HCI Evaluation Part One

Dr Jon Bird
jon.bird@bristol.ac.uk

Thanks to Stuart Gray, Pete Bennett, Simon Lock, Thomas Bale, Harry Field who developed some of these slides

Today’s Lecture

- What is HCI evaluation?
- Why is it important
- The Think Aloud evaluation technique
- Heuristic evaluation

HCI Evaluation

- Evaluation is a crucial part of the user-centred development process – we want to ensure our software meets our users' requirements
- The focus of this lecture is on Think Aloud technique and Heuristic Evaluation, which are two of the most widely used evaluation methods in industry
- They are methods that we recommend you carry out on your game as part of your group project – you can write up the results in your report

Why is evaluation important?

- *“Iterative design, with its repeating cycle of design and testing, is the only validated methodology in existence that will consistently produce successful results. If you don’t have user-testing as an integral part of your design process you are going to throw buckets of money down the drain.”*

Bruce Tognazzini (we’ll meet him later in the lecture)

The Think Aloud evaluation technique

- Users are asked to verbalise what they are thinking and doing as they perform a task using your software
- The Think Aloud technique provides insights into the user experience of using your software
- It can identify issues with the software e.g. navigation problems or content that can be improved
- It can be used as part of the software development process to iteratively improve software or used with a finished product

Benefits of Think Aloud

- Cheap
- Relatively easy
- It provides insight into people's experiences as they interact with your product
- It can be carried out with low numbers of participants
- Fits in with most software development processes

Drawbacks of Think Aloud

- it relies on people verbalising thoughts and impressions, rather than objective measures
- Participants may say what they believe to be the right answer rather than what they really think (social desirability). This can distort your results and conclusions

Planning a Think Aloud evaluation

- Decide what questions you want your study to answer. For example, whether users can find particular content or what their understanding is of the information presented.
- Write down the tasks you want the user to complete while using your software
- Decide how many participants you want to recruit and how long you want the sessions to last (45 to 90 minutes works well)

Carrying out a Think Aloud evaluation 1

- Have a facilitator to run the evaluation and one or two observers to take notes on what the user says
- Explain to the participants how a think aloud works: they should tell you their thoughts, reactions and emotions as they occur while they are performing the task
- Explain that there is no right answer and it’s fine to be critical

Carrying out a Think Aloud evaluation 2

- Ask the participants to complete the tasks you have planned. This should be **uninterrupted** as far as possible, although the facilitator will probably need to give some prompts.
- If the user goes silent then prompt them to verbalise their thoughts by saying “what are you thinking”

Analysing a Think Aloud evaluation

- Put the written notes together from both observes in to one document
- Organise the notes into meaningful categories e.g. what features helped users; what features led to problems; any additional features that users wanted.
- You can make your own meangingul categories
- Count the number of times users comment about different categories to identify the biggest issues

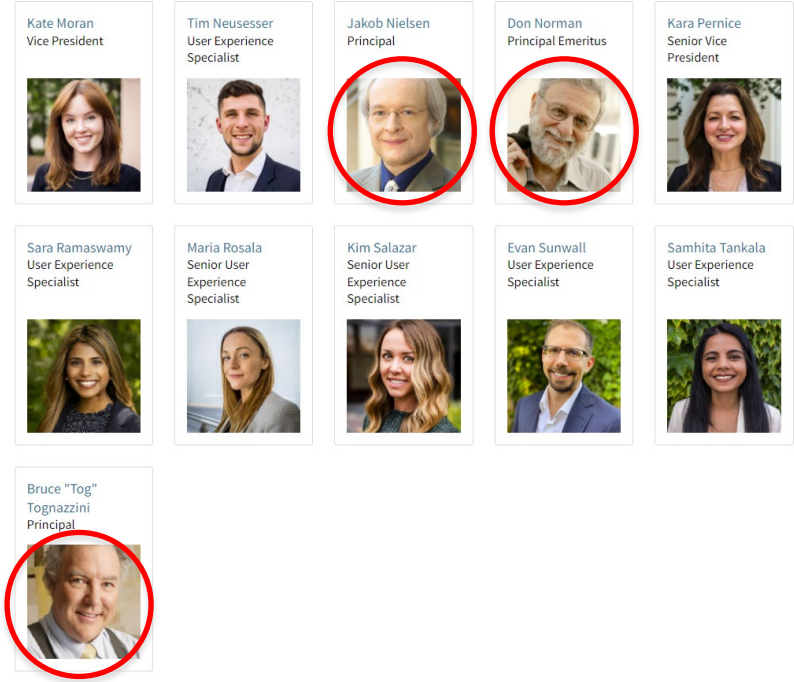
Jakob Nielsen – heuristic evaluation

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *CHI'90*, 249-256.

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

Nielsen Norman group http://www.nngroup.com/

- The Nielsen Norman group is a UX research and consulting firm
- It was founded by two big figures in the HCI world:
  - Don Norman coined the term “user experience” and developed a set of design heuristics
  - Jakob Nielsen also developed a set of usability heuristics and was a pioneer of heuristic evaluation



What is a heuristic?

- A rule of thumb
- Experienced-based strategies
- E.g. if you're doing some DIY then 'measure twice, cut once' is a useful heuristic

Heuristic evaluation 1

- An evaluation technique conducted **without** users
- Also known as **expert** evaluation as it's sometimes carried out by external experts (sometimes by the development team) aka evaluators
- It's a type of **analytical** evaluation, that is, based on a set of principles or a model…
- …rather than by observing users (which is known as **empirical** evaluation)

Heuristic evaluation 2

- It’s an **inspection** method – it involves inspecting a design to find usability problems
- This involves asking whether the design complies with **usability principles** (a set of heuristics)

Heuristic evaluation is widely used because…

- It's **cheap** (only needs a small number of evaluators and no specialist equipment or labs)
- Relatively **easy** to carry out (can do it after a few hours of training)
- **Instant gratification** – lists of problems are **available immediately** after the inspection
- It **fits in** with most software development processes used in industry
- It's a very **cost effective**: benefit-cost ratio of 48: cost of $10,500; expected benefits $500,000 (Nielsen 1994).

Where are the users?

- Heuristic evaluation is based on HCI researchers' extensive experience of designing and evaluating interfaces
- By focusing on users, HCI researchers learned what works and what doesn't
- Their experience is distilled into **usability principles** (a set of heuristics)
- The principles represent the findings from thousands of user studies
- They have been used for over 30 years

What are Nielsen's 10 principles of heuristic evaluation?

- visibility of system status
- match between system and 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 users recognise, diagnose and recover from errors
- help and documentation

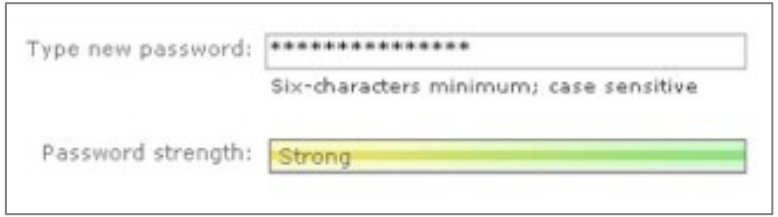
Nielsen's 10 principles of heuristic evaluation (minimal information)

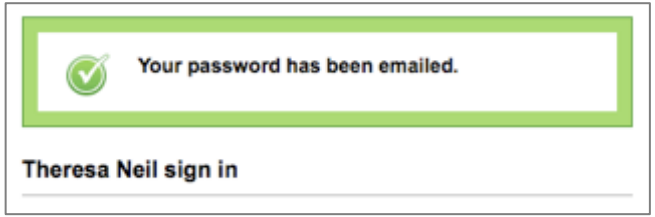
- feedback
- metaphor
- user control and freedom
- consistency
- error prevention
- recognition not recall
- flexible use
- minimal information
- error recognition and recovery
- help

Visibility of system status - feedback

- Inform the user about what's going on:
  - show appropriate feedback and progress
  - do not show blank screens
  - do not show static "load" or progress messages

Visibility of system status: examples



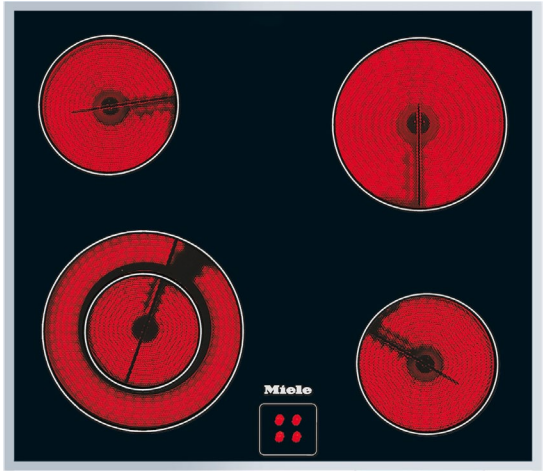


Microsoft Live
Password strength is shown as the password is entered. Colors are used to augment the message.

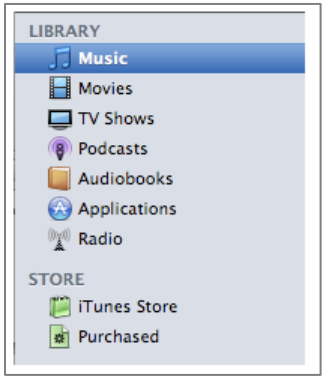
Tick
A feedback message is displayed when an action is performed

Match between system and real world - metaphor

- There must be a match between the system's interface controls 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



Match between system and real world - examples

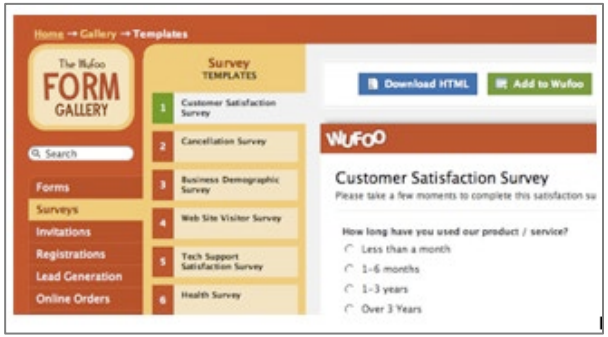


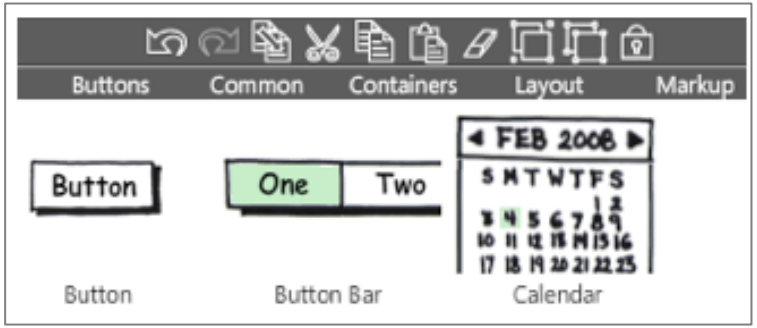
iTunes
Organized as a library that contains your media library: music, movies, TV shows, audiobooks. Beneath the Library is the Store where you can buy more media to put in your Library.

User control and freedom - navigation

- 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 dialog.
- Support undo and redo and a clear way to navigate.
- Provide bread crumbs to clearly show where the user is.

User control and freedom - examples



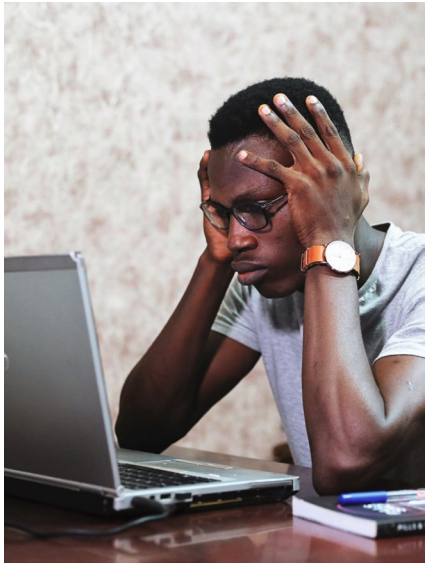


Wufoo
Clearly marks where the person is and where they can go by showing the selection in each menu

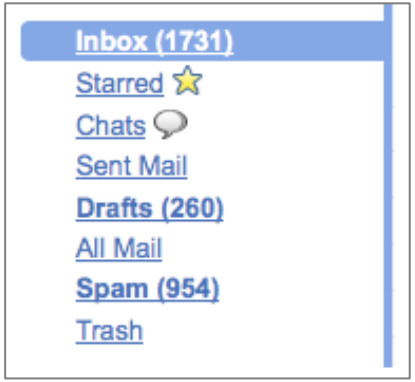
Balsamiq
Undo and Redo buttons are available in the toolbar, and can also be accessed with the standard keyboard shortcuts

Consistency and standards

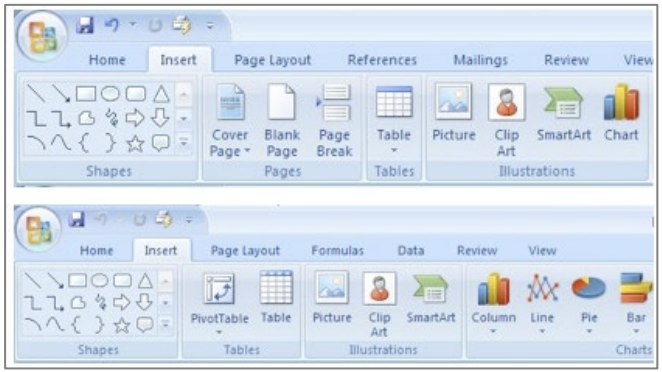
- Users should not have to wonder whether different words, situations, or actions mean the same thing.
- Follow platform conventions.



Consistency: examples



Gmail
When Gmail was designed, they based the organizational folders on the same ones used in other client email applications: Inbox, Drafts, Sent Mail.



Microsoft Office
Word, Excel, and PowerPoint all use the same style toolbar with the same primary menu options: Home, Insert, Page Layout.

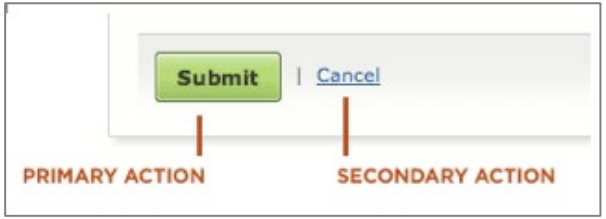
Error prevention

- 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.

Error prevention: examples

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

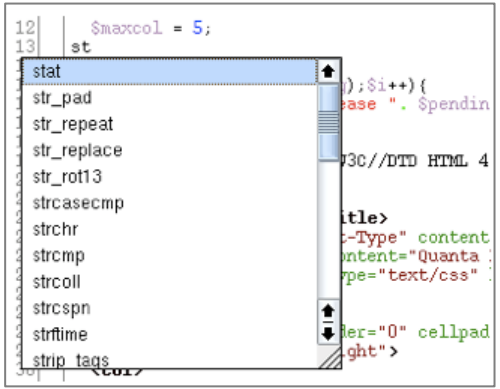


Example from "Web form Design:Filling in the Blanks" by Luke W.
Make the primary action prominent with a larger click area. Cancel and other secondary actions are just shown as links

Recognition rather than recall

- Minimize the user’s memory load.
- Make 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.

Recognition: examples

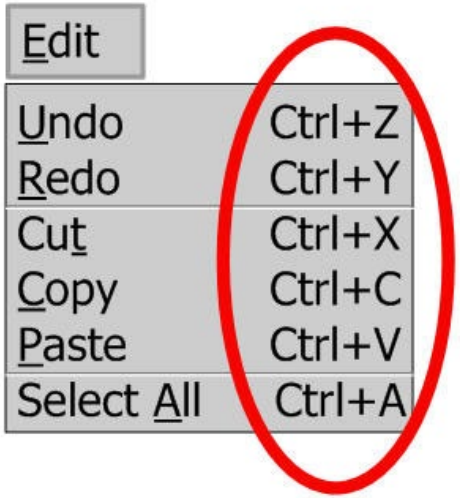


Quanta IDE
Auto completion for coding in a development environment

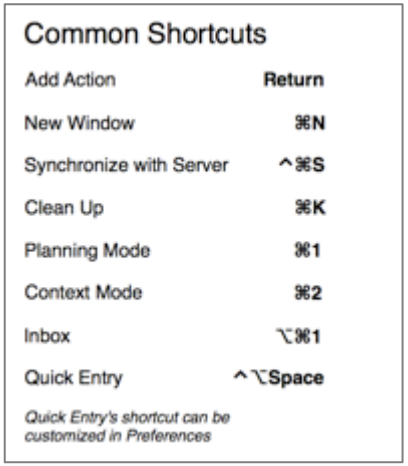
Keynote
Previews the fonts you can pick from, instead of just the font name

Flexibility and efficiency of use

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



Flexibility and efficiency: examples

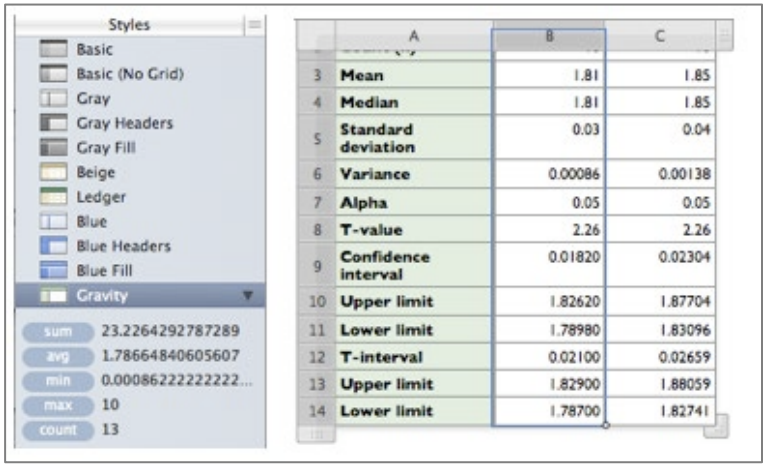


Common Shortcuts

Add Action	Return
New Window	⌘N
Synchronize with Server	^⌘S
Clean Up	⌘K
Planning Mode	⌘1
Context Mode	⌘2
Inbox	⌥⌘1
Quick Entry	^⌥Space

Quick Entry's shortcut can be customized in Preferences

OmniFocus
List of keyboard shortcuts and accelerators

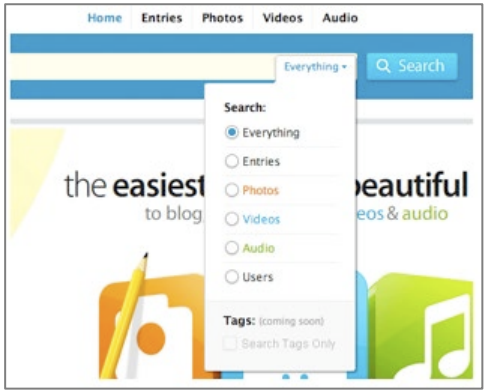


Numbers by Apple
Previews common function results on the left when a column is selected, more efficient than clicking on an action in the toolbar

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
- Visual layout should respect the principles of contrast, repetition, alignment, and proximity.

Aesthetics: example

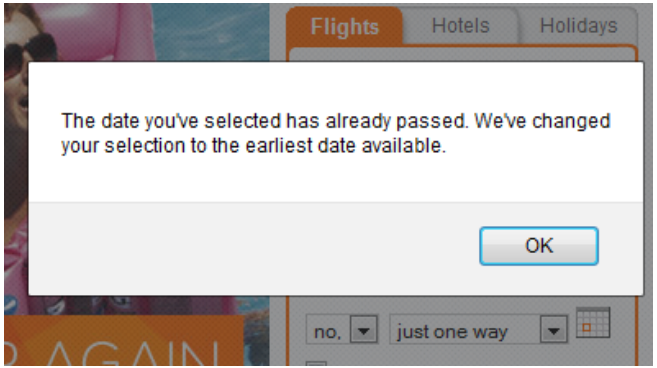


Kontain's search menu exemplifies the four principles of visual design:

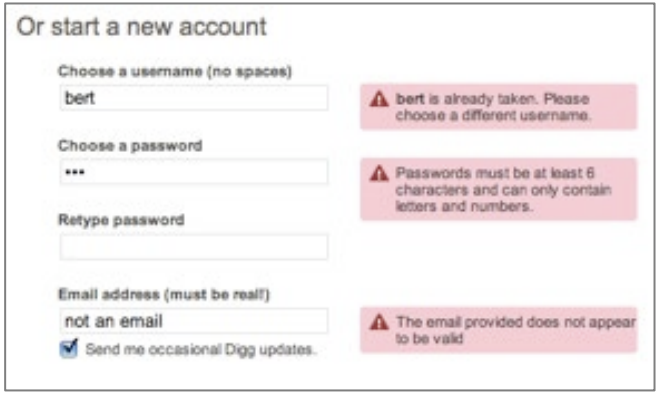
1. Contrast: bold text is used for the two labels in the search
2. Repetition: the orange, blue, and green text match the media types
3. Alignment : strong left alignment of text, right aligned drop down
4. Proximity: a light rule is used to separate tags from the other options

Help users recognise, diagnose and recover from errors

- Help users recognize, diagnose, and recover from errors.
- Error messages should be expressed in plain language (no jargon), precisely indicate the problem, and constructively suggest a solution.



Error recognition and recovery: examples

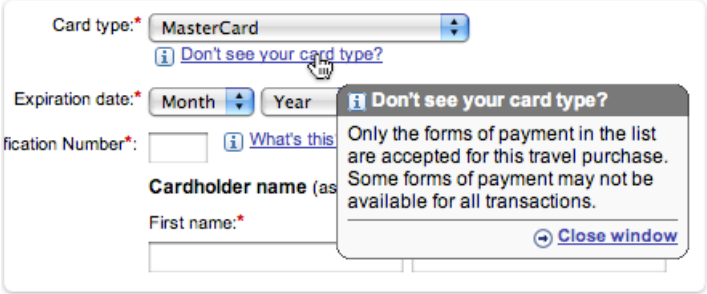


Digg
Provides immediate feedback with specific instructions

Humorous ‘ Page Not Found’ Error
Uses a funny image and text, but provides viable alternatives (article listings and blog link) and a course of action (report it)

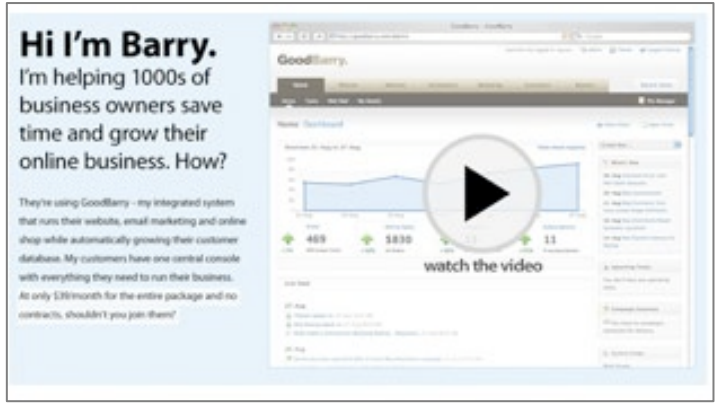
Help and documentation

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



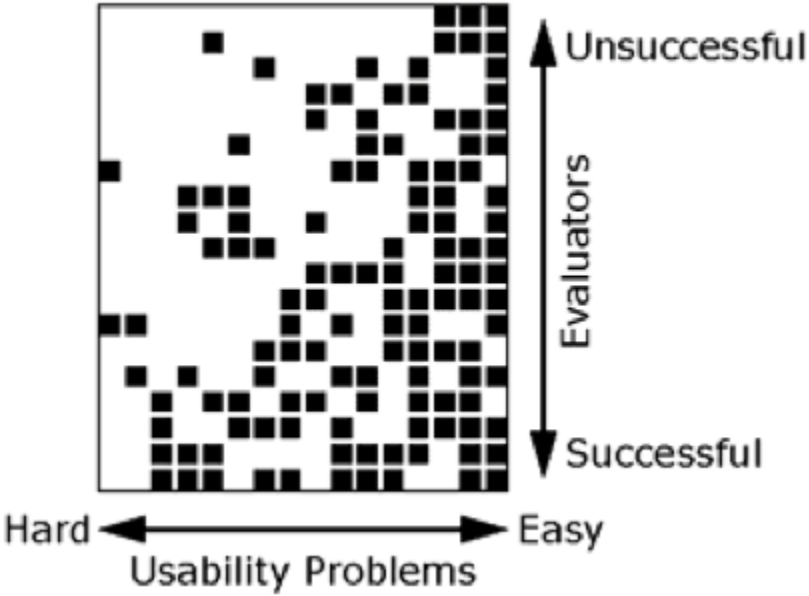
Help and documentation: examples

Picnik
Contextual tips in Picnik are clear and easy to navigate

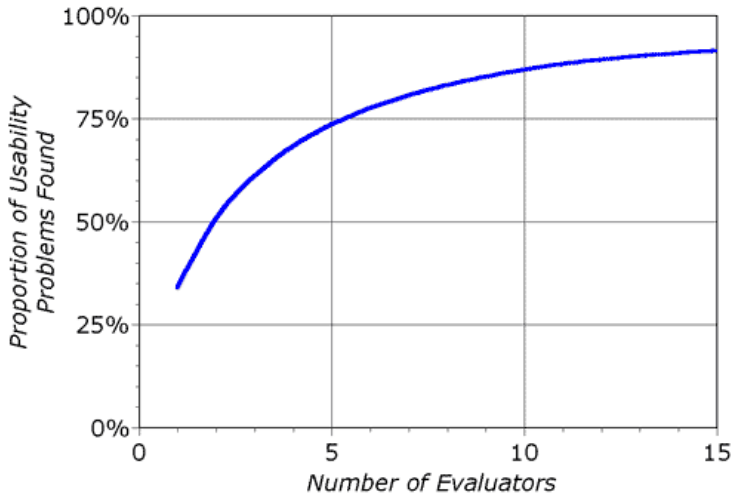


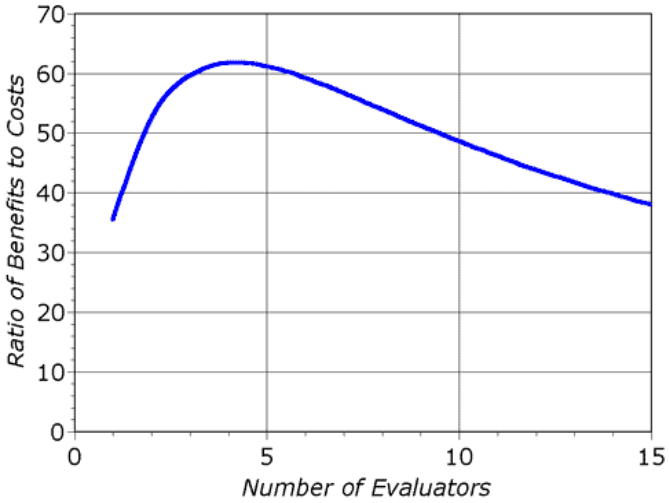
GoodBarry
Embedded videos can be used to showcase features as well as get people started using the product

How many evaluators are needed for heuristic evaluation?



Practical considerations





How to run a heuristic evaluation 1

- Each of the 3 – 5 evaluators does a heuristic evaluation of an interface alone
- Sometimes an observer can record the evaluator's comments, sometimes the evaluator does it
- Observers **can** answer evaluators' questions, in contrast to traditional user testing, particularly if it's not a walk up and use system
- Heuristic evaluation can be done on paper prototypes

How to run a heuristic evaluation 2

- Heuristic evaluations typically last 1 – 2 hours, but it does depend on the complexity of the software
- The expert goes through the interface several times – first time to get a feel for the system, second time to focus on specific elements
- Evaluators can be given scenarios that describe typical usage scenarios (built from a task analysis of users)
- Evaluators produce a list of usability problems: the usability principle and the design feature that violated it

Benefits of heuristic evaluation

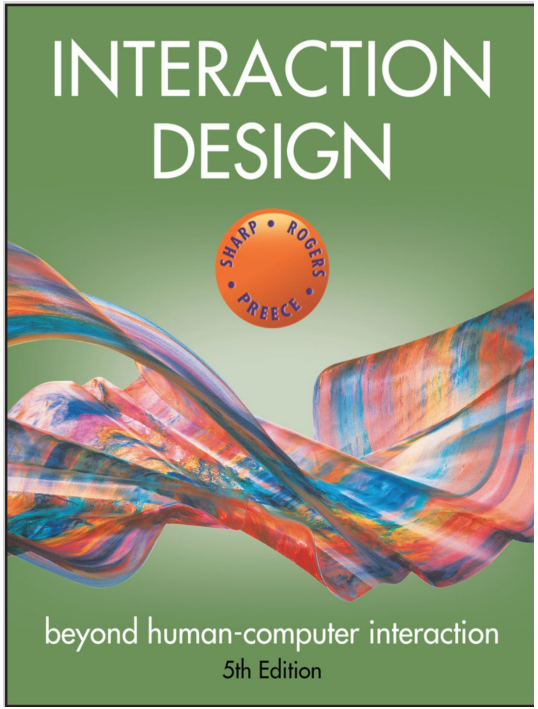
- Cheap
- Relatively easy
- Instant gratification lists of problems are available immediately after the inspection
- It can be carried out with low numbers of participants
- Fits in with most software development processes
- Cost effective

Drawbacks of heuristic evaluation

- Important issues may get missed
- Might identify false issues
- Many trivial issues are often identified, making it seem overly critical
- Experts have biases

Reading

- *Interaction Design: Beyond Human-Computer Interaction* covers all HCI evaluation techniques. It's available through the university library as an eBook. Read about the evaluation techniques covered in this lecture to deepen your understanding
- Read the original Nielsen paper on heuristic evaluation: https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/



Reading 2

- Explore the materials (papers, articles and videos) on heuristic evaluation on the Nielsen Norman group website:

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

Before the workshop today

- Please review the lecture materials on the Think Aloud and Heuristic Evaluation techniques

- Your workshop activities will involve evaluating your games using these two techniques