

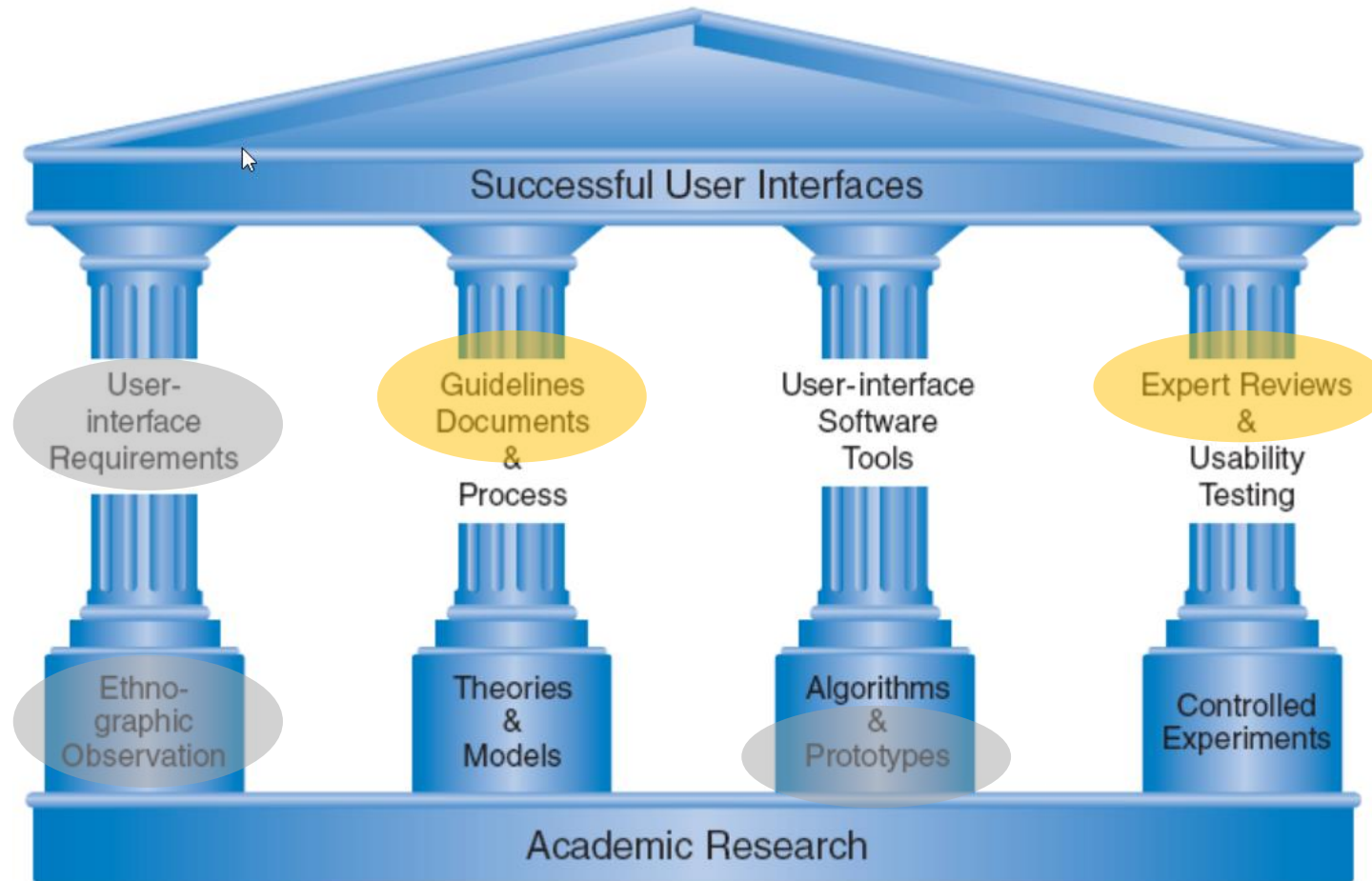
Design Evaluation: Heuristics

Human Computer Interaction

Fulvio Corno, Luigi De Russis

Academic Year 2019/2020

The Four Pillars of Design



Ben Shneiderman & Catherine Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction

Goals

Generating design solutions



☐ Guidelines

☐ Principles

☐ Theories

Evaluating generated designs



☐ Expert reviews and heuristics

☐ Usability testing

☐ Controlled experiments

Evaluation

Testing the usability, functionality and acceptability of an interactive system

Goal

- Evaluation: «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 the different usability dimensions
 - Using a range of different techniques
- Identify and correct usability issues as soon as possible

Evaluation approaches

- Evaluation may take place:
 - In the laboratory
 - In the field
- Involving users (Empirical Evaluation):
 - 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 evaluation:
 - Simulation and software measures
 - Formal evaluation with models and formulas
 - Especially for low-level issues

Cognitive walkthrough

A simple technique to analyze all individual step in an interaction path

Cognitive Walkthrough

- Step-by-step revision of a sequence of actions (interaction steps) to perform a given task
- Evaluators examine each step, looking for possible problems
- Particularly suited for systems designed for learning-by-exploration

Walkthrough organization

Walkthrough specification

- A specification or prototype of the system
- A description of the task the user is to perform on the system
- A complete, written list of the actions needed to complete the task
- An indication of who the users are (experience, knowledge)

For each step, you must check

- Is the *effect* of the action the same as the *user's goal* at that point?
- Will users see that the action is available?
- Once users have found the *correct action*, will they know it is the one they need?
- After the action is taken, will users *understand the feedback* they get?

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

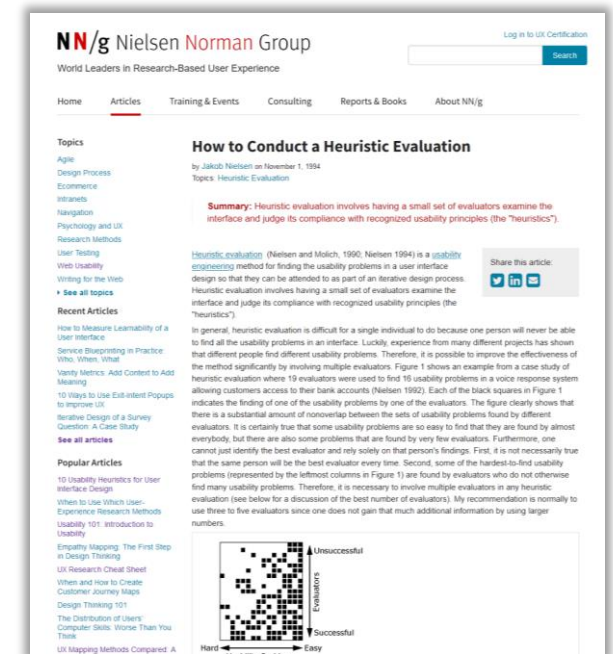
Heuristic Evaluation

- 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, ...)
 - Goal: find usability problems in a design
- Also popularized as “Discount Usability”



Basic idea

- Define a set of heuristics (or principles)
- 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



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

Heuristics

- Nielsen proposed 10 heuristic rules
 - Good at finding most design problems
 - Inspired and connected to the Design Principles (→Guidelines)
- 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

- Pre-evaluation training
 - Give evaluator information about the domain and the scenario to be evaluated
- Evaluation
 - Individual
- Severity Rating
 - First, individually
 - Then, aggregate and find consensus
- 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 if 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

Evaluation (III)

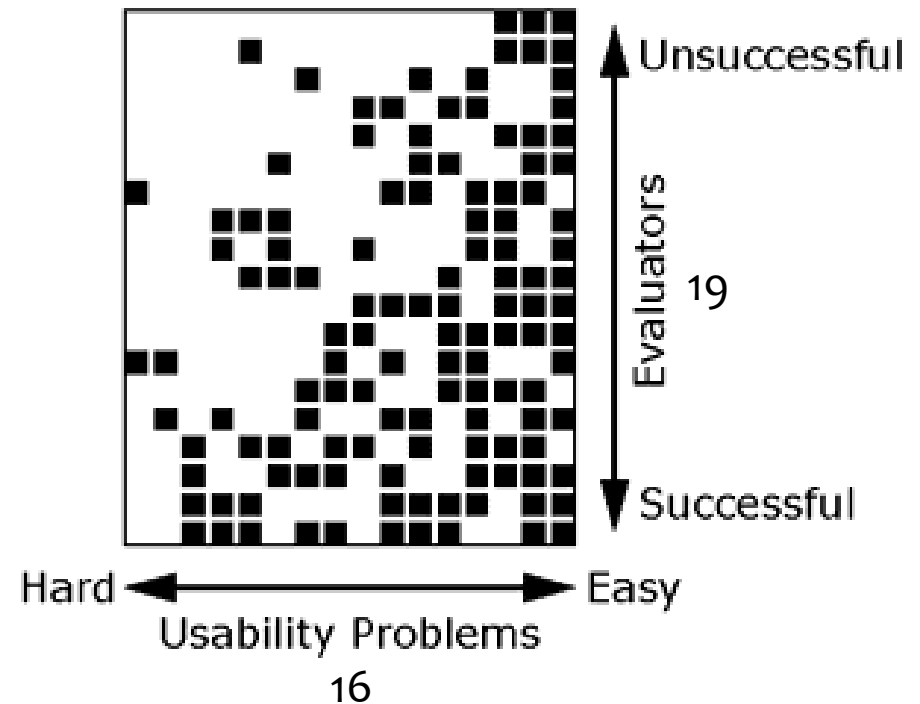
- Where problems may be found
 - A single location in the UI
 - Two or more locations that need to be compared
 - Problem with the overall UI structure
 - Something is missing
 - May be due to prototype approximation
 - May still be unimplemented



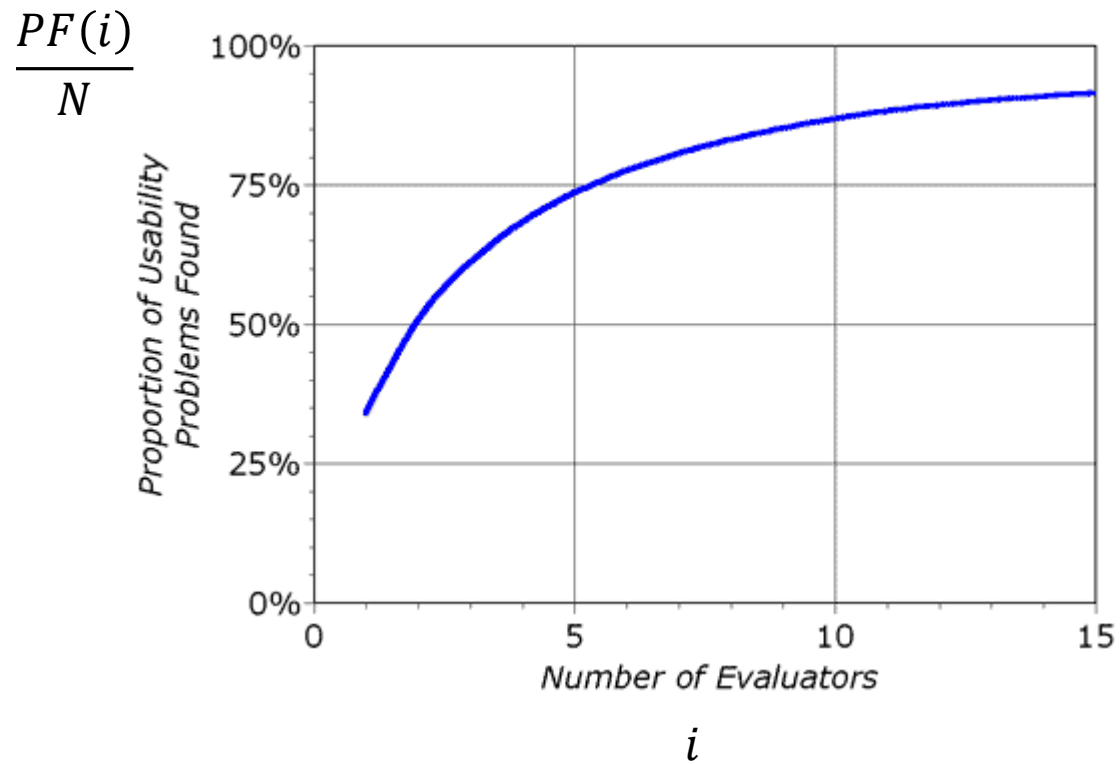
<https://www.nngroup.com/articles/usability-problems-found-by-heuristic-evaluation/>

Multiple evaluators

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



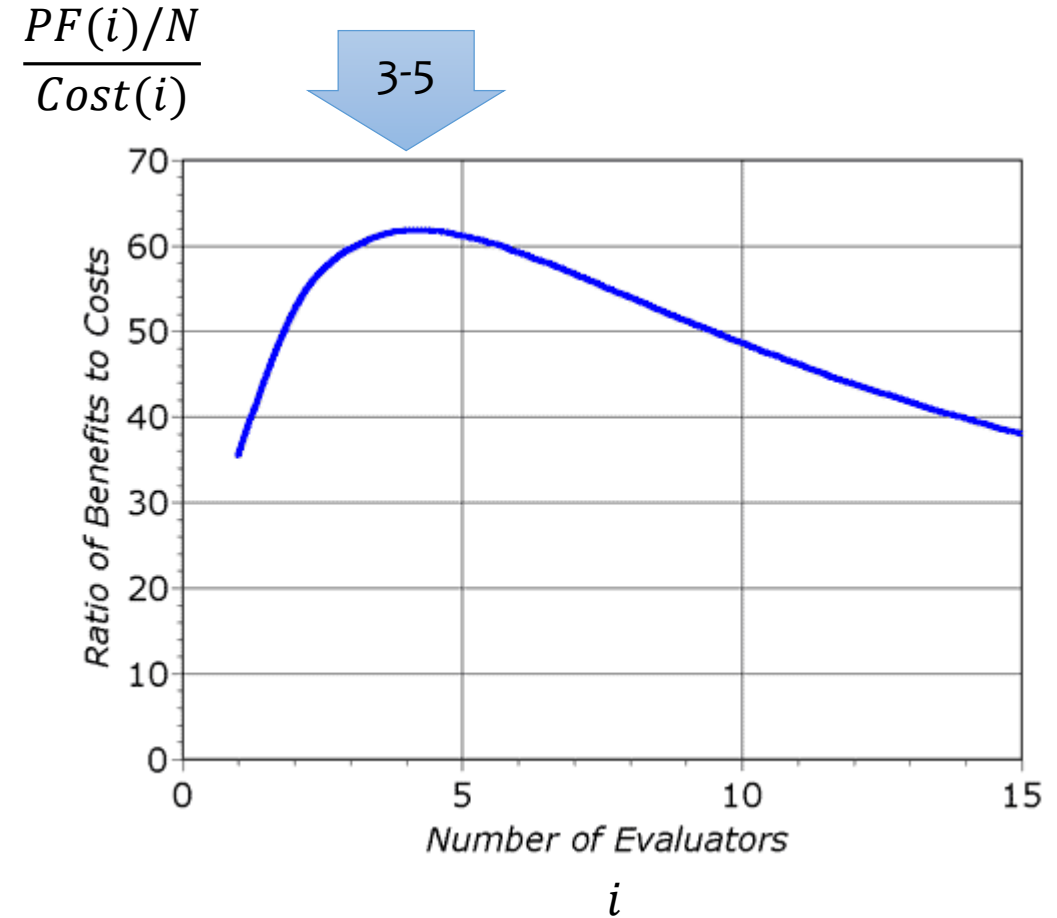
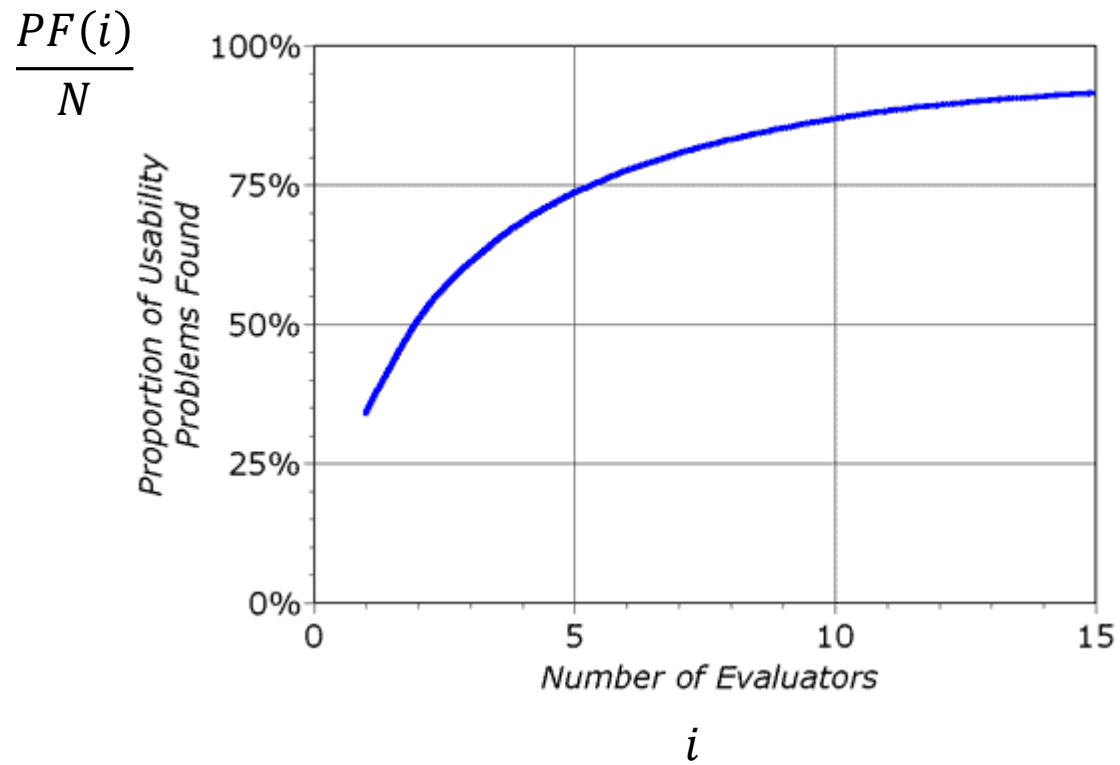
How many evaluators?



- $PF(i) = N(1 - (1 - l)^i)$
- $PF(i)$: 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

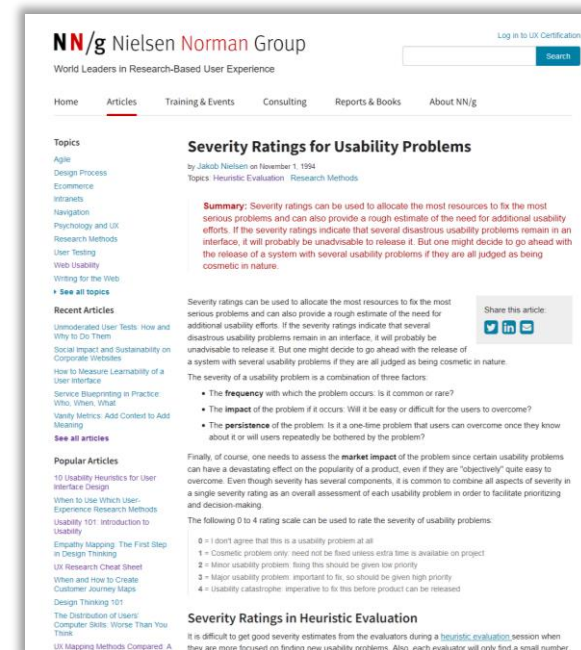
How many evaluators?

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



Severity Rating

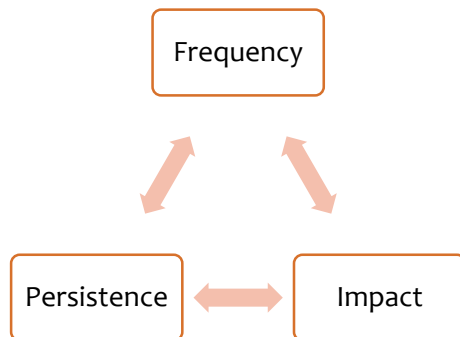
- We need to allocate the most resources to fix the most serious problems
- We need to understand if additional usability efforts are required
- **Severity** is a combination of:
 - **Frequency** with which the problem occurs: common or rare?
 - **Impact** of the problem if it occurs: easy to overcome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



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

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 sw, and prepare the set-up
- More accurate (by definition!)
 - Actual users and tasks

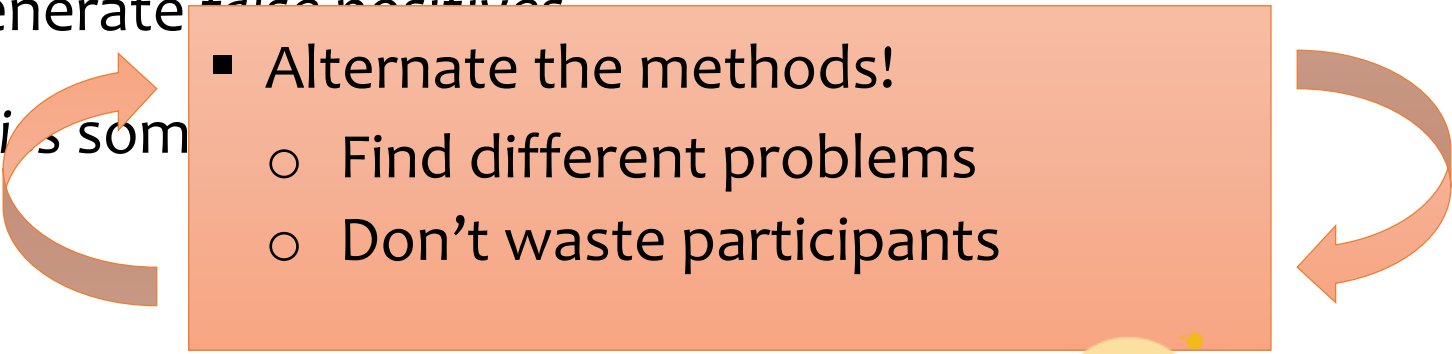
Heuristic Evaluation vs User Testing

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 sw, and prepare the set-up
- More accurate (by definition!)
 - Actual users and tasks

- 
- Alternate the methods!
 - Find different problems
 - Don't waste participants




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

Nielsen's Usability Heuristics

10 Usability Principles to be used in Heuristic Evaluation

10 Nielsen's Usability Heuristics



The 10 Usability Heuristics

11 videos • 9,192 views • Last updated on Oct 6, 2019

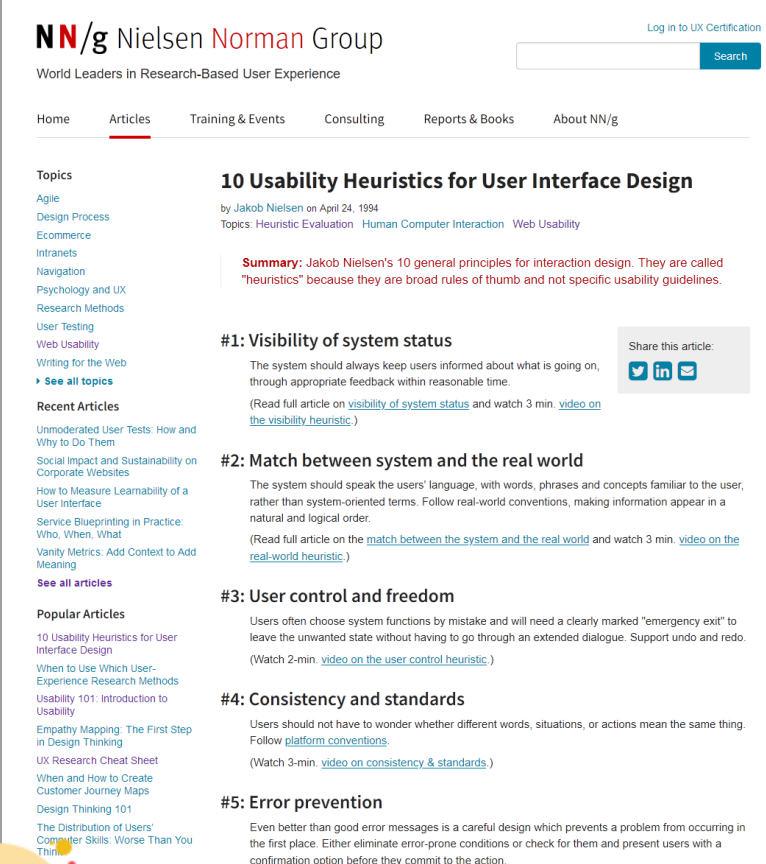
The 10 basic principles for designing a good user experience: these have remained true for decades, since they were introduced for heuristic evaluation of user interfaces. More info: <https://www.nngroup.com/articles/ten-usability-heuristics/>

#UX #HeuristicEvaluation

Subscribe

- 1 Usability Heuristic 1: Visibility of System Status
NN/g 2:37
- 2 Usability Heuristic 2: Match Between the System and the Real World
NN/g 3:09
- 3 Usability Heuristic 3: User Control & Freedom
NN/g 2:16
- 4 Usability Heuristic 4: Consistency and Standards
NN/g 2:38
- 5 Usability Heuristic 5: Error Prevention
NN/g 2:53
- 6 Usability Heuristic 6: Recognition vs. Recall in User Interfaces
NN/g 2:49
- 7 Usability Heuristic 7: Flexibility and Efficiency of Use
NN/g 2:55
- 8 Usability Heuristic 8: Aesthetic and Minimalist Design
NN/g 1:58
- 9 Usability Heuristic 9: Help Users Recognize, Diagnose and Recover from Errors
NN/g 2:20
- 10 Usability Heuristic 10: Help & Documentation
NN/g 2:47

 https://www.youtube.com/playlist?list=P_LJOFJ3Ok_idtb2YeifXlG1-TYoMBLoG6I



NN/g Nielsen Norman Group
World Leaders in Research-Based User Experience

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

Log in to UX Certification Search

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

Recent Articles
Unmoderated User Tests: How and Why to Do Them
Social Impact and Sustainability on Corporate Websites
How to Measure Learnability of a User Interface
Service Blueprinting in Practice: Who, When, What
Vanity Metrics: Add Context to Add Meaning
See all articles

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
Mapping Methods Compared: A

10 Usability Heuristics for User Interface Design

by Jakob Nielsen on April 24, 1994
Topics: Heuristic Evaluation Human Computer Interaction Web Usability

Summary: Jakob Nielsen's 10 general principles for interaction design. They are called "heuristics" because they are broad rules of thumb and not specific usability guidelines.

Share this article:
[Twitter](#) [LinkedIn](#) [Email](#)

#1: Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
(Read full article on [visibility of system status](#) and watch 3 min. [video on the visibility heuristic](#).)

#2: 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.
(Read full article on the [match between the system and the real world](#) and watch 3 min. [video on the real-world heuristic](#).)

#3: 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.
(Watch 2-min. [video on the user control heuristic](#).)

#4: Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow [platform conventions](#).
(Watch 3-min. [video on consistency & standards](#).)

#5: 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.

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



10 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

#1: Visibility of system status

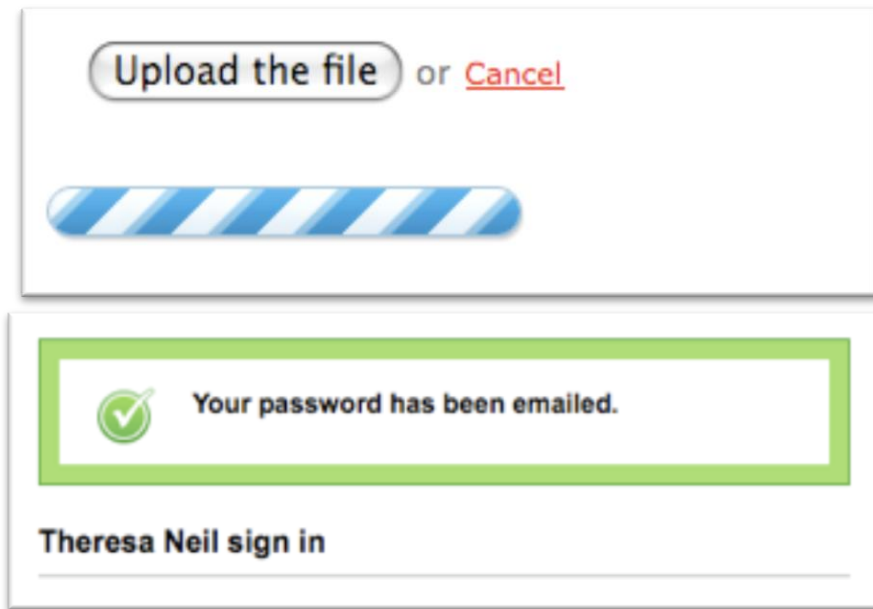
- The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.



<https://www.nngroup.com/articles/visibility-system-status/>

#1: Visibility of system status

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



#2: Match between system and the real world

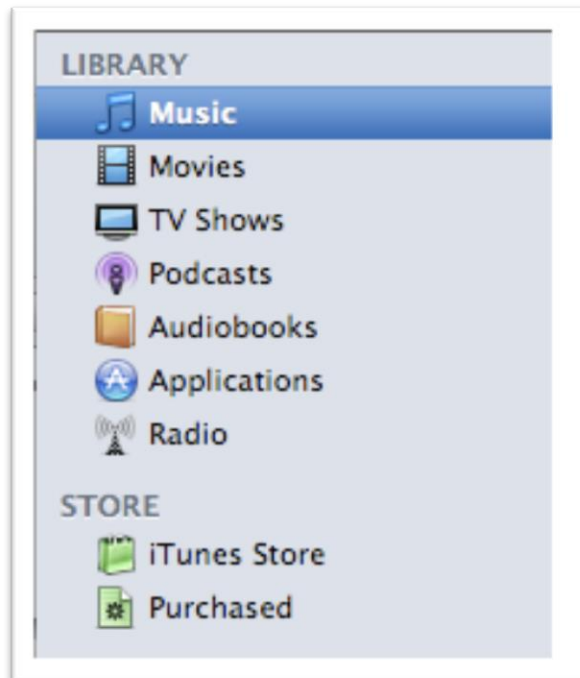
- The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.



<https://www.nngroup.com/articles/match-system-real-world/>

#2: 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.

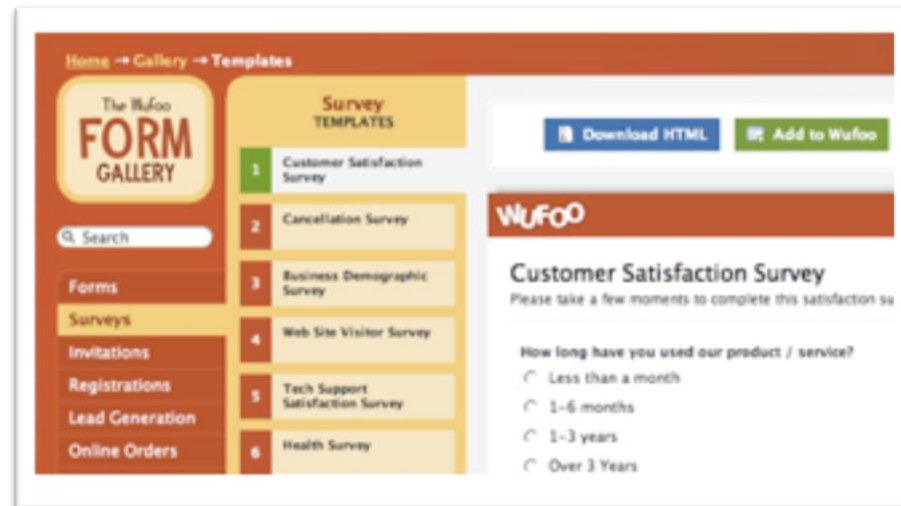
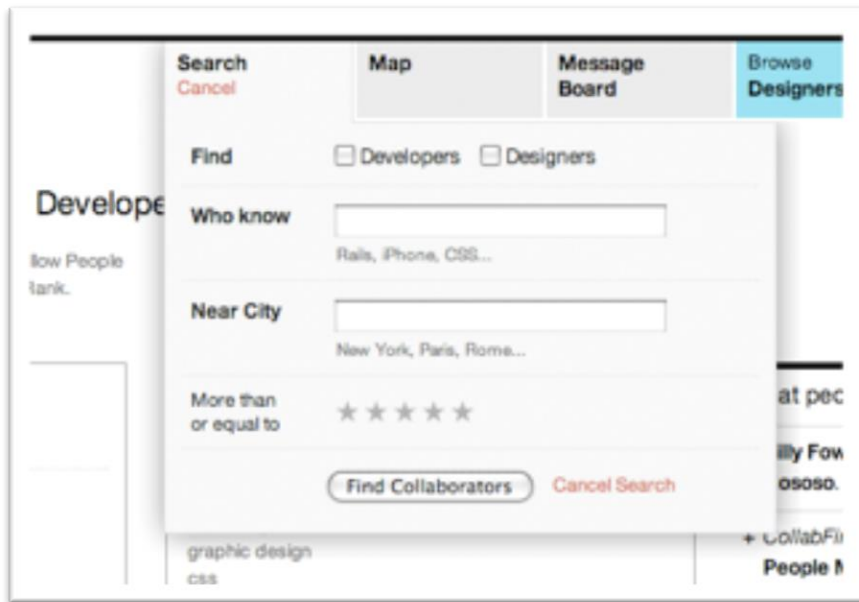


#3: User control and freedom

- 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.

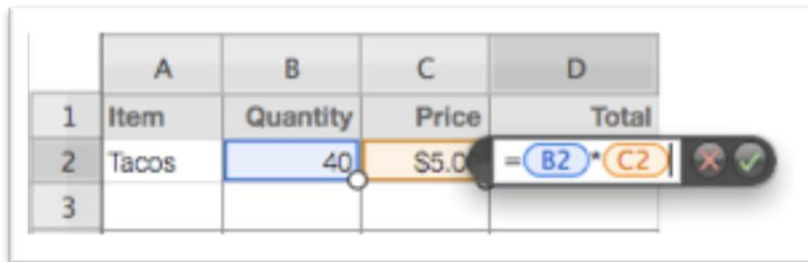
#3: User control and freedom

- 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.



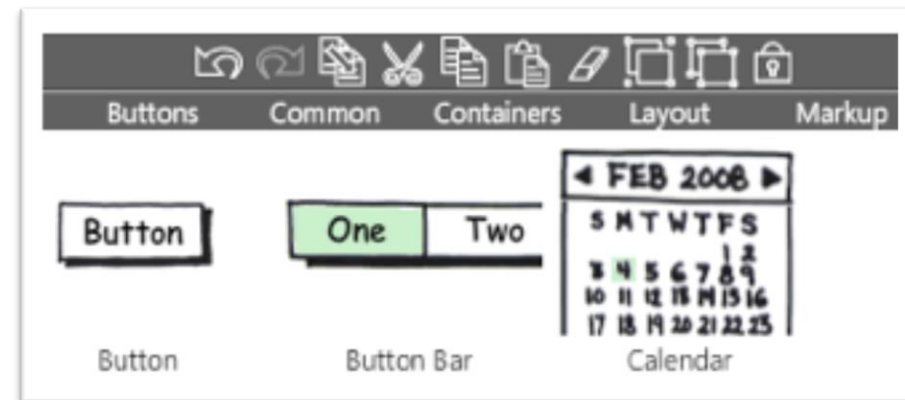
#3: User control and freedom

- 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.



| | A | B | C | D |
|---|-------|----------|--------|--------|
| 1 | Item | Quantity | Price | Total |
| 2 | Tacos | 40 | \$5.00 | =B2*C2 |
| 3 | | | | |

A small dialog box is overlaid on the formula bar, containing a red 'X' button and a green checkmark button.

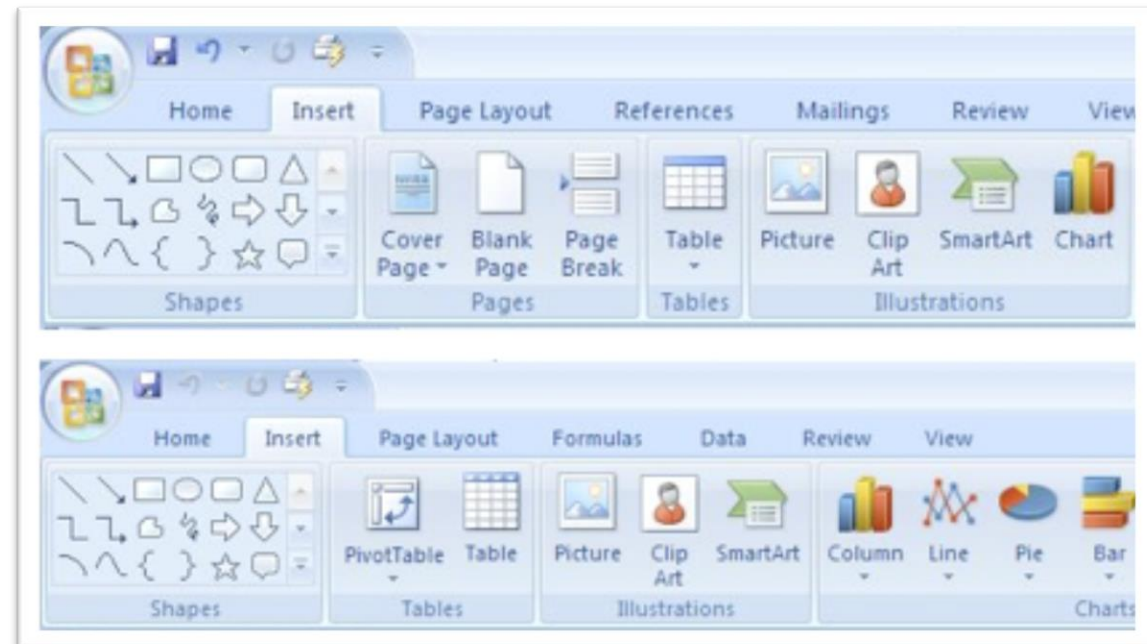
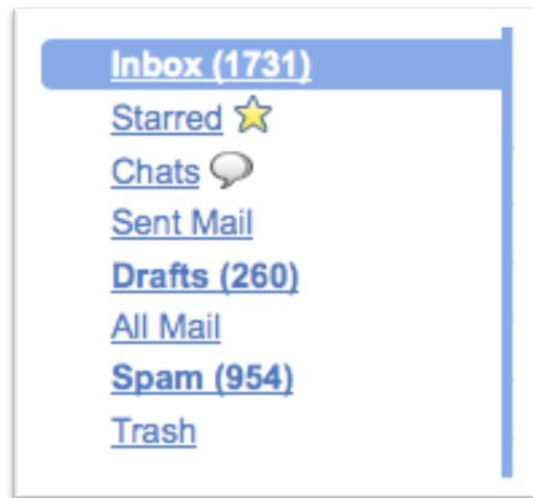


#4: Consistency and standards

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

#4: Consistency and standards

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



#5: Error prevention

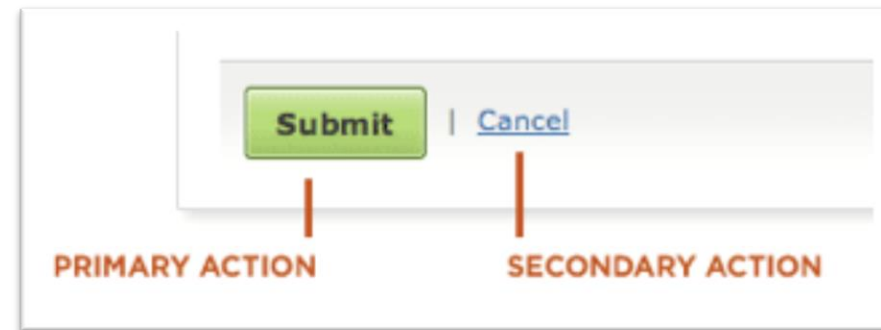
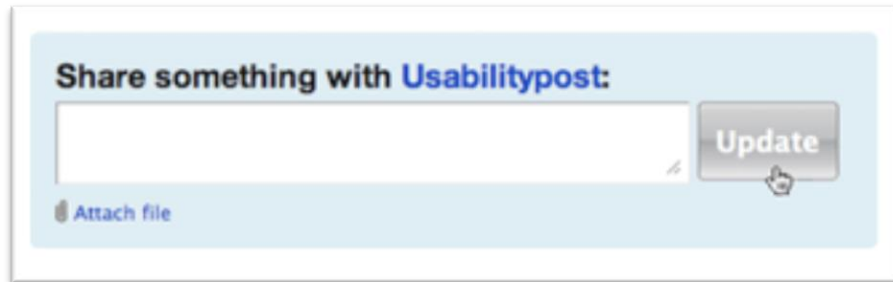
- Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.



<https://www.nngroup.com/articles/slips/>

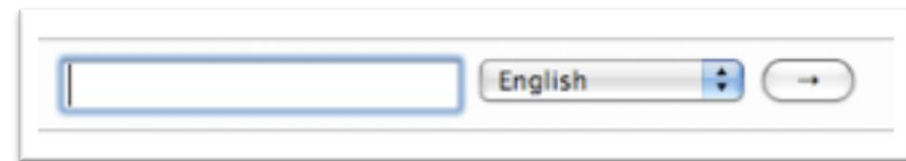
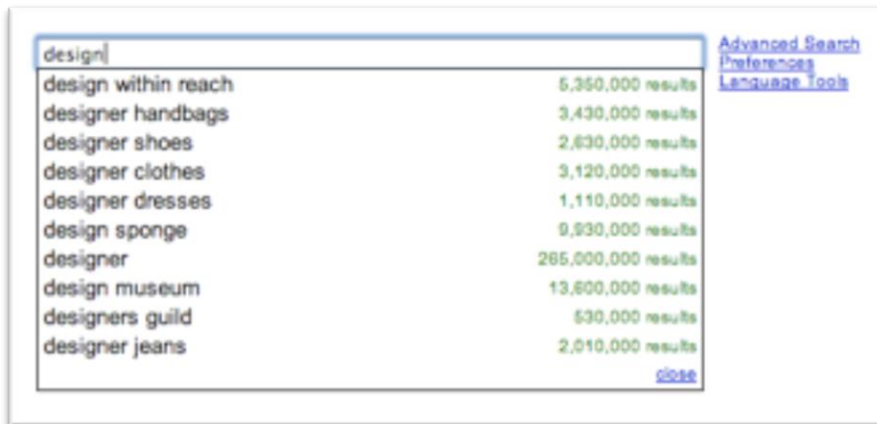
#5: Error prevention

- 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

- 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.



#6: Recognition rather than recall

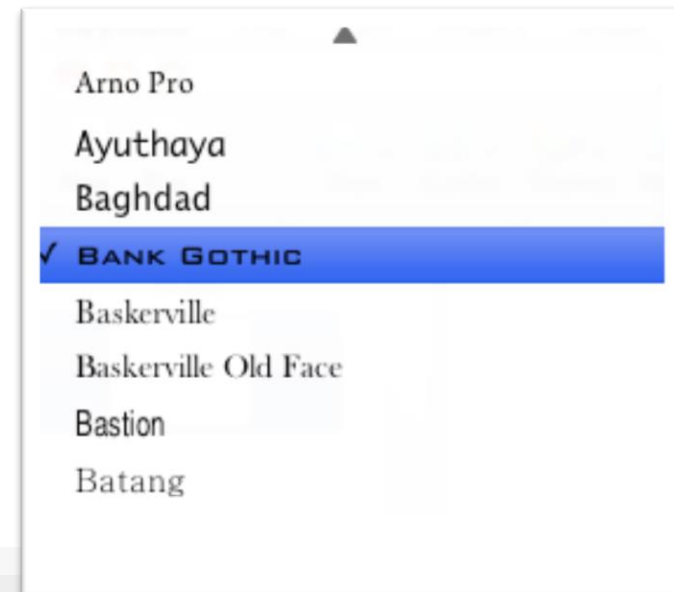
- Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.



<https://www.nngroup.com/articles/recognition-and-recall/>

#6: 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.

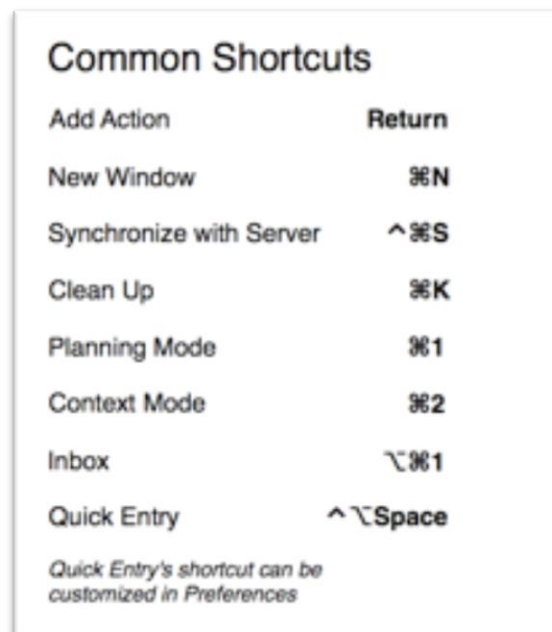


#7: Flexibility and efficiency of use

- 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 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.



Styles

- Basic
- Basic (No Grid)
- Gray
- Gray Headers
- Gray Fill
- Beige
- Ledger
- Blue
- Blue Headers
- Blue Fill
- Gravity**

| | A | B | C |
|----|---------------------|---------|---------|
| 3 | Mean | 1.81 | 1.85 |
| 4 | Median | 1.81 | 1.85 |
| 5 | Standard deviation | 0.03 | 0.04 |
| 6 | Variance | 0.00086 | 0.00138 |
| 7 | Alpha | 0.05 | 0.05 |
| 8 | T-value | 2.26 | 2.26 |
| 9 | Confidence interval | 0.01820 | 0.02304 |
| 10 | Upper limit | 1.82620 | 1.87704 |
| 11 | Lower limit | 1.78980 | 1.83096 |
| 12 | T-interval | 0.02100 | 0.02659 |
| 13 | Upper limit | 1.82900 | 1.88059 |
| 14 | Lower limit | 1.78700 | 1.82741 |

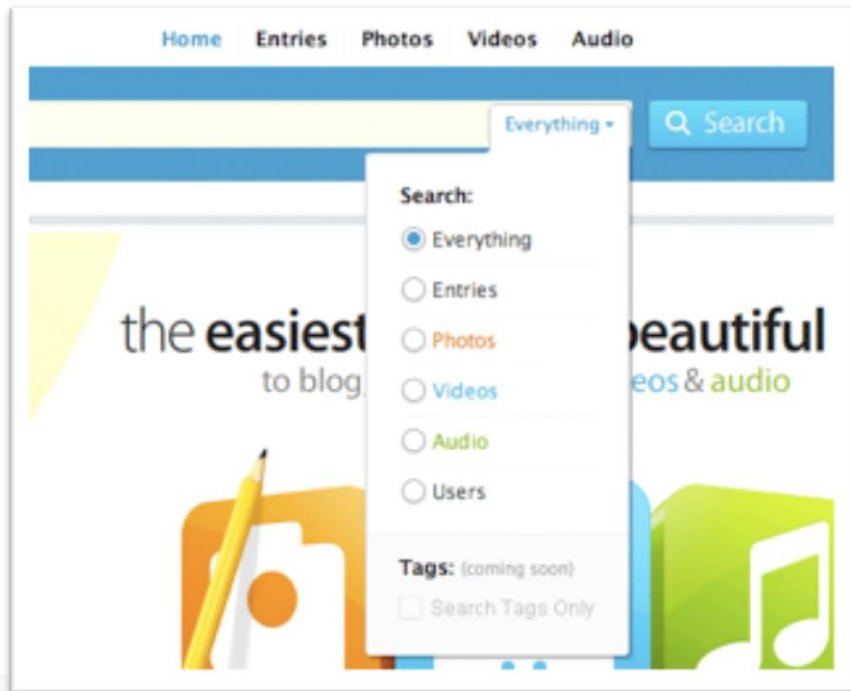
sum 23.2264292787289
avg 1.78664840605607
min 0.000862222222222...
max 10
count 13

#8: 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 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.



The screenshot shows a timesheet application for Theresa Neil. The title is 'Timesheet for Theresa Neil'. The date range is '04 May 2009 - 10 May 2009'. The interface includes a table with columns for days of the week and a total column. The table contains data for various projects, with hours logged for each day and a total for each project. The total hours for all projects are 40.50.

| CLIENT - PROJECT (TASK) | Mon May 04 | Tue May 05 | Wed May 06 | Thu May 07 | Fri May 08 | Sat May 09 | Sun May 10 | TOTAL |
|-------------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|--------------|
| Project 1 | | | | | 4.00 | | | 4.00 |
| Project 2 | | | | | 2.50 | | | 2.50 |
| Project 3 | | | 4.00 | | | | | 4.00 |
| Project 4 | | | 1.00 | | | | | 1.00 |
| Project 5 | | | 1.00 | | | | | 1.00 |
| Project 6 | | | | 4.50 | | | | 4.50 |
| Project 7 | | | 1.00 | | | | | 1.00 |
| Project 8 | | | | 1.50 | 1.00 | | | 2.50 |
| Project 9 | 10.00 | 6.00 | | | | | | 16.00 |
| Project 10 | | | | | 2.00 | 2.00 | | 4.00 |
| Total | 10.00 | 6.00 | 7.00 | 6.00 | 9.50 | 2.00 | 0.00 | 40.50 |

#9: 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: 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.

Or start a new account

Choose a username (no spaces)

bert

Choose a password

Retype password

Email address (must be real!)

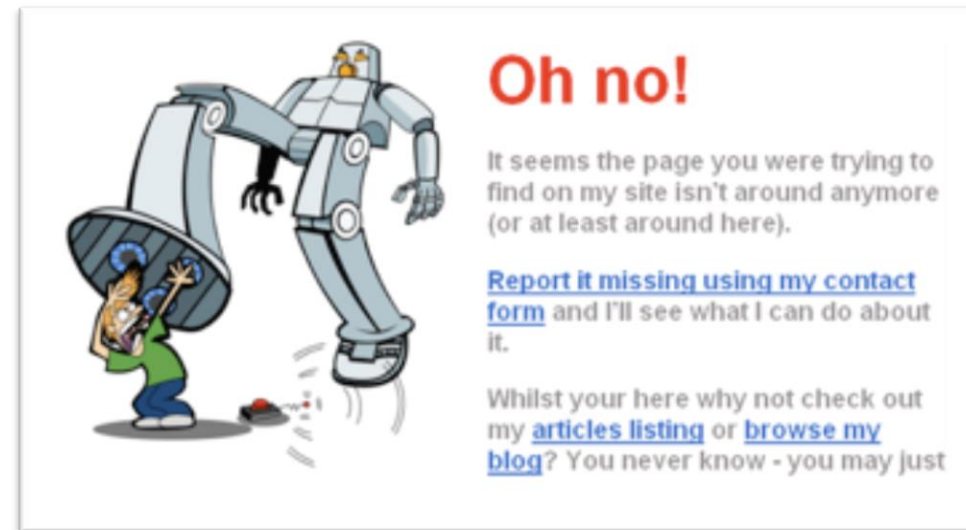
not an email

☒ Send me occasional Digg updates.

bert is already taken. Please choose a different username.

Passwords must be at least 6 characters and can only contain letters and numbers.

The email provided does not appear to be valid

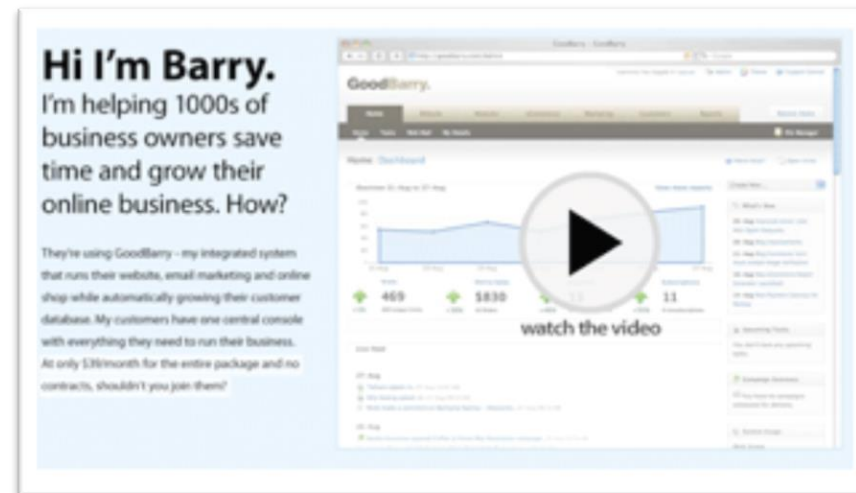


#10: Help and documentation

- 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.

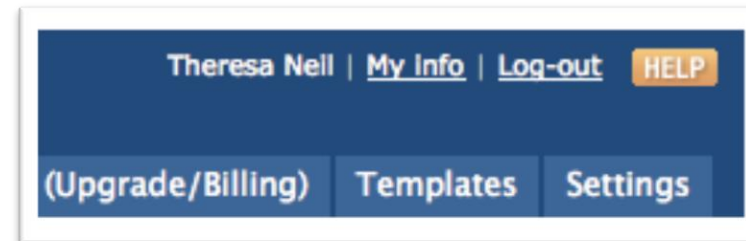
#10: Help and documentation

- 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 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.



References

- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale: Human Computer Interaction, 3rd Edition
 - Chapter 9: Evaluation Techniques
- Ben Shneiderman, Catherine Plaisant, Maxine S. Cohen, Steven M. Jacobs, and Niklas Elmqvist, Designing the User Interface: Strategies for Effective Human-Computer Interaction
 - Chapter 5: Evaluation and the User Experience
- COGS120/CSE170: Human-Computer Interaction Design, videos by Scott Klemmer, https://www.youtube.com/playlist?list=PLLsT5z_DsK_nusHL_Mjt87THSTlgrsyJ

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/>

