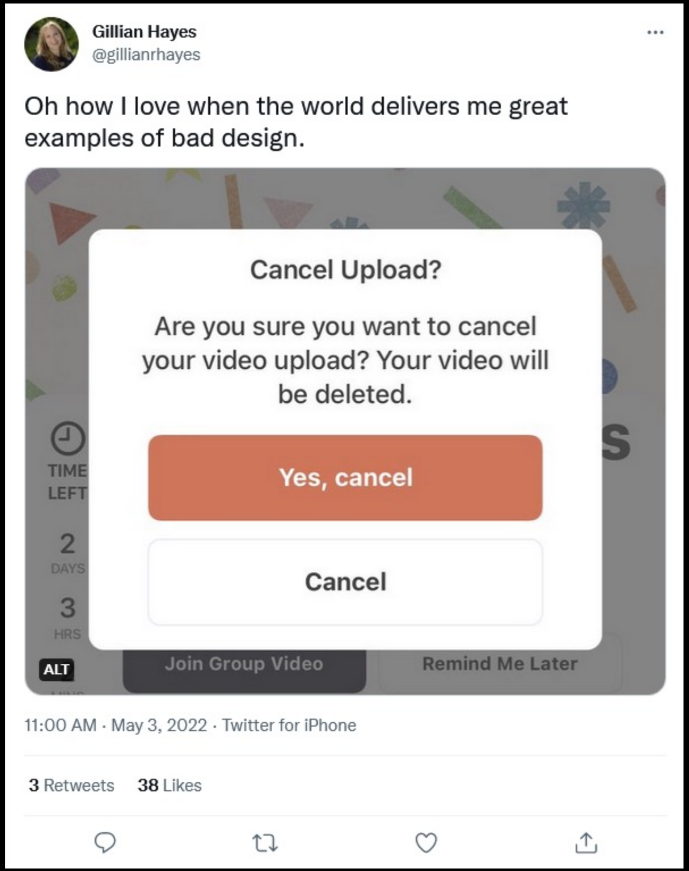
Visibility of system status

pay attention to response time

0.1 sec:	no special indicators needed (why?)
1.0 sec:	person tends to lose track of data
10 sec:	maximum duration if person to stay focused
	longer delays require progress bars

Violations can also be what is **not** in an interface

Heuristics



Gillian Hayes
@gillianrhayes

Oh how I love when the world delivers me great examples of bad design.

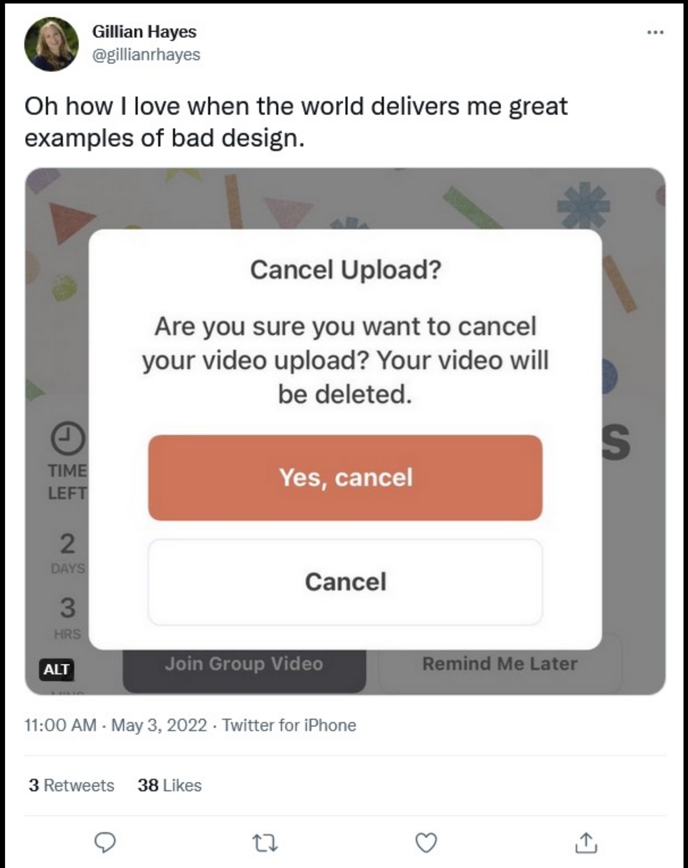
Cancel Upload?

Are you sure you want to cancel your video upload? Your video will be deleted.

11:00 AM · May 3, 2022 · Twitter for iPhone

3 Retweets 38 Likes

Heuristics



Gillian Hayes
@gillianrhayes

Oh how I love when the world delivers me great examples of bad design.

Cancel Upload?

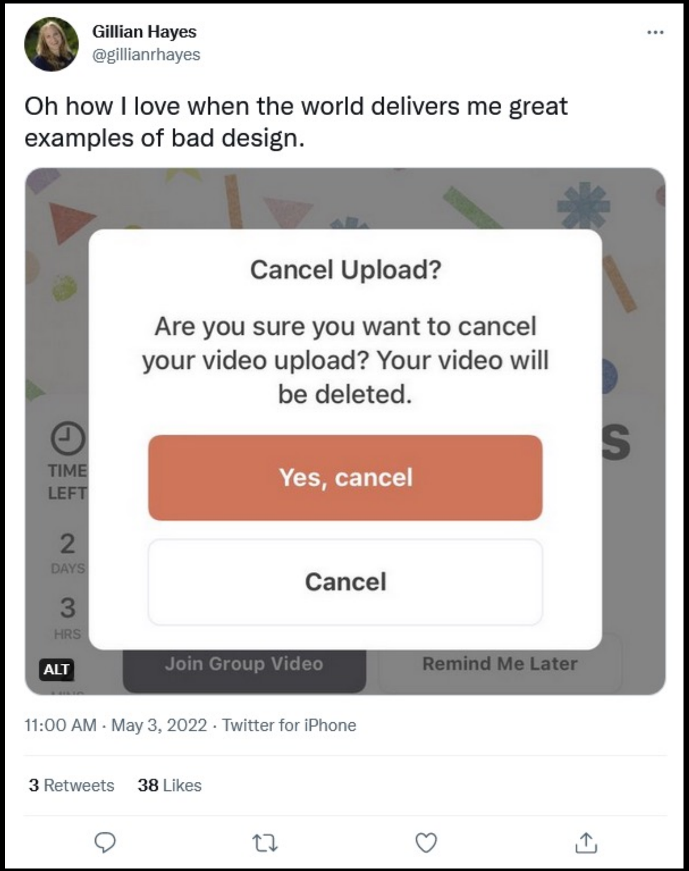
Are you sure you want to cancel your video upload? Your video will be deleted.

11:00 AM · May 3, 2022 · Twitter for iPhone

3 Retweets 38 Likes

Error prevention

Heuristics



Gillian Hayes
@gillianrhayes

Oh how I love when the world delivers me great examples of bad design.

Cancel Upload?

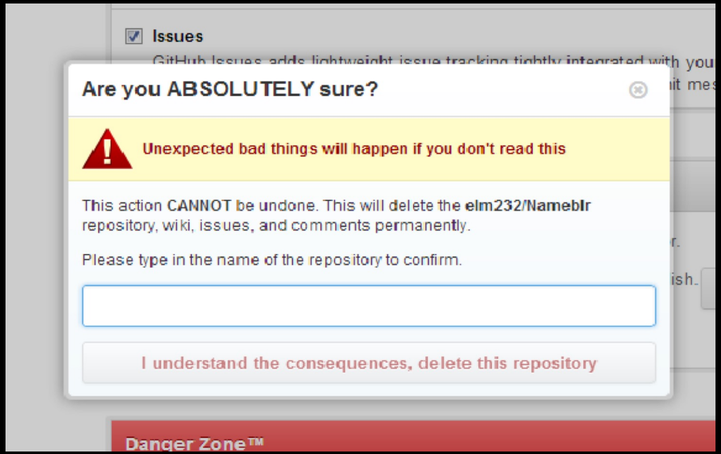
Are you sure you want to cancel your video upload? Your video will be deleted.

11:00 AM · May 3, 2022 · Twitter for iPhone

3 Retweets 38 Likes

Error prevention

Contrast with:



☑ Issues

GitHub Issues adds lightweight issue tracking tightly integrated with you

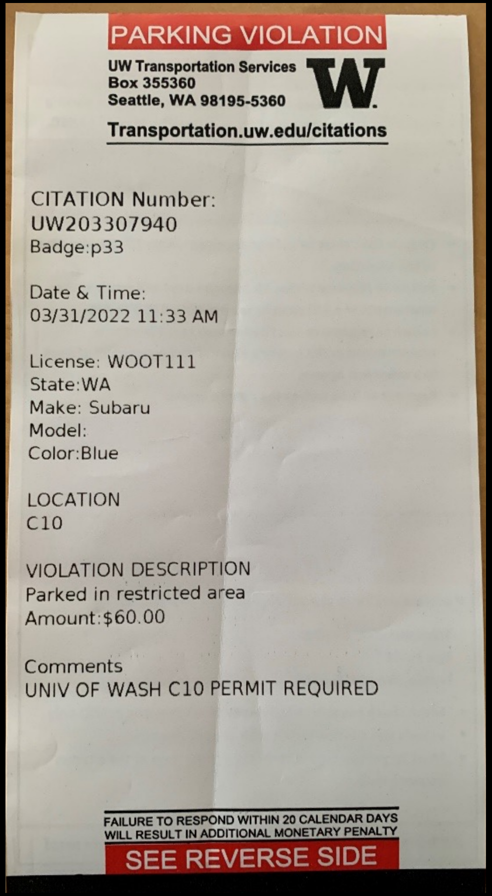
Are you ABSOLUTELY sure?

Unexpected bad things will happen if you don't read this

This action CANNOT be undone. This will delete the **elm232/Nameblr** repository, wiki, issues, and comments permanently.

Please type in the name of the repository to confirm.

Heuristics



PARKING VIOLATION

UW Transportation Services
Box 355360
Seattle, WA 98195-5360

Transportation.uw.edu/citations

CITATION Number:
UW203307940
Badge:p33

Date & Time:
03/31/2022 11:33 AM

License: WOOT111
State:WA
Make: Subaru
Model:
Color:Blue

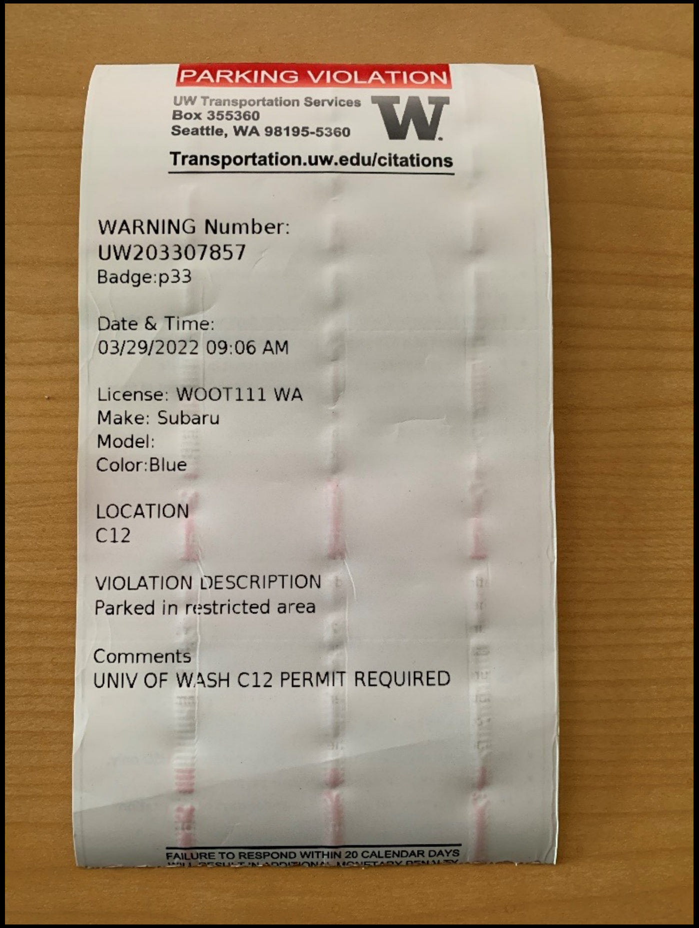
LOCATION
C10

VIOLATION DESCRIPTION
Parked in restricted area
Amount:$60.00

Comments
UNIV OF WASH C10 PERMIT REQUIRED

FAILURE TO RESPOND WITHIN 20 CALENDAR DAYS
WILL RESULT IN ADDITIONAL MONETARY PENALTY

SEE REVERSE SIDE



PARKING VIOLATION

UW Transportation Services
Box 355360
Seattle, WA 98195-5360

Transportation.uw.edu/citations

WARNING Number:
UW203307857
Badge:p33

Date & Time:
03/29/2022 09:06 AM

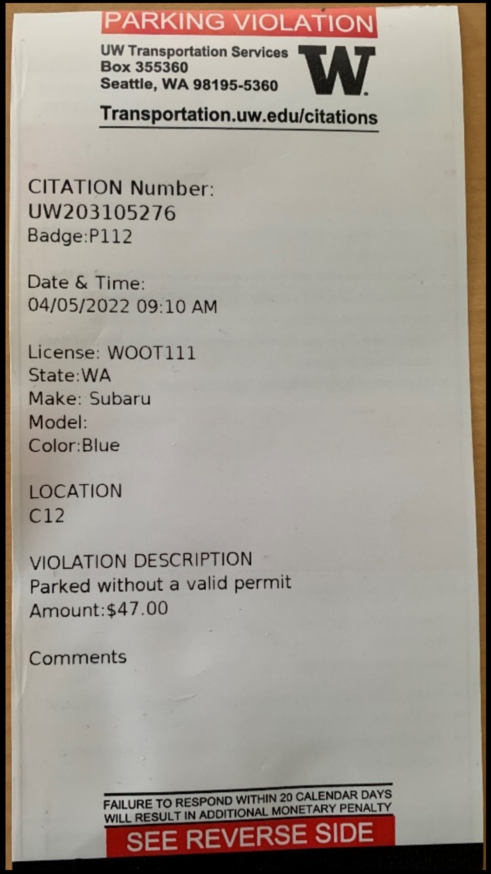
License: WOOT111 WA
Make: Subaru
Model:
Color:Blue

LOCATION
C12

VIOLATION DESCRIPTION
Parked in restricted area

Comments
UNIV OF WASH C12 PERMIT REQUIRED

FAILURE TO RESPOND WITHIN 20 CALENDAR DAYS
WILL RESULT IN ADDITIONAL MONETARY PENALTY



PARKING VIOLATION

UW Transportation Services
Box 355360
Seattle, WA 98195-5360

Transportation.uw.edu/citations

CITATION Number:
UW203105276
Badge:P112

Date & Time:
04/05/2022 09:10 AM

License: WOOT111
State:WA
Make: Subaru
Model:
Color:Blue

LOCATION
C12

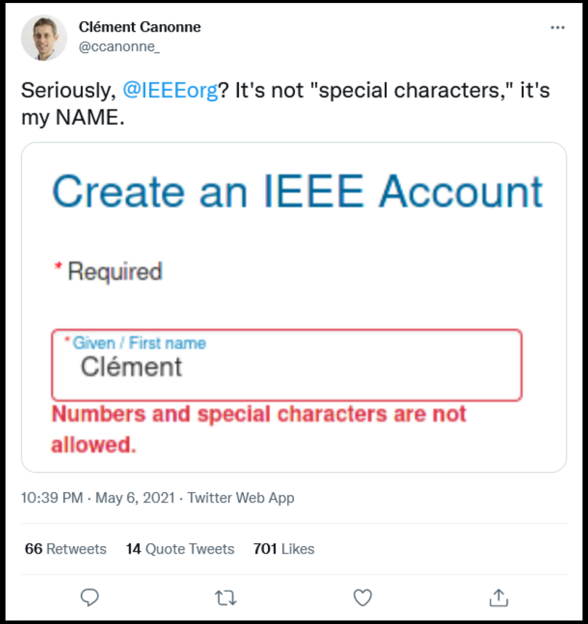
VIOLATION DESCRIPTION
Parked without a valid permit
Amount:$47.00

Comments

FAILURE TO RESPOND WITHIN 20 CALENDAR DAYS
WILL RESULT IN ADDITIONAL MONETARY PENALTY

SEE REVERSE SIDE

Heuristics



Clément Canonne
@ccanonne_

Seriously, @IEEEorg? It's not "special characters," it's my NAME.

Create an IEEE Account

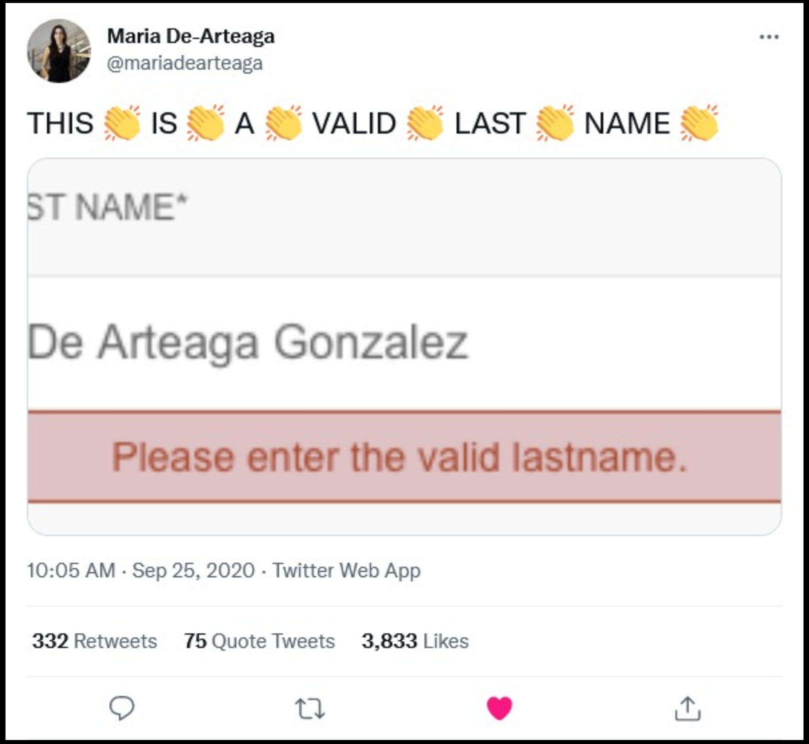
* Required

* Given / First name
Clément

Numbers and special characters are not allowed.

10:39 PM · May 6, 2021 · Twitter Web App

66 Retweets 14 Quote Tweets 701 Likes



Maria De-Arteaga
@mariadearteaga

THIS 👏 IS 👏 A 👏 VALID 👏 LAST 👏 NAME 👏

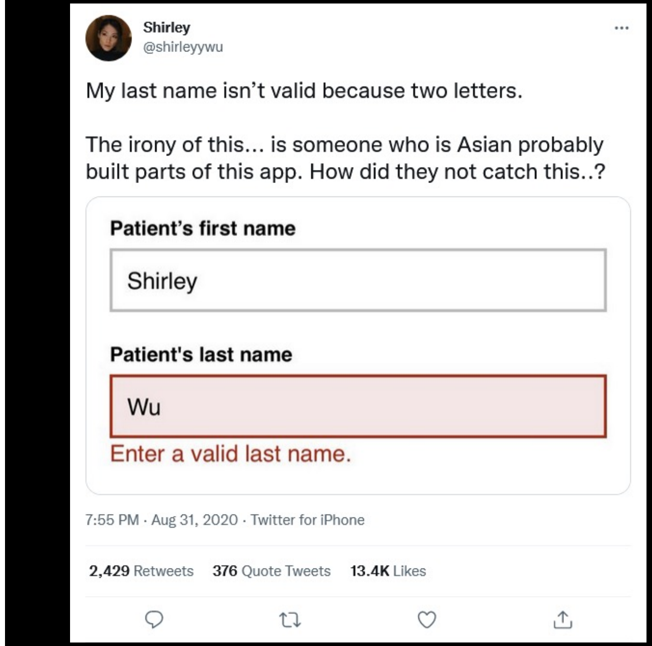
ST NAME*

De Arteaga Gonzalez

Please enter the valid lastname.

10:05 AM · Sep 25, 2020 · Twitter Web App

332 Retweets 75 Quote Tweets 3,833 Likes



Shirley
@shirleyywu

My last name isn't valid because two letters.

The irony of this… is someone who is Asian probably built parts of this app. How did they not catch this..?

Patient's first name
Shirley

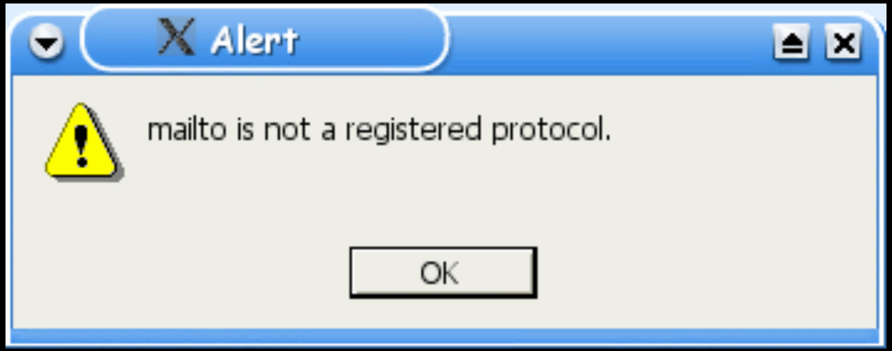
Patient's last name
Wu

Enter a valid last name.

7:55 PM · Aug 31, 2020 · Twitter for iPhone

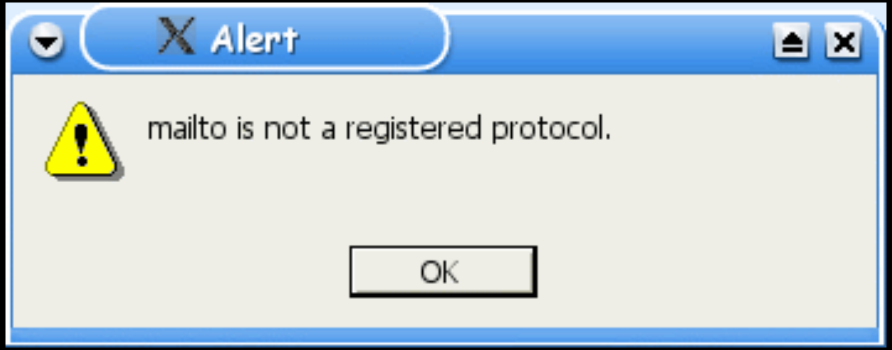
2,429 Retweets 376 Quote Tweets 13.4K Likes

Heuristics



mailto is not a registered protocol.

Heuristics



mailto is not a registered protocol.

“Mailto”, “protocol”?

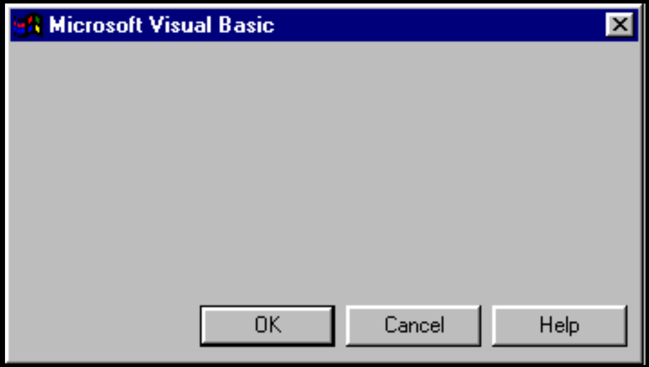
Match system to real world

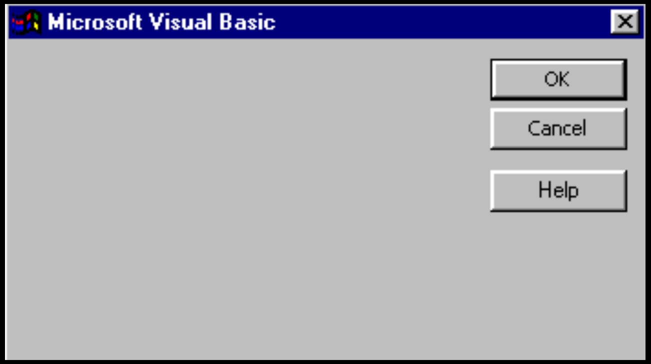
Speak the person’s language

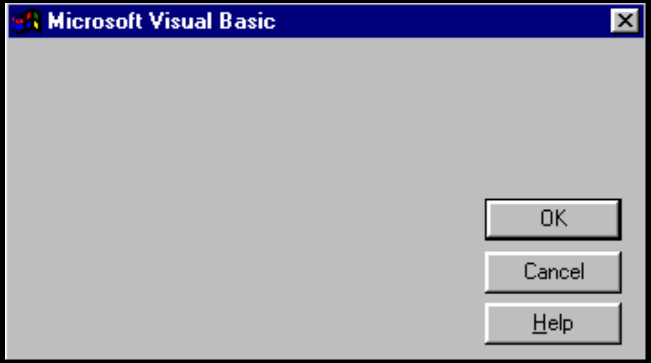
Heuristics

Consistency & Standards

Which is the standard layout for these buttons?







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

Provide any knowledge an evaluator may need

- might not be necessary (e.g., walk up an use, is domain expert)
- could be supplied as scenarios

Each evaluator produces list of problems

- explain why with reference to heuristic
- be specific and list each problem separately

Example Heuristic Violation

1. [H4 Consistency]

The interface used the string "Save" on the first screen for saving the person’s file, but used the string "Write file" on the second screen.

People may be confused by this different terminology for the same function.

How to Perform Heuristic Evaluation

Why separate listings for each violation?

- risk of a ‘fix’ repeating some 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 with paper prototypes, but sometimes just “not yet implemented”

Severity Rating

Used to allocate resources to fix problems

Combination of

- frequency
- impact
- persistence (one time or repeating)

Can be calculated after all evaluations

Can 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 people from completing tasks. Highly confusing or unclear. Difficult to avoid. Likely to occur more than once.

4 - Critical usability problem.

People will not be able to accomplish their goals.
People 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 person’s file, but used the string "Write file" on the second screen.

People may be confused by this different terminology for the same function.

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 4]

The interface used the string "Save" on the first screen for saving the person’s file, but used the string "Write file" on the second screen.

People may be confused by this different terminology for the same function.

Fix: Change second screen to "Save".

Severity:	Bigger # = More Severe
Fix:	Bigger # = More Easily Fixable

These scales allow multiplying Severity and Fix as a crude indication of priority

Big Impact X Easy to Fix = Top Priority

Phases of Heuristic Evaluation

1) Pre-Evaluation Training

   give evaluators needed domain knowledge & scenario information

2) Evaluation

   individuals evaluate interface and make lists of problems

3) Severity Rating

   Individually determine how severe each problem is

4) Aggregation

   group meets to aggregate and rate problems

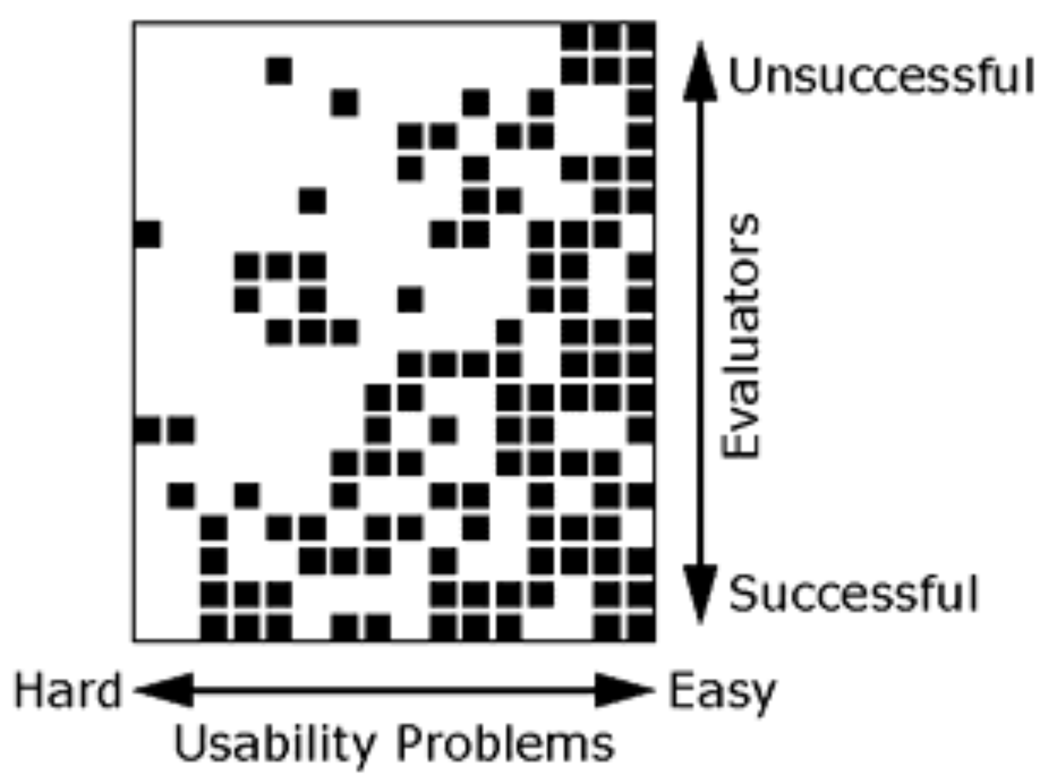
5) Debriefing

   discuss the outcome with design team

Why Multiple Evaluators?

Every evaluator does
not find every problem

Good evaluators find
both easy and hard to identify problems



Results of Heuristic Evaluation

Discount method: benefit-cost estimate of 48

- cost was $10,500 for benefit of $500,000
- how might we calculate this value?
  - in-house productivity; market sales

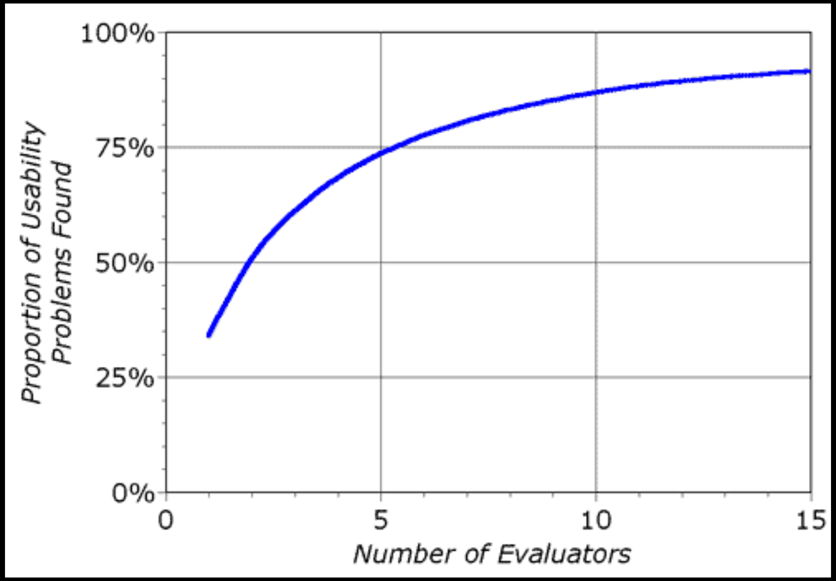
Single evaluator achieves poor results

- only finds 35% of usability problems
- 5 evaluators find ~ 75% of usability problems
- why not more evaluators?

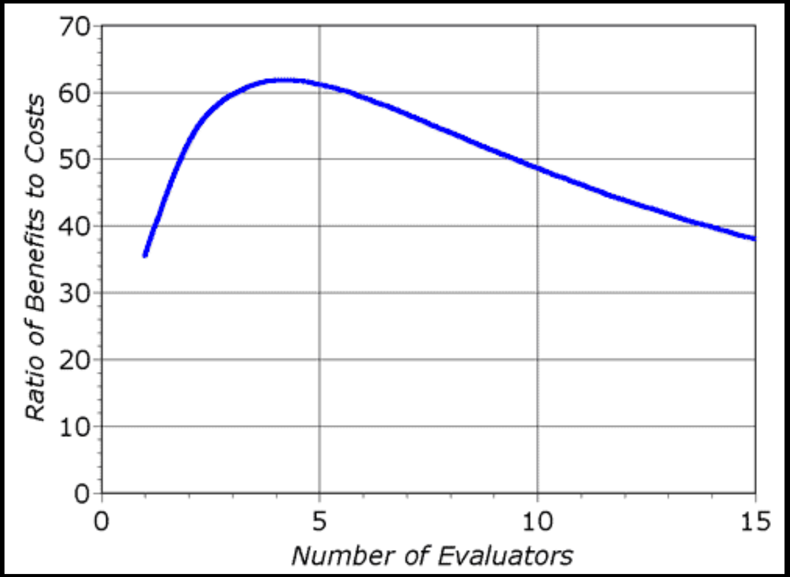
Nielsen, 1994

Decreasing Returns

problems found



benefits / cost

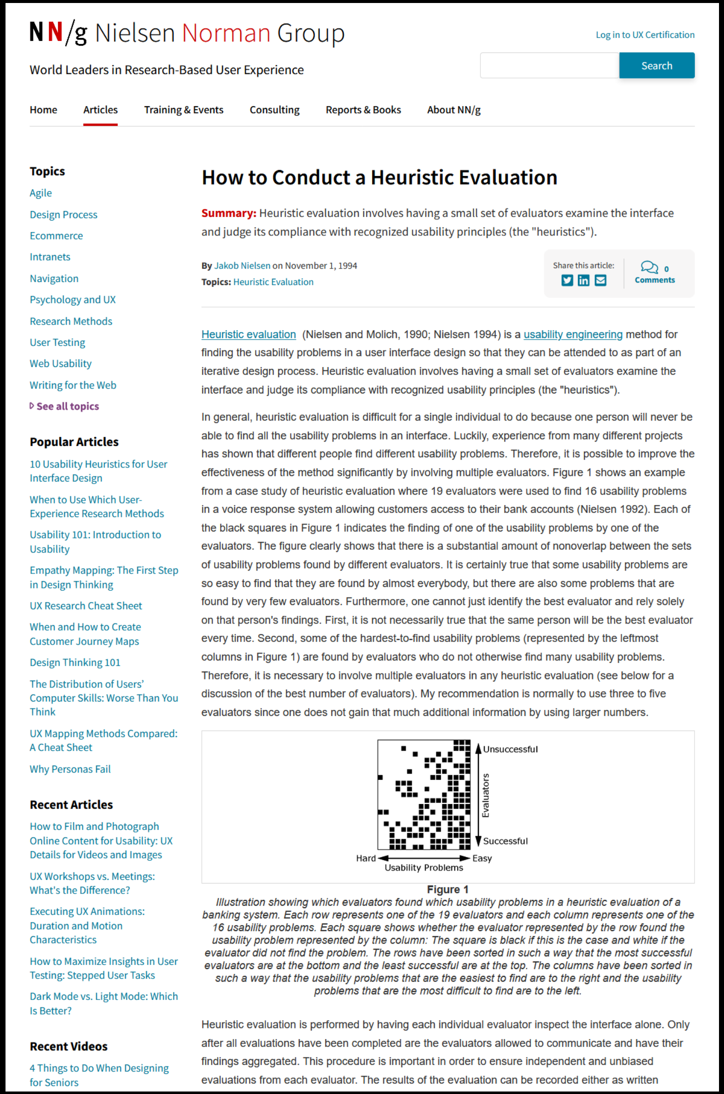


Nielsen, 1994

Additional Reading

Introduces heuristic evaluation and its development

Makes basic argument for cost-effectiveness



NN/g Nielsen Norman Group

Log in to UX Certification

World Leaders in Research-Based User Experience

Search

Home Articles Training & Events Consulting Reports & Books About NN/g

How to Conduct a Heuristic Evaluation

Summary: Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics").

By Jakob Nielsen on November 1, 1994
Topics: Heuristic Evaluation

Share this article: 0 Comments

Heuristic evaluation (Nielsen and Molich, 1990; Nielsen 1994) is a usability engineering method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process. Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics").

In general, heuristic evaluation is difficult for a single individual to do because one person will never be able to find all the usability problems in an interface. Luckily, experience from many different projects has shown that different people find different usability problems. Therefore, it is possible to improve the effectiveness of the method significantly by involving multiple evaluators. Figure 1 shows an example from a case study of heuristic evaluation where 19 evaluators were used to find 16 usability problems in a voice response system allowing customers access to their bank accounts (Nielsen 1992). Each of the black squares in Figure 1 indicates the finding of one of the usability problems by one of the evaluators. The figure clearly shows that there is a substantial amount of nonoverlap between the sets of usability problems found by different evaluators. It is certainly true that some usability problems are so easy to find that they are found by almost everybody, but there are also some problems that are found by very few evaluators. Furthermore, one cannot just identify the best evaluator and rely solely on that person's findings. First, it is not necessarily true that the same person will be the best evaluator every time. Second, some of the hardest-to-find usability problems (represented by the leftmost columns in Figure 1) are found by evaluators who do not otherwise find many usability problems. Therefore, it is necessary to involve multiple evaluators in any heuristic evaluation (see below for a discussion of the best number of evaluators). My recommendation is normally to use three to five evaluators since one does not gain that much additional information by using larger numbers.

Figure 1

Illustration showing which evaluators found which usability problems in a heuristic evaluation of a banking system. Each row represents one of the 19 evaluators and each column represents one of the 16 usability problems. Each square shows whether the evaluator represented by the row found the usability problem represented by the column: The square is black if this is the case and white if the evaluator did not find the problem. The rows have been sorted in such a way that the most successful evaluators are at the bottom and the least successful are at the top. The columns have been sorted in such a way that the usability problems that are the easiest to find are to the right and the usability problems that are the most difficult to find are to the left.

Heuristic evaluation is performed by having each individual evaluator inspect the interface alone. Only after all evaluations have been completed are the evaluators allowed to communicate and have their findings aggregated. This procedure is important in order to ensure independent and unbiased evaluations from each evaluator. The results of the evaluation can be recorded either as written

Topics

- Agile
- Design Process
- Ecommerce
- Intranets
- Navigation
- Psychology and UX
- Research Methods
- User Testing
- Web Usability
- Writing for the Web
- See all topics

Popular Articles

- 10 Usability Heuristics for User Interface Design
- When to Use Which User-Experience Research Methods
- Usability 101: Introduction to Usability
- Empathy Mapping: The First Step in Design Thinking
- UX Research Cheat Sheet
- When and How to Create Customer Journey Maps
- Design Thinking 101
- The Distribution of Users' Computer Skills: Worse Than You Think
- UX Mapping Methods Compared: A Cheat Sheet
- Why Personas Fail

Recent Articles

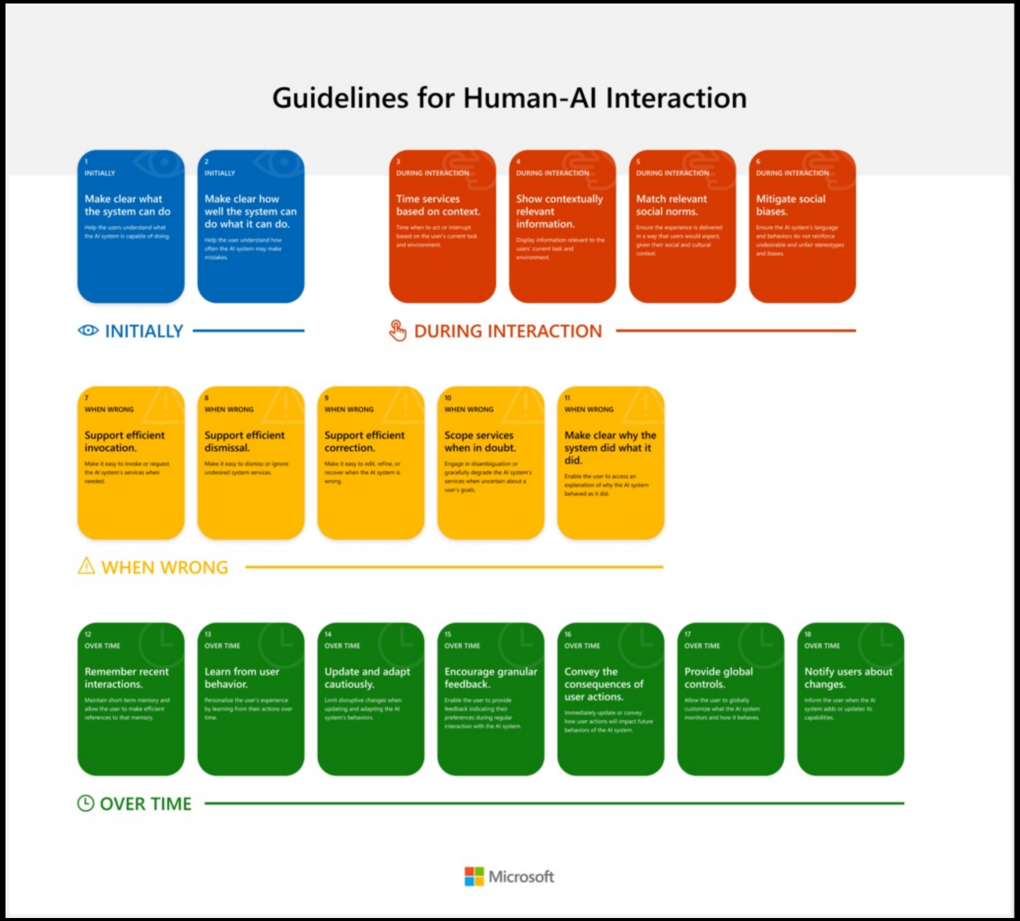
- How to Film and Photograph Online Content for Usability: UX Details for Videos and Images
- UX Workshops vs. Meetings: What's the Difference?
- Executing UX Animations: Duration and Motion Characteristics
- How to Maximize Insights in User Testing: Stepped User Tasks
- Dark Mode vs. Light Mode: Which Is Better?

Recent Videos

- 4 Things to Do When Designing for Seniors

Nielsen, 1994

More Focused Heuristics



Guidelines for Human-AI Interaction

1
INITIALLY
Make clear what the system can do
Help the users understand what the AI system is capable of doing.

2
INITIALLY
Make clear how well the system can do what it can do.
Help the user understand how often the AI system may make mistakes.

3
DURING INTERACTION
Time services based on context.
Time when to act or interrupt based on the user's current task and environment.

4
DURING INTERACTION
Show contextually relevant information.
Display information relevant to the users' current task and environment.

5
DURING INTERACTION
Match relevant social norms.
Ensure the experience is delivered in a way that users would expect, given their social and cultural context.

6
DURING INTERACTION
Mitigate social biases.
Ensure the AI system's language and behaviors do not reinforce undesirable and unfair stereotypes and biases.

INITIALLY

DURING INTERACTION

7
WHEN WRONG
Support efficient invocation.
Make it easy to invoke or request the AI system's services when needed.

8
WHEN WRONG
Support efficient dismissal.
Make it easy to dismiss or ignore undesired system services.

9
WHEN WRONG
Support efficient correction.
Make it easy to edit, refine, or recover when the AI system is wrong.

10
WHEN WRONG
Scope services when in doubt.
Engage in disambiguation or gracefully degrade the AI system's services when uncertain about a user's goals.

11
WHEN WRONG
Make clear why the system did what it did.
Enable the user to access an explanation of why the AI system behaved as it did.

WHEN WRONG

12
OVER TIME
Remember recent interactions.
Maintain short-term memory and allow the user to make efficient references to that memory.

13
OVER TIME
Learn from user behavior.
Personalize the user's experience by learning from their actions over time.

14
OVER TIME
Update and adapt cautiously.
Limit disruptive changes when updating and adapting the AI system's behaviors.

15
OVER TIME
Encourage granular feedback.
Enable the user to provide feedback indicating their preferences during regular interaction with the AI system.

16
OVER TIME
Convey the consequences of user actions.
Immediately update or convey how user actions will impact future behaviors of the AI system.

17
OVER TIME
Provide global controls.
Allow the user to globally customize what the AI system monitors and how it behaves.

18
OVER TIME
Notify users about changes.
Inform the user when the AI system adds or updates its capabilities.

OVER TIME

Microsoft

Inspection vs. Usability Testing

Inspection

- Is much faster
- Does not require interpreting participant actions
- May miss problems or find false positives

Usage data and usability testing

- More accurate by definition
- Account for actual people and tasks

One effective approach is to alternate between them

- Find different problems, conserve participants

Alternative Inspection-Based Methods

Cognitive Walkthrough

- Surfaces different types of usability problems
- Consider as a complement to heuristic evaluation

Action Analysis

- Low-level modeling of expert performance
- Be aware of GOMS, but may never encounter it

Assignment 3b

Split up your team

- 2 people receive a heuristic evaluation
- Others find teams to give a heuristic evaluation

Iterate and obtain a second heuristic evaluation

Keep detailed records of revision

- “Before” design, identified issue, “After” design
- You will need this record in later assignments

3c: One usability test and iteration due Wednesday (5/15)

CSE 440:
Introduction to HCI

14: Inspection

May 9, 2024

Jesse J. Martinez | Avery Mack | Simona Liao