

User Centered Design

[8] Design Guidelines, Patterns and Principles

Dr. Agnes Lisowska Masson
Human-IST Institute, University of Fribourg
November 6th, 2018



What makes a bad design?

What makes a good design?

Motivating good and careful design

“Everything is best for something and worst for something else. The trick is knowing what is what, for what, when, for whom, where, and most importantly, why.”

– Bill Buxton

“Does human error cause accidents? Yes, but we need to know what led to the error: in the majority of instances it is inappropriate design of equipment or procedures.”

– Donald Norman

Questions to ask

- **Who** will interact with it?
- **Why** will they interact with it?
- **What** will they do with it?
- **Where** will they interact with it?
- **When** will they interact with it?
- **How** can they interact with it?

Why do we choose a technology?

- Function
 - what the technology can help us do
- Need/want
 - why we need or want the technology
- Aesthetics
 - how the technology looks
- Social reasons
 - what owning a technology means to us and to other people
 - how the technology can help us connect to others

You can design...

- The functionalities
 - what tasks the system/technology will let you do
- The behaviour
 - how the system lets you do those tasks
 - which input and output devices to use
 - what type of feedback to give
- The look and feel
 - how the system/technology will look visually
 - which materials will be used

But don't forget...

- People think and behave differently....so don't expect them to think and behave like you do
- People have different physical characteristics
- You don't necessarily represent a good average user of the system you design

So realistically...

- It is rarely possible to accommodate all people perfectly
 - design is often a compromise
 - ✓ e.g. ceiling height : 2.35 m
 - ✓ but tallest man : ~2.50 m
- Rule of thumb
 - design should cater for 95% of audience
 - but that means 5% of the population might be disadvantaged
- Designing for the average is a mistake
 - may exclude half the audience

Design patterns, principles, guidelines and standards

UI design patterns

■ Design patterns

- solutions to recurring problems in a particular context
- described by architect Christopher Alexander in the 1960s
- capture design practice, but not the theory

Action Panel

- **What:** Instead of using menus, present a large group of related actions on a UI panel that's richly organized and always visible.
- **When:** You need to present many actions -- too many for a Button Group. You could put them into a menu, but maybe you don't have a menu bar at all, or you'd rather make the actions more obvious.
- **Why:** Visibility and freedom of presentation.
- **How:** Putting the action panel on the UI

http://designinginterfaces.com/Action_Panel

Sort By Column

Design pattern

Problem summary

The user needs to search or scan a table for values that is of interest

Example



Usage

- Use when there are many rows in the table (above 10), which makes it hard to single out one row and its relation to other rows
- Use when there are more than one page to show
- Use when you want to be able to compare rows in a table – for instance numbers.
- Do not use if the amount of rows are few and the table is easy to search or scan.

Solution

Each column headline/label is a link. When the label is clicked, the rows in the table are ordered ascending by the specific column's values. If the same label is clicked again, the order is reversed: the rows in the table are now ordered descending by the specific column's values.

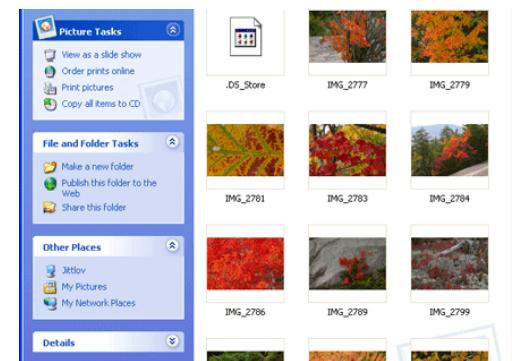
When the rows of a table has been sorted by a specific column, an arrow is often showed beside the column's label indicating the direction the rows has been sorted in. It is also often seen that the column's label is presented in another font color or font weight (bold / regular).

Rationale

The pattern provides an easy way to get an overview and compare rows in a table. Furthermore, the pattern is also well known from desktop applications dealing with rows of data.

More example images of the 'Sort By Column' pattern

<http://ui-patterns.com/patterns>



Principles, guidelines and standards

■ Standards

- “A standard is a document that provides requirements, specifications, guidelines or characteristics that can be used consistently to ensure that materials, products, processes and services are fit for their purpose”

www.iso.org

- set by international bodies to ensure compliance by a large community

■ Principles:

- tend to be very general and could be applied across different technologies and systems
- suggest how to increase usability
- are useful checklists for good design
- derived from a mix of theory-based knowledge, experience and common-sense

■ Guidelines:

- tend to be more specific to a device or system

From: <http://alandix.com/blog/2016/03/31/principles-vs-guidelines/>

Principles, guidelines and standards

■ Authority

lowest	principles guidelines	agreed by community, but not mandated proposed by manufacturer, but rarely enforced
highest	standards	mandated by standards authority

■ Applicability

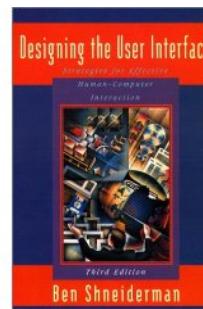
highest	principles guidelines	very broad (e.g. observability')
		more specific but still allow interpretation
lowest	standards	very tight

From: <http://alandix.com/blog/2016/03/31/principles-vs-guidelines/>

Examples of design principles/rules

Norman's Design Principles

1. Visibility
2. Feedback
3. Constraints
4. Mapping
5. Consistency
6. Affordance



Schneiderman's 8 Golden Rules of Design

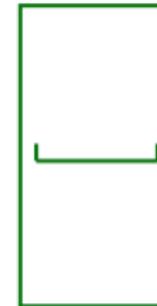
1. Strive for consistency
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialogs to yield closure
5. Offer simple error handling
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short-term memory load

Visibility

- Let the user know
 - what they can do
 - what is relevant/important
 - how they can do it



Push or pull ?



Which side ?



*Can only push,
side to push clearly visible*

- Why do it?
 - decrease memory load
 - ✓ use knowledge in the world



Feedback

- Let user know
 - what has been done
 - what the result is



- Can be
 - visual
 - auditory
 - haptic
 - combination

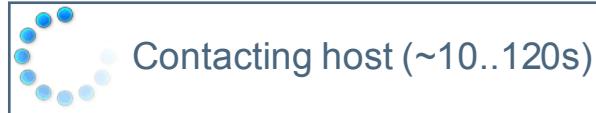


- Ideally
 - as specific as possible depending on user's input
 - within the context of an action

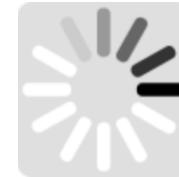


Feedback

- Response times are very important
- How users perceive delays
 - < 0.1 s : perceived as "instantaneous"
 - 1 s : user's flow of thought stays uninterrupted, but delay is noticed
 - 10 s : limit for keeping user's attention focused on the dialog
 - > 10 s : user will want to perform other tasks while waiting
- Dealing with long delay
 - wait cursor
 - ✓ for short transaction
 - percent done dialog
 - ✓ time left
 - ✓ estimated time
 - random animation
 - ✓ unknown time



Contacting host (~10..120s)

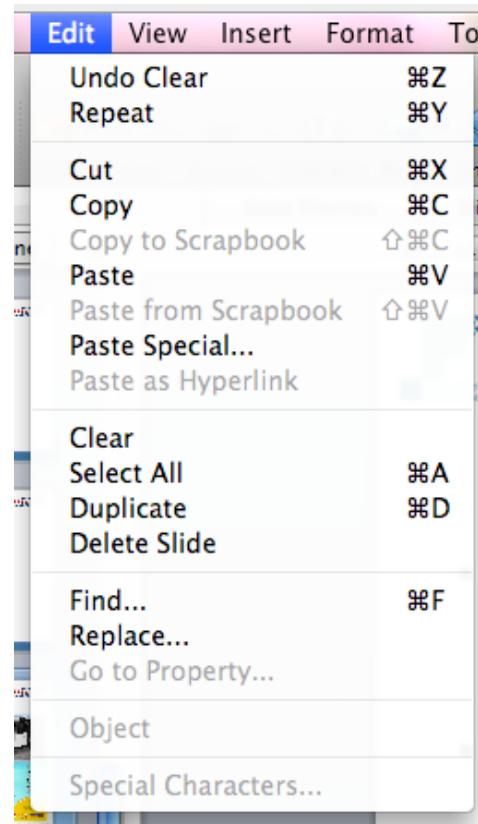


Constraints

- Ways to restrict the type of interaction that can take place at a specific moment

- Types
 - logical
 - cultural
 - semantic
 - physical

- Why do it?
 - help reduce errors
 - give clues



<http://city.lego.com/en-US/Products/Default.aspx#3177>

Constraints

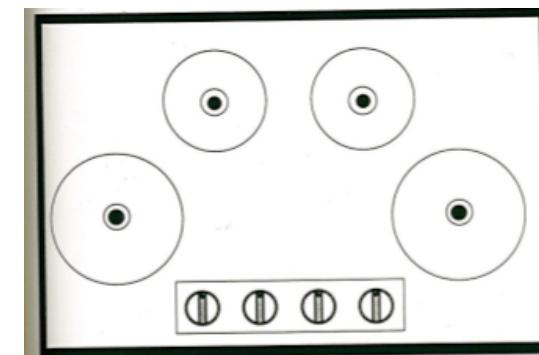
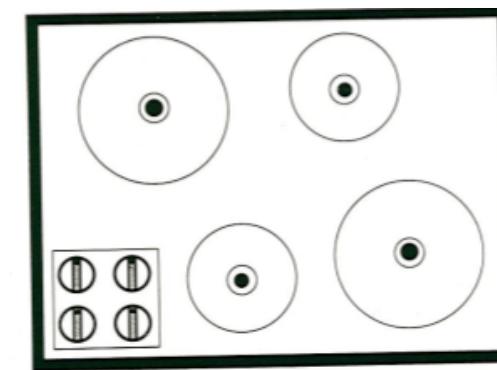
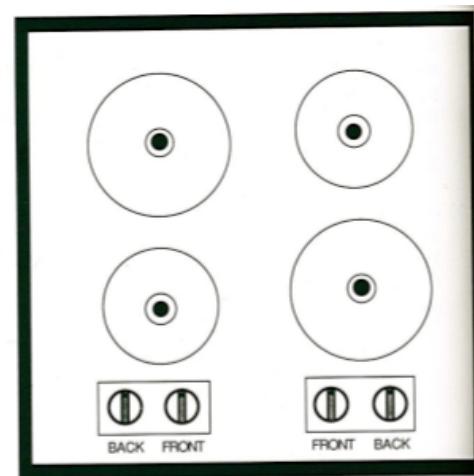
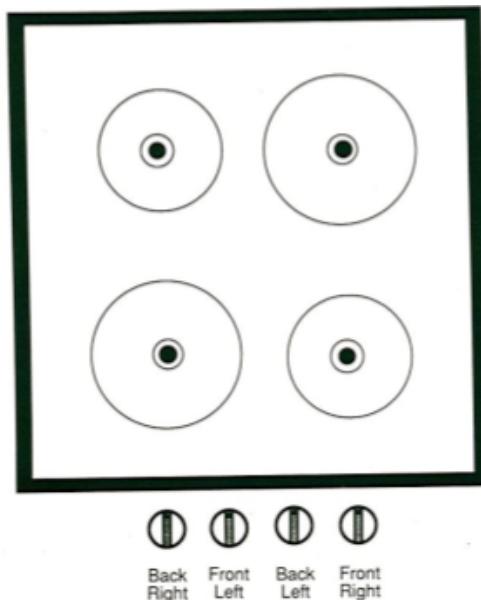
- Dates on forms

The screenshot shows a Windows application window titled "Form1". Inside, there is a label "Date:" followed by a standard Windows-style date input field consisting of three separate text boxes for Month, Day, and Year. Below this is another date input field where "May" is selected in the Month dropdown, "22" in the Day dropdown, and "1997" in the Year dropdown. At the bottom is a single combined input field containing "May 22 1997" with dropdown arrows on either side of the day and year parts.

The screenshot shows a travel booking form with several date and time inputs. On the left, there are two sections: "Outbound" with the value "03.01.2012" and "Return" with the value "30.01.2012". Below these are two more date inputs labeled "Departure date" and "Return date", each with a placeholder "DD/MM/YYYY". To the right of these date inputs are two time inputs labeled "from:" with a placeholder "mm/dd/yyyy" and a calendar icon, and two dropdown menus for hours ("10") and minutes ("00").

Mapping

- Set of relations between objects



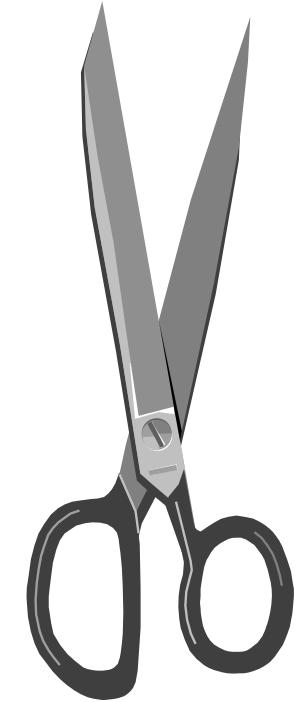
Consistency

- Similar operations or similar elements used to do similar tasks
 - shortcut keys (Ctrl-S, Ctrl-V,...), right click
- 2 kinds
 - internal consistency – within an application
 - external consistency – across applications
- Consistency
 - encourages predictability
 - encourages learnability (reusability to skills)
 - improves user's trust



Affordance

- Attribute of an object that gives a clue about how to use it
- 2 kinds
 - real /physical
 - ✓ perceptually obvious, doesn't have to be learned
 - perceived/virtual
 - ✓ learned conventions



Enable frequent users to use shortcuts

- Experienced users want to perform frequent operations quickly
- Strategies:
 - keyboard and mouse accelerators
 - ✓ mnemonics / abbreviations
 - ✓ command completion
 - ✓ context menus
 - ✓ function keys
 - ✓ ...
 - navigation jumps
 - ✓ e.g., going to window/location directly, and avoiding intermediate nodes
 - history systems
 - ✓ web : ~60% of pages are revisited
 - bookmarks

Offer simple error handling

- People WILL make errors
 - don't blame the user...try to make errors impossible
 - ✓ e.g. widgets so that users can only enter legal data
- Two types of errors
 - mistakes
 - ✓ conscious thought processes lead to an error instead of a correct solution
 - slips
 - ✓ unconscious thought processes take over conscious ones
 - e.g. drive to store, end up in the office
 - ✓ shows up frequently in skilled behavior
- Ideal error messages should contain
 - clear explanation of the error using user's language
 - indication of cause of the error
 - indication of how to correct the error

Offer simple error handling

■ Methods

➤ warn

- ✓ warn people that an unusual situation is occurring
- ✓ but becomes irritating when overused (e.g., audible bell, alert box)

➤ undo

- ✓ make actions undoable
- ✓ allow reconsideration of action by user (E.g. open trash to undelete a file)

➤ do nothing

- ✓ user has to infer what went wrong
- ✓ forbid the user from continuing (e.g. login/password)
- ✓ E.g. enter letter into a numeric-only field (key clicks ignored) or put a file icon on top of another file icon (returns it to original position)

➤ self-correct

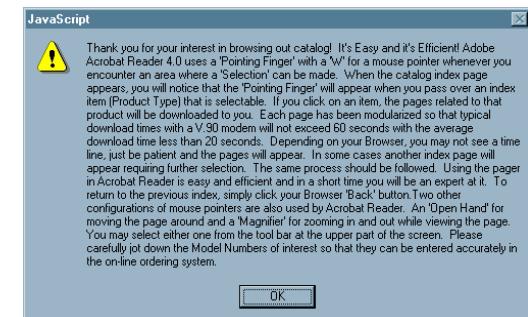
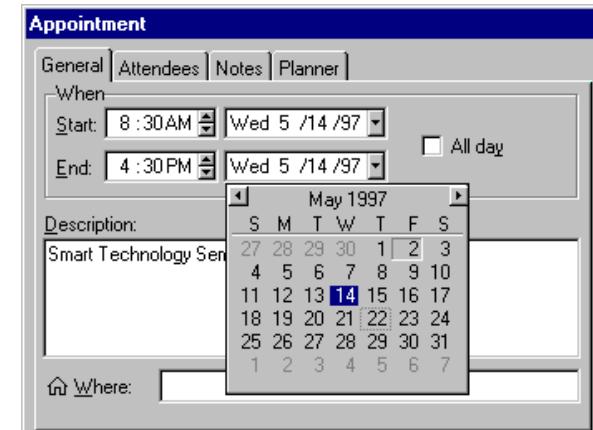
- ✓ system guesses legal action and does it instead (but this leads to a problem of trust, e.g. spelling corrector)

Permit easy reversal of actions

- Users don't like to feel trapped by the computer
 - should offer an easy way out of as many situations as possible
- Strategies:
 - cancel button (for dialogs waiting for user input)
 - universal Undo (can get back to previous state)
 - interrupt (especially for lengthy operations)
 - quit (for leaving the program at any time)
 - defaults (for restoring a property sheet)

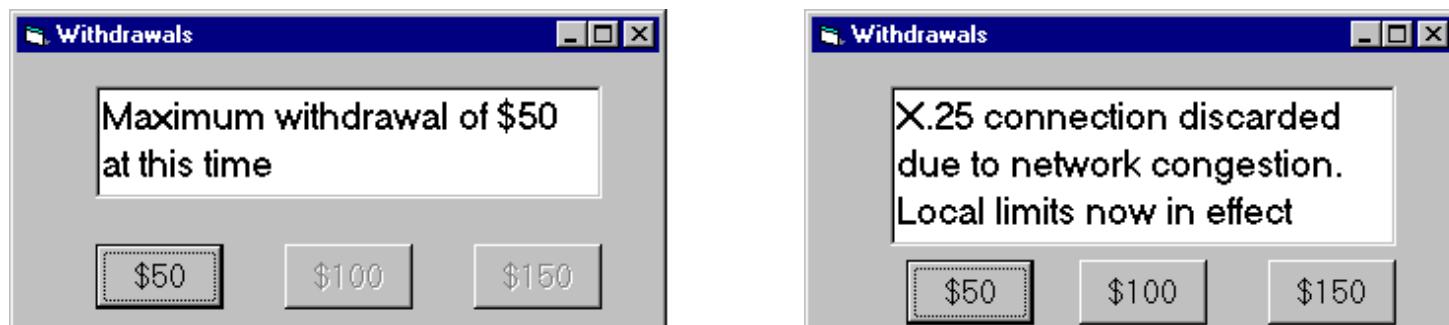
Reduce short-term memory load

- People aren't good at remembering, computers are
 - promote recognition over recall
- Use generic commands
 - same command can be applied to all interface objects (context of use)
 - copy, cut, paste, drag & drop, ... for words, paragraphs, images, files..
- Avoid long (verbose) messages
- Give input format, example and default
- Use meaningful mnemonics, icons, tooltips



Simple and natural dialogs

- Match the user's conceptual model and task sequence
- Present exactly the information the user needs
 - less is more (less to learn, to get wrong, to distract...)
 - information should appear in natural order
 - ✓ related information should be graphically clustered
 - ✓ matches user's expectations
 - remove or hide irrelevant or rarely needed information
- Use terminology based on users' language for task
 - e.g. withdrawing money from a bank machine

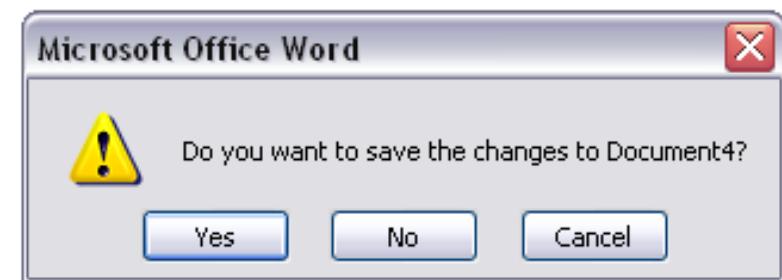
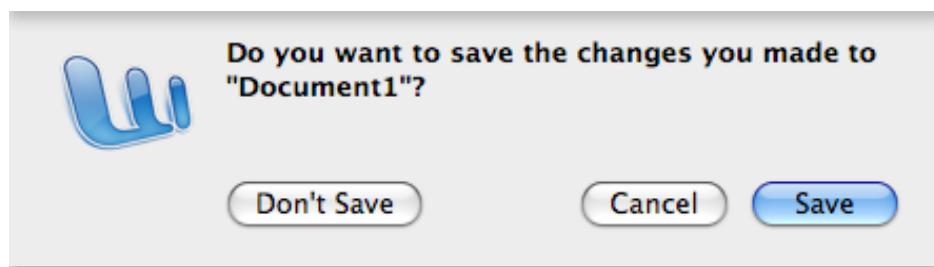


Other things to consider...

- Transfer effects

- positive – previous learning applies in new situation
- negative – previous learning conflicts with new situation

- But...transfer usually isn't 100%



Other things to consider...

- Navigation
 - Where am I? Where am I going? Where was I before? How do I get back?

NEWS
23 September 2011 Last updated at 10:06 GMT

Home | UK | Africa | Asia-Pac | Europe | Latin America | Mid-East | South Asia | US & Canada | Business | Health | Sci/Environment

Magazine | In Pictures | Also in the News | Editors' Blog | Have Your Say | World Radio and TV | Special Reports



Deutsch | Français



Home

Produits

Support

Applis, Cartes, etc.

Partenaires Nokia

My Nokia

Toute la collection

Accessoires

Newsletter

Accessoires

Libérez le musical



HUMAN-IST
HUMAN-CENTRED
INTERACTION
SCIENCE
& TECHNOLOGY

User Centered Design – 8
A. Lisowska Masson / UniFr



Other things to consider...

- Idioms and cultural differences

- are learned
 - are difficult to change



- Examples:

- colours

- ✓ red = stop/danger, green = go/safe

- faucets

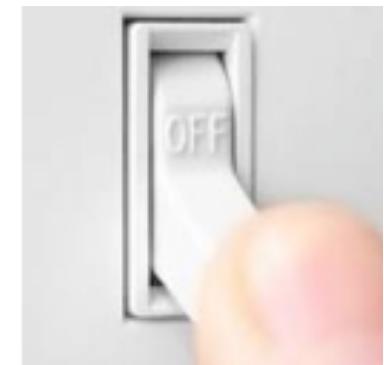
- ✓ North America: anti-clockwise on
 - ✓ UK: anticlockwise off



- light switch

- ✓ North America: down is off
 - ✓ UK: down is on

- keyboards (QWERY, QWERTZ)



Other things to consider...

■ Visual Design

➤ Colour

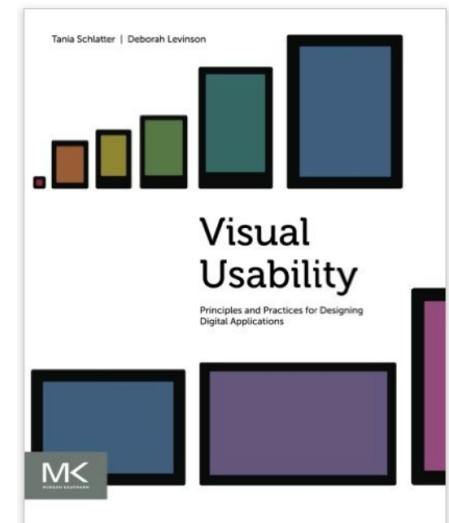
- ✓ can be used to help people know where they are, what they can do, and what attributes are associated with an application

➤ Personality

- ✓ ‘An application’s personality helps build expectations about what the application does and who it’s meant for’

➤ Imagery

- ✓ can be used to create contrast, draw attention, provide information in a simpler way

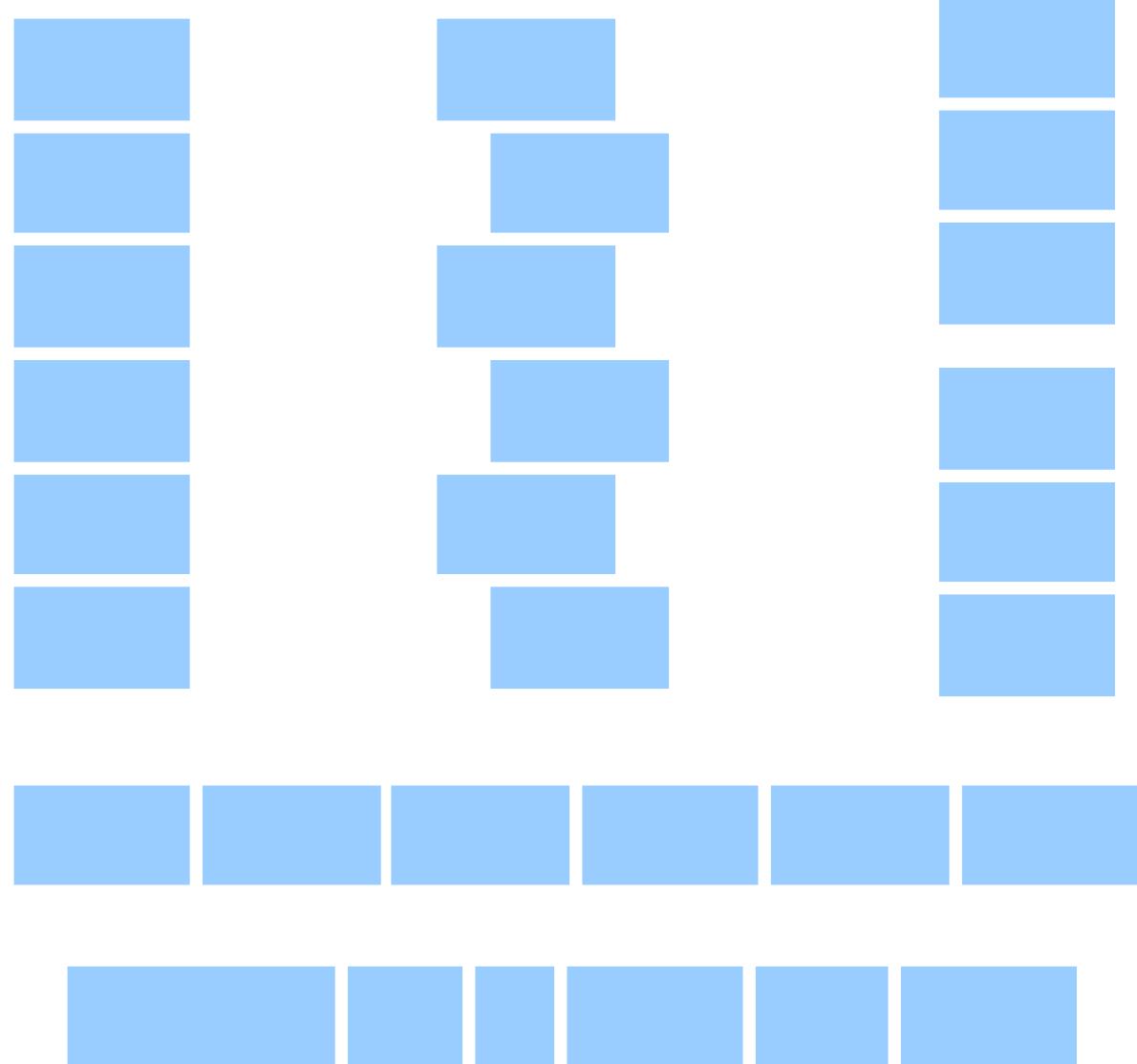


Visaul Usability
– Schlatter & Levinson

Other things to consider...

- Visual Design

- Layout – position, alignment and whitespace can be used to affect perception of how things are grouped together



Other things to consider...

- Visual Design

- typography

EXAMPLE 1

Example 2

Example 3

Example 4

Example 5

EXAMPLE 6

Example 7

- visual hierarchy

Facebook focuses on media sharing and adds timeline

Facebook has outlined plans to encourage users to share more of the media they consume - including music and movies - with friends.

Its founder Mark Zuckerberg also unveiled a dramatic redesign to the website, replacing user profiles with an audio visual timeline of their life.

The updates were revealed at Facebook's annual F8 developer conference.

The screenshot shows a Facebook news feed item from Spotify. It features a green circular icon with three white dots, followed by the text "Log into Spotify" and "Play and discover music for FREE!". Below this, there is a section titled "Add Spotify to your Timeline" which includes a description of what it does and who can see it. It also lists "Spotify needs:" (e-mail address and birthday) and shows a list of top songs: "On Melancholy Hill" by Gorillaz, "The Lazy Song" by Bruno Mars, and "Last Friday Night (T.G.I.F.)" by Keri Hilson featuring Pitbull.

Spotify is one of the companies to take advantage of Facebook's enthusiasm for media sharing

Other things to consider...

- Sounds

- can express things like size, speed, distance, frequency, importance...

Dog



Bell



Drip



Boing



Lots of resources exist...

- ISO standards
 - http://www.usabilitynet.org/tools/r_international.htm
 - use in context, interface and interaction, ...
- Neilson Norman Group
 - <http://www.nngroup.com/reports/>
 - web, intranet, application, email, mobile, social media, different audiences...
 - Bruce Tognazzini (software), Jakob Nielsen (web), Don Norman (tech)
- Ben Shneiderman
 - 8 Golden Rules of Design
 - book: Designing the User Interface
- W3C
 - Accessibility, usability and inclusion
 - ✓ (<http://www.w3.org/WAI/intro/usable>)
- Many others...

Heuristic Evaluation

Heuristic evaluation

- What:
 - systematic analytical evaluation of an interface to see if complies with principles/guidelines/standards
 - helps avoid common design pitfalls and usability problems
- How:
 - 3-5 evaluators (usability engineers, experts)
 - each evaluator inspects the interface (on their own)
 - ✓ ~1..2 hours for simple interfaces
 - compare notes afterwards
 - ✓ Single evaluator only catches ~35% of usability problems
 - ✓ 5 evaluators catch ~75%
- When:
 - at different stages: on paper, on prototypes and on final working systems

Heuristic evaluation

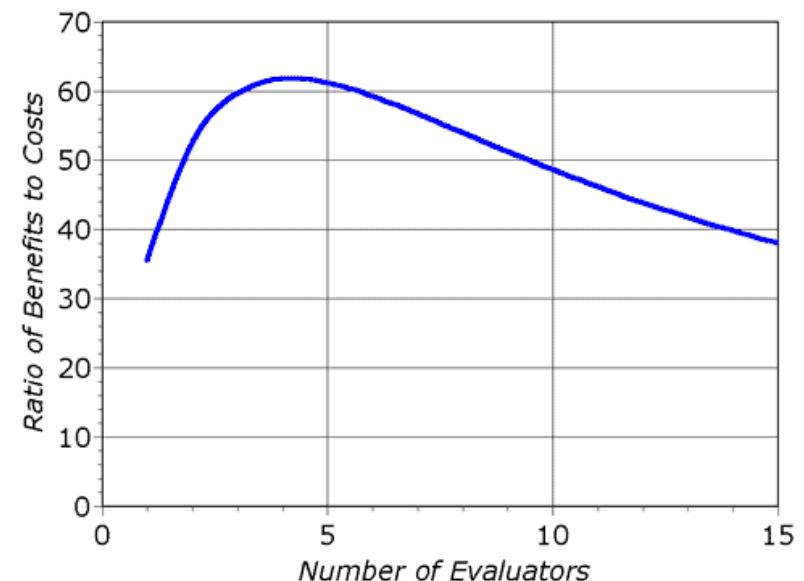
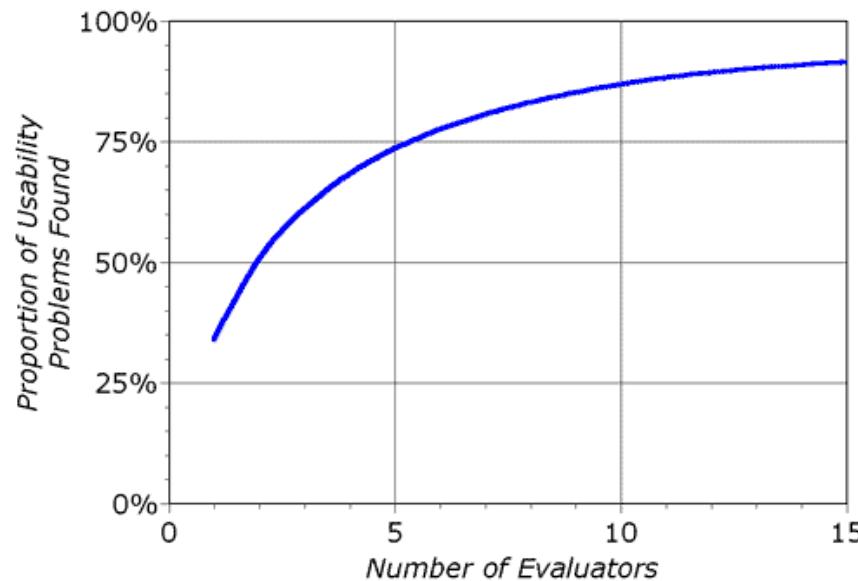
- Advantages:
 - minimalist approach
 - ✓ a few principles/guidelines can identify many common usability problems
 - ✓ easily remembered, easily applied with little effort
 - discount usability engineering
 - ✓ doesn't require end users
 - ✓ cheap and fast way to evaluate a system
 - ✓ can be done by usability experts, double experts
- Disadvantages:
 - principles are subject to interpretation
 - ✓ They're more than a simple checklist (expertise needed)
 - usually doesn't involve domain expertise
 - ✓ May be missing factors crucial for a domain
- Should not replace a usability evaluation with representative end users

Heuristic evaluation: single evaluator

- Problems found by a single evaluator
 - average over six case studies:
 - ✓ 35% of all usability problems
 - ✓ 42% of the major problems
 - ✓ 32% of the minor problems
 - not great, but
 - ✓ finding some problems with one evaluator is much better than finding no problems with no evaluators
 - results vary depending on the difficulty of interface and expertise of evaluator
 - average problems found by :
 - ✓ novice evaluators (no usability expertise) 22%
 - ✓ regular specialists (expertise in usability) 41%
 - ✓ double specialists (experience in usability + particular kind of interface being evaluated) 60%
 - Tradeoff
 - ✓ Novices provide poorer results, but are cheaper!

Heuristic evaluation: multiple evaluators

- 3-5 evaluators find 66-75% of all usability problems
 - different people find different usability problems
 - only small overlap between the sets of problems found (except major)



- Better if evaluators inspect the interface alone
 - evaluation is not influenced by others
 - greater variability in the kinds of errors found

“in anything at all, perfection is finally attained not when there is no longer anything to add, but when there is no longer anything to take away”

- Antoine de St. Exupéry