Strategies for Digital Wellbeing

Heuristic Evaluation

Alberto Monge Roffarello

Evaluation Goal (from an HCI perspective)

- «Evaluation tests the usability, functionality, and acceptability of an interactive system»
  - According to the design stage (sketch, prototype, … final)
  - According to the initial goals
  - Alongside different dimensions
  - Using a range of different techniques
- Very wide (and a little bit vague) definition
- The idea is to identify and correct problems as soon as possible

Evaluation Approaches

- Evaluation may take place:
  - In the laboratory
  - In the field
- Involving users:
  - Experimental methods
  - Observational methods
  - Query methods
  - Formal or semi-formal or informal
- Based on expert evaluation:
  - Analytic methods
  - Review methods
  - Model-based methods
  - Heuristics
- Automated:
  - Simulation and software measures
  - Formal evaluation with models and formulas
  - Especially for low-level issues

Lab Studies

- In lab studies, users are taken out of their normal work environment to take part in **controlled** tests. They are typically adopted in the early stages of design (e.g., to compare alternatives, you don't need a working implementation).
  - 👍 simulation of dangerous environments
  - 👍 suitable for specific tasks within a system
  - 👎 lack of context
  - 👎 unnatural situations leading to biases
  - 👎 not suitable for all the tasks

Field Studies

- Field studies takes the designer or evaluator out into the **user's work environment** in order to observe the system in action.
  - 👍 open nature: the "real" context
  - 👍 users are in their natural environment
  - 👎 low degree of control
  - 👎 higher costs (you need a working implementation)
  - 👎 longer duration

Expert Evaluations

- Evaluation may be based on **expert evaluation:**
  - Analytic methods
  - Review methods
  - Model-based methods
  - Heuristics
- It is useful to identify any areas that are likely to cause difficulties because they violate known cognitive principles, or ignore accepted empirical results

  👍 it can be used at any stage in the development process

  👍 it is relatively cheap, since it does not require user involvement

  👎 it does not assess actual use of the system

Heuristic Evaluation

Experts check potential issues on your design, by referring to a set of heuristic criteria

When Is Design Critique Useful?

- Before user testing
  - To save effort
  - Solving easy-to-solve problems
  - Leaving user testing for bigger issues
- Before redesigning
  - Identify the good parts (to be kept) and the bad ones (to be redesigned)
- To generate evidence for problems that are known (or suspected)
  - From ‘murmurs’ or ‘impressions’ to hard evidence
- Before release
  - Smoothing and polishing

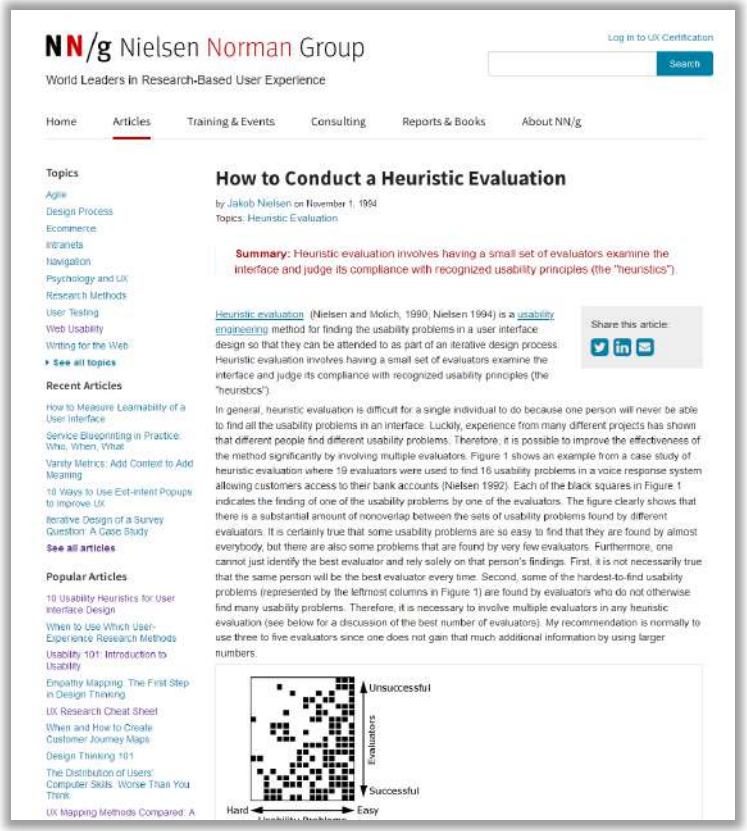
Heuristic Evaluation

- A method developed by Jacob Nielsen (1994)
  - Structured design critique
  - Using a set of simple and general heuristics
  - Executed by a small group of experts (3-5)
  - Suitable for any stage of the design (sketches, UI, …)
  - Original goal: find usability problems in a design
- Also popularized as “Discount Usability”

https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

Basic Idea

- Define a set of heuristics (or principles):
  - a heuristic is a guideline or general principle or rule of thumb that can guide a design decision or be used to critique a decision that has already been made.
- Give those heuristics to a group of experts
  - Each expert will use heuristics to look for problems in the design
- Experts work independently
  - Each expert will find different problems
- At the end, experts communicate and share their findings
  - Findings are analyzed, aggregated, ranked
- The discovered *violations* of the heuristics are used to fix problems or to re-design



Heuristics

- Nielsen proposed 10 heuristic rules
  - Good at finding most design problems
- In a specific context, application domain, or for specific design goals …
  - … new heuristics can be defined
  - … some heuristic can be ignored

Phases of Heuristic Evaluation

1. Pre-evaluation training
   - Give evaluator information about the domain and the scenario to be evaluated
2. Evaluation
   - Individual
3. Severity Rating
   - First, individually
   - Then, aggregate and find consensus
4. Debriefing
   - Review with the design team

Evaluation (I)

- Define a set of tasks, that the evaluators should analyze
- For each task, the evaluator should step through the design several times, and inspect the UI elements
  - On the real design, or on a preliminary prototype
- At each step, check the design according to each of the heuristics
  - 1st step, get a general feeling for the interaction flow and general scope
  - 2nd step (and following), focus on specific UI elements, knowing where they fit in the general picture
- Heuristics are used as a "reminder" of things to look for
  - Other types of problems can also be reported

Evaluation (II)

- Comments from each evaluator should be recorded or written
  - There may be an observer, taking notes
  - The observer may provide clarifications, especially it the evaluator is not a domain expert
- Session duration is normally 1h – 2h
- Each evaluator should provide a list of usability problems
  - Which heuristic (or other usability rule) has been violated, and why
    - Not a subjective comment, but a reference to a known principle
  - Each problem reported separately, in detail

https://www.nngroup.com/articles/usability-problems-found-by-heuristic-evaluation/

Evaluation (III)

- Where problems may be found
  - A single location in the UI
  - Two or more locations that need to be compared
  - Problem with the overall UI structure
  - Something is missing
    - May be due to prototype approximation
    - May still be unimplemented

What is a Tasks?

- «A **task** is a **goal** together with some ordered set of **actions.**» (Benyon)

Goal

- A state of the application domain that a work system (user+technology) wishes to achieve.
- Specified at particular levels of abstraction.

Task

- A structured set of activities required, used, or believed to be necessary by an agent (human, machine) to achieve a goal using a particular technology.
- The task is broken down into more and more detailed levels of description until it is defined in terms of actions.

Action

- An action is a task that has no problem solving associated with it and which does not include any control structure.
- Actions are 'simple tasks'.

All About Tasks

- Task: the structured **set of activities**/high-level actions required to **achieve** a user goal.
  - It says what a person *wants to do*, not how, and describe a *complete* goal.

- Often, given a domain, you have a **mix** of tasks with different **complexity**
  - Simple tasks – common or introductory
  - Moderate tasks
  - Complex tasks – infrequent or for power/extreme users

Sample Task: To Clean The House (I)

- **Steps**:
  - get the vacuum cleaner out
  - fix the appropriate attachments
  - clean the rooms
  - when the dust bag gets full, empty it
  - put the vacuum cleaner and tools away
- Must know and use different **artifacts**:
  - vacuum cleaners, their attachments, dust bags
  - cupboards, rooms
  - …

Sample Task: To Clean The House (II)

- **Goals:**
  - Here your *point of view* comes in
  - Removing dust? -> narrow goal
  - Tidying up the house after a party?
  - Hosting people for the dinner?
  - Having a satisfying evening? -> wide goal

Sample Task: To Clean The House (III)

- **Pain points**:
  - Narrow version: Why I need to empty the dust bag?
  - Broader version: Why I need a vacuum cleaner to have the house cleaned up?

Example of Good Tasks

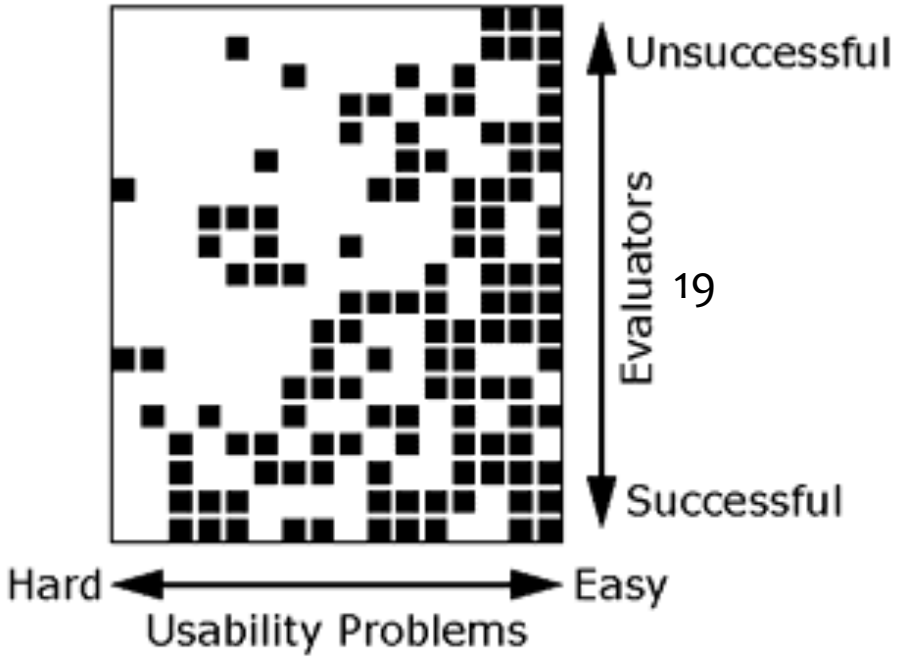
- Service/App: Uber
- Simple task: signaling for a ride
  - *Is it a task? Why is it simple?*
- Moderate task: reach out to the driver to get a forgotten object
  - *Is it a task? Why is it moderate?*
- Complex task: become a driver for Uber
  - *Is it a task? Why is it complex?*

Example of Bad Tasks

- Service/App: Uber
- Open the app and tap on “Travel”
  - *Is it a task? Why is it bad?*
- Go into your account settings, check the messages, and then send a present
  - *Is it a task? Why is it bad?*
- …

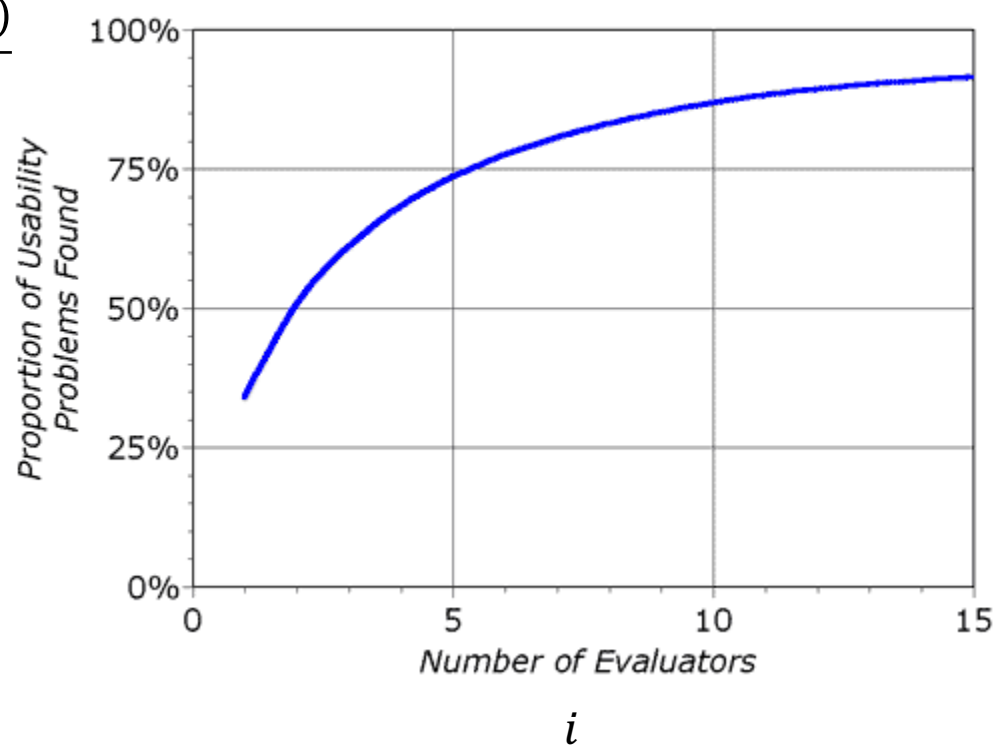
Multiple Evaluators

- No evaluator finds all problems
  - Even the best one finds only ~1/3
- Different evaluators find different problems
  - Substantial amount of nonoverlap
- Some evaluators find more problems than others



How Many Evaluators

$\frac{PF(i)}{N}$



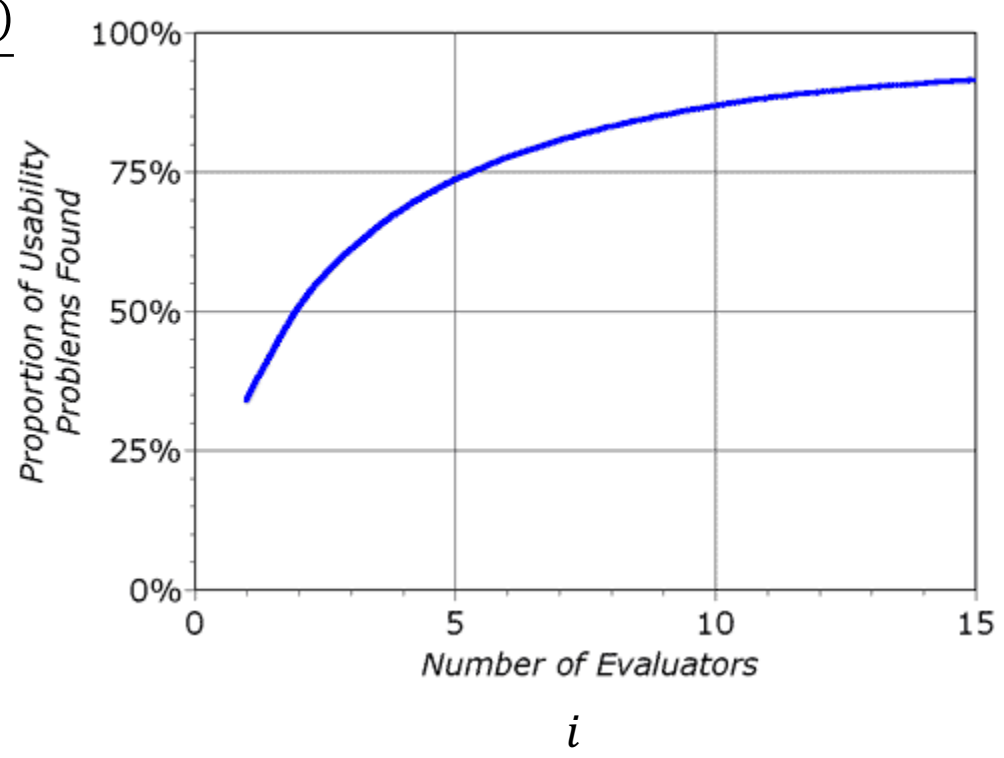
i

- $PF(i) = N\left(1 - (1 - l)^i\right)$
- $PF(i)$: problems found
- i: number of *independent* evaluators
- N: number of existing (but unknown) usability problems
- l: ratio of usability problems found by a single evaluator

How Many Evaluators

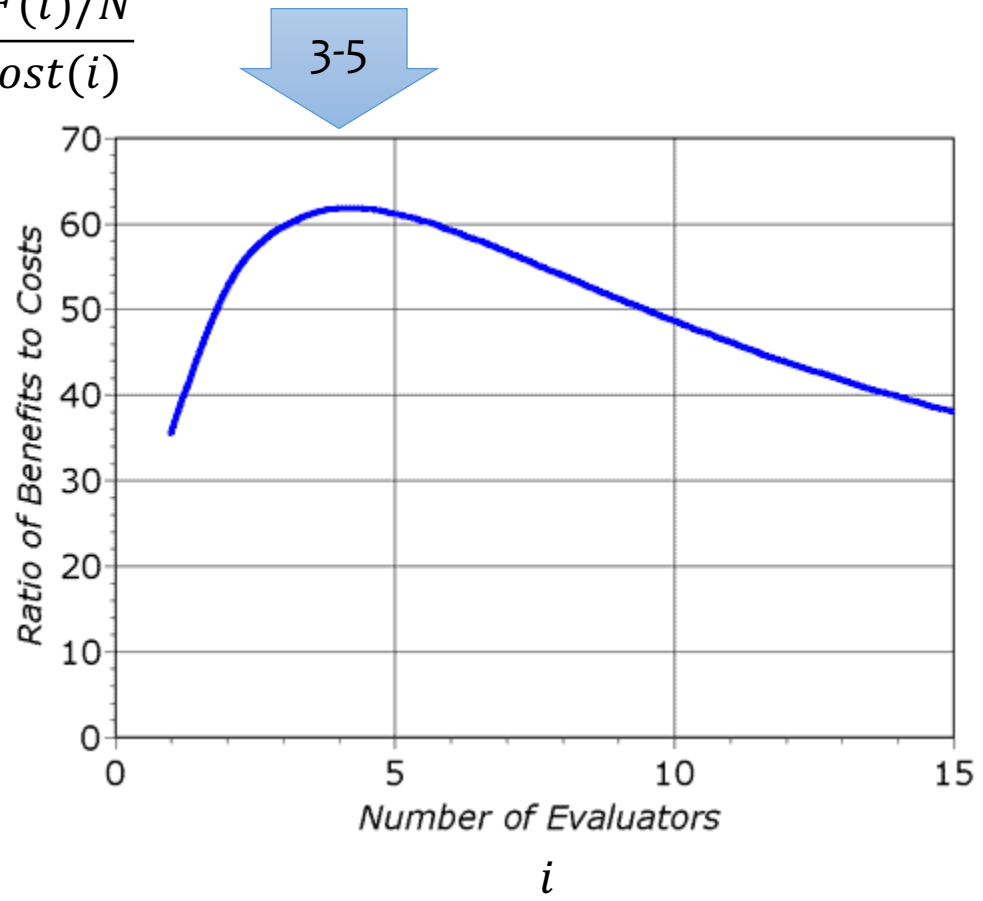
$$Cost(i) = \text{Fixed} + \text{Fee} \times i$$

$\frac{PF(i)}{N}$



i

$\frac{PF(i)/N}{Cost(i)}$

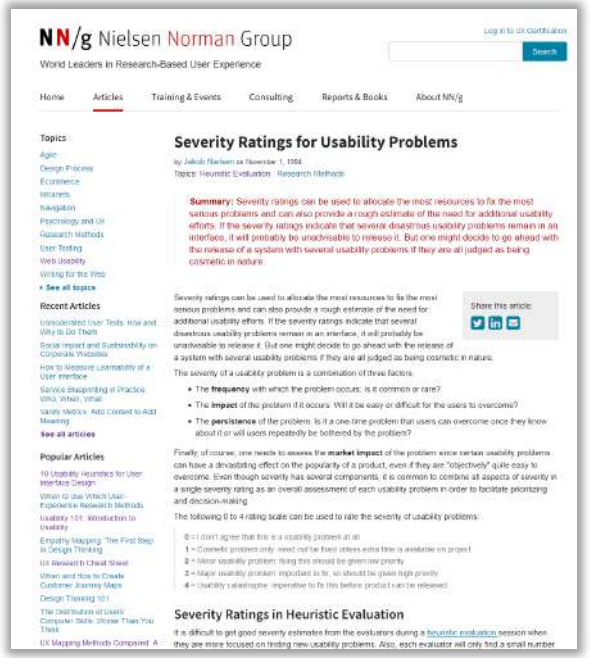


i

https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/

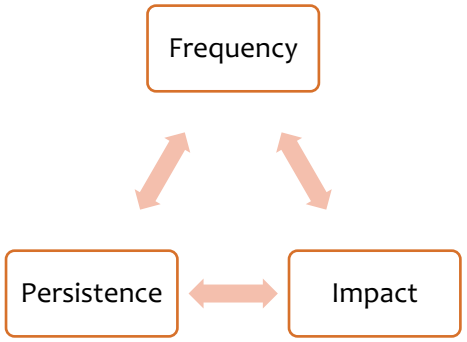
Severity Rating

- We need to allocate the most resources to fix the most serious problems
- We need to understand if additional usability efforts are required
- **Severity** is a combination of:
  - **Frequency** with which the problem occurs: common or rare?
  - **Impact** of the problem if it occurs: easy to overcome or difficult?
  - **Persistence**, is it one-time or will it occur many times to users?
- Define a *combined severity rating*
  - Individually, for each evaluator



Severity Ratings scale

0	No problem	I don't agree that this is a usability problem at all
1	**Cosmetic** problem only	need not be fixed unless extra time is available on project
2	**Minor** usability problem	fixing this should be given low priority
3	**Major** usability problem	important to fix, so should be given high priority
4	Usability **catastrophe**	imperative to fix this before product can be released



Combined Severity Ratings

- Severity ratings from *one* evaluator have been found *unreliable*, they should not be used
- After all evaluators completed their rankings
  - Either let them discuss, and agree on a consensus ranking
  - Or just compute the average of the 3-5 ratings

Debriefing

- Meeting of all evaluators, with observers, and members of the *development* team
- Line-by-line analysis of the problems identified
  - Discussion: how can we fix it?
  - Discussion: how much will it cost to fix it?
- Can also be used to brainstorm general design ideas

Heuristic Evaluation vs. User Testing

Heuristic Evaluation

- Faster (1-2h per evaluator)
- Results are pre-interpreted (thanks to the evaluators)
- Could generate *false positives*
- Might *miss* some problems

User Testing

- Need to develop software, and prepare the set-up
- More accurate (by definition!)
  - Actual users and tasks

Heuristic Evaluation vs. User Testing

Heuristic Evaluation

- Faster (1-2h per evaluator)
- Results are pre-interpreted (thanks to the evaluators)
- Could generate *false positives*
- Might *miss* some

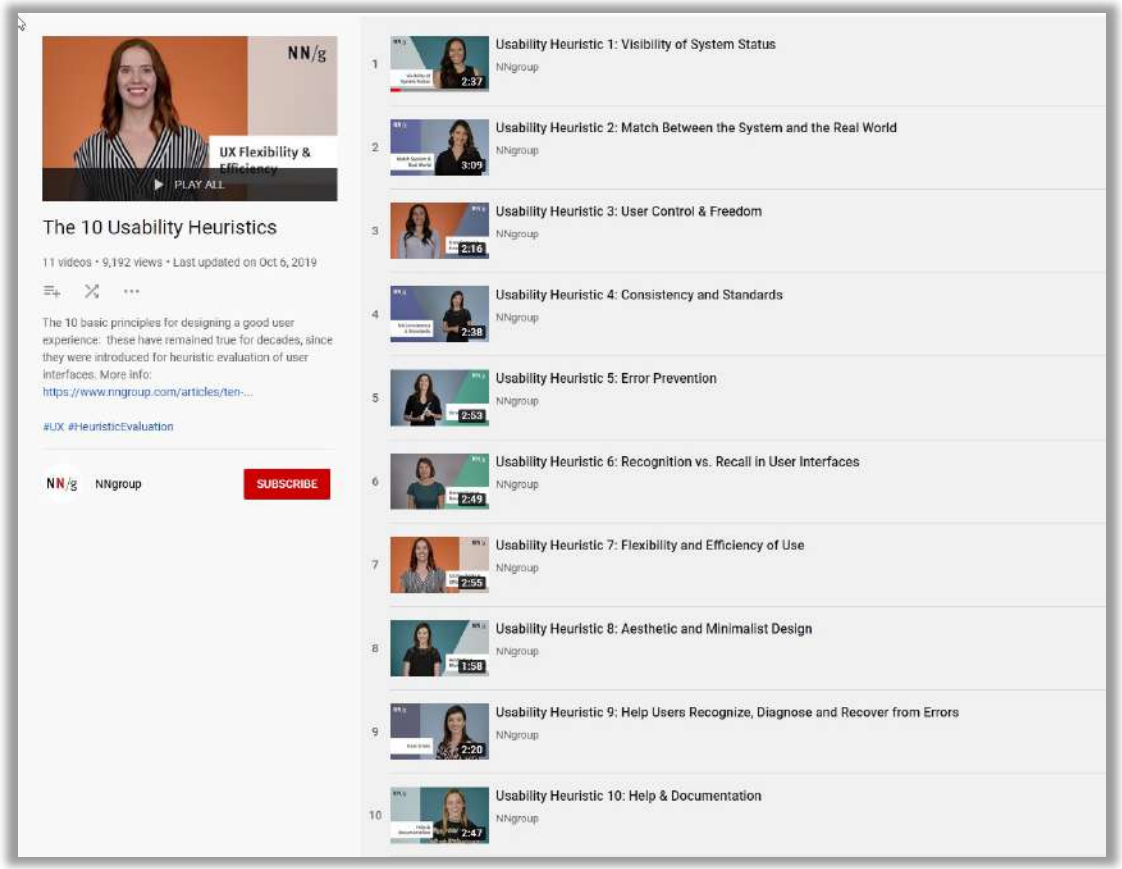
User Testing

- Need to develop software, and prepare the set-up
- More accurate (by definition!)
  - Actual users and tasks

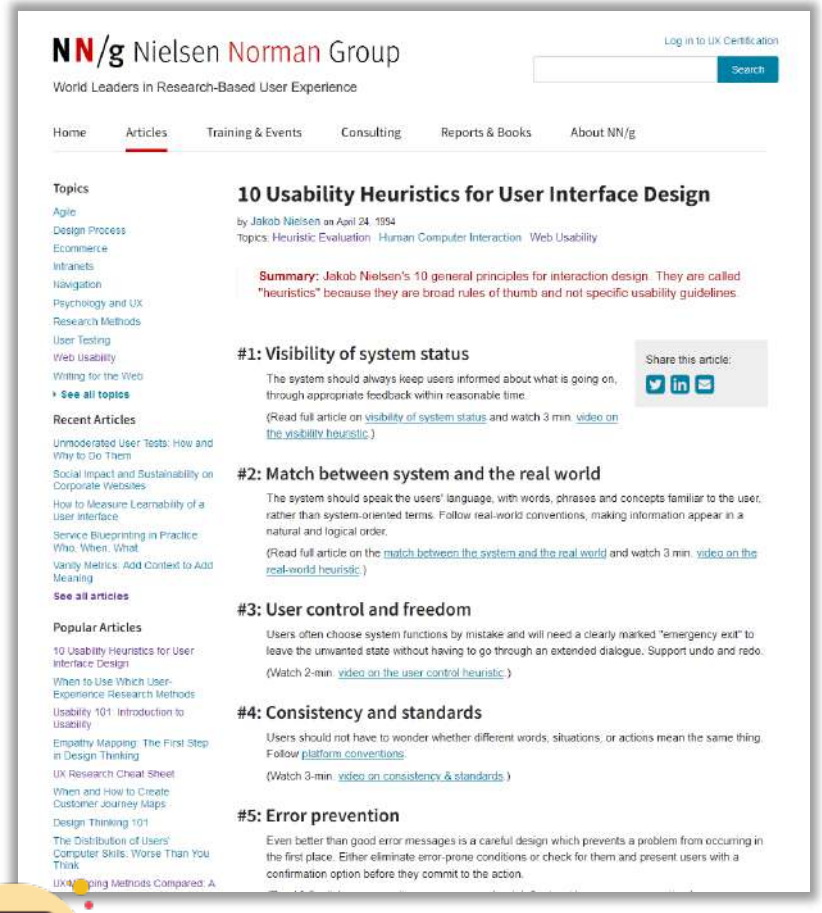
- Alternate the methods!
  - Find different problems
  - Do not waste participants

https://www.nngroup.com/articles/usability-problems-found-by-heuristic-evaluation/

10 Nielsen's Usability Heuristics



https://www.youtube.com/playlist?list=PLJOFJ3Ok_idtb2YeifXlG1-TYoMBLoG6I

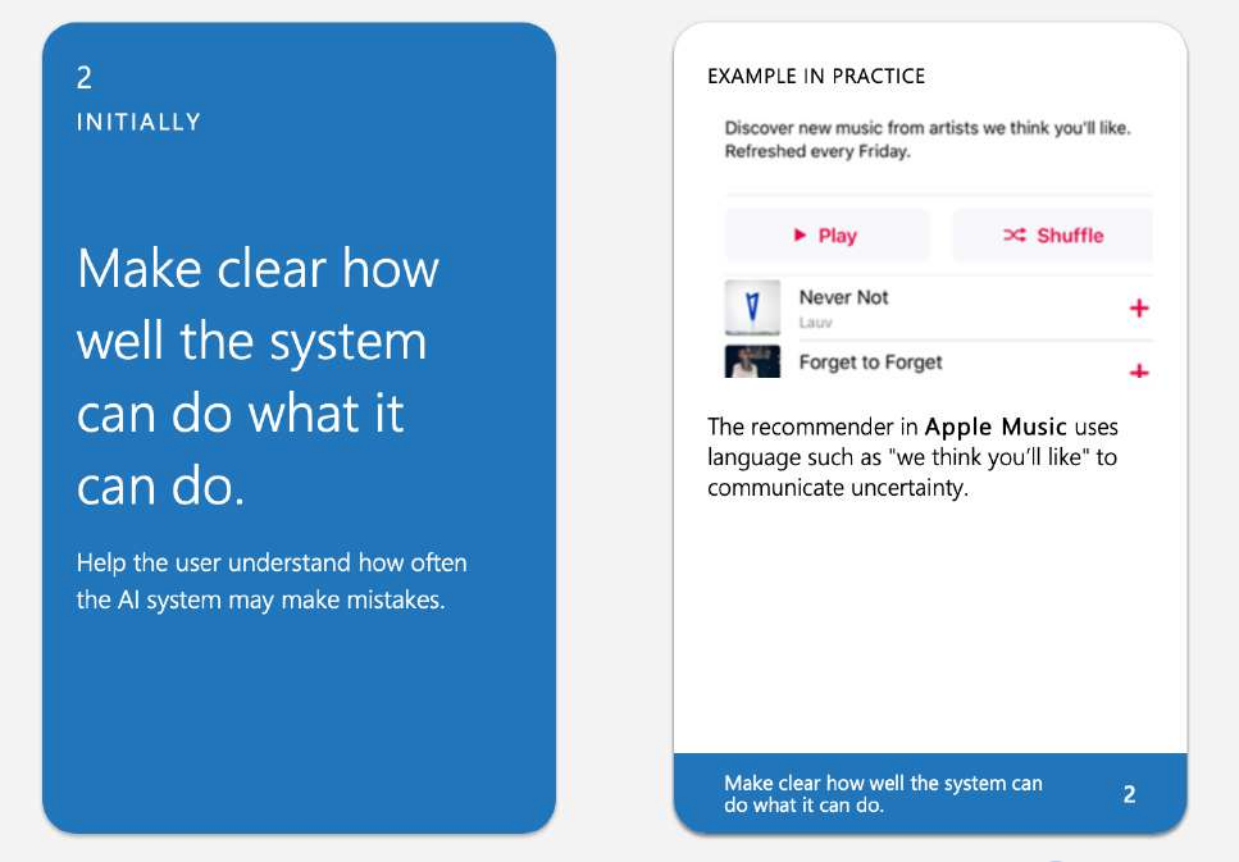


https://www.nngroup.com/articles/ten-usability-heuristics/

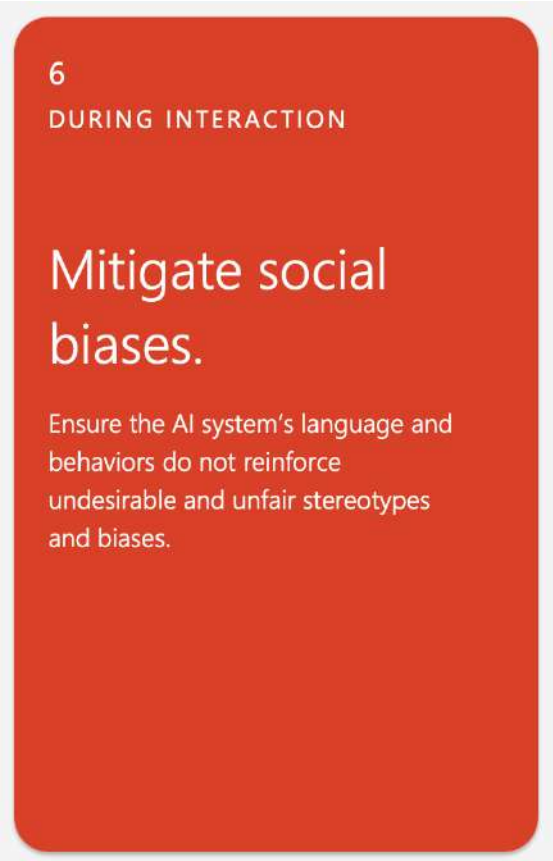
“Custom” Heuristic Evaluations: Guidelines for Human-AI Interaction

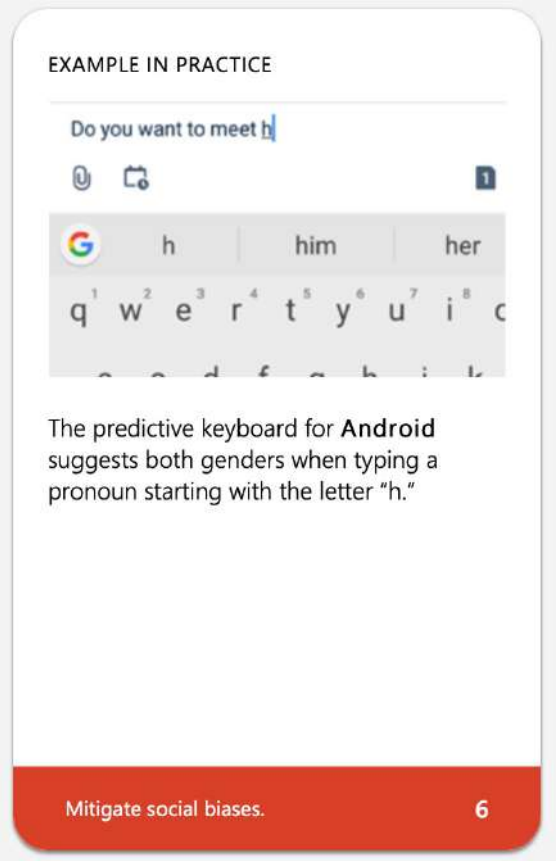
By Microsoft Research: https://www.microsoft.com/en-us/research/project/guidelines-for-human-ai-interaction/

“Custom” Heuristic Evaluations: Guidelines for Human-AI Interaction

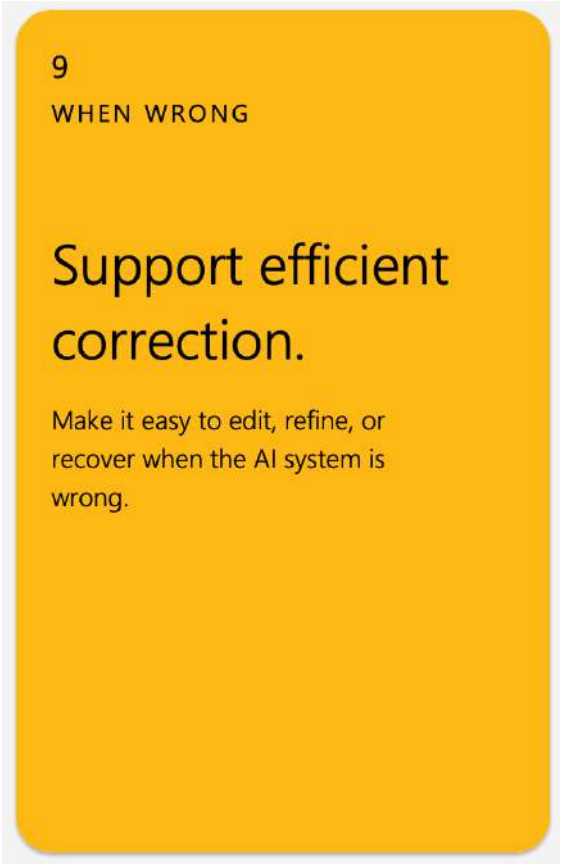


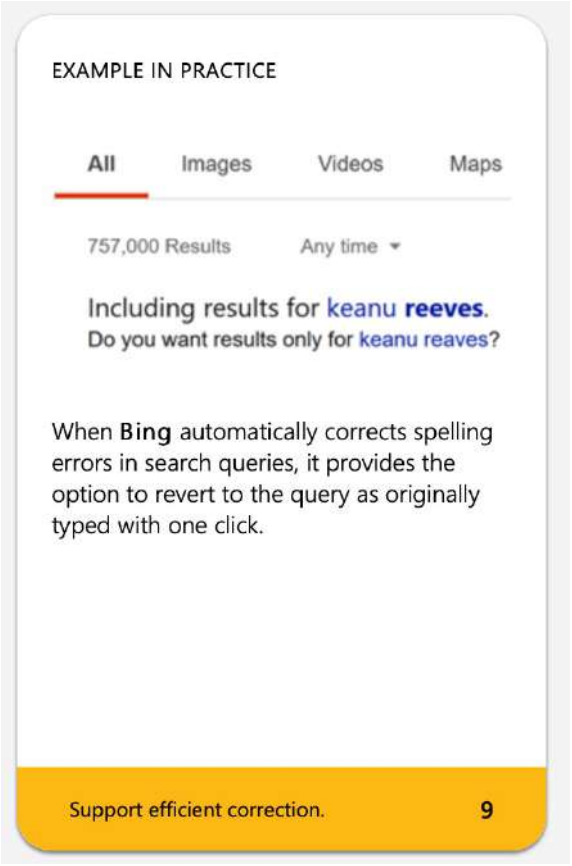
“Custom” Heuristic Evaluations: Guidelines for Human-AI Interaction



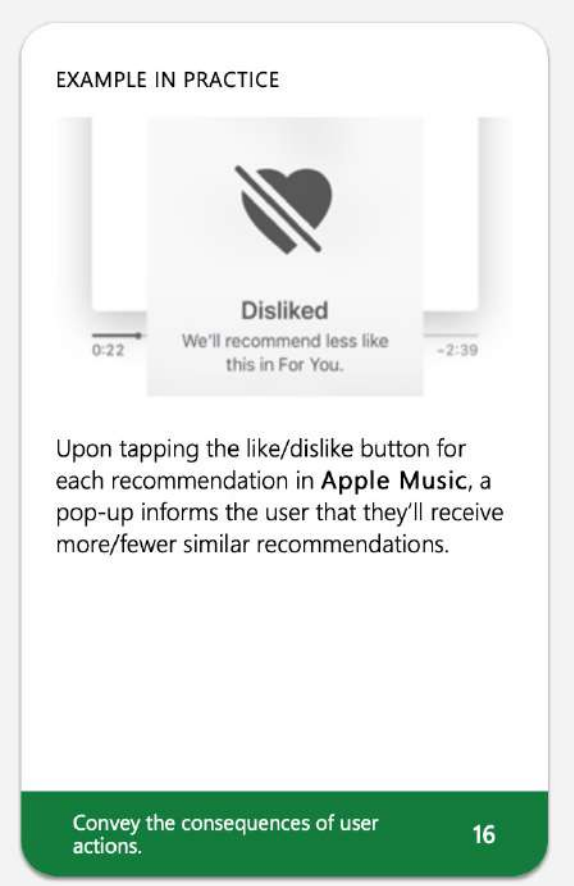


“Custom” Heuristic Evaluations: Guidelines for Human-AI Interaction





“Custom” Heuristic Evaluations: Guidelines for Human-AI Interaction

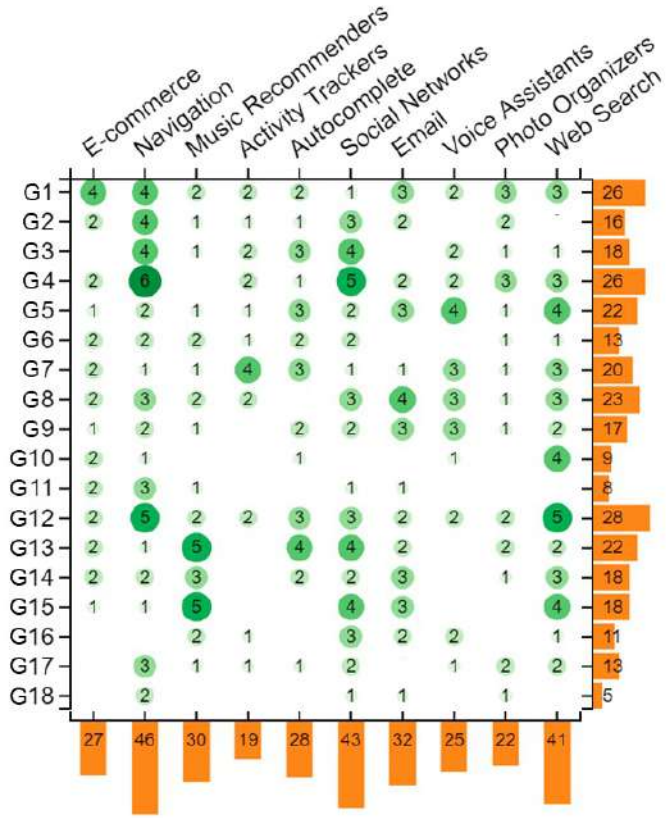


“Custom” Heuristic Evaluations: Guidelines for Human-AI Interaction

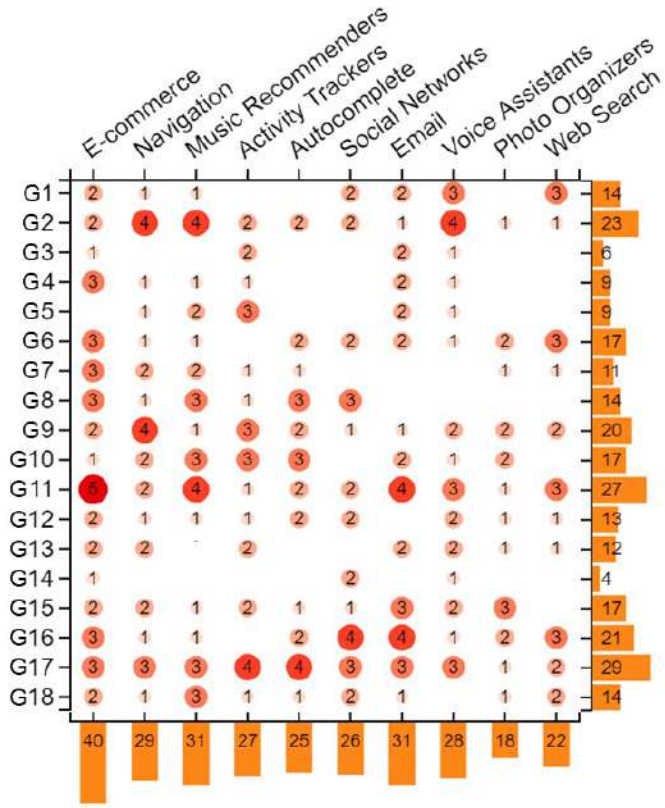
- Each participant was assigned to an AI-driven feature of a product they were familiar with and asked to find examples (applications and violations) of each guideline;
- For each guideline, researchers asked participants first to determine if it “does not apply” to their assigned feature (i.e., irrelevant or out of scope).
- If relevant, researchers asked participants to provide their examples of applications and violations of the guideline, rating the extent of the application or violation on a 5-point semantic differential scale from “clearly violated” to “clearly applied,” along with an explanation of the rating.

Amershi et al., *Guidelines for Human-AI Interaction,* CHI 2019, https://dl.acm.org/doi/10.1145/3290605.3300233

“Custom” Heuristic Evaluations: Guidelines for Human-AI Interaction



Counts of “clear application” or “application”



Counts of “clear violation” or “violation” responses.

Amershi et al., *Guidelines for Human-AI Interaction,* CHI 2019, https://dl.acm.org/doi/10.1145/3290605.3300233

Acknowledgements and Thanks

- Slides on heuristic evaluation are from the Human-Computer Interaction course of the Politecnico di Torino (http://bit.ly/polito-hci)

License

- These slides are distributed under a Creative Commons license "**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**"

- **You are free to:**
  - **Share** — copy and redistribute the material in any medium or format
  - **Adapt** — remix, transform, and build upon the material
  - The licensor cannot revoke these freedoms as long as you follow the license terms.
- **Under the following terms:**
  - **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - **NonCommercial** — You may not use the material for commercial purposes.
  - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.
- https://creativecommons.org/licenses/by-nc-sa/4.0/