



Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Usability Principles and Paradigms

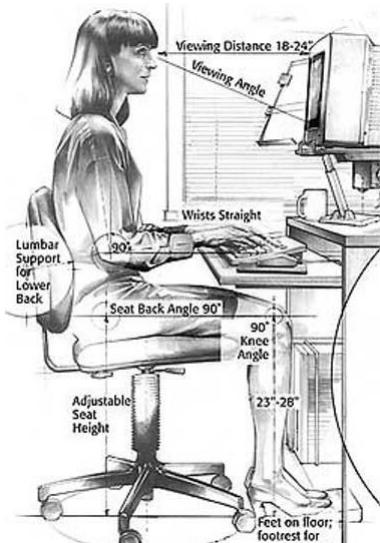


(Donald Norman, Design of everyday things)

- Introduction
- History
- Usability and standards
- Principles
- Paradigms

- During and after the World War II the following disciplines emerged:

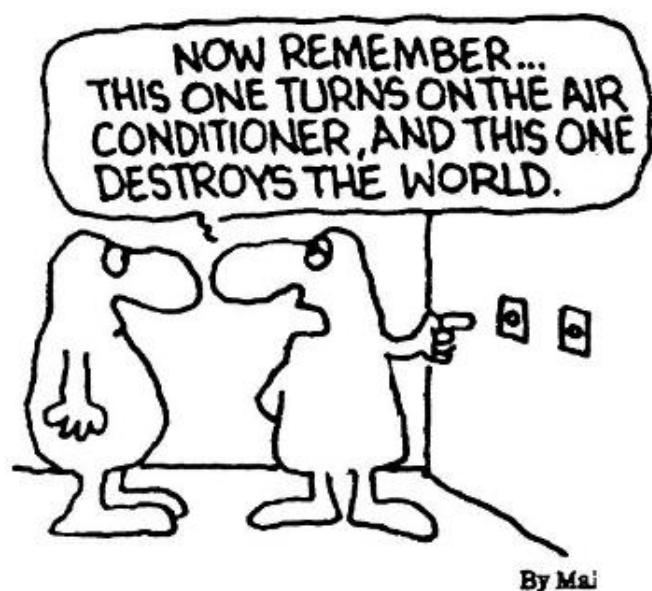
Ergonomics – more focused on physical aspects **Human factors** – also cognitive aspects



Ergonomics- The science of fitting workplace conditions and job demands to the capabilities of the working population

<https://www.cdc.gov/niosh/topics/ergonomics/ergoprimer/default.html>

Ergonomics and Human Factors



- Interaction emerged as new independent field within Computing in the 80s, mainly due to:
 - Lower price of technology
 - Technology migration
 - Need to increase users' productivity

Man-Machine Interaction  **Human-Computer Interaction**
(ninties)

- It expanded rapidly
- It is currently an interdisciplinary field
- Human-Centered Computing is an ACM scientific area within Computing
(also at the University of Aveiro)

Interactive systems design

- Interactive systems include a “part” which we don’t control:

The user, who:

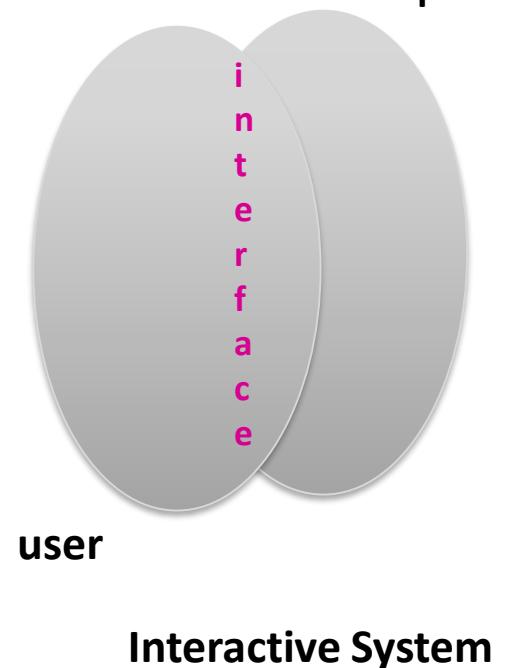
- is very complex
- not well known
- we cannot control



(and users may be very different)

This makes design difficult

- User Interface (UI) is the means by which the user and a computer system interact
- To the user **“the interface is the system”**
- The user interface design involves a considerable effort



Interactive system design – Human-Centered design

- Involves knowing:

Methods

Usability principles (independent from technology)

Usability paradigms (more technology dependent)

- We must know the **success examples** (usability paradigms)
- Understand **why they work** (usability principles)
- Use the **adequate methodology** (user-centered approach) and methods
- And **test**, re-design,
 test, redesign
 ...
 until we attain the usability goals

- **Usability** is, according to ISO 9241-11:

“the extent to which a product can be used by **specified users** to achieve specified goals with **effectiveness, efficiency** and **satisfaction** in a **specified context of use**”

- Effectiveness + efficiency -> **ease of use**
- **Satisfaction** is also very important

Standards evolve:

- ISO 9241-11's three factors of usability have become five by **ISO 25010's quality in use** factors:
- Effectiveness
- Efficiency
- Satisfaction
- Freedom from risk
- Context coverage

- **User Experience (UX)** is:

“ person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service”

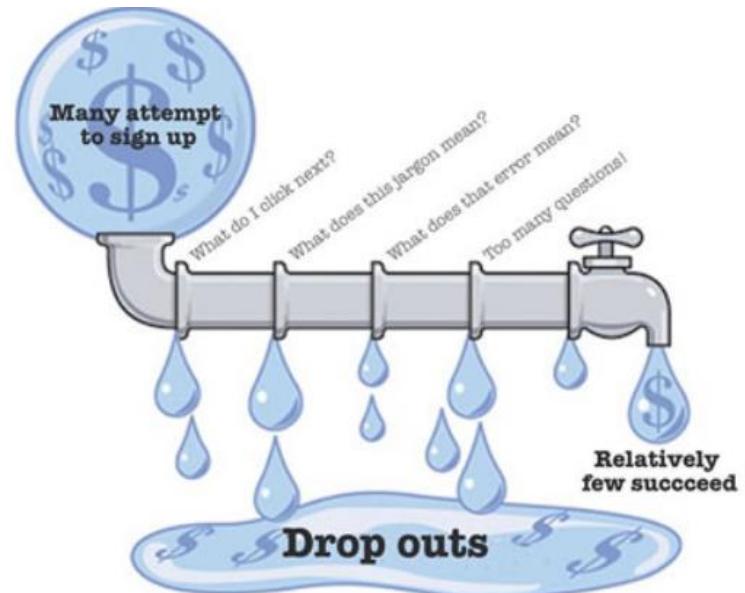
- UX includes all the users' emotions, preferences, perceptions, physical and psychological responses, ... that occur before, during and after use
- UX is **broader than usability**, it includes other aspects...
- Usability criteria can be used to assess aspects of user experience.

<https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-1:v1:en>

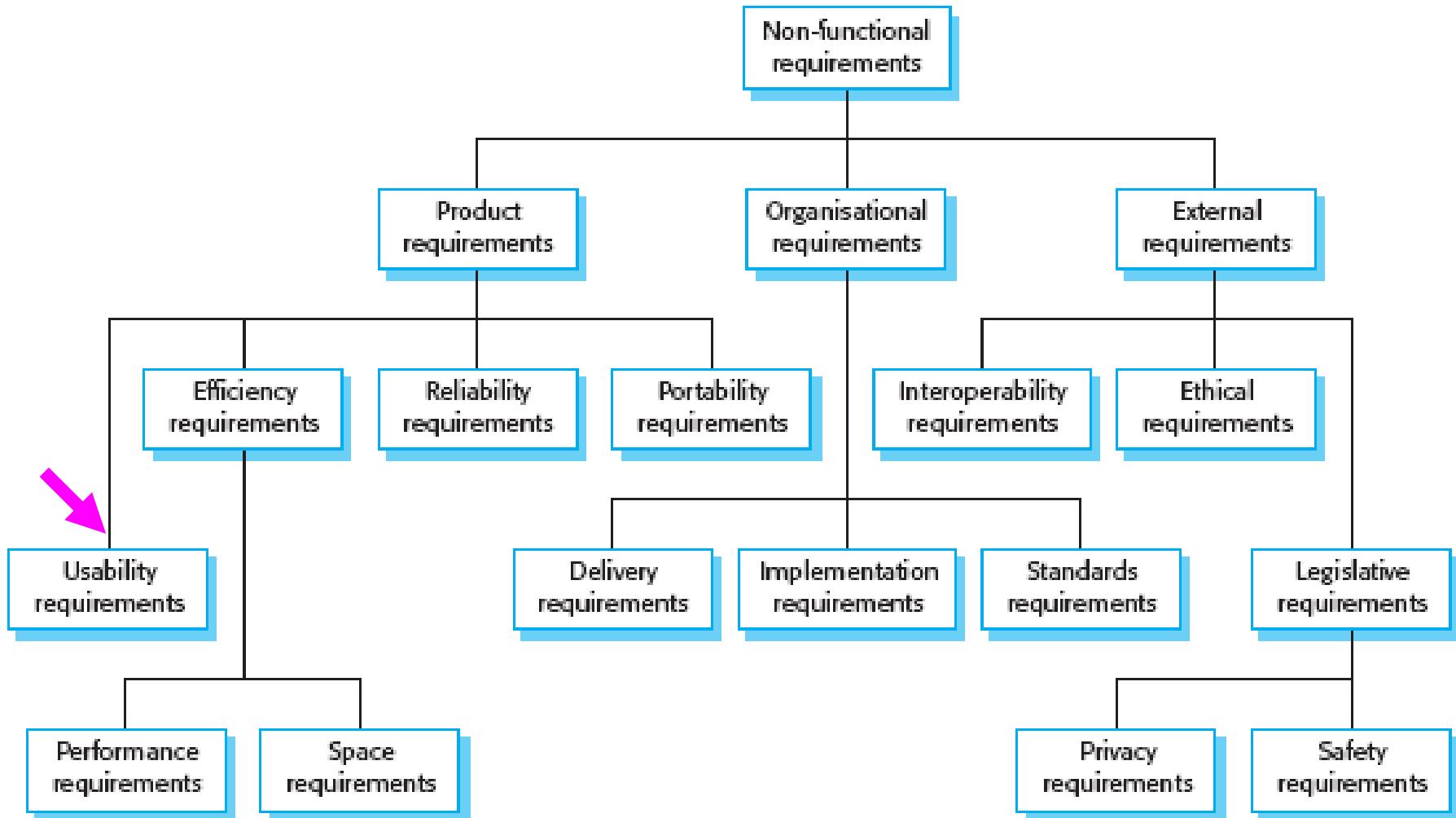
Usability

- Is directly related to the system capacity to allow users **attaining their goals through its usage**
 - Fundamental aspects:
 - **easy to use** (fast and with few errors) (efficiency, efficacy-> performance)
 - **satisfaction**
- Is defined in a **context of use**: is a system property of allowing specific users to perform specific tasks efficiently with efficacy and satisfaction
- Easy to learn and remember (learnability, memorability) is a related aspect

- Main usability benefits:
 - Higher user performance and satisfaction
 - Lower development costs
 - Lower support costs ...
- **Higher profits for everyone!**



Usability is a non-functional requirement





- ISO 13407 -> ISO 9241-210 (2010) addresses:
- ... Four Principles of Human-Centered Design:
 - active involvement of users
 - appropriate allocation of function to system and to user
 - iteration of design solutions
 - multi-disciplinary design
- ... and Four Human-Centered Design Activities:
 - understand and specify the context of use
 - specify user and organizational requirements
 - produce more than one candidate design solution
 - evaluate designs against requirements



- **ISO 9241-112:2017**
- **Ergonomics of human-system interaction — Part 112:**
- **Principles for the presentation of information**
- ... establishes ergonomic design principles for interactive systems related to the software-controlled presentation of information by user interfaces.
- It applies to the three main modalities
visual, auditory, tactile/haptic
- These principles apply to the perception and understanding of presented information
- are applicable in analysis, design, and evaluation of interactive systems
- ...

<https://www.iso.org/standard/64840.html>

Paradigms

- Examples of creative insight that enhanced interaction along the history of computing
- Inspirations for a conceptual model
- General approach adopted by a community for carrying out research
 - Shared assumptions, concepts, values, and practices
 - For example, desktop, ubiquitous computing, in the wild

Some usability paradigms (along the history of computing)



(VDUs)

Video Display Units (VDUs) (1950s)

Time sharing (1960s)

WIMP (Windows, Icons, Menus, Pointers) (1980s)

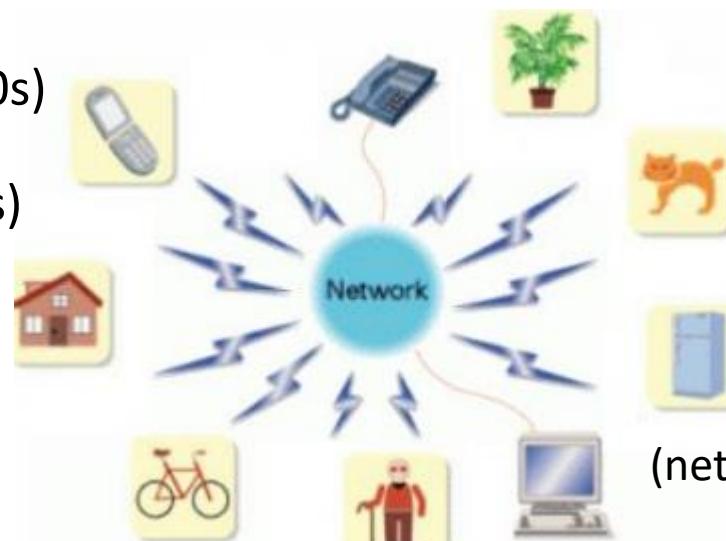
Direct manipulation (1980s)

WWW (1990s)



(WIMP)

Ubiquitous computing (1990s)



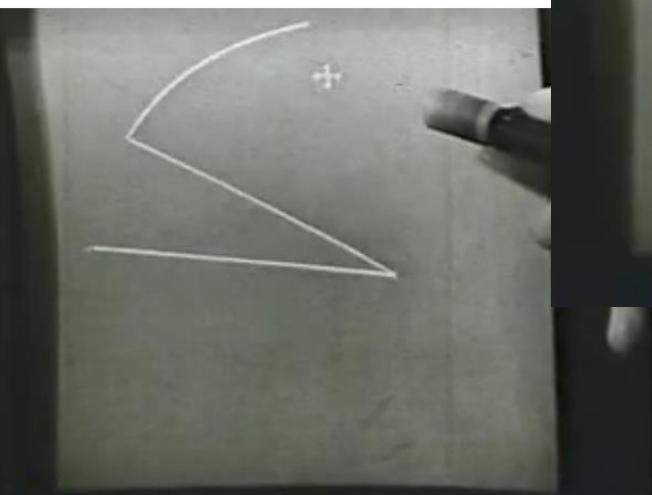
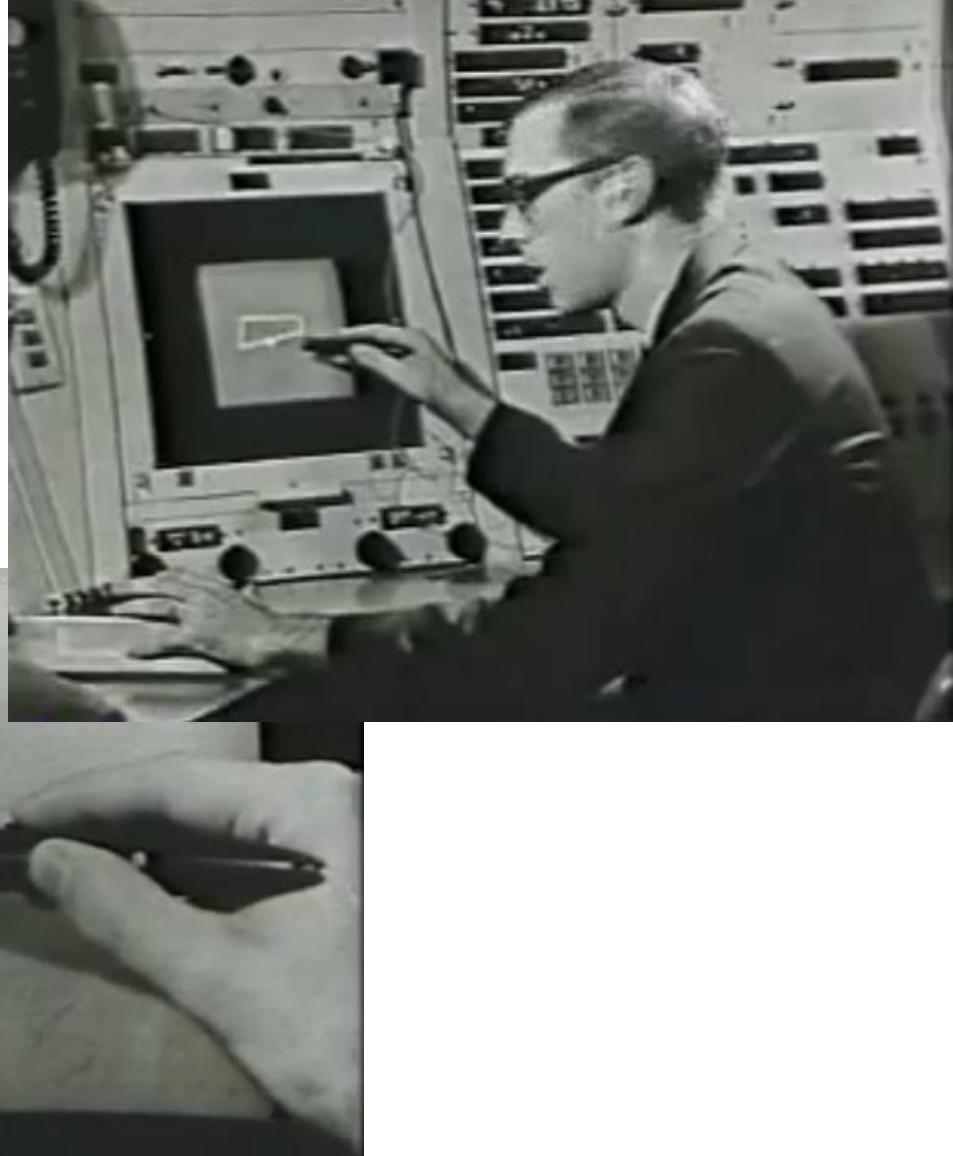
(anytime, anywhere...)



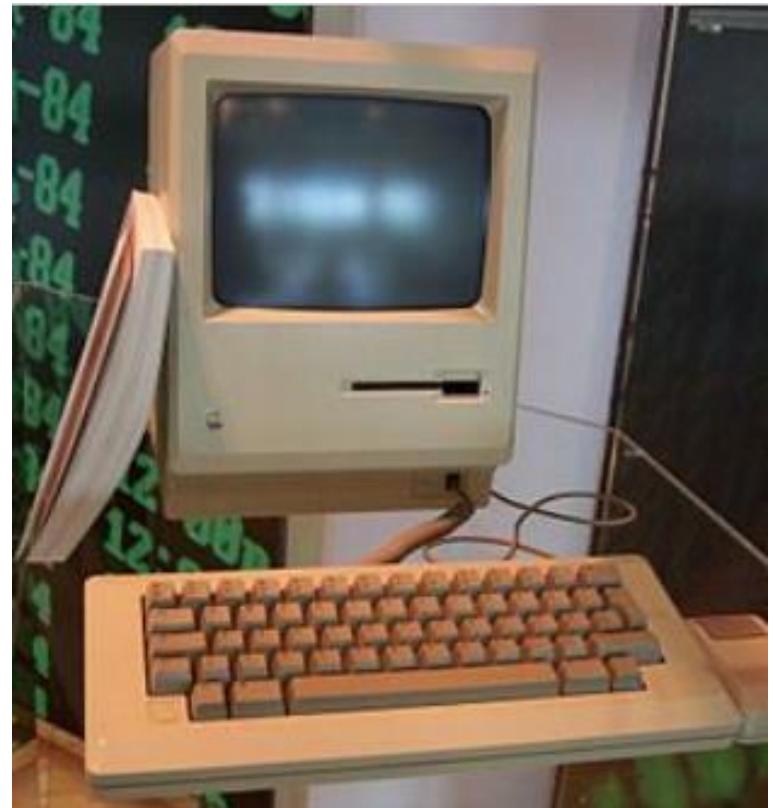
Wearable Computing (1990s)

(networking everything ...)

Sketchpad (Ivan Sutherland, 1963)



Alto and Macintosh



^

Apple Macintosh 512KB, 1984

< Xerox PARC, 1973

Ubiquitous computing (Ubicomp)

Mark Weiser, “The Computer for the 21st Century”, Scientific American, Sept 1991, pp. 94-104

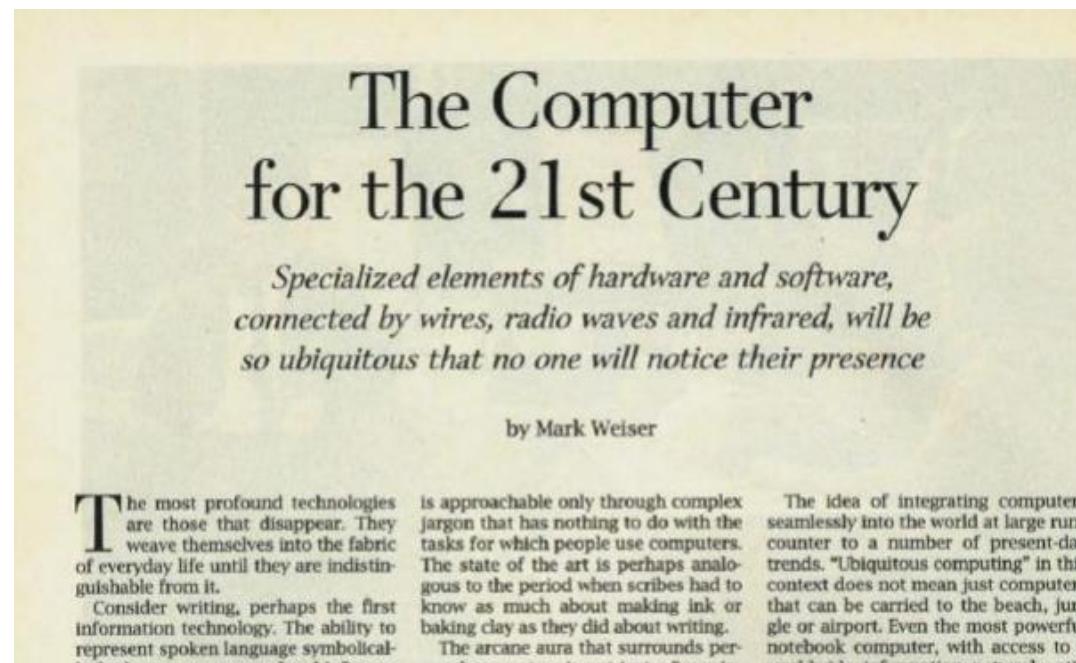
<https://www.ics.uci.edu/~corps/phaseii/Weiser-Computer21stCentury-SciAm.pdf>

<https://dl.acm.org/doi/10.1145/329124.329126>

- Computing everywhere and anywhere

- Related concepts:

- Pervasive computing
- Ambient intelligence
- Cyber-physical computing
- Internet of things
- Haptic computing



- Ubiquitous computing involves:
 - small, inexpensive, robust networked processing devices
 - distributed at all scales throughout everyday life
- Examples:
 - refrigerators "aware" of their suitably tagged contents
 - domestic control illumination and heating, continuously and imperceptibly considering the occupants
- Ubiquitous computing presents challenges across computer science:
 - in systems design and engineering, in systems modelling , in user interfaces

<https://www.youtube.com/watch?v=TrffIHzupTY>

Wearable computing



Steve Mann's 'GlassEye™' (aka EyeTap)

"the study or practice of inventing, designing, building, or using miniature body-borne computational and sensory devices. Wearable computers may be worn under, over, or in clothing, or may also be themselves clothes, i.e. "Smart Clothing" (Mann, 1996a).

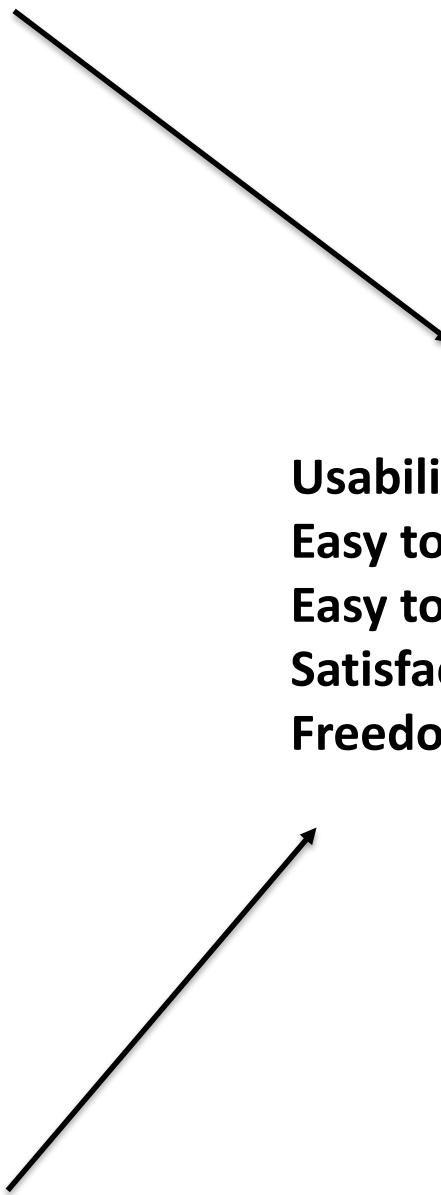
Other terms: "Body-Borne Computing" or "Bearable Computing"

<https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/wearable-computing>

Usability principles (a possible list)

- User compatibility
- Task compatibility
- Work-flow compatibility
- Product compatibility
- Feedback
- Coherence
- Familiarity
- Simplicity
- Flexibility
- Control
- Technology invisibility
- Robustness
- Error protection

Usability goals:
Easy to learn and memorise
Easy to use
Satisfaction
Freedom from risk



Principles should be used in interactive computing systems...

More conventional ...



other devices ...



less conventional interactive computing systems...



and critical interactive computing systems...

- E.g. medical devices:

[https://criticalsoftware.com/multimedia/critical
/de/KNrWVSj87-UxD-Pocket-Guide.pdf](https://criticalsoftware.com/multimedia/critical/de/KNrWVSj87-UxD-Pocket-Guide.pdf)

From the cockpit displays in planes to the dashboard layouts in cars, design can literally be a matter of life and death.

- or avionics:

<https://rightware.com/blog/modernizing-aircraft-cockpits-with-automotive-ui-design-knowhow/>

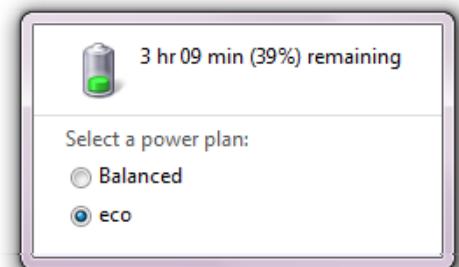
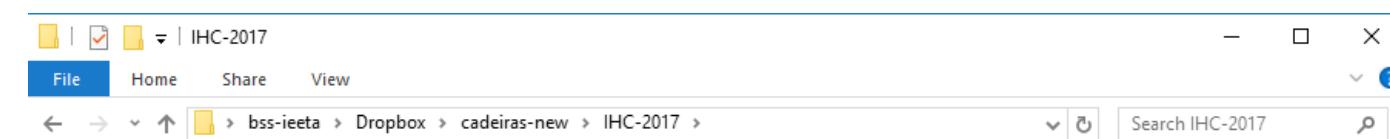
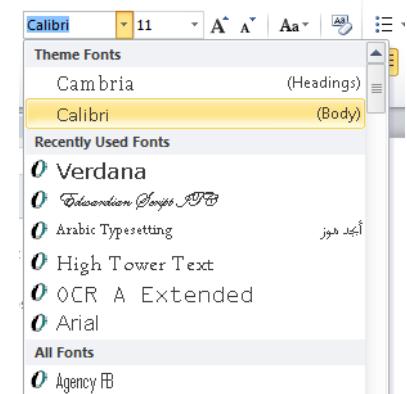
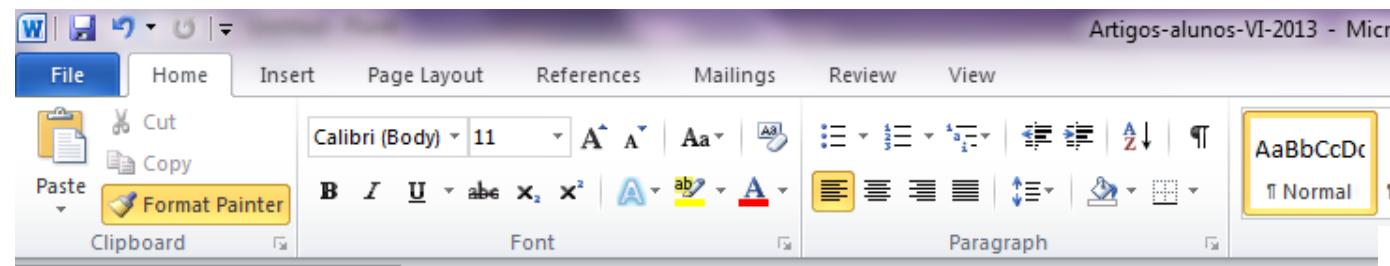
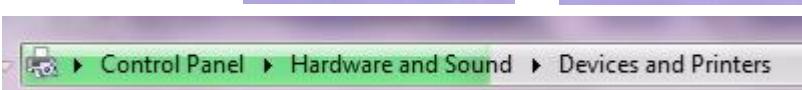
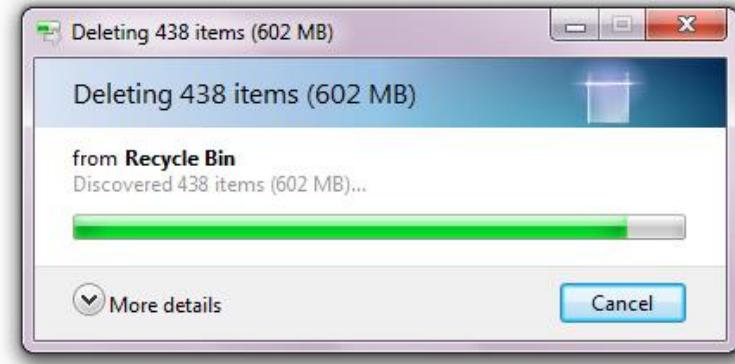


Everything you wanted to know about UxD for critical systems

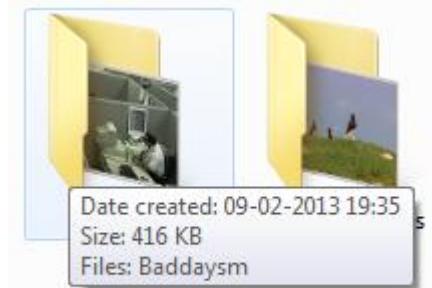
...but were afraid to ask



Visibility of the system status, Feedback (in more conventional platforms)



Name	Date modified	Type	Size
Escrever relatórios	09-02-2017 21:24	File folder	
outras-Universidades	09-02-2017 21:24	File folder	
assign3-task-analysis-esm.pdf	05-02-2017 12:28	Foxit Reader PDF ...	151 KB
Avaliação-paco.jpg	19-02-2017 09:05	JPG File	101 KB
avaliação-paco.png	19-02-2017 16:44	PNG File	41 KB
CHI2017-recommended sessions.docx	11-02-2017 23:49	Documento do Mi...	20 KB
CHI2017-recommended sessions.pdf	11-02-2017 23:48	Foxit Reader PDF ...	224 KB
course17-TA-Methods.pdf	05-02-2017 11:09	Foxit Reader PDF ...	211 KB
Heuristic evaluation-topics-2016.docx	12-02-2017 12:12	Documento do Mi...	14 KB



Feedback

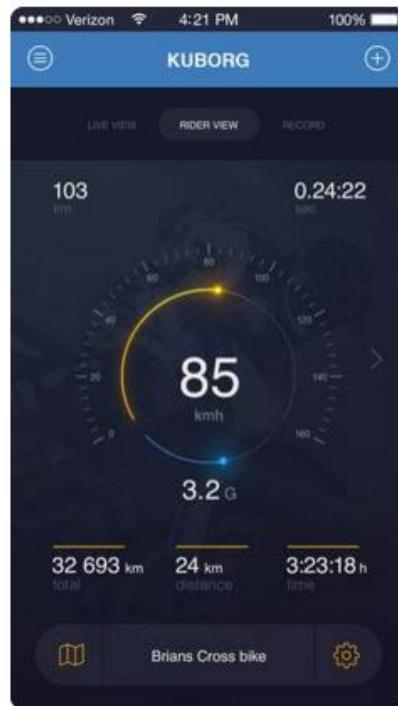
Visibility of the system status



TV off



ON



ON

TV on



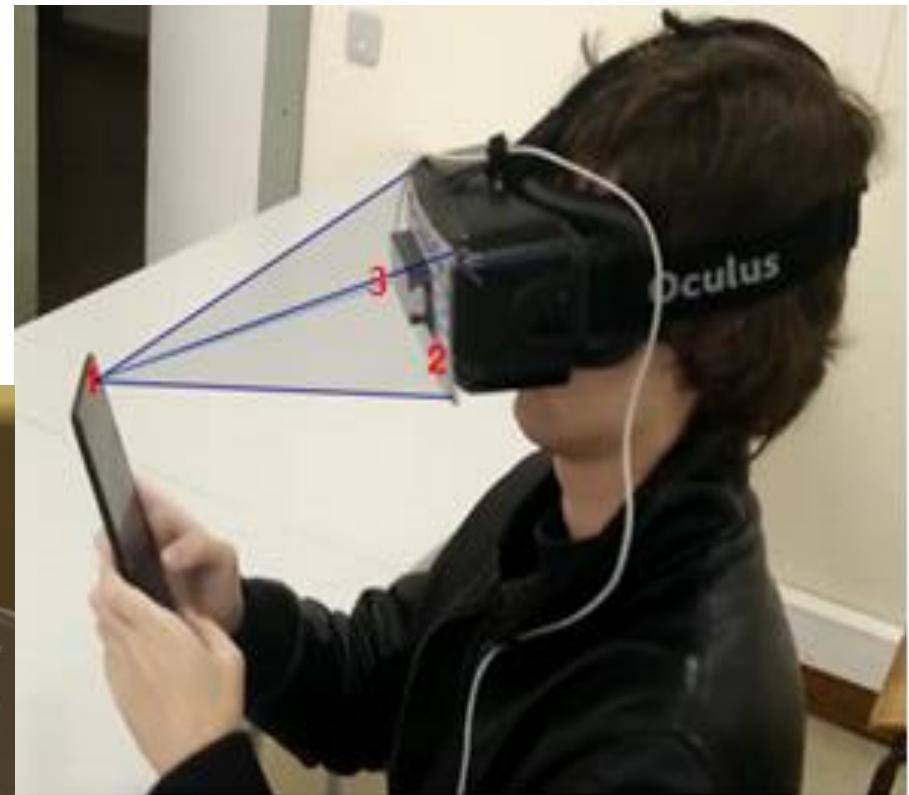
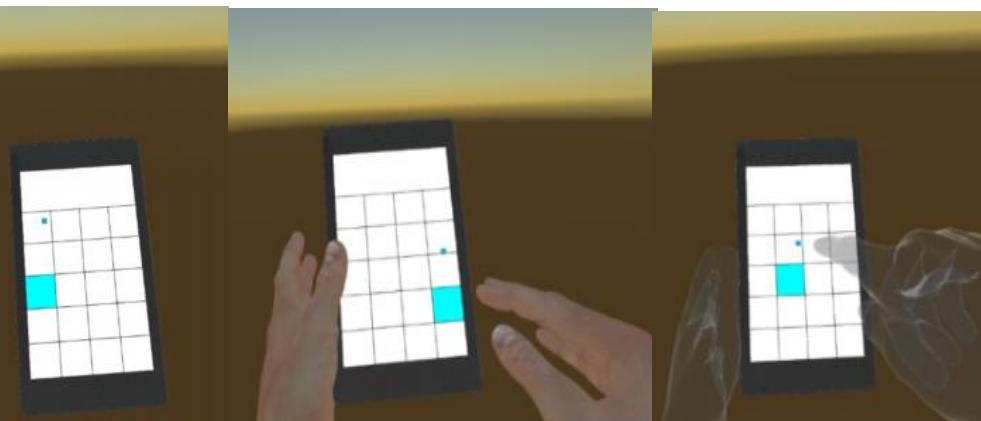
Feedback

Visibility of the system status

In a virtual reality system it is important to have:

- Feedback in tasks (navigation, manipulation, selection ...)
- Visibility concerning body position (avatar)
- ...

No avatar Realistic avatar Translucent avatar



Simplicity (hide complexity)

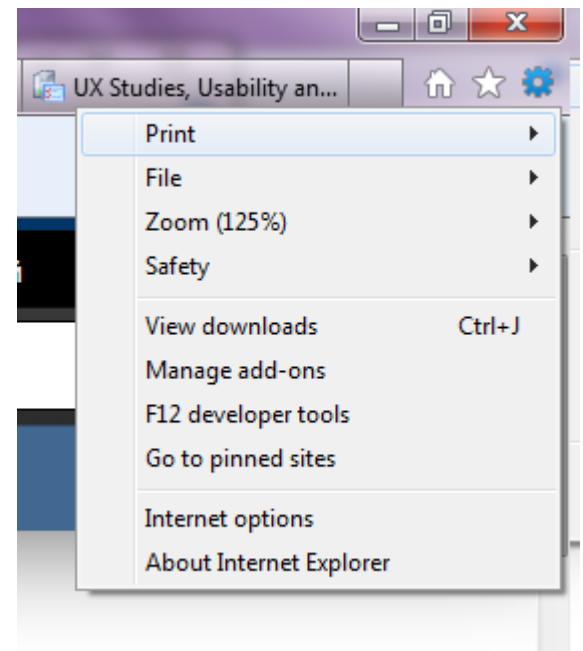
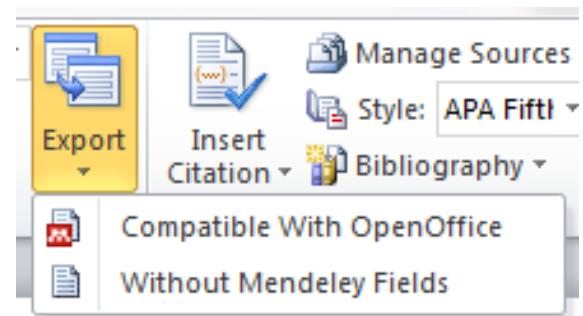
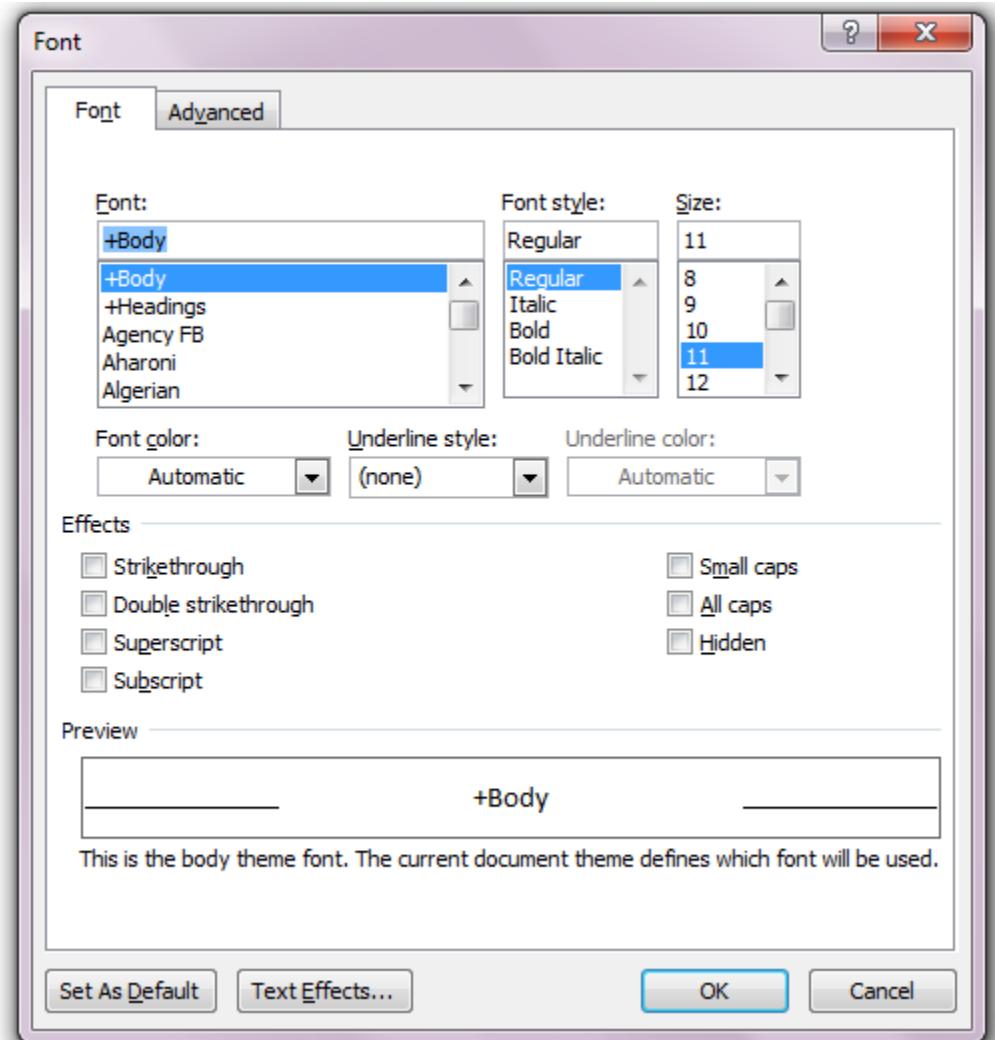
Avoid providing a large number of choices
and try solving problems using the
simplest solutions possible



A screenshot of a software interface, likely a web browser or document editor, showing a context menu. The menu includes options like 'Export', 'Insert Citation', 'Manage Sources', 'Style: APA Fifth', 'Compatible With OpenOffice', and 'Without Mendeley Fields'. Below the menu is a toolbar with icons for file operations. The main window shows a list of items under 'UX Studies, Usability an...'. The menu is open over the 'Print' option.

A screenshot of the Windows 'Font' dialog box. The 'Font' tab is selected. In the 'Font' section, '+Body' is selected from a dropdown list. The 'Font style:' dropdown shows 'Regular' is selected. The 'Size:' dropdown shows '11' is selected. Below these are sections for 'Font color:', 'Underline style:', and 'Underline color:'. The 'Effects' section contains checkboxes for 'Strikethrough', 'Double strikethrough', 'Superscript', 'Subscript', 'Small caps', 'All caps', and 'Hidden'. A 'Preview' section shows the word '+Body' in the chosen font. At the bottom are buttons for 'Set As Default', 'Text Effects...', 'OK', and 'Cancel'.

Simplicity (defaults hide complexity)



Familiarity

(profit from the user's experience)



User Documents



Private Folder
Closed



Private Folder
Open



Calendar



Contacts



Dashboard



Recycle Bin



CutePDF Writer



Familiarity

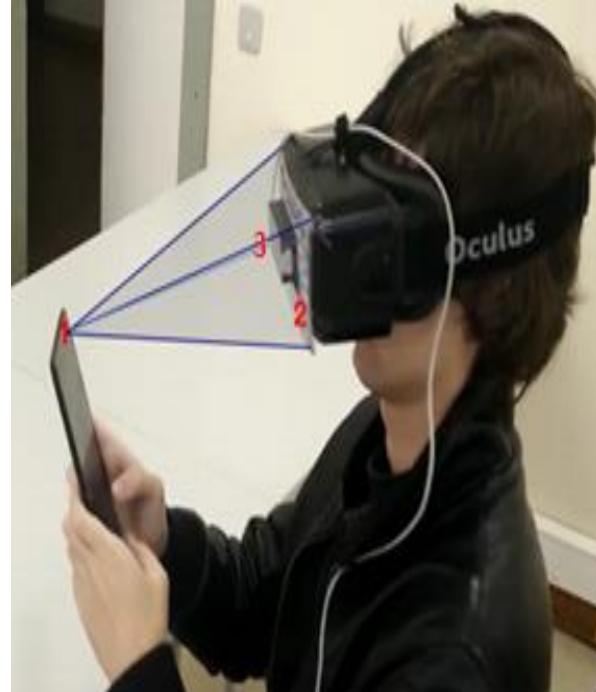
Familiar Icons



Familiarity

In less conventional interactive systems it is important to have:

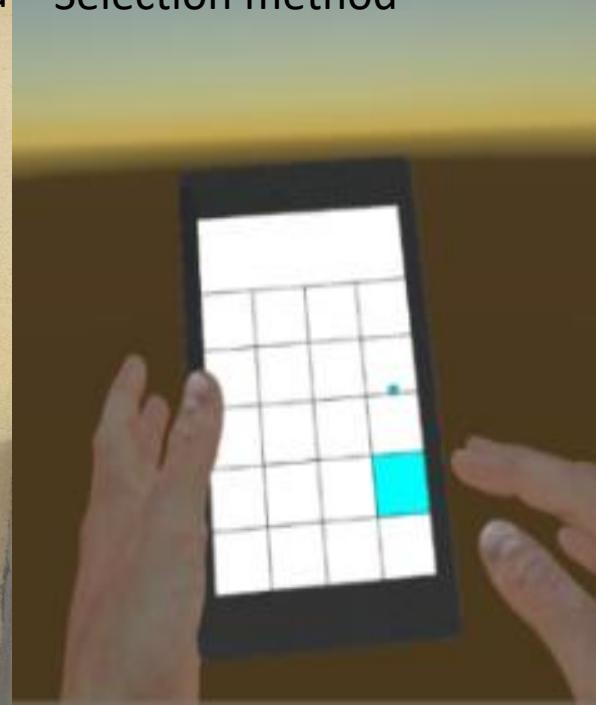
- Familiar gestures to perform tasks (navigation, manipulation, selection ...)



Bike navigation method



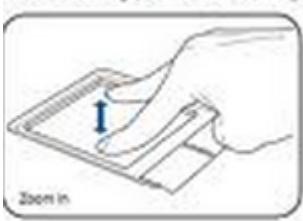
Selection method



Manipulation method

Flexibility

(let the user choose)



Taskbar and Start Menu

[Customize the Start menu](#) | [Customize icons on the taskbar](#) |
[Change the picture on the Start menu](#)



Ease of Access Center

[Accommodate low vision](#) | [Use screen reader](#) |
[Turn on easy access keys](#) | [Turn High Contrast on or off](#)

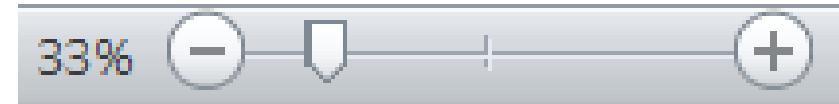


Folder Options

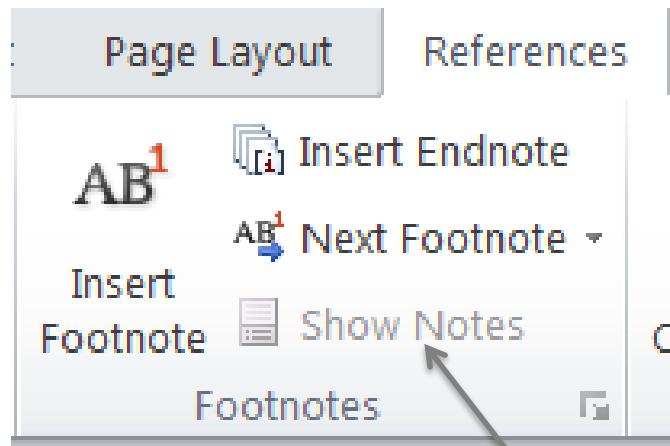
[Specify single- or double-click to open](#) |
[Show hidden files and folders](#)



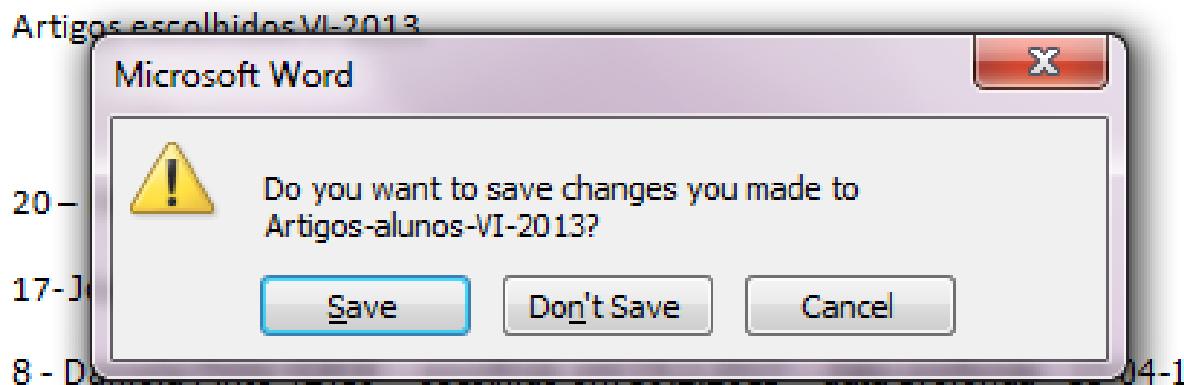
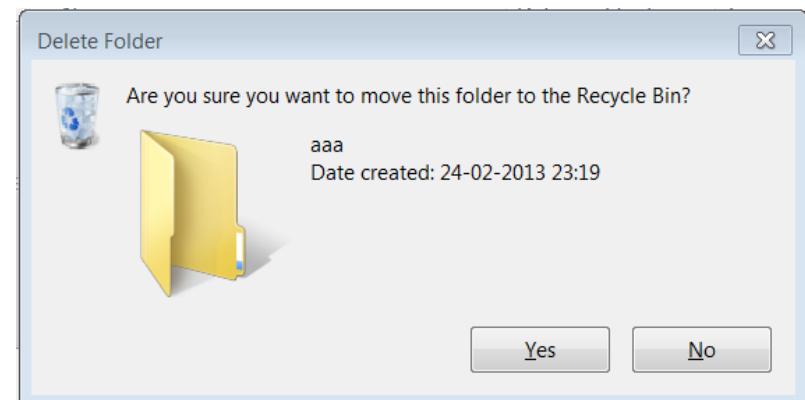
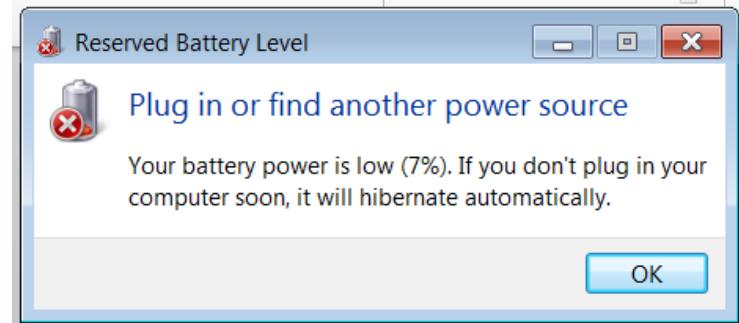
Let the user use different devices,
Or select the volume ...



Robustness and error prevention



Not accessible (in grey)



Old usability problems @ DETI (already solved!)

Solved : lights control @ room 4.1.02



Usability problems @ home



How does it open?



Wrong affordance!

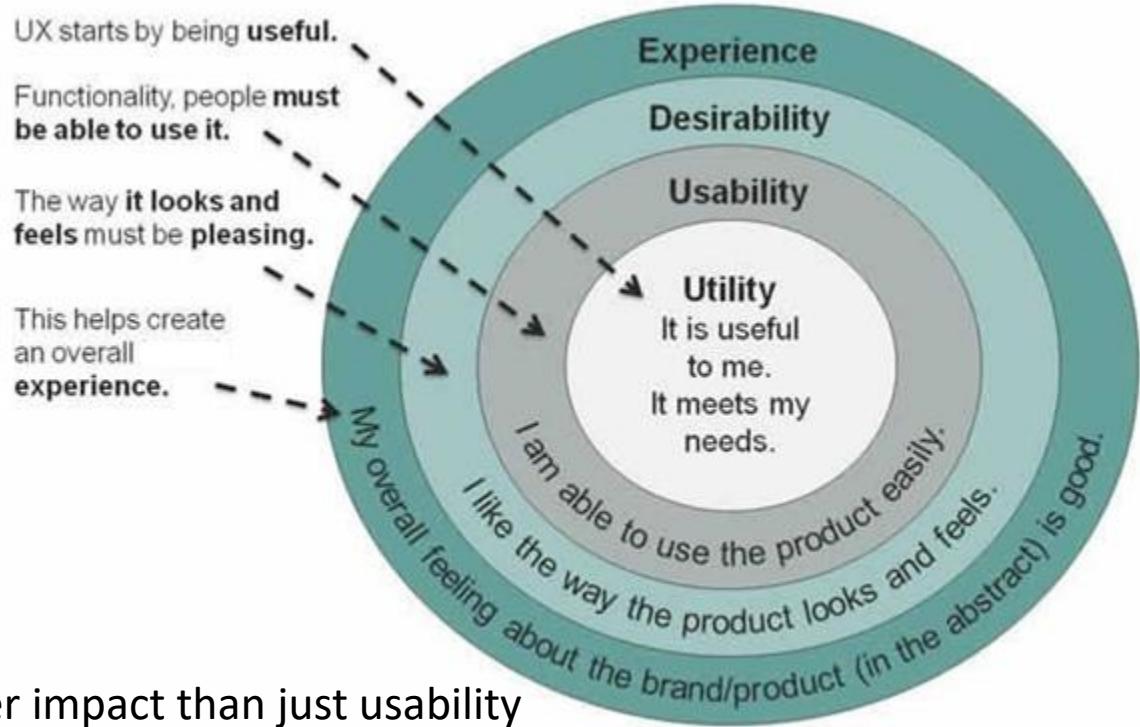


User Experience (UX)



- The ease in which people interact with a system to achieve specific goals
- The experience a person has when he/she interacts with a product (encompasses all aspects)

Usability -> function



A positive UX has a much greater impact than just usability

<https://www.nngroup.com/articles/ux-research-cheat-sheet/>

- **Usability** is concerned with the “effectiveness, efficiency and satisfaction with which specified **users** achieve specified goals in particular environments”
- **User experience** is concerned with “all aspects of the **user's experience** when interacting with the product, service”
- User experience (UX) involves a person's:
 - behaviors,
 - attitudes,
 - and emotions about using a particular product, system or service
- It includes the practical, experiential, affective, meaningful and valuable aspects of human-computer interaction and product ownership
- and also a person's perceptions of system aspects such as utility, ease of use and efficiency
- may be considered subjective and is dynamic as it is constantly modified over time

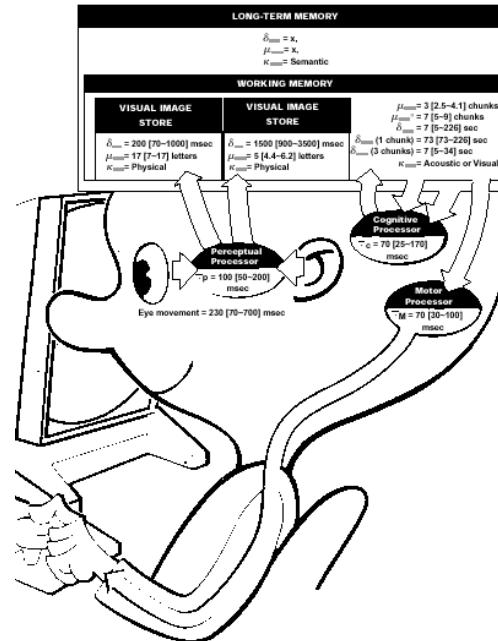
Main bibliography

- Sharp, H., Preece, J., and Rogers, Y., *Interaction Design- beyond Human-Computer Interaction*, Wiley, 2019
- Dix, A., J. Finley, G. Abowd, B. Russell, *Human Computer Interaction*, 3rd. ed., Prentice Hall, 2004
- Shneiderman, B., Plaisant, C., Cohen, M., and Jacobs, S., *Designing the User Interface: Strategies for Effective Human-Computer Interaction* , 5th ed., Addison-Wesley, 2009
- The Encyclopedia of Human Computer Interaction, 2nd ed., Interaction Design Foundation. <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed>



The User

User profile and Human Information Processing Systems



www.id-book.com

What we see does not depend only of the stimuli!

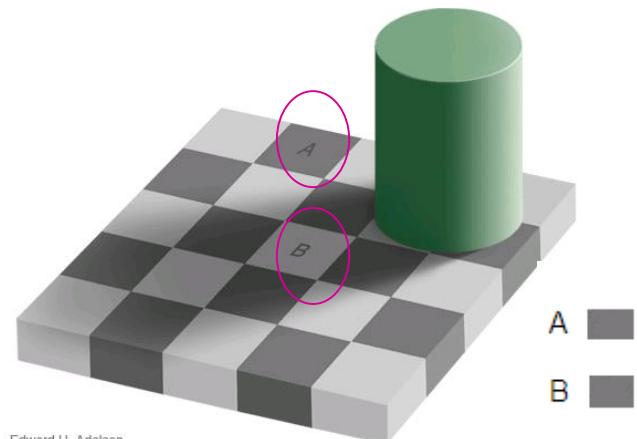
Two very powerful visual illusions:

The Ames' room



<https://www.youtube.com/watch?v=aS-vzPuZzuk>

The Adelson's illusion



Edward H. Adelson

<https://michaelbach.de/ot/lum-adelsonCheckShadow/index.html>

What we see does not depend only of the stimuli!

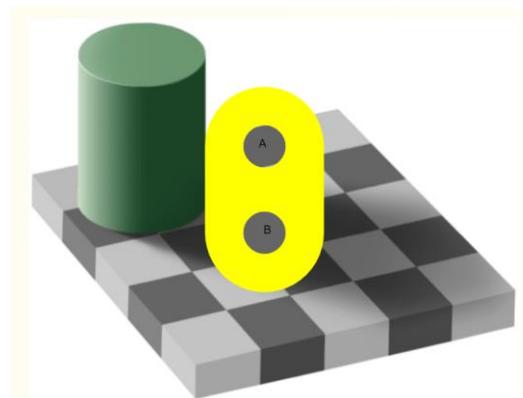
Two very powerful visual illusions:

The Ames' room



<https://www.youtube.com/watch?v=aS-vzPuZzuk>

The Adelson's illusion



<https://michaelbach.de/ot/lum-adelsonCheckShadow/index.html>

Outline

- Users Profile – relevant characteristics for interactive systems
- Human Information Processing System (HIPS)
 - Perceptual sub system
 - Senses
 - Sight, Hearing, Touch, Smell, Taste
 - + Proprioception, Kinesthesia ...
 - Cognitive sub-system
 - Memory
- Some implications on the design of interactive systems

Users' profile

- Human Information Processing System (HIPS)

- Knowledge and experience

- Work and task

- Physical characteristics

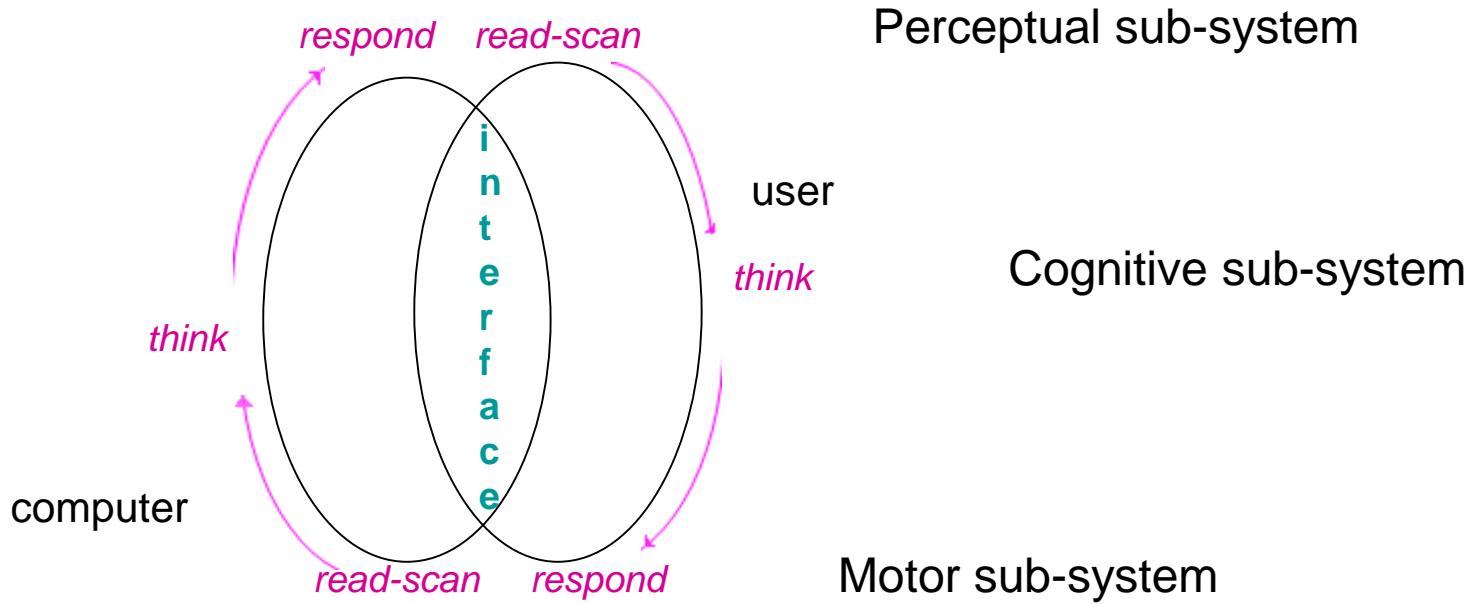
- Environment

- Tools

More variable among users

**There are many user models to be used in the design of Interactive systems
(e.g. personas, GOMS, KLM, ...)**

Dialog in an interactive system



Human Information Processing System (HIPS)

- Humans have different capabilities that might be considered when designing interactive systems
- Information is received through various I/O channels
- Information is stored in memory
- Emotions may influence capabilities
- Users share common characteristics but differences cannot be ignored

Human Information Processing System (HIPS): main aspects relevant to interactive systems design

Perceptual sub-system

memory – perceptual buffer (iconic, echoic, ...)

process – pattern recognition

Cognitive sub-system

memory -

short term/working memory (STM)

long term memory (LTM)

processes -

selective attention

problem resolution
learning

...

Motor sub-system

HIPS
bottleneck

Perceptual sub-system – I/O



VISION HEARING TOUCH



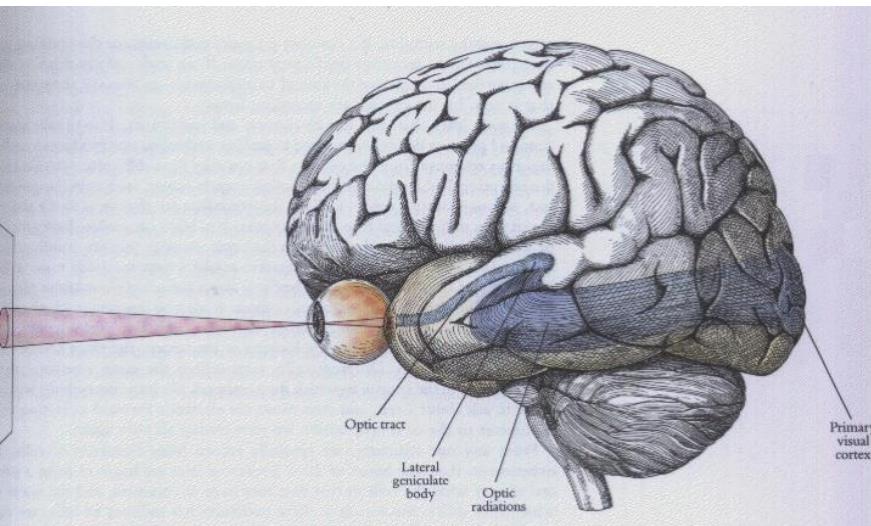
SMELL TASTE

The five Aristotelian senses

- Input: 5+ senses
Sense is a physiological capacity of organisms that provides data for perception
 - Some more relevant than other
 - For HCI, vision is preferred, but hearing and touch are more and more important ...
- Proprioception (more important in some types of systems: Virtual reality)
- Output: communication system
 - vocal, gestures, eye gaze, ,...

Vision

- Relevant for HCI:
 - Compensation of movements and illumination changes
 - Context used to solve ambiguities
 - May be tricked: Visual illusions come from excessive compensation.



Eye – sensor

Brain - processor

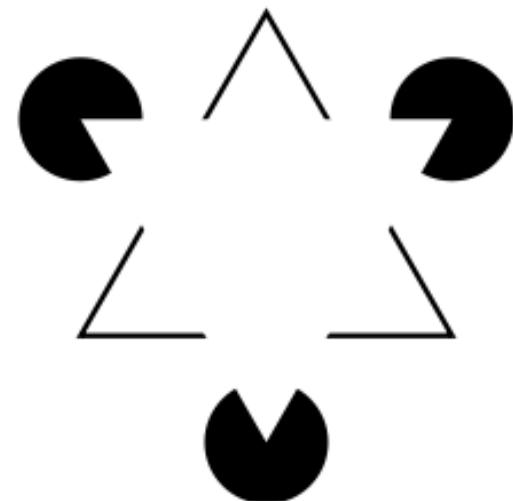
(Hubel, 1988)

Pattern Recognition

Process that matches information from a stimulus with information retrieved from memory

- It is a very powerful process
- It is subconscious
- It does not use only current data
- It solves ambiguities
- Occurs also in other senses

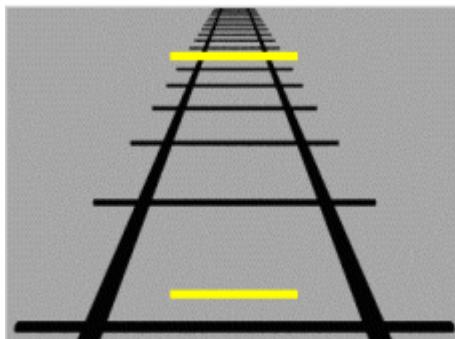
The quick brown
fox jumps over the
the lazy dog.



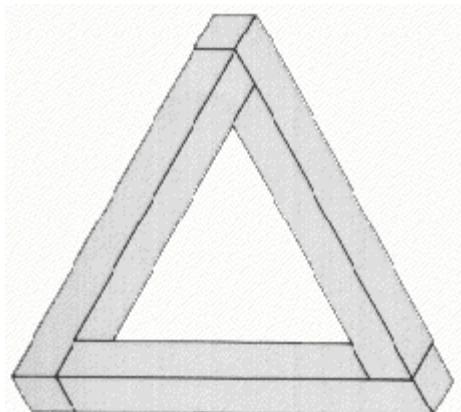
(Kaniza illusion, Wikipedia)

Visual Illusions illustrate that **what we see does not depend only of the stimulus**

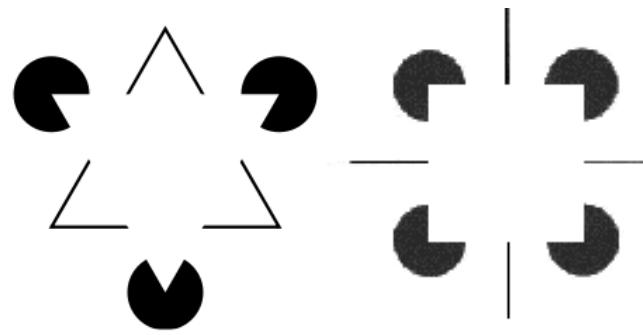
- bring out good adaptations of our visual system to standard viewing situations
- under artificial manipulations can cause inappropriate interpretations of the scene



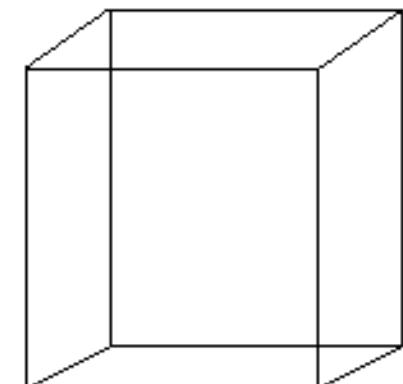
Ponzo illusion



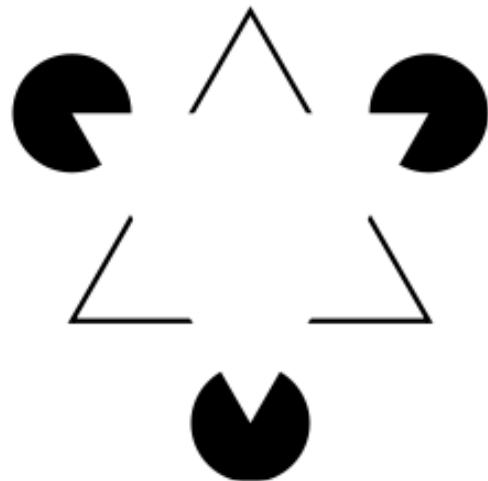
Penrose triangle:
Impossible object?



Kanizsa illusion



Necker cube



Kanizsa illusion:

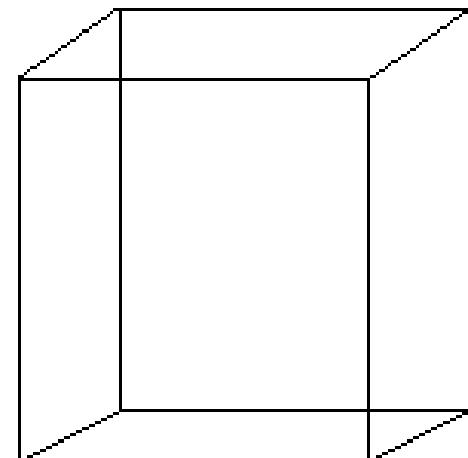
Although there are no actual triangles a sort of pattern recognition phenomenon is triggered and the image is interpreted as two overlapping triangles (simple explanation)

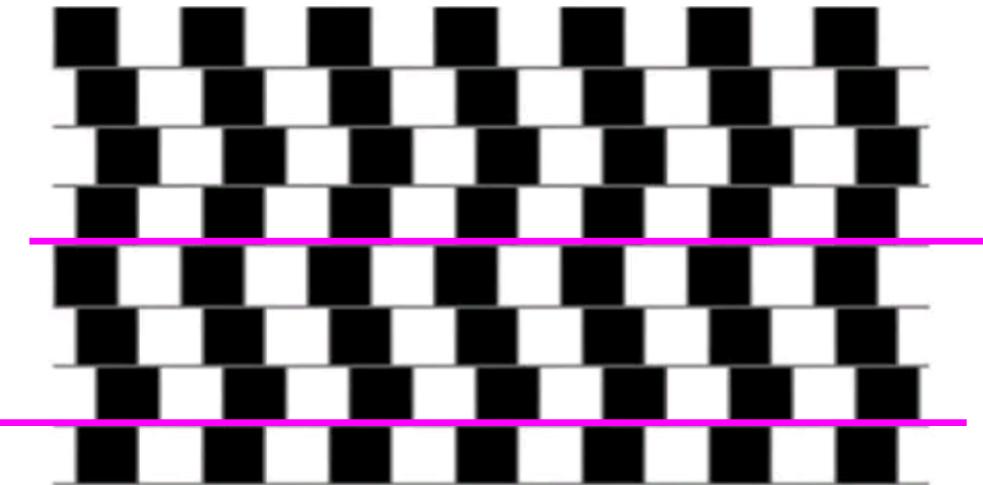
https://en.wikipedia.org/wiki/Illusory_contours

Necker cube:

Cube with no visual cues as to its orientation; it can be interpreted to have either the lower-left or the upper-right square as its front side

https://en.wikipedia.org/wiki/Necker_cube



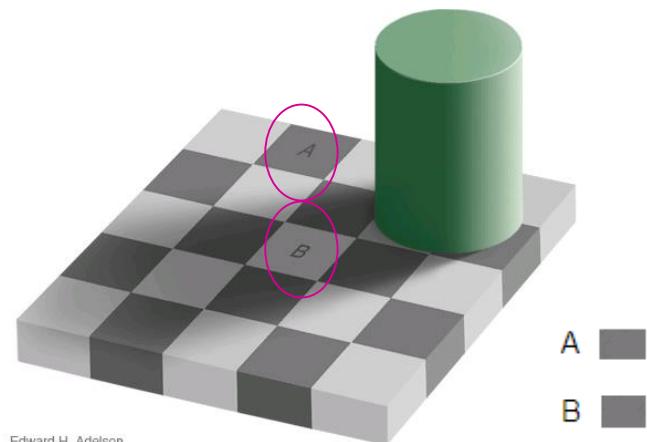


<http://www.youtube.com/watch?v=2XqPOJLUM1s>



<http://www.youtube.com/watch?v=URLRdcnU6Hk&NR=1>

https://www.youtube.com/watch?v=-IWk5NkxQF8&ebc=ANyPxKoRFxfOSCgdPavBoMpgPrXjRRVqZmhiAvIBDgThnPfndq-gheNYZ-6cNRv2yYwN5SqX52DQGDWjvBnzUQQ-_N6iCVgMdQ

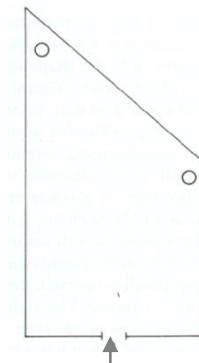


<http://www.michaelbach.de/ot/>

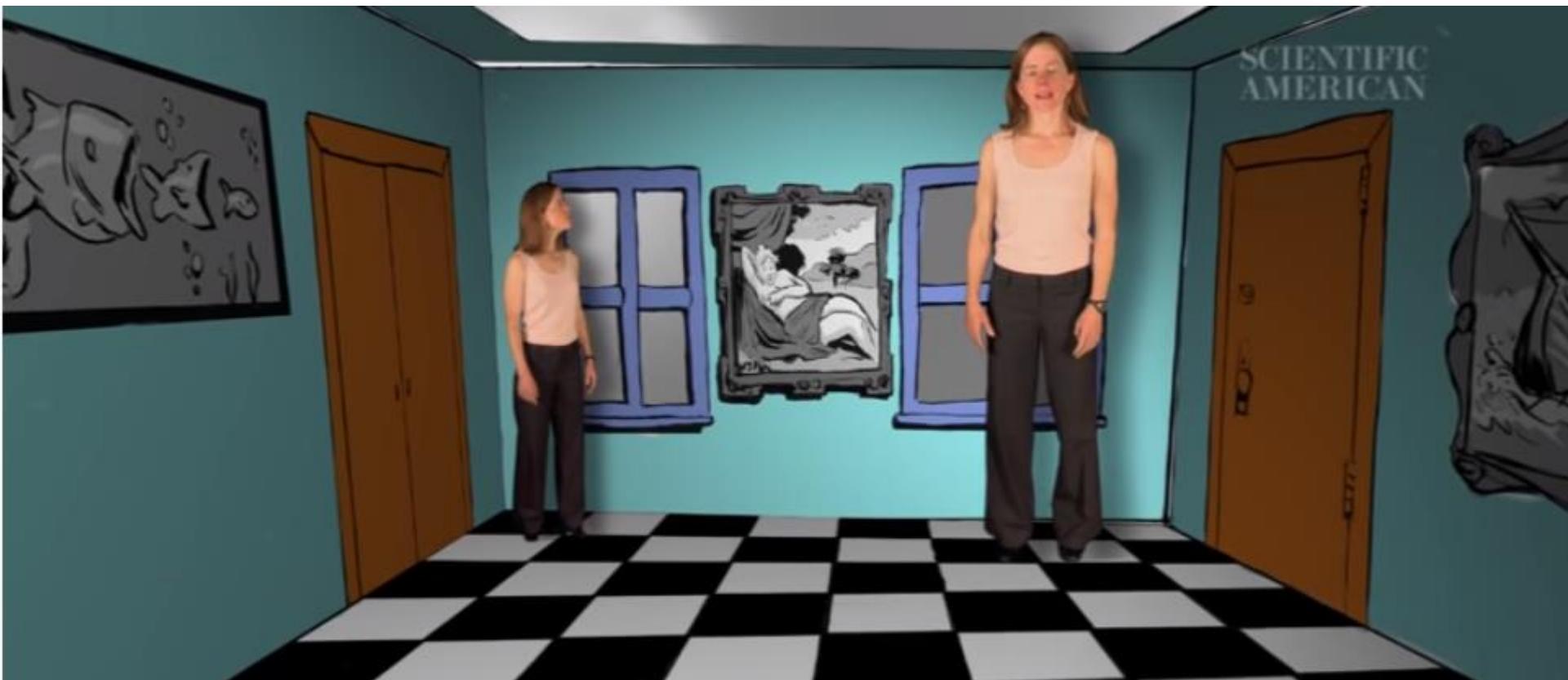
Ames Room

(what we see does not depend only of the stimulus)

A room that pushes the boundaries of human perception...



Point of view



<https://www.youtube.com/watch?v=qJhyu6nlGt8>

17

<https://www.youtube.com/watch?v=aS-vzPuZzuk>

Other senses

- Hearing
 - Information on direction, objects and distance
 - Only sense that is really 3D
 - Cannot be “turned off”
 - Human hearing - 20Hz to 15KHz
 - Filtering is possible (Background noise – “cocktail party” example)



[https://en.wikipedia.org/
wiki/Sense](https://en.wikipedia.org/wiki/Sense)

Other senses

- Touch
 - Important feedback
 - Key senses for people with sight problem
 - Several receptors in skin:
 - Termoreceptors: cold and hot
 - Nociceptor: pain
 - Mecanoreceptor: pressure
 - Some areas more sensitive (fingers)



[https://en.wikipedia.org/
wiki/Sense](https://en.wikipedia.org/wiki/Sense)

Simulators are complex interactive systems that stimulate several senses...

<https://www.the737experience.co.uk/>

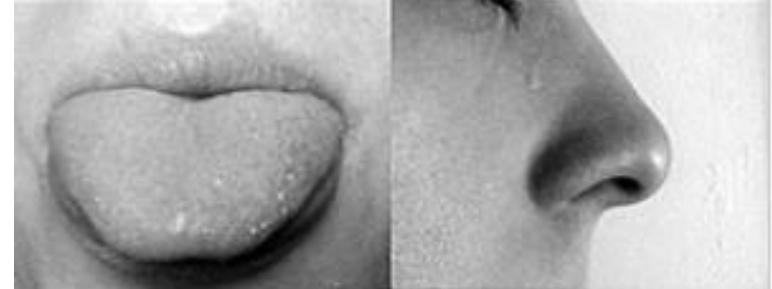
https://en.wikipedia.org/wiki/Flight_simulator



<https://surgicalscience.com/systems/lapsim/>

Other senses

- Smell and Taste
 - Complex chemical senses
 - High latency
 - Difficult to use in HCI
 - Some experimental work exists (and some commercial products)
- And others (as proprioception - awareness of your body position)



<https://en.wikipedia.org/wiki/Sense>

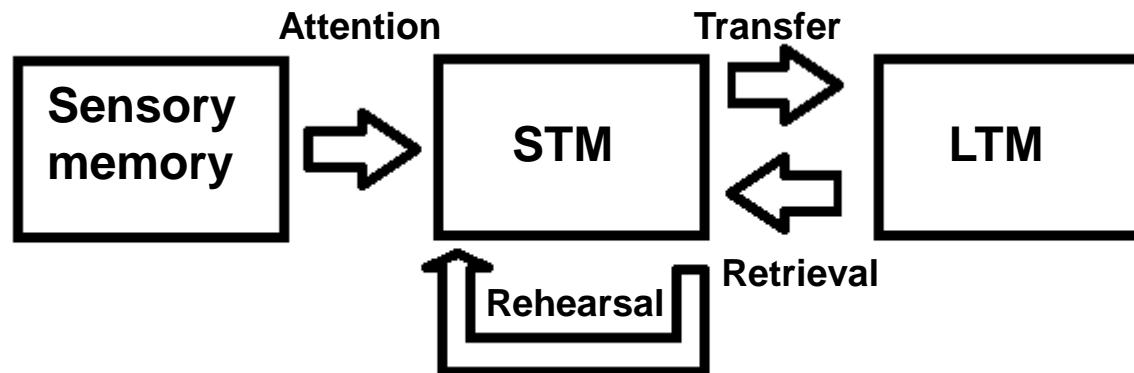
<https://www.khanacademy.org/test-prep/mcat/processing-the-environment/somatosensation/v/proprioception-kinesthesia>

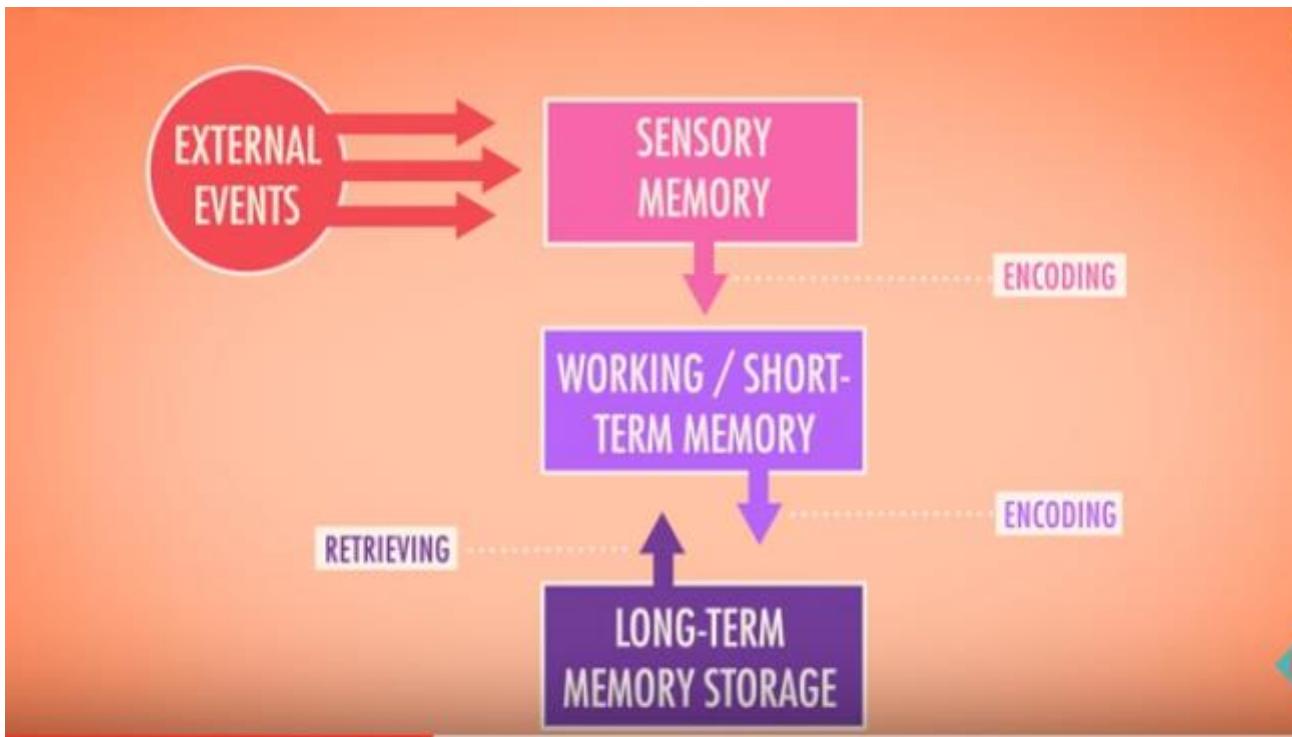
<https://neuro-marseille.org/en/blog/our-true-sixth-sense-on-arte/>

Memory

The **Atkinson–Shiffrin model** (1968) (a.k.a. multi-store model or modal model) asserts that human memory has three components:

- Sensory memory
- Short-term memory -> working memory (more recent concept)
- Long-term memory





How we make memories - Crash Course Psychology #13

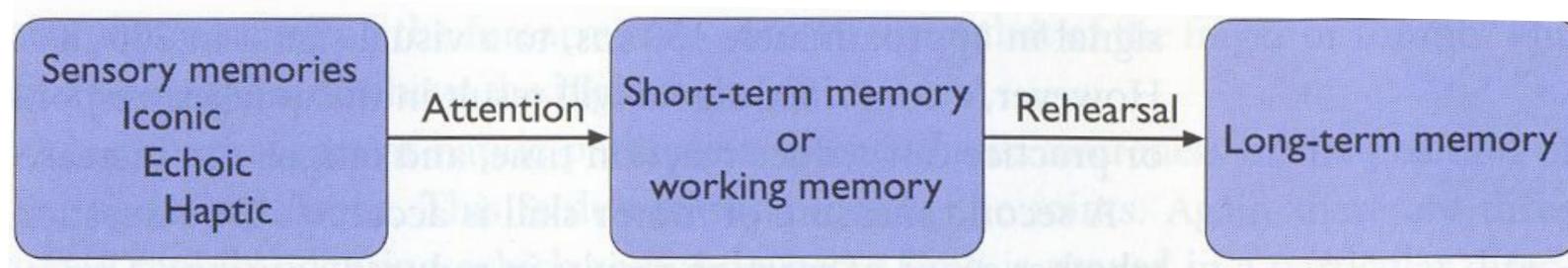
<https://www.youtube.com/watch?v=bSycdIx-C48>

How we remember and forget - Crash Course Psychology #14

<https://www.youtube.com/watch?v=HVWbrNIls-Kw>

Memory

- The Atkinson–Shiffrin model (1968) memory has three components
 - Sensory memory / iconic memory, very short
 - A few seconds
 - Short-term memory / working memory
 - +/- 18 seconds, 7+/-2 items
 - Long-term memory
 - ~Infinite capacity



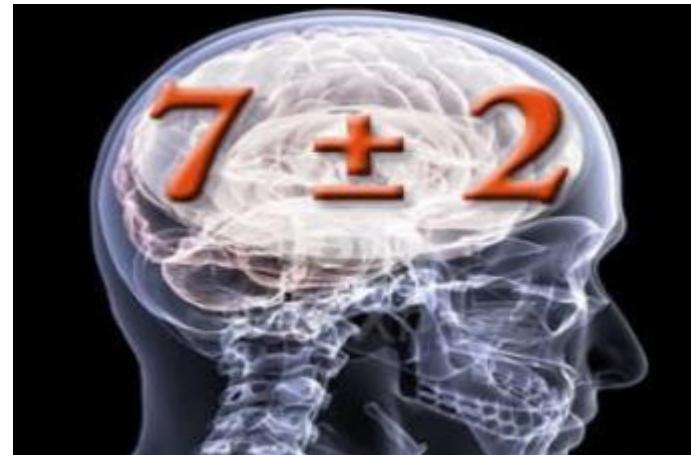
Short Term Memory (STM) / Working memory

- Working memory:
 - Mental calculus (6×35)
 - Reading (understand a sentence)
 - ...
- Characteristics:
 - Quick access +/- 70ms
 - Quick forget
 - Limited capacity 7 +/- 2
 - Can be increased with **chunking**, to enter long-term memory

Short Term Memory (STM) / Working memory

a few characteristics:

- Short duration: a few seconds (<30s)
- Limited capacity: 7 ± 2 elements



what is an element?

example: try to memorize the following numbers 649325401741
 111122223333

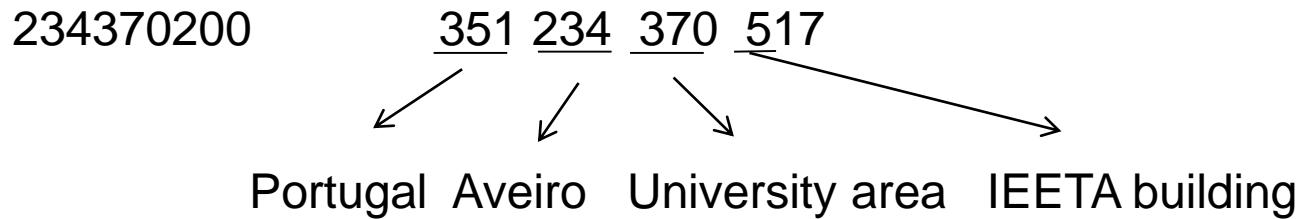
which is easier?

and this one: 351234370517

Short Term/ working Memory (STM)

649325401741 → 12 digits

111122223333 → 3 digits and a rule

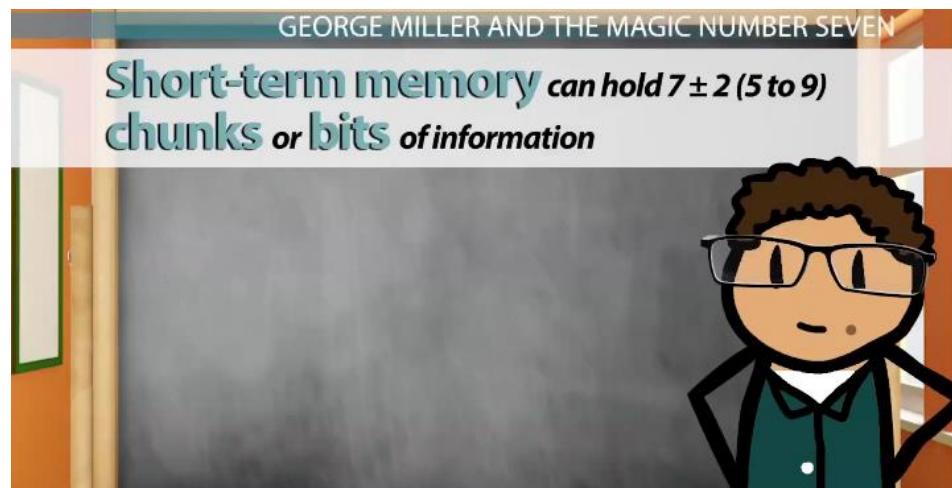


These numbers correspond to different “chunks”

Chunk: the largest meaningful unit that a person recognizes; depends on the person knowledge

- **Chunking** refers to an approach for making more efficient use of short-term memory by grouping information
- Resulting chunks are easier to commit to memory than a longer uninterrupted string of information.
- Can be used for making more efficient use of short-term memory by grouping information

<http://study.com/academy/lesson/george-miller-psychologist-theories-on-short-term-memory-lesson-quiz.html>



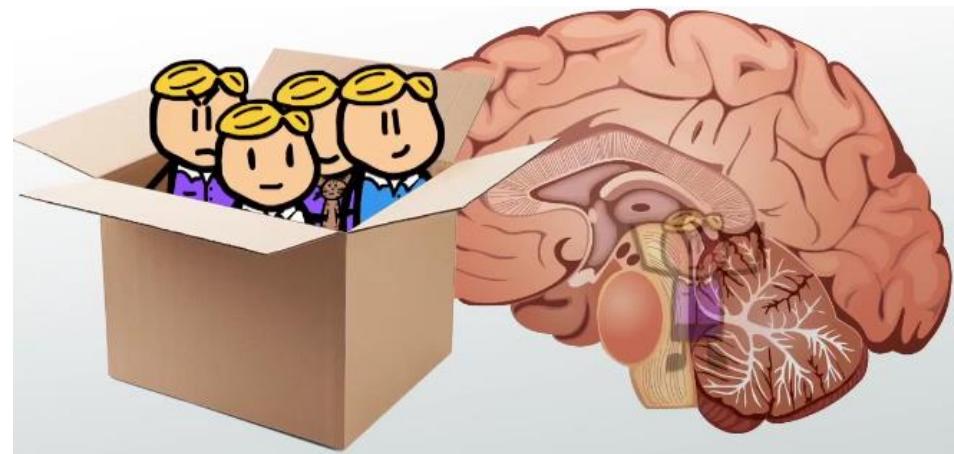
Long Term Memory

Stage of the dual memory model (Atkinson-Shiffrin memory model)

Informative knowledge can be stored for long periods of time

Main characteristics:

- “Infinite” capacity and duration
- non reliable access



<http://study.com/academy/lesson/long-term-memory-definition-types-examples.html>

Long-term Memory - Recognition vs. Recall

- Information retrieval - Recognition vs. Recall
 - Recognition: remembering with the help of a visible cue
 - aka “Knowledge in the world”
 - Recall: remembering with no help
 - aka “Knowledge in the head”
- Recognition is much easier
 - so e.g. menus are more learnable than command languages

Design Implications

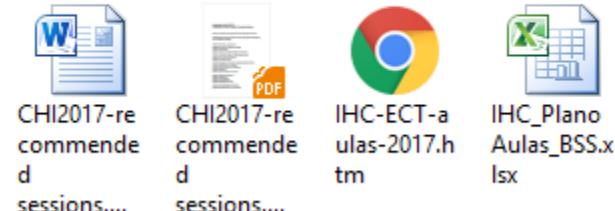
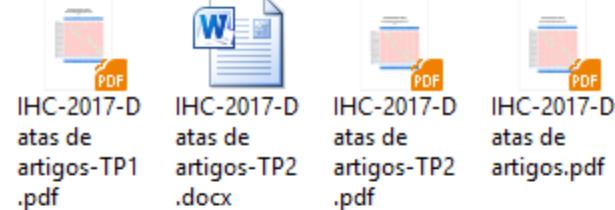
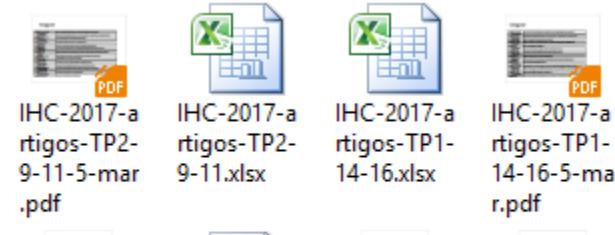
Memory

- Reduce cognitive load by avoiding long and complicated procedures for carrying out tasks.
- Design interfaces that promote recognition rather than recall by using familiar interaction patterns, menus, icons, and consistently placed objects.
- Provide users with a variety of ways of labeling digital information (e.g. files, emails, and images) to help them easily identify it again through the use of folders, categories, color, tagging, time stamping, and icons.

Selective attention



Occurs when we block out certain features of our environment and focus on one particular feature



It may be:

- Voluntary

- Involuntary

Both can be (and are) exploited in UIs

Calling your attention to an application



<https://dictionary.apa.org/selective-attention>

<https://www.khanacademy.org/test-prep/mcat/processing-the-environment/attention-language/v/selective-attention>

Design Implications

Attention

- Consider context. Make information salient when it requires attention at a given stage of a task.
- Avoid cluttering visual interfaces with too much information.
- Consider designing different ways of supporting effective switching and returning to a particular interface.

Design Implications

Learning

Design interfaces that encourage exploration

Design interfaces that constrain and guide users to select appropriate actions when initially learning.

Design Implications

Problem solving

- Provide information and help pages that are easy to access for people who want to understand more about how to carry out an activity more effectively (e.g., web searching).
- Use simple and memorable functions to support rapid decision-making and planning.
- Enable users to set or save their own criteria or preferences

HIPS Strengths and weaknesses (*versus computer*)

Strengths

- LTM ~infinite capacity
- LTM duration and complexity
- Capacity to learn
- Powerful selective attention
- Powerful pattern recognition process

Weakesses

- STM limited capacity
- STM limited duration
- Error prone processing
- Non reliable access to LTM
- Slow processing

Recommendation:

Assign tasks between user and computer according to the capacities of each

Example: minimize the users STM load
do not ask the user to perform computations

Emotion

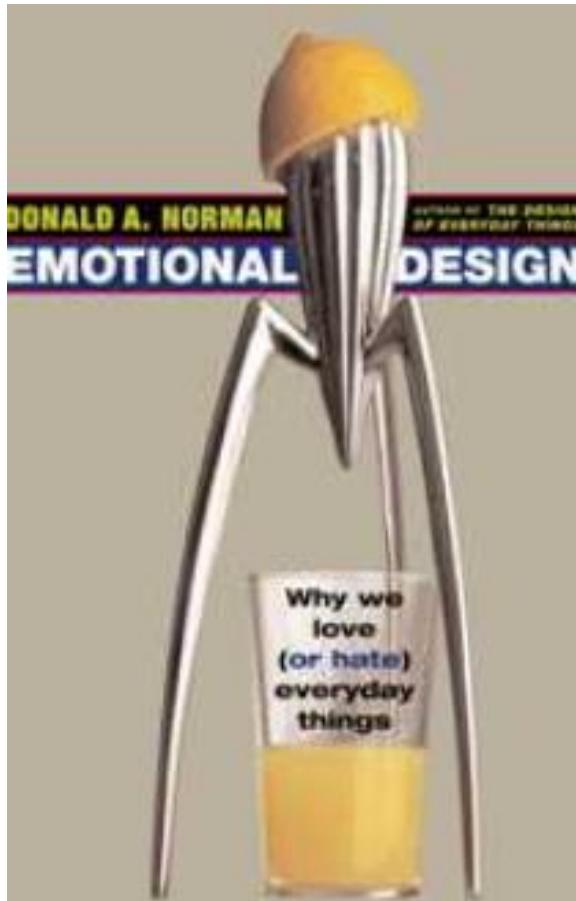
- Various theories on how it works
- Involves cognitive and physical response to stimuli
- Biological response to a physical stimuli is called affect
- Affect Influences how we react to situations

“Negative affect can make it harder to do even easy tasks; positive affect can make it easier to do difficult tasks”

(Donald Norman)

Not only the cognitive system is important

But also the **emotional system**



- Emotional design is a critical part of design
- Other Donald Norman videos:
<http://www.youtube.com/watch?v=Wl2LkzIkacM>
http://www.youtube.com/watch?v=_PM3uqPNrWY
<https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/affective-computing>

- As ... moved from designing and evaluating work-oriented applications towards dealing with **leisure-oriented applications**, ... we have had to consider e.g. what constitutes an *experience*, how to deal with users' ***emotions***, and understanding aesthetic practices and experiences"



<https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/affective-computing>

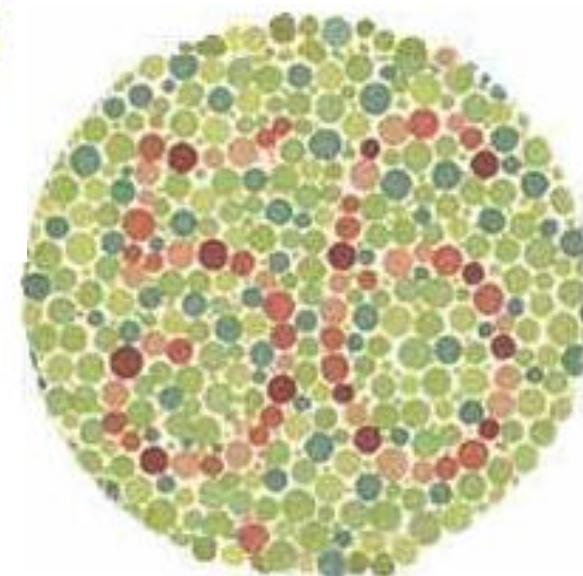
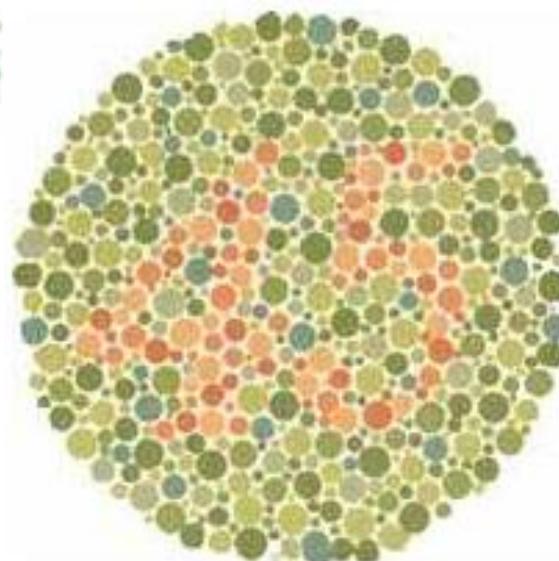
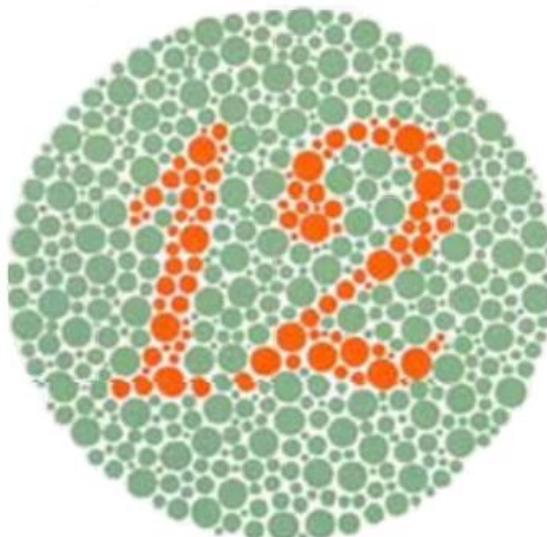
User Profile- other characteristics (besides HIPS)

- Experience and knowledge - education and reading level
 - experience with the system and task
 - mother language
 - computer literacy ...
- Work and task - usage frequency
 - training
 - usage type (mandatory, optional)
 - usage of other systems ...
- Physical Characteristics – color vision deficiencies
 - physical deficiencies
 - handedness
 - age ...

Don't forget cultural aspects!!...

Color blindness

Ishihara test for red-green blindness (daltonism or deutanopia)



(Shinobu Ishihara, 1917)

- The test includes 38 figures

User Profile - overview

- Human Information Processing System
- Experience and knowledge - education and reading level
 - experience with the system and task
 - mother language
 - computer literacy ...
- Work and task - usage frequency
 - training
 - usage type (mandatory, optional)
 - usage of other systems ...
- Physical Characteristics – color vision and physical disabilities
 - handedness
 - age ...

Don't forget Cultural aspects!!...

How do these characteristics influence UI design?

- Much system experience, but low task experience -> more semantic help
- Much task experience but low system experience -> more syntactic help
- High usage frequency -> easy to use
- Low usage frequency -> easy to learn and remember
- Mandatory -> easy to use
- Optional -> easy to learn and remember
- Color (particularly red and green) should not be used as only cue to convey information
 - Etc., ...

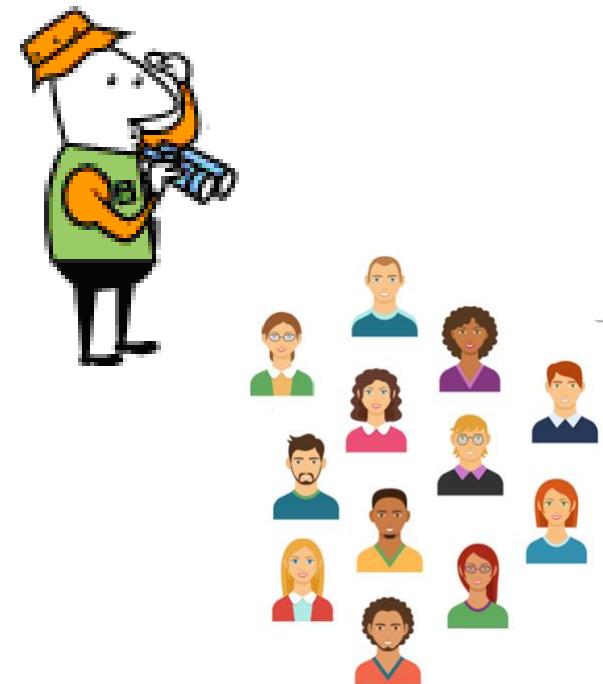
The take away:

- Users are much different from designers/developers
- Users vary a lot among themselves
- Users change along time (evolve, forget...)

Final recommendations:

We should consider the users as an unknown species and study them scientifically

Remember: You are not the user!



Main bibliography

- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, *Human-Computer Interaction*, 3rd edition, Prentice Hall, 2004, chap. 1
- Jenny Preece and Helen Sharp, *Interaction Design – Beyond Human-Computer Interaction*, 5th edition, John Wiley, 2019
(http://www.id-book.com/chapter3_teaching.php)
- John Carroll, Human Computer Interaction - brief intro, *The Encyclopedia of Human-Computer Interaction*, 2nd edition <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/affective-computing>

Paper Presentation Assessment

Students attending the presentation may vote via a link available in Moodle

- Using the following scale:

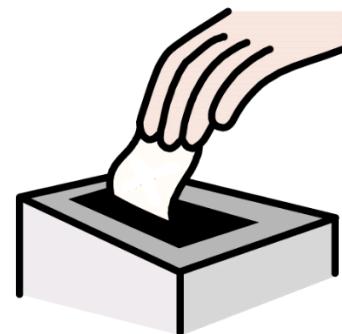
5- Excellent
4- Very good
3- Good
2- Sufficient
1- Poor



[Vote on a paper presentation](#)

- Taking into consideration:

- Adequacy
- Slides
- Presentation
- Discussion
- Exceeded time?



Paper presentation evaluation and voting

- Consider the following aspects + presentation duration:
 - Selection of the paper:
 - is the topic understandable by this audience?
 - Slides:
 - are well organized?
 - include the needed information?
 - have the right amount of text and figures?
 - Oral presentation:
 - was fluid and enthusiastic/motivating?
 - was balanced between the two presenters?
 - Personal conclusions and discussion:
 - the personal conclusions were adequate
 - the presenters provided good answers

Paper presentation evaluation and voting

- Voting scale on overall performance:

5- Excellent

4- Very good

3- Good

2- Fair

1- Poor

Paper Presentation

Vote on a paper presentation

1 POOR	2 FAIR
3 GOOD	4 VERY GOOD
5 EXCELLENT	

- Presenters, please submit the slides via Moodle:



Submission of paper presentation slides (the day of your presentation)

Please submit the presentation slides just before your presentation using the following file name format:

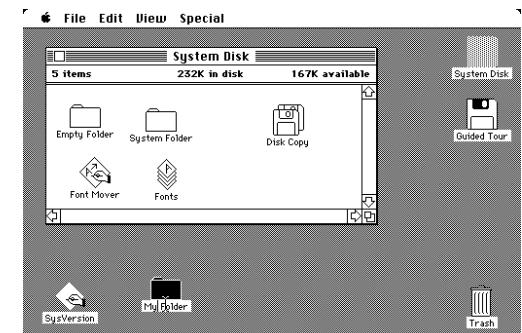
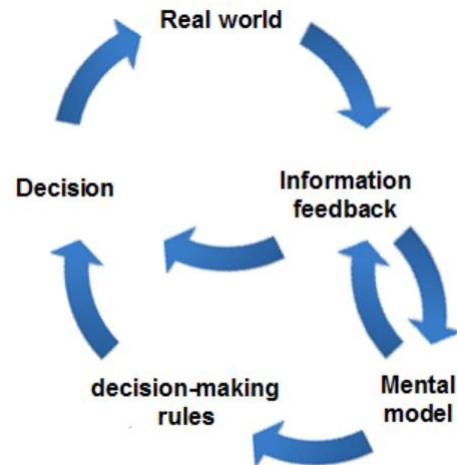
TPX_nmec1+nmec2_presentation date;

e.g.:

TP1_120000+120001_March_1.pdf



Mental and conceptual models



Relevant issues:

- What are mental models?
- How do we construct them?
- What is known about them?
- What are they used for?
- What are conceptual models?
- Guidelines to obtain good conceptual models (promoting good mental models)

[What are Mental Models? | IxDF \(interaction-design.org\)](#)

Mental models and Conceptual models

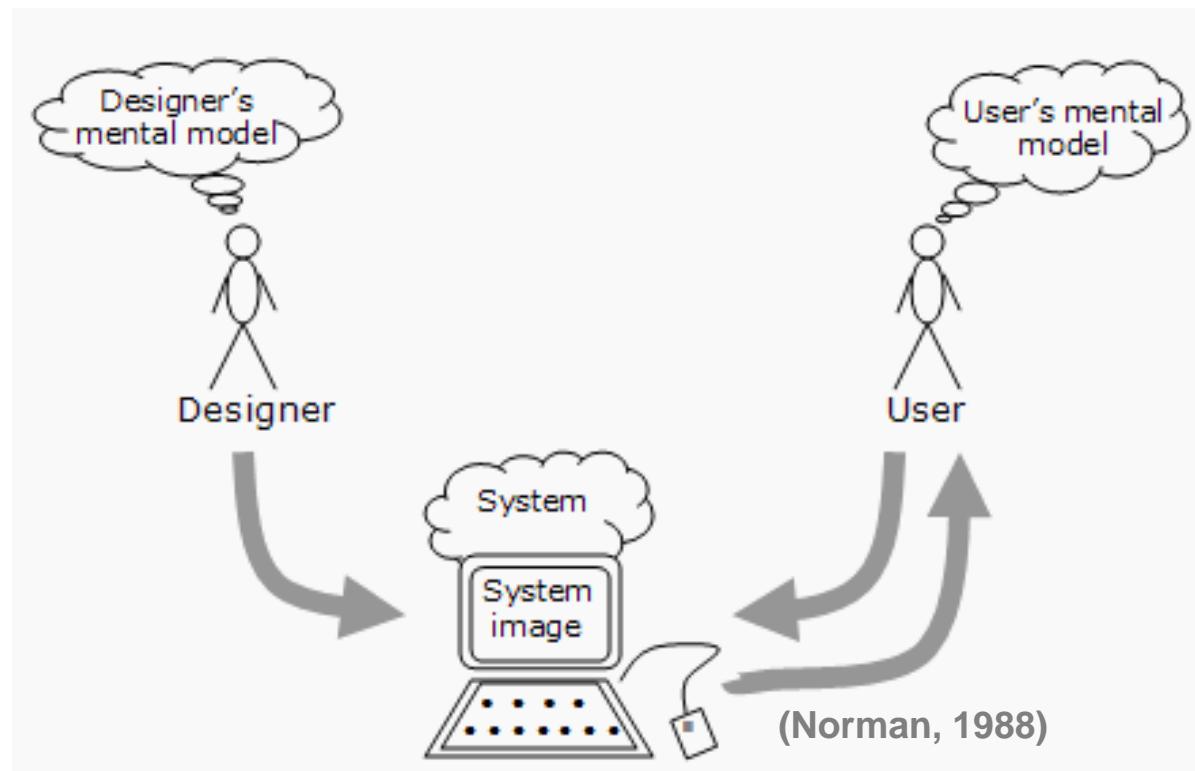
- The **conceptual model** is the UI highest level
- The conceptual model is the conceptual framework in which the functionality is provided to the user
- To understand how to design a good conceptual model it is necessary to understand mental models
- A **mental model** (in a simple way) is the **user's internal representation** of the current conceptualization and understanding of the system
- A **conceptual model** is the designer's attempt to foster good mental models through UI aspects

Mental models

Mental models are representations of the world that help us understand complex concepts and make better decisions

- based on past experiences, beliefs, and assumptions to understand how the world works
- essential for decision-making, problem-solving, and learning, as well as effective communication and collaboration in group settings

- The user develops a **mental model** of how he/she thinks the system works
- And uses it to:
 - reason about the system
 - anticipate system behavior
 - explain why the system reacts as it does



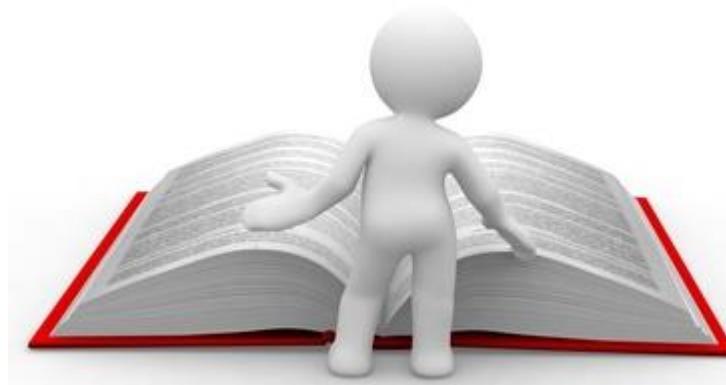
<https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/mental-models>

How do we create a mental model?

- Using the system
- Observing others using the system
- Reading documentation
- ...



and thus all these are important ways to train the user to use a system



Mental models allow:

- Make predictions
- Determine causes of observed events
- Determine adequate actions to produce the wanted changes
- Understand analogous devices
- ...

“What users believe they know about a UI strongly impacts how they use it. Mismatched mental models are common, especially with designs that try something new.”

<https://www.nngroup.com/articles/mental-models/>

Which button shall I press?

Example: remote controller of some projectors
@ DETI

- This device has a different UI from the others I am used to (not complying with the consistency and standards heuristic)
- I will try to infer how to use it based on the mental models I have
- But it is ambiguous and it does not give prompt feedback (not complying with the visibility of the system status heuristic)
- Determining adequate actions to produce the wanted changes fails!



I press one button, nothing happens, I press the other ...

... **low efficiency, low efficacy, low satisfaction**

poor usability and UX



Mental models:

- Are **incomplete**
- Are **unstable** (people easily forget the details)
- Are **not scientific** (maintain "superstitious" behaviours)
- **Don't have specific limits** (mistake similar devices and operations)
- ...

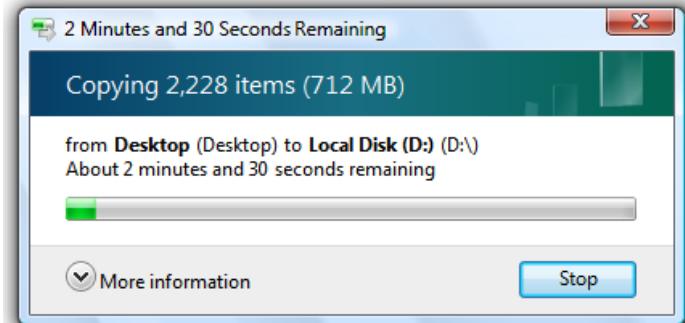
Often people do a lot of extra actions instead of planning,
which would avoid those actions!

“We must give up finding elegant mental models, and instead learn how to understand the incomplete and confuse structures people have”

Donald Norman

Main guidelines to obtain a good conceptual model (that fosters a good mental model)

- Make visible invisible parts and processes



- Give *feedback*



- Use coherence (colors names, command syntax, dialog styles, information location on the screen, etc., etc.).

Desktop metaphor →

- Use a metaphor (optional)



All this may help the user to understand better how the system works

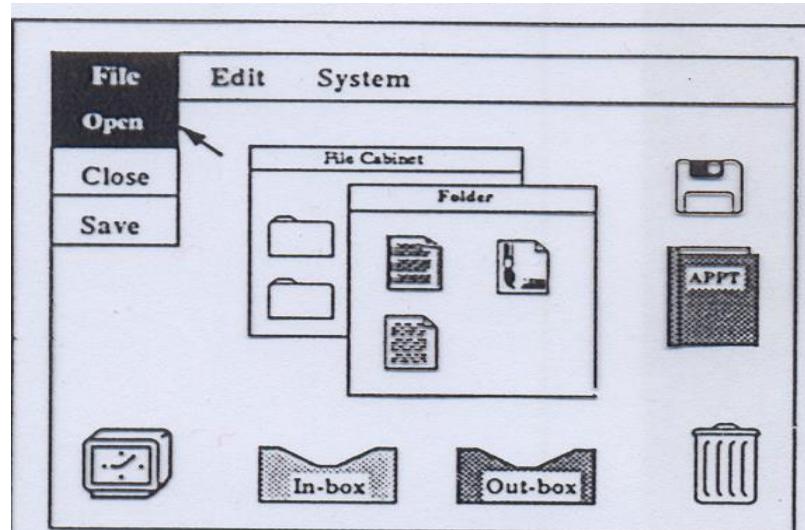
Metaphors

- Exploit existing mental models of the real world

Metaphors **can be misleading** since the “the essence of metaphor is understanding and experiencing one kind of thing in terms of another” (Lakoff and Johnson, 1983)

- Which, by definition, makes a metaphor different from what it represents or points to

Example: The Desktop metaphor:



<https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro>

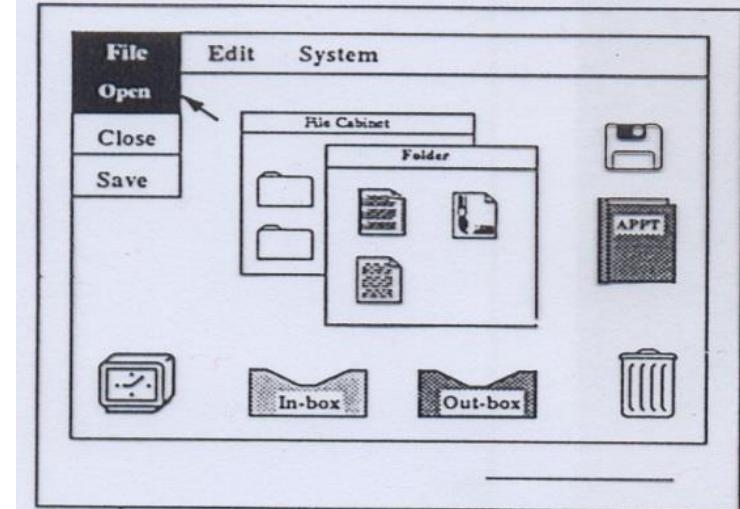
Another example: the bulletin board (Trello)

The image shows a side-by-side comparison of a digital Trello board and its physical analog. On the left is a screenshot of a Trello board titled "The Great Kitchen Redesign". The board is organized into four main columns: Ideas, To Do, Doing, and Done. The "Ideas" column contains cards like "Get a new window valence to match the cabinet colors" with a photo of a pot rack. The "To Do" column has cards such as "Adjust water pressure of the sink" and "Remove old refrigerator and stove". The "Doing" column includes "Pick countertop colors" and "Buy new kitchen cart". The "Done" column lists "Install new sink", "Install new flooring", "Design new kitchen space", and "Buy paint for cabinets". A sidebar on the right shows activity logs for the board. On the right is a photograph of a real corkboard with several yellow sticky notes pinned to it, representing the physical version of the same task management system.

<https://trello.com/tour>

Potential problems in using metaphors

- Incomplete metaphors may **confound** the user
- Risk of **under-utilization** of the system's capacities
- Less experienced users (e.g. children) seem to expect more “**literal**” metaphors
- Sophisticated users seem to expect more “**magical**” metaphors



Thus, the use of a metaphor should be carefully pondered ...

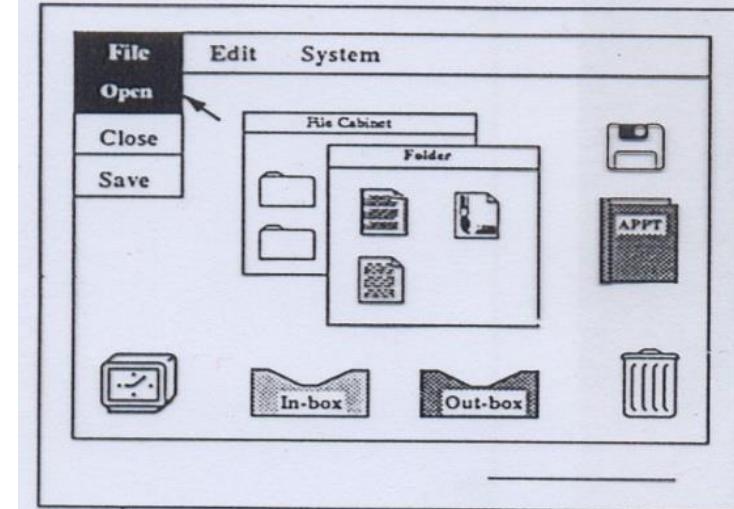


Examples of using metaphors

- In Apple's original desktop metaphor:

remove the diskette from the system ->

-> drag it to the recycling bin !! (unlike the real world...)



- Navigation/locomotion in VR systems:

Magical metaphor -> “teleportation”

Less magical (more literal) -> “physically” walking



Conceptual models - summary

- A conceptual model is a **high-level description** of a product in terms of:
 - what users can do with it
 - the concepts they need to understand how to interact with it
- Developing a conceptual model involves:
 - Understanding the problem space
 - Specifying how the proposed design will support users
- Conceptual models **must foster good mental models**
- Paradigms, visions, theories, models, and frameworks
 - Provide ways of framing design and research

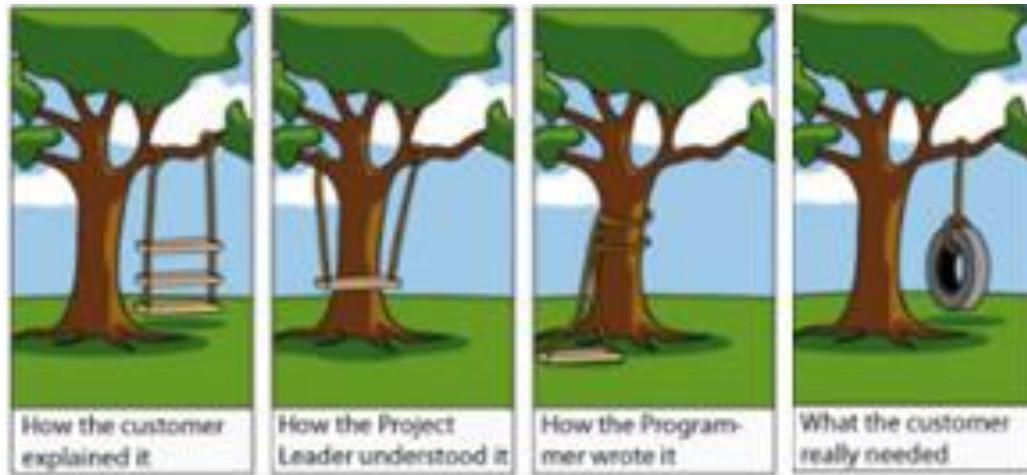
Main bibliography

- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, *Human-Computer Interaction*, 3rd edition, Prentice Hall, 2004
- Jenny Preece and Helen Sharp, *Interaction Design – Beyond Human-Computer Interaction*, 5th edition, John Wiley, 2019
(http://www.id-book.com/chapter3_teaching.php)
- John Carroll, Human Computer Interaction - brief intro, *The Encyclopedia of Human-Computer Interaction*, 2nd edition <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/affective-computing>
- Mental Models, *The Encyclopedia of Human-Computer Interaction*, 2nd edition, <https://www.interaction-design.org/literature/topics/mental-models>
<https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/mental-models>



Design of Interactive Systems

Human-Centered Design of Interactive Systems and Usability Engineering Lifecycle



The problem of interactive systems design...



How the customer
explained it



How the Project
Leader understood it



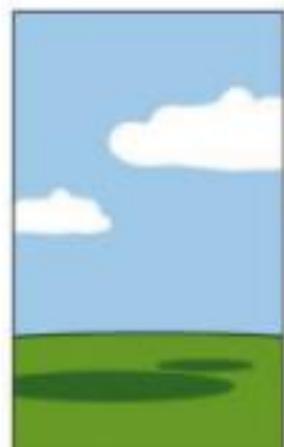
How the Analyst
designed it



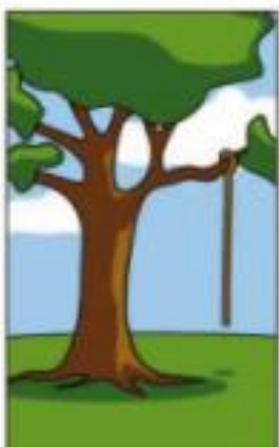
How the Program-
mer wrote it



How the Business Con-
sultant described it



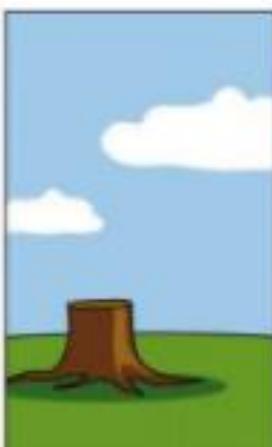
How the project was
documented



What operations
installed



How the customer
was billed



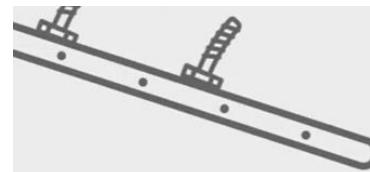
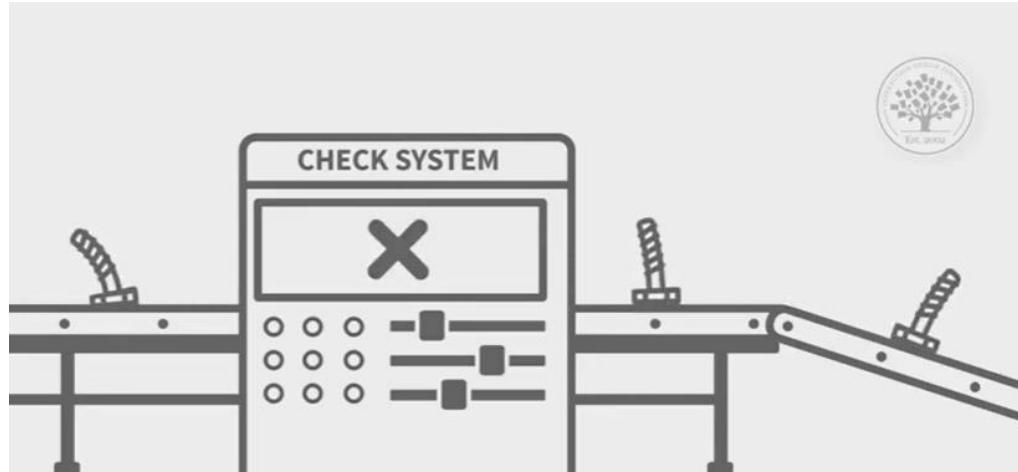
How it was
supported



What the customer
really needed

Proper requirements management is fundamental ...

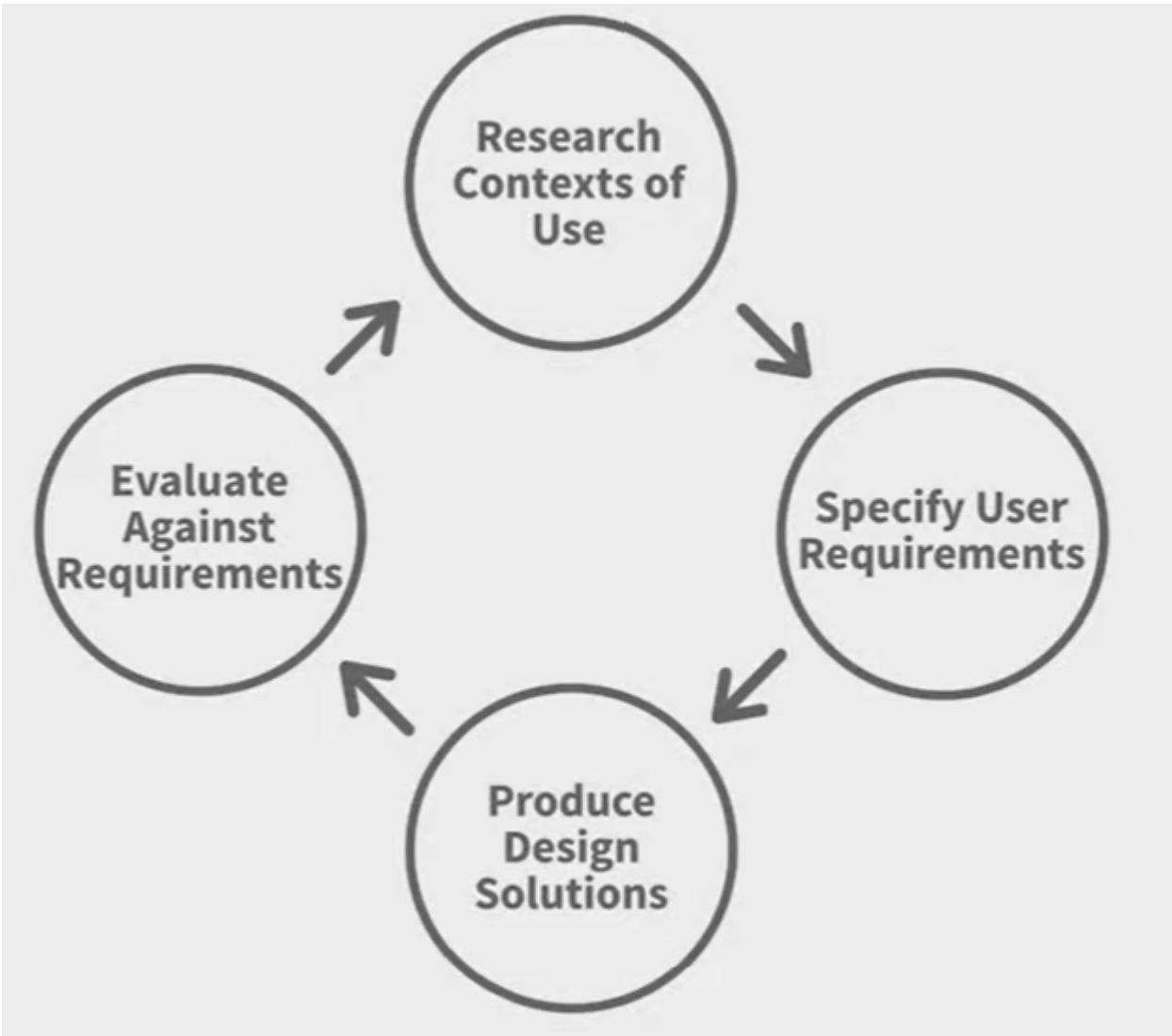
**Evaluation is very important tool but cannot be the only one ...
We need good design methods...**



<https://www.youtube.com/watch?v=SC7KL-TGe4s>



What is Human-Centered Design of Interactive Systems?



https://www.youtube.com/watch?v=KkUor_NTuDA

Human-Centred Design (HCD)

- **Approach to systems design and development** that aims to make interactive systems more usable by focusing on the use of the system and applying human factors/ergonomics and usability knowledge and techniques
- “**human-centred design**” is preferable to “user-centred design”; emphasizing it also impacts on other stakeholders (not just users)
- In practice, are often used synonymously

<https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en>

Benefits of Human-Centred Design

- Usable systems can provide a number of benefits:
 - improved productivity,
 - enhanced user well-being,
 - avoidance of stress,
 - increased accessibility,
 - reduced risk of harm.

Benefits of Human-centered design (cont.)

- Following the best practices, helps to identify challenges upfront so that a solution can be found early
- By putting a larger emphasis on principles and practices, iterative improvements can be made and avoid costly large scale rework
- The “10%” rules:
 - 10% of IT staff should be user experience (UX) professionals
 - 10% of budget dedicated to UX.



<http://www.usability.gov/what-and-why/benefits-of-ucd.html>

Human-Centred Design involves

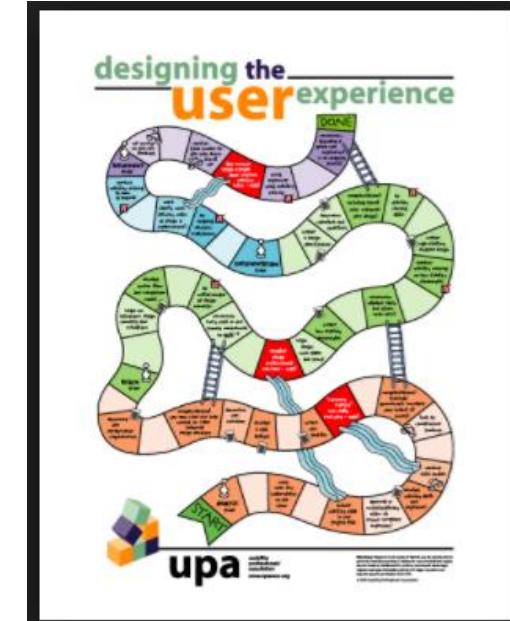
- **Understanding users' needs, motivations, contexts and activities**
- Understanding business, technical, and domain opportunities, requirements, and constraints
- Using this knowledge as a grounds to create products whose form, content, and behavior is **useful, usable, economically viable and technically feasible**

Human centered design processes for interactive systems:

"Human-centered design is an approach to interactive system development that focuses specifically on making systems usable. It is a multi-disciplinary activity."

- There are **several** proposals of HCD methodologies
- All are **iterative**
- And **include usability evaluation in iterations**

We must consider the situation at hand and ponder which are the best fitting and how to use them



<https://www.w3.org/WAI/redesign/ucd>

<http://www.usability.gov/how-to-and-tools/methods/user-research/index.html>

Human-centred design for interactive systems

- helps to identify and plan effective and timely human-centred design activities
- complements existing systems design approaches
- It can be incorporated in approaches as diverse as:
 - object-oriented,
 - waterfall,
 - rapid application development
- The principles of human-centred design and the related activities have not changed substantially since [ISO 13407](#) and have been **validated by many years of application.**

Example: study about why IT projects fail (and several causes are avoidable using adequate HCD methods)

Comparing with the situation in aviation an **IT accident** may be defined as:

“project with significant damage, e.g. large cost or schedule overrun, failing business goals, low usability, etc.”

Five accident investigations of large government IT projects in Denmark

Identified 37 different causes, each causing damage to one or more projects, e.g.

- surprises with system integration
- wrong estimate of human performance

Only one cause related to programming

Proposes 22 cures that in combination could have prevented most of the damages
Half are familiar to developers, but were ignored in the project (e.g. usability test)

Some causes for IT projects failures

(and several causes are avoidable using adequate HCD/UCD methods)

Analysis

- Doesn't identify user needs and win-win
- Requirements don't cover customer needs
- Describes solution in detail. No freedom to supplier
- Makes heavy demands and believes it is for free
- Oversells technology, e.g. SOA, web-based
- Multi-vendor strategy - supplier independent ! ?
- Wants everything at once, e.g. cover all types of debt
- Doesn't plan the new work processes
- No feasible solution, e.g. data missing, performance dubious.
- Surprising rule complexity.

Design

- Doesn't ensure usability, even when they know how
- Designs user screens too late
- Accepts the solution description without understanding it
- Cannot see how far the supplier is

Test

- Deploys the system with insufficient testing

Deployment

- Deploys system with insufficient support/training
- The system is not used as intended
- Wrong estimate of human performance



Study as-is and plan to-be

Problem oriented requirements (SL-07)



Plan to-be, SL-07 reqs (sections B1 and C)



Early prototypes & usability tests

Early prototypes & usability tests

Early prototypes & usability tests

Early prototypes , Monitor remaining work hours



Pilot test, Ask expert developers



Usability tests, Pilot test, Deploy part-by-part

Observe at pilot test, Follow-up study

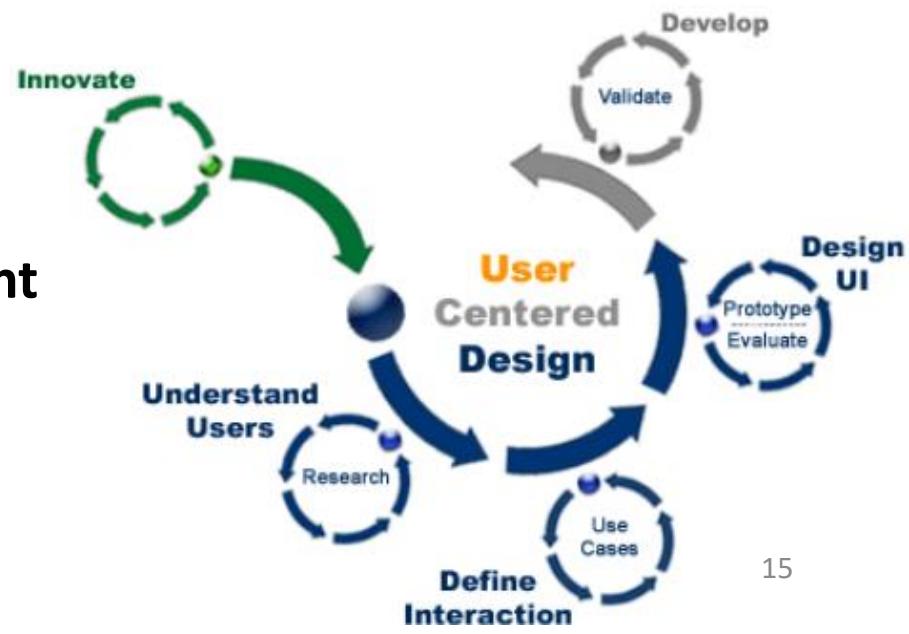
Check at POC, Check at pilot test

Iteration

- A key characteristic of HCD enables continual refinement through rapid prototyping and testing through repetition:
 - Observe and study,
 - Decide what the problem might be
 - Determine which parts of the design work and which don't

iterate again...

- **Fosters getting the requirements right**

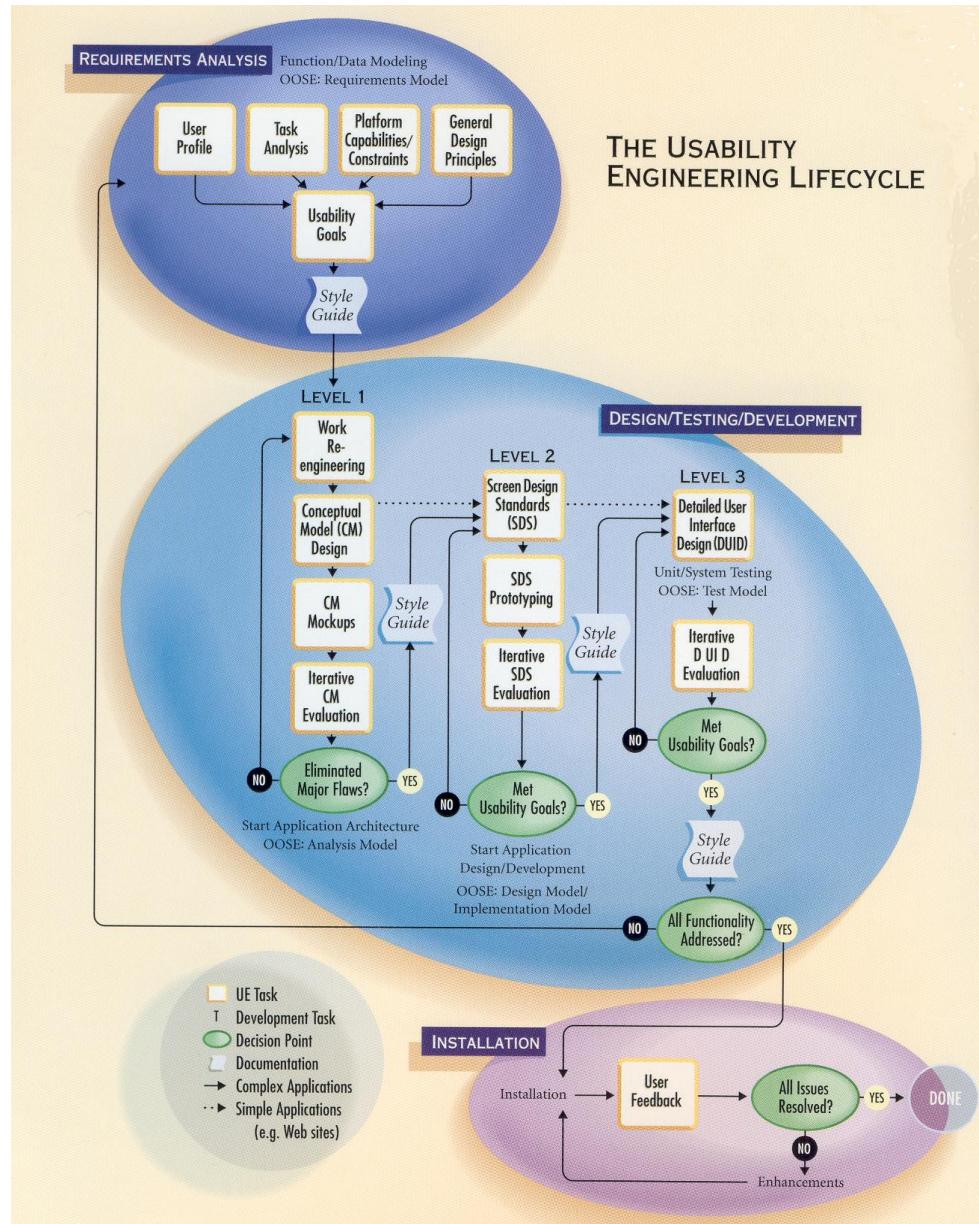


Examples of several approaches

- The Usability Engineering Lifecycle ([Mayhew, 1999](#))
- Goal-directed design ([Cooper, 2007](#))
- Activity-oriented design ([Norman, 2013](#))
- Etc.

One of the first Human-Centered Design approaches:

The Usability Engineering Lifecycle (Mayhew, 1999)



Requirement Analysis

User Profiles

Establish user characteristics important for UI design

Contextual Task Analysis

Obtain a user-centered model of work as it is currently done; extract the product usability requirements

Usability Goal Setting

Establish specific quantitative and qualitative usability goals to drive UI design

Platform Capabilities and Constraints

Establish capabilities and constraints of the technology platform which limit UI design alternatives

General Design Principles

Identify principles and guidelines that may be relevant for the product under development

Design, Testing, Development – Level 2

Screen Design Standards

Establish a set of design standards to set the stage for detailed UI Design

Screen Design Standards Prototyping

Support the evaluation, refinement and validation of the Screen Design Standards

Iterative Screen Design Standards Evaluation

Evaluate, refine, and validate the Screen Design Standards

Style Guide Development

Document the Conceptual Model Design, the Screen Design Standards and the output of Requirement Analysis

Design, Testing, Development – Level 3

Design

User Interface Design

Design the complete, detailed product User Interface

User Interface Evaluation

Evaluate, refine, and validate key subsets of the detailed User Interface Design

Installation

User Feedback

Obtain usability data after a product has been installed and used

Inform the UI Design for later releases or related products

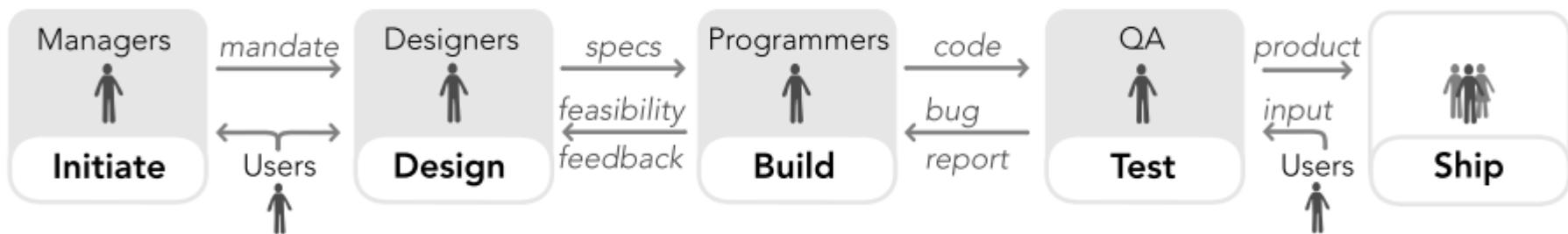
Evolution of the software development process

Early days of the S/W industry



...

Goal directed S/W development



(Cooper, 2007)

Goal-directed design (Cooper, 2007)

- **Works on any platform, in any subject area, and with any users**
- **Is based on understanding the users and their desired end-state**
- Not on any particular technology
- It involves **personas and workflows**

Activity-based design (Norman, 2013)

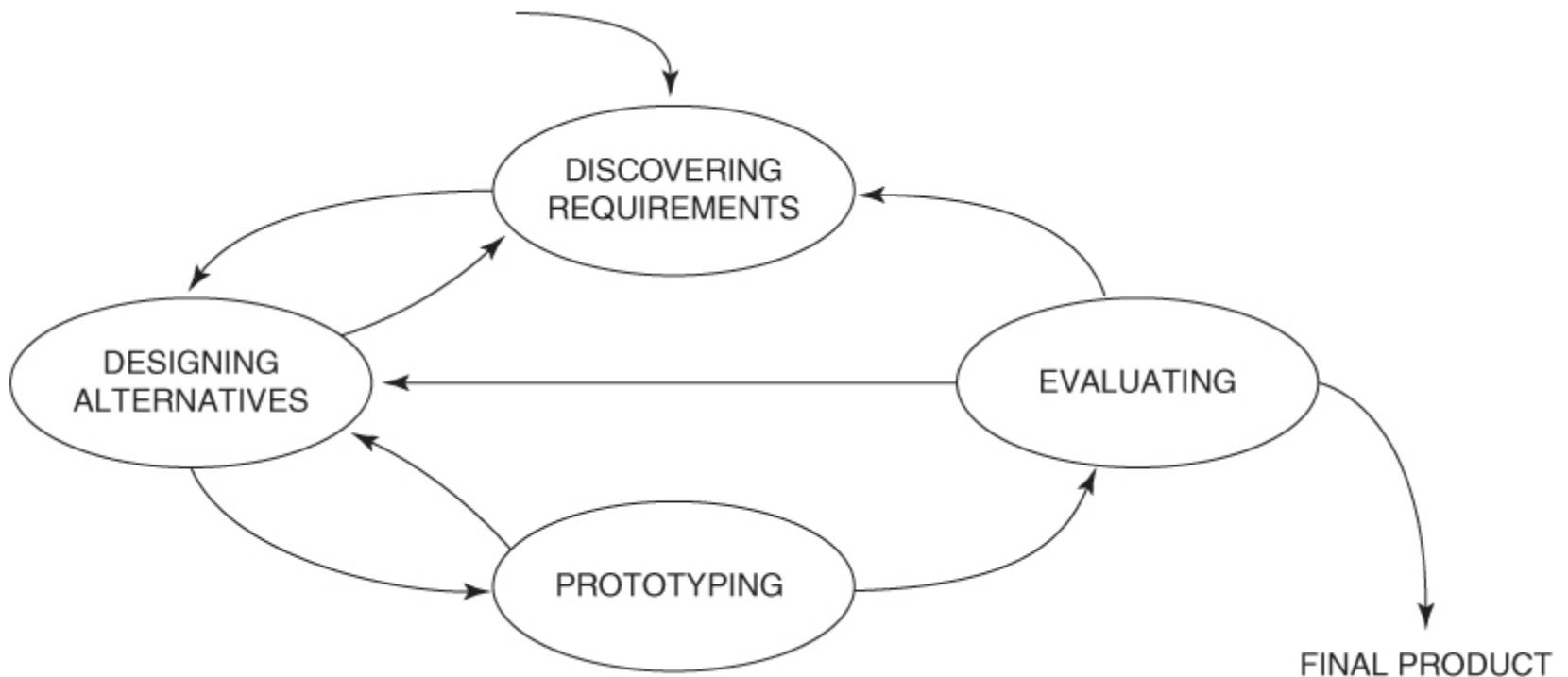
- Let the activity define the product
- Let the conceptual model be built around the activities
- Activity is a set of tasks performed together toward a goal

It is different from task

Is a high-level structure (e.g. “go shopping”)

- Tasks are lower level components (e.g. walk to the shop, ask for a product...)

Simple Interaction design lifecycle model



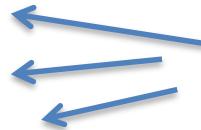
(Sharp et al, 2019)

Start by doing user research to discover requirements...

User Research Methods

There are **a lot of methods**; we must consider the situation at hand to select and adapt the ones we use:

- Context interviews
- Focus groups
- Individual interviews
- On-line surveys
- **Personas**
- **Scenarios**
- **Task analysis**
- Activity analysis
- First click tests
- ...



**Methods to use in the mini-project
(2nd assignment)**

<http://www.usability.gov/how-to-and-tools/methods/user-research/index.html>

Requirement analysis for the mini-project



Lab 6 - User Centered Design (UCD). 2nd Assignment

- Introduction to requirement analysis: using personas, scenarios and task analysis
- Introduction to the technologies to be used



How to perform a requirements analysis

(available in Moodle Lab classes area)

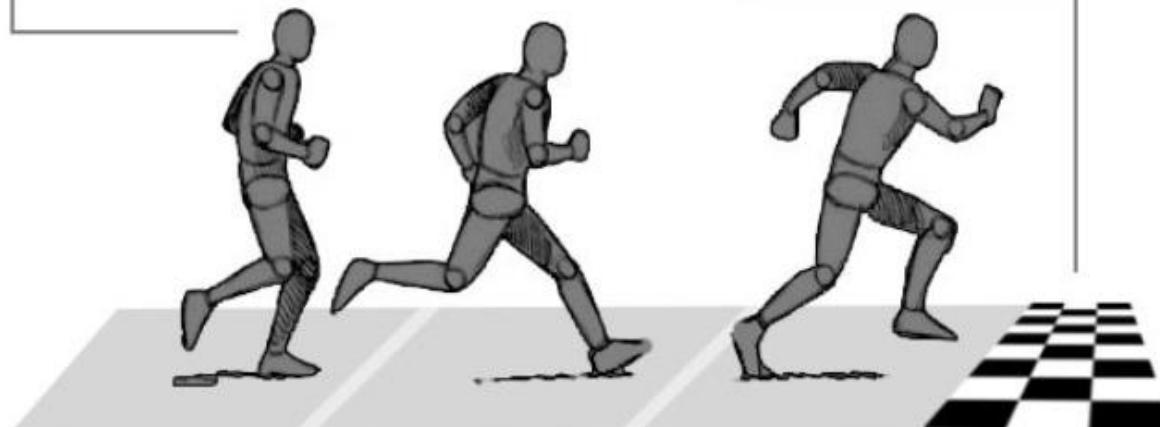
Personas and scenarios

1. Persona

Defines who the story is about. This main character has attitudes, motivations, goals, and pain points, etc.

3. Goal

Defines what the persona wants or needs to fulfill. The goal is the motivation of why the persona is taking action. When that goal is reached, the scenario ends.



2. Scenario

Defines when, where, and how the story of the persona takes place. The scenario is the narrative that describes how the persona behaves as a sequence of events.

<https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>

Personas

- **Fictional characters based upon user research** to represent the different user types that might use a service/product in a similar way
- make the design task less complex,
and guide the ideation process
- help:
 - understand users' needs, experiences, behaviors and goals
 - step out of oneself and recognize that different people have different needs and expectations
 - uncovering universal features and functionality
 - create a good user experience for your target users



History of Personas

Stem from IT system development during the late 1990s

How to best communicate an understanding of the users?

Various concepts emerged:

user archetypes, user models, lifestyle snapshots, ...

Alan Cooper (1999) proposed personas to describe fictitious users

There is **no single definition** of what a persona should contain

Nor a unified understanding of how to apply the method

Benefits of Personas

- Offer a quick and inexpensive way to test and prioritize features throughout the development process
- Help
 - Focus decisions by adding a layer of real-world consideration
 - Stakeholders evaluate new feature ideas
 - Information architects develop informed wireframes, and interface behaviors
 - Designers create the overall look and feel
 - System engineers/developers decide which approaches to take based on user behaviors

Types of Personas

- Several types (most based on previous user research):
 - **Goal-directed** Personas ([Cooper, 2007](#))
 - **Role-based** Personas (goals + behavior)
 - **Engaging** Personas (goals + behavior + backgrounds)
 - **Fictional** Personas (based on assumptions, not user research)
- **Fictional personas can only be used as an initial sketch of user needs**

<https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>

Best Practices for Developing Personas

- Create 2-4 personas of the product/service main audiences
- Conduct user research:
 - Who are the users
 - Why are they using the system?
 - What behaviors, assumptions, and expectations?
- Develop the appropriate descriptions of each persona's: background, motivations, and expectations
- Do not include too personal information
- Be relevant and serious

Elements of a persona

- Persona Group (i.e. web manager)
- Fictional name
- Job titles and major responsibilities
- Demographics such as age, and education
- The goals and tasks they are trying to complete using the product
- Their physical, social, and technological environment

**Personas have no value in themselves, until they become part of a scenario
they do not have real value!**

Example of a Persona

Persona:

USDA Senior Manager Gatekeeper



Photo:

Fictional name:

Matthew Johnson

Job title/
major
responsibilities:

Program Staff Director, USDA

Demographics:

- 51 years old
- Has a Ph.D. in Agricultural Economics.

Goals and
tasks:

Spends his work time:

- Requesting and reviewing research reports,
- preparing memos and briefs for agency heads, and
- supervising staff efforts in food safety and inspection.

Environment:

He is comfortable using a computer and refers to himself as an intermediate Internet user. He is connected via a T1 connection at work and dial-up at home. He uses email extensively and uses the web about 1.5 hours during his work day.

Example of using personas in VR

Model the people who will be using the VR application

Help to prevent the design from being driven by design/ engineering convenience

Personas should

- **not be too detailed**

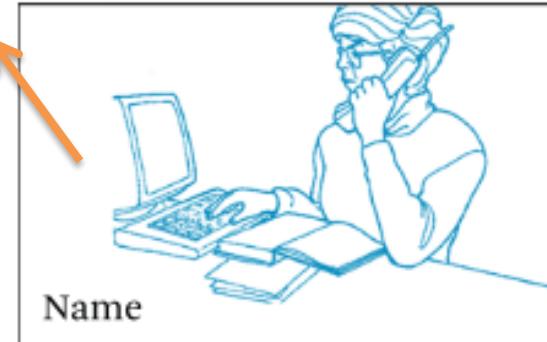
- **be validated** in later stages

(Jerald, 2016)

 Name	<ul style="list-style-type: none">• Job• Experience• Activities• Attitude• Competencies• Age
<ul style="list-style-type: none">• Problems• Pain points• Needs• Concerns• Fears• Desires	<ul style="list-style-type: none">• Knowledge of VR• Dream VR system• Vision of VR• VR hardware access• Budget for VR• Activities that fit VR

Describe 2–4 characters representing the range of targeted users

Sketch/photo and name



Name

Basic description of the person

- Job
- Experience
- Activities
- Attitude
- Competencies
- Age

- Problems
- Pain points
- Needs
- Concerns
- Fears
- Desires

- Knowledge of VR
- Dream VR system
- Vision of VR
- VR hardware access
- Budget for VR
- Activities that fit VR

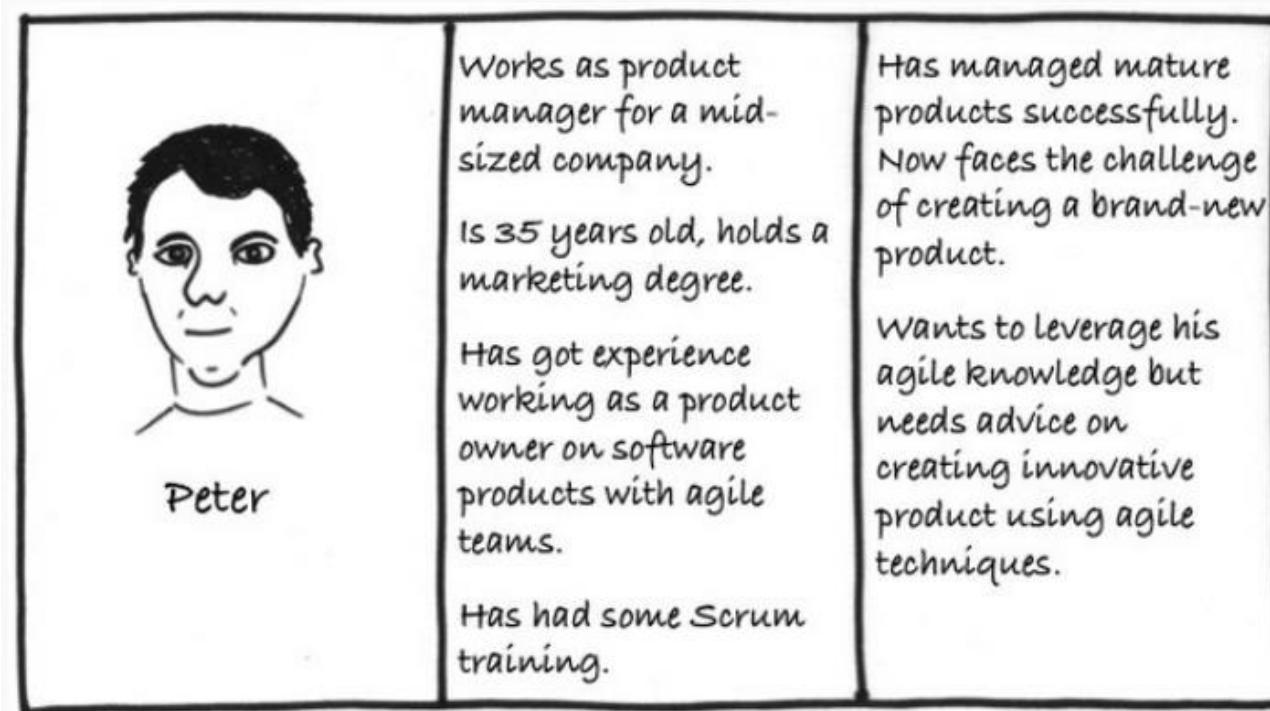
Challenges the person has



Relation to type of system (VR in this case)

If personas are especially important (e.g., for therapy applications), then data should be very carefully collected with interviews and/or questionnaires

Another example of a Persona

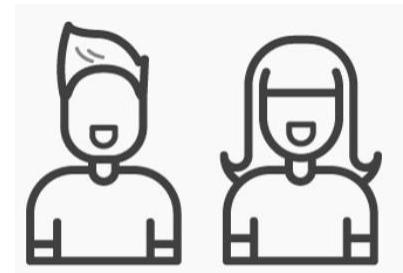


- A main difficulty of the persona method is getting the team members to use it
- The 10-step process of creating a persona can help

<https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>

10 steps to Creating Personas and Scenarios (and getting them used by the team)

- 1- Collect data - and knowledge about the users
- 2- Form a hypothesis - general idea of the various users
- 3- Everyone accepts the hypothesis
- 4- Establish a number– the final number of personas
- 5- Describe the personas - to be able to develop solutions
- 6- Prepare scenarios for the personas to describe solutions
- 7- Obtain acceptance from the organization and participants
- 8- Disseminate knowledge for the participants to use them
- 9- Everyone prepares scenarios- Personas have no value in themselves
- 10- Make ongoing adjustments- revise the descriptions often



Personas: The Take Away

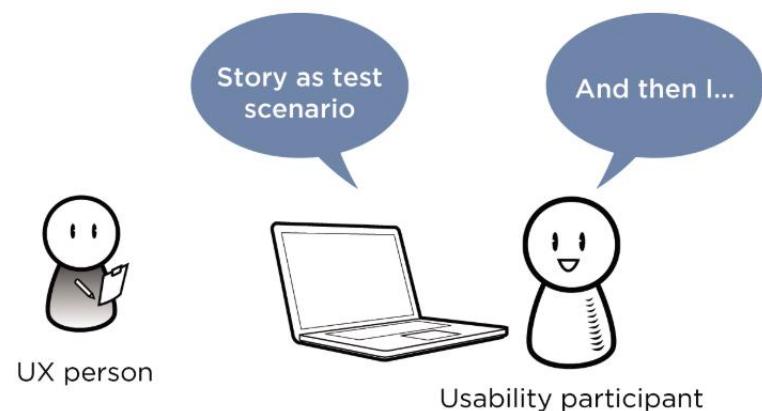
- Personas are user models, **fictional characters based on user research** to help understand:
 - users' needs, and experiences
 - behaviors and goals
- Make the design task at hand less complex
- Guide the ideation processes, and help to achieve the goal of creating a good user experience for the target user group
- The 10-step process covers the entire process from the preliminary data collection, through active use, to continued development of personas.



<https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>

Scenarios

- **Stories and contexts about how the user groups use** a future product/service
- Note the goals and questions to be achieved and sometimes define the possibilities of how the user(s) can achieve them on the product/service
- Scenarios are critical for
 - **designing**
 - **usability testing**



<https://www.usability.gov/how-to-and-tools/methods/scenarios.html>

<https://www.interaction-design.org/literature/topics/user-scenarios>

- Scenarios should be used in the ideation phase of a project
- Need to be ***based on research*** with users
- Do ***not*** represent ***all*** possible users
- Typically account for the ***most common*** users or user motivations
- Are commonly ***based on personas***
- Can be used to determine the most important areas to test during usability testing, and to provide guidance to the test

<https://www.interaction-design.org/literature/topics/user-scenarios>

What to Consider When Writing Scenarios

- Good scenarios are concise but answer the following questions:
 - **Who is the user?** Use the **personas**
 - **Why does the user uses the product?** Note what motivates the user and their expectations, if any
 - **What goals does s/he have?** Use **task analysis**
 - **How can the user achieve their goals with the product?**

<https://www.usability.gov/how-to-and-tools/methods/scenarios.html>

Types of Scenarios

- **Goal/Task-based Scenarios** state only what the user wants to do

Example: You are traveling to Paris for your job next week and you want to check on the amount you can be reimbursed for meals and other expenses

- **Elaborated Scenarios** give more user story details
- **Full Scale Task Scenarios** include the steps to accomplish the task

<https://www.usability.gov/how-to-and-tools/methods/scenarios.html>

Scenarios: The Take Away

- User scenarios are a **great way of communicating the key tasks a user will perform with a system**
- They can also **help define the usability testing regime**
- To create user scenarios is a simple process and **should be used for developing and iterating interactive products**

<https://www.interaction-design.org/literature/topics/user-scenarios>



Task Analysis

- The process of **learning about users** by observing them in action to **understand in detail how they perform their tasks** and achieve their intended goals.
- Helps identify the tasks that product/service must support
- Helps support other aspects of the user-centered design process
- It is important to perform a task **analysis early in your process**, in particular prior to design work

<https://www.usability.gov/how-to-and-tools/methods/task-analysis.html>

Types and benefits of Task Analysis

- Several types
 - Hierarchical task analysis (studied in more detail later)
 - ...
 - Cognitive task analysis
- Help support several aspects of the user-centered design process, including:
 - Requirements gathering
 - Developing structure
 - Prototyping
 - Usability testing

<https://www.usability.gov/how-to-and-tools/methods/task-analysis.html>

- **Task analysis is useful for** understanding:
 - **Users' goals** and what they are trying to achieve
 - The **steps that users currently take** to achieve their goals
 - The personal, social and cultural experiences that **users bring to the tasks**
 - The **influence of the physical environment** on the users while attempting to meet a goal

<https://www.interaction-design.org/literature/article/task-analysis-a-ux-designer-s-best-friend>

- **Task analysis may be performed:**
 - in a **more formal way** (e.g. HTA - Hierarchical Task Analysis)
 - or
 - in a **more informal way**:
 - using several different methods
- First **use the 11 questions (at least 1, 2, 3 and 5)**
- Then **decompose the main tasks**

Standard/Informal Questions to be answered

1. Who is going to use the system?
2. What tasks do they now perform?
3. What new tasks are desired?
4. How are the tasks learned?
5. Where are the tasks performed?
6. What is the relationship between customer and data?
7. What other tools does the user have?
8. How do users communicate with each other?
9. How often are the tasks performed?
10. What are the time constraints on the task?
11. What happens when things go wrong?

Minimum set of questions to be answered

1. Who is going to use the system?

- Use **all the information obtained previously about the users** (e.g. to develop the personas), concerning:
age, needs, motivations, background, experience, technology literacy, physical characteristics...

↙

2. What tasks do they now perform?

- Identify the **tasks that users perform currently**, without using the system under development, including:
relative importance, frequency of performing the tasks, if they are performed by one or more users, ...

3. What new tasks are desired?

- Identify new tasks that might empower the users taking advantage of the new way of performing the tasks
- Be careful and prioritize the new tasks to support ...

↙

5. Where are the tasks performed?

- Observe the environment where users currently perform the tasks
- Identify other activities, the type of space (office, shop floor, hospital, class room, shopping mall, ...), noise, light and dust conditions, stress level, ...

How to analyze the tasks

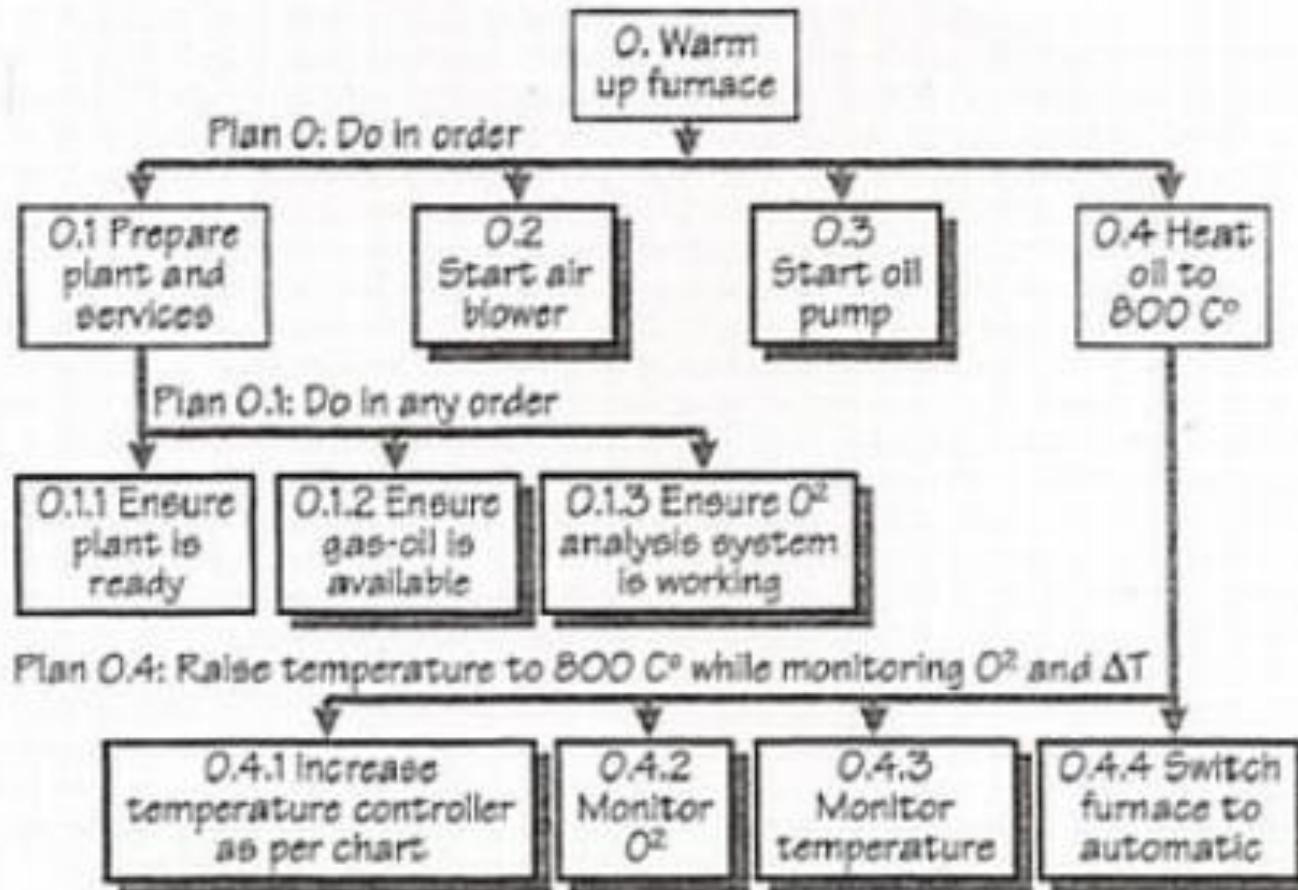
- Decompose a high-level task into the following steps:
 - Identify the task to be analyzed
 - Break this high-level task down into 4 to 8 subtasks
 - Draw a layered task diagram of each subtasks
 - Produce a written account as well
 - Present the analysis to someone else who knows the tasks
- The decomposition level of detail should be coherent across subtasks

<http://www.usabilitybok.org/task-analysis>

Example of a Hierarchical Task Analysis

(it will be addressed in more detail)

Diagram for the goal: warm up a furnace



Task Analysis: The Take Away

- Is **one of the most powerful tools in UX design**
- It is not hard to get to know how to do it
- The difficult part is remembering to keep the user's perspective
- It is **useless when it is not backed by rigorous user research**
- is not a one-off process; **can be repeated** later in the process
- **It requires time, resources, people and budget. Be sure to have a sufficient amount of all**
- Like any other activity in UX design!

<https://www.interaction-design.org/literature/article/task-analysis-a-ux-designer-s-best-friend>

Other methods...

- The previous methods should be used in the 2nd assignment (mini-project)
- But there are other methods that can be used in a requirements analysis (and that you may already know...):
 - User stories
 - Use cases
 - Story boards
 - Etc.

User Stories

Emerged from agile development methods, are **short concepts or descriptions of features customers would like to see**

Should:

not go into too much detail

be written:

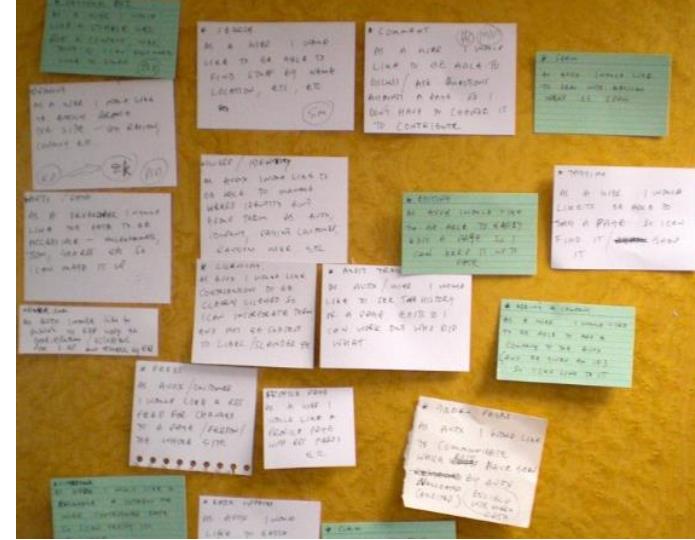
- from the **user's point of view**
- with the **client and team members**

[What are User Stories? | IxDF \(interaction-design.org\)](https://interaction-design.org)



User Stories

Short statements *about* a feature written from a user's perspective



"As a <type of user> I want <some goal> so that <some reason>."

Should be written with the **minimum amount of detail** necessary to fully **encapsulate the value that the feature** is meant to deliver

<https://manifesto.co.uk/how-much-detail-should-a-user-story-have/>

<https://www.interaction-design.org/literature/topics/user-stories>

<https://www.interaction-design.org/literature/article/user-stories-capturing-the-user-s-perspective-quickly-and-simply>

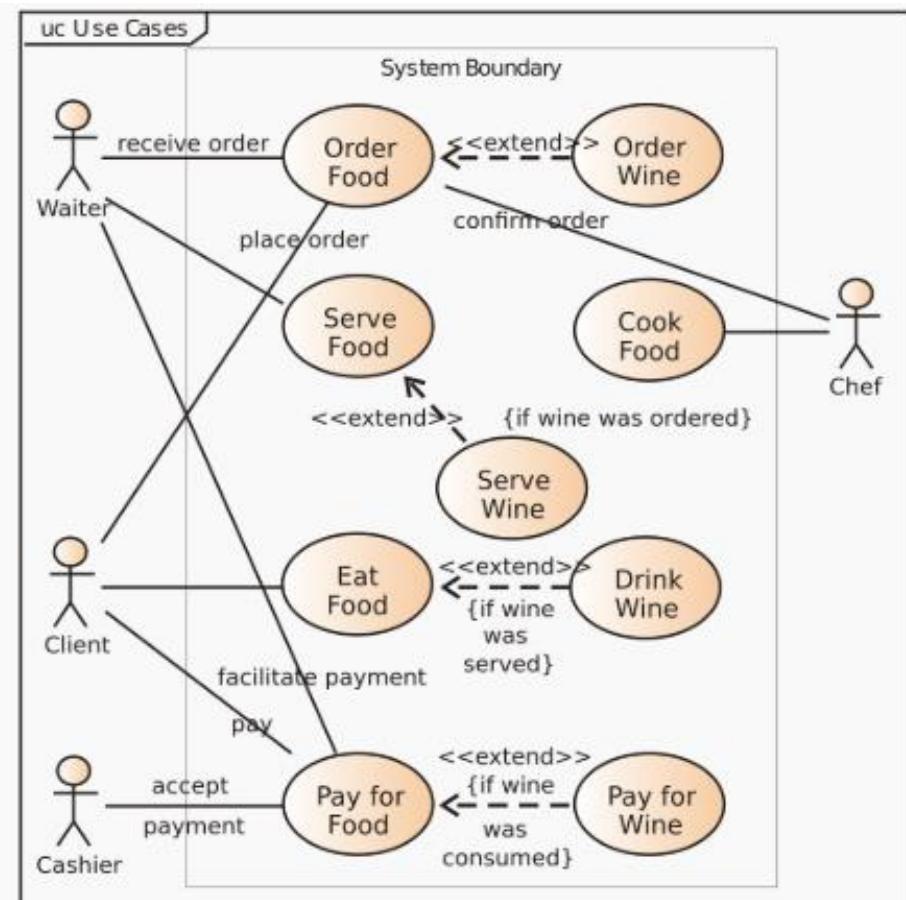
Use Cases

Written description of how users will perform tasks

Outlines a system's behavior responding to a request from a user's point of view

A common way for developers to explain user scenarios using UML

<https://www.usability.gov/how-to-and-tools/methods/use-cases.html>



Scenarios, User stories and Use cases

- Scenarios are created by user researchers to help communicate with the design team
- User stories are created by project/product managers to define the requirements prior to a sprint in agile development
- Use cases are created for developers to help with testing
- The difference in target audience means that the structure and information contained in the three approaches also varies.

Example

Scenario

“Jim, an internal medicine intern at Mount Pleasant Hospital, walks into the room of his patient, Andrew. Since Andrew stayed the night in the hospital, Jim needs to review Andrew’s medical records to see if the nurses on the night shift had checked in and recorded any changes in Andrew’s condition.”

User Story

As a doctor, I need to get up to medical date records so that I know how to proceed with my patients’ treatment

(it does not reflect the context of use)

Use case: Review Records

Actor: Doctor

Steps:

Doctor walks into room

Doctor sees patient in bed

Doctor identifies patient in bed

Doctor sees medical charts on foot of bed

Doctor gets medical charts from foot of bed

Doctor opens medical charts

Doctor reads medical charts

Doctor changes pages to continue reading

Doctor closes medical chart

<https://www.akendi.com/blog/scenarios-user-stories-and-use-casesoh-my/>

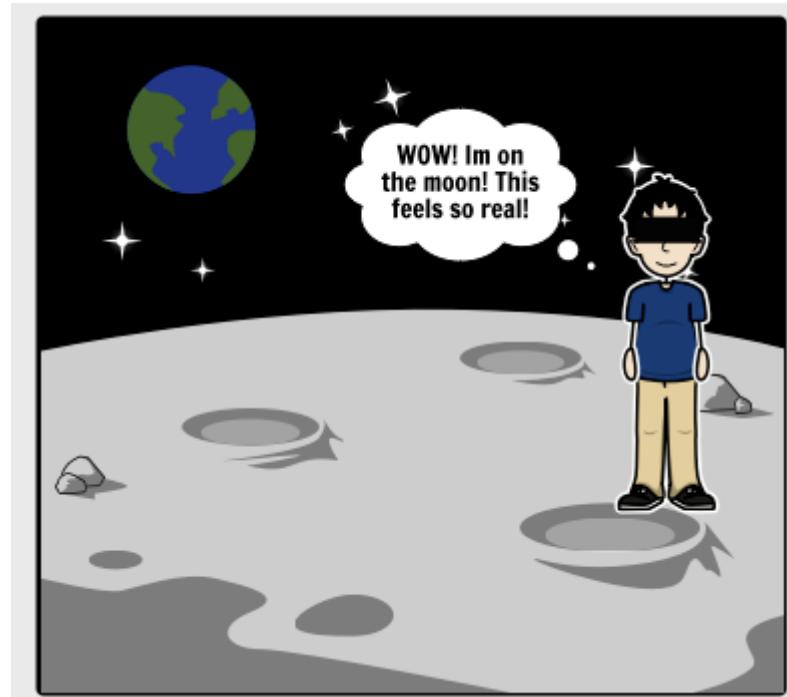
Storyboards

Are early visual forms of an experience

Derived from the film industry

Particularly useful for VR

The user can be shown directly interacting with objects



<https://www.storyboardthat.com>

<http://usabilitybok.org/storyboard>

Prototyping for the mini-project



Lab 8 - Test with Low Fidelity Prototype

Each group shall prepare a usability test with a paper prototype for the application under development (test should take +/- 15min). To prepare this class, use the notes about prototyping available. Main results of this evaluation shall be reported in the final presentation.



Notes on paper prototyping

(available in Moodle Lab classes area)

Prototyping

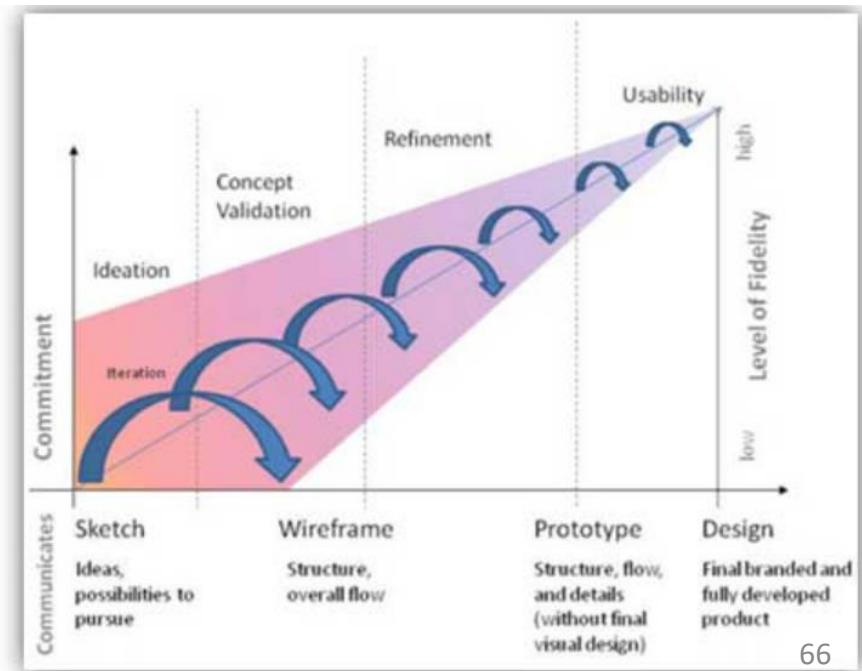
Prototype is a **draft version** of a product to:

- **explore ideas**
- show the intention behind a feature or the **overall design concept**

before investing time and money into development

low-fidelity → high-fidelity

<https://www.usability.gov/how-to-and-tools/methods/prototyping.html>



Low fidelity prototypes

- Fast and inexpensive
- Valuable to test the UI conceptual model
- Easy to modify even during user tests
- Elicit users feedback concerning general aspects
- Estimated to allow detecting up 80% of the usability issues



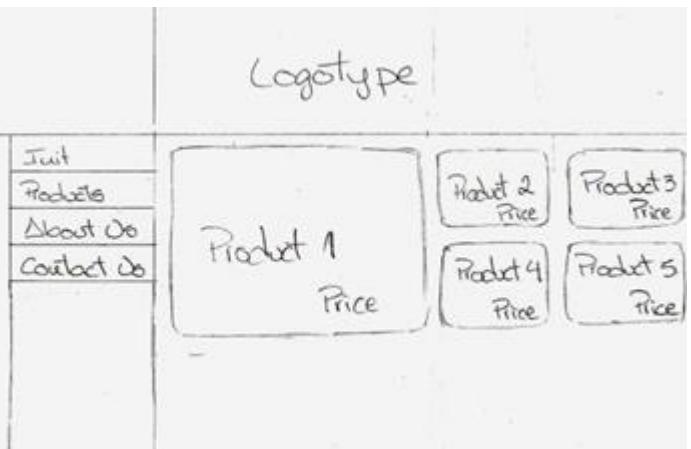
- Specifically adequate to get feedback concerning:

- Concepts and terminology

- Navigation

- Contents

- Functionality



E-mail application

http://

#	FROM	TOPIC	DATE
4	xxx@gmail.com	Topic4	01.01.2013 12:00
3	xxx@gmail.com	Topic3	01.01.2013 12:00
2	xxx@gmail.com	Topic2	01.01.2013 12:00
1	xxx@gmail.com	Topic1	01.01.2013 12:00

MAILBOX WRITE MAIL CONTACTS QUIT

NEXT PAGE

E-mail application

http://

DATE: 01.01.2013 12:00

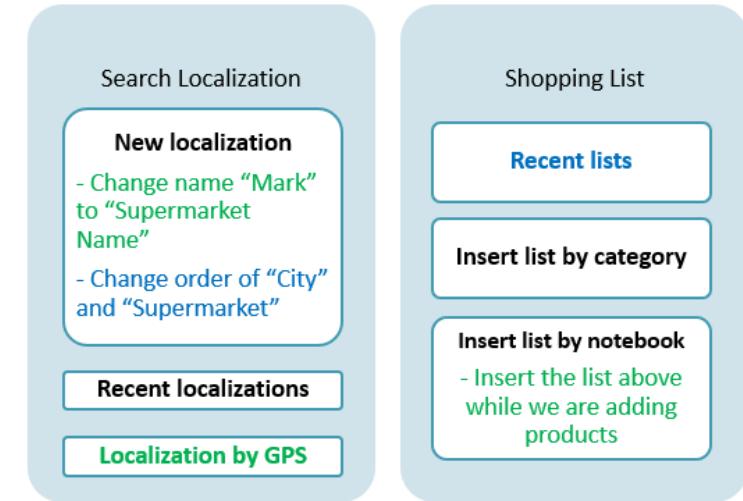
FROM: xxx@gmail.com

TOPIC: topic

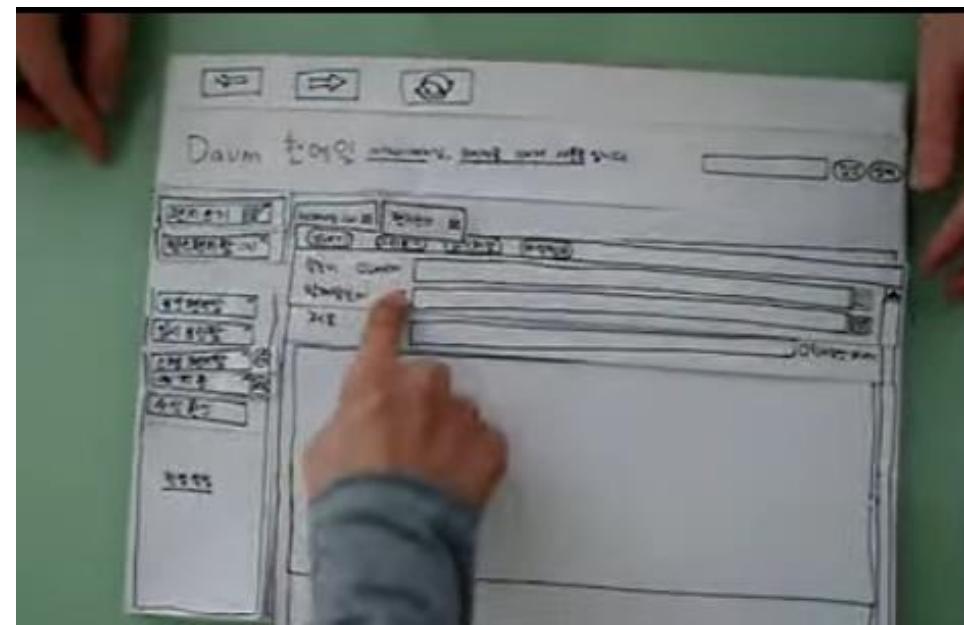
Content of the e-mail.

xxx

REPLY DELETE 68



- Do not need to have much detail, nor to be very realistic, e.g.:
 - Sizes of windows, fonts, etc. don't need to be final
 - Text may be replaced by some lines
 - Images may be replaced by words
 - In general no colour is needed



What can be used to make it?

- Paper, pens, markers, ...
- Applications such as:
 - Balsamiq
 - Figma
 - Pencil
 - Proto.io

Patient n° xxxxx

System Logo

Name

Address

Phone Number

Email

Foto Photo

Identify Card

Health System No

Birth Date

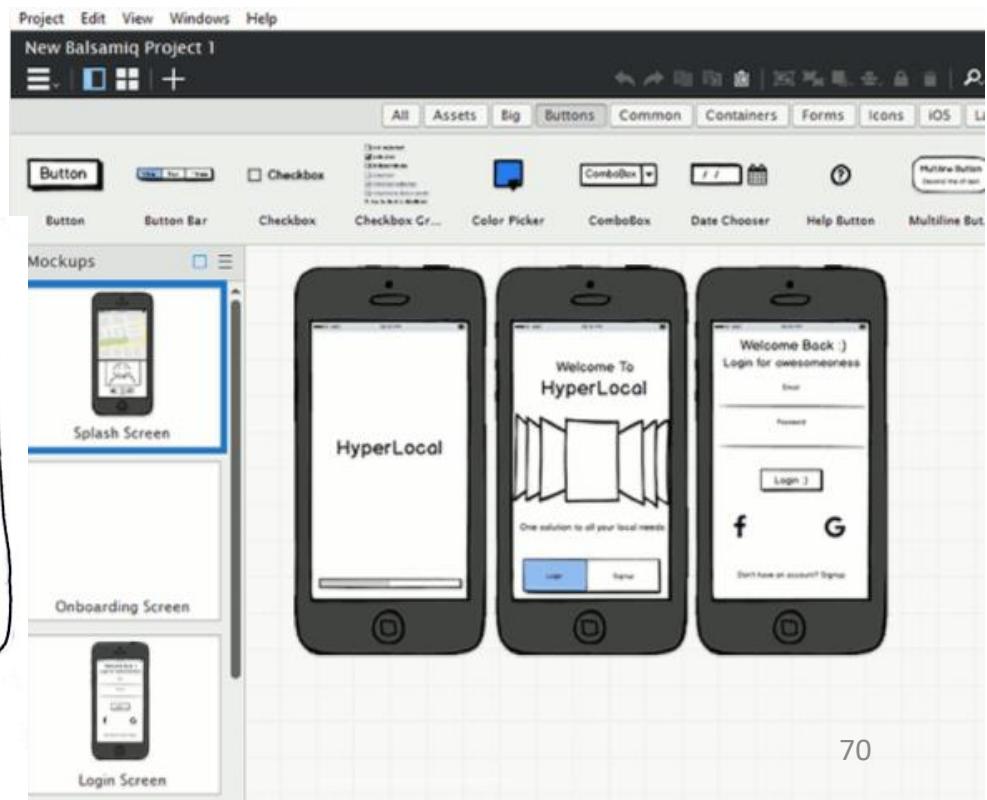
Gender

Height weight

Blood Group

Notes

Continue >



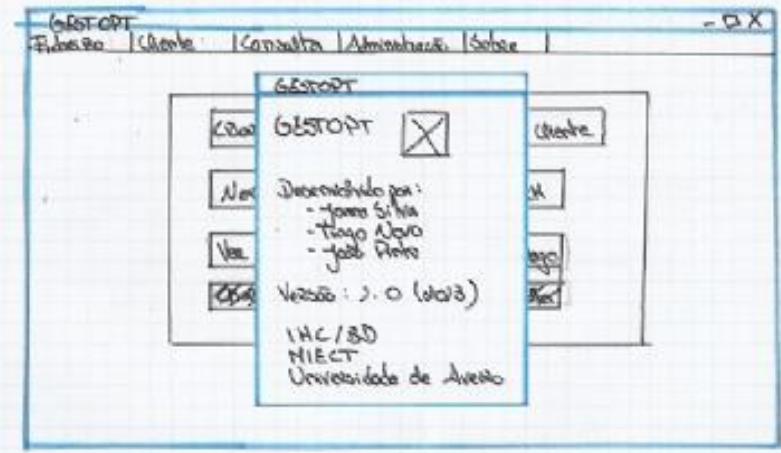
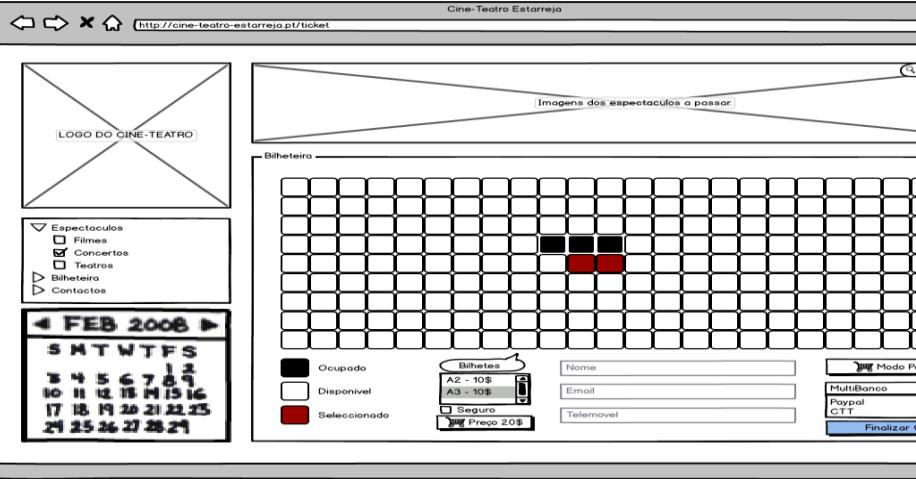
Performing a usability **test early** in the process can have **huge returns**
a paper prototype allows to do this with a **minimal time investment**



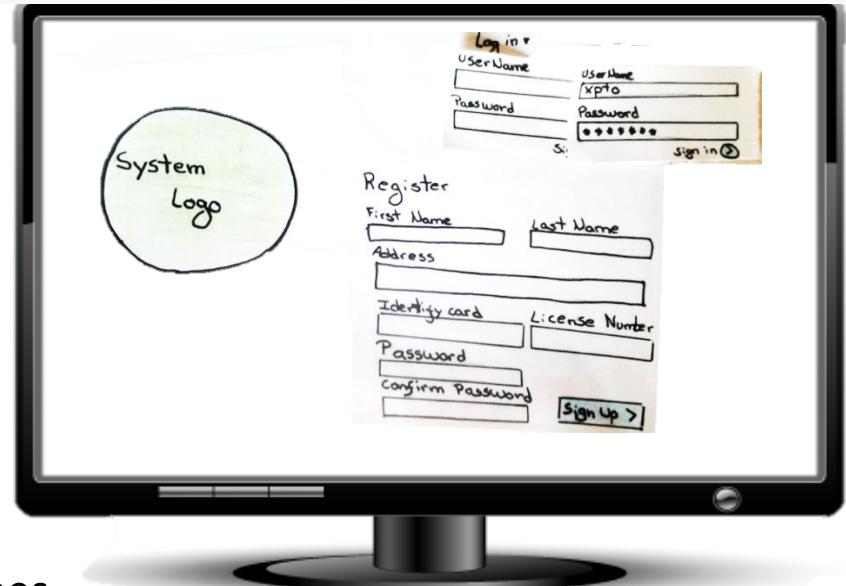
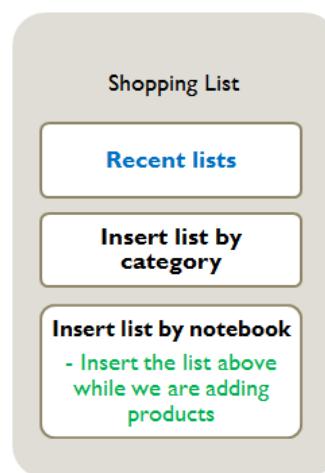
What is Paper Prototyping? – Interaction Design Foundation

<https://www.interaction-design.org/literature/topics/paper-prototyping>

Examples of lower or higher fidelity prototypes



→ O botão 'Crie Ficheiro' só está disponível se o Login for feito por um Administrador



Acknowledgment

To all students who have used low fidelity prototypes in previous editions of the Human-Computer Interaction course and colleagues who supervised them

Even for less conventional applications



Wizard of Oz prototyping

- A prototype that only works by having someone behind-the-scenes “pulling the levers and flipping the switches” (named after the classical film)
- A user interacts with an interface without knowing that the responses are given by someone



The “wizard” was a “man behind-the-scene”

<https://www.nngroup.com/articles/wizard-of-oz/>

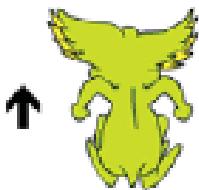
https://en.wikipedia.org/wiki/Wizard_of_Oz_experiment





Example of using the Wizard of Oz method in other situations

- Definition of a set of gestures to use in a game



Höysniemi, J., Hämäläinen, P., Turkki, L., and Rouvi, T. 2005. "Children's intuitive gestures in vision-based action games". *Commun. ACM* 48, 1, Jan. 2005, 44-50

Example of using the Wizard of Oz method in other situations



- Haptic Wizard of Oz Prototyping aids designers in rapidly designing and testing interactive hardware like this above car cockpit

D. Leithinger, C. Zheng, and E. Y. Do, “Haptic Wizard of Oz Prototyping in VR,” in *VR/MR Workshop*, 2018.

Wizard of Oz @ HCI-UA

Paulo Dias, T. Sousa, J. Parracho, I. Cardoso, A. Monteiro, Beatriz Sousa Santos
“Student Projects Involving Novel Interaction with Large Displays”, *IEEE Computer Graphics and Applications*, vol.34, no.2, Mar.-Apr. 2014, pp.80-86

Used to get insight
on which gestures
might be more
intuitive to control
a Pac-Man game



Main bibliography

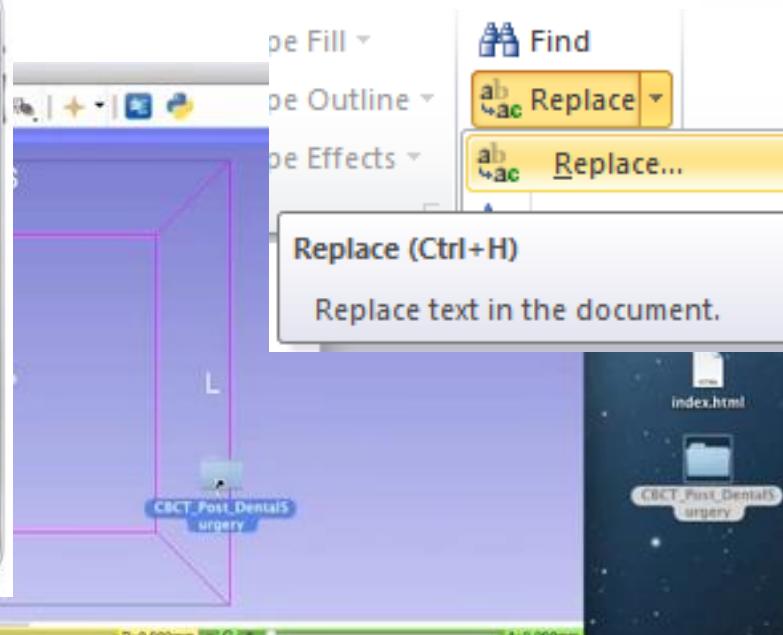
- Cooper, A., R. Reimann, and D. Cronin, *About Face 3 The Essentials of Interaction Design*. Wiley Publishing, Inc., 2007
- The Encyclopedia of Human-Computer Interaction, 2nd Ed., Interaction Design Foundation <https://www.interaction-design.org/literature>
- Mayhew, D., *The Usability Engineering Lifecycle*, Morgan Kaufmann, 1999
- [https://books.google.pt/books/about/The Design of Everyday Things.html?id=I1o4DgAAQBAJ&printsec=frontcover&source=kp read button&redir_esc=y#v=onepage&q&f=false](https://books.google.pt/books/about/The%20Design%20of%20Everyday%20Things.html?id=I1o4DgAAQBAJ&printsec=frontcover&source=kp_read_button&redir_esc=y#v=onepage&q&f=false)
- Sharp, H., Y. Rogers, *Interaction Design : Beyond Human-Computer Interaction*, 5th ed., Wiley, 2019
[https://books.google.pt/books/about/Interaction Design.html?id=HreODwAAQBAJ&redir_esc=y](https://books.google.pt/books/about/Interaction_Design.html?id=HreODwAAQBAJ&redir_esc=y)
- Sommerville,, I., *Software Engineering*, 9th ed., Addison Wesley, 2010
[http://ifs.host.cs.st-andrews.ac.uk/Books/SE9/WebChapters/PDF/Ch 29%20Interaction design.pdf](http://ifs.host.cs.st-andrews.ac.uk/Books/SE9/WebChapters/PDF/Ch_29%20Interaction_design.pdf)

Interesting Links

- <https://www.nngroup.com/>
- <http://www.usability.gov/>
- <https://uxpa.org/about-ux/>
- <https://www.usability.gov/how-to-and-tools/index.html>
- <https://www.w3.org/WAI/redesign/ucd>



Interaction styles



Interaction Styles

“The concept of Interaction Styles refers to all the ways the user can communicate or otherwise interact with the computer system.”

Soegaard, Mads. Interaction Styles (Retrieved March 2020)

http://www.interactiondesign.org/encyclopedia/interaction_styles.html

There are a lot of studies and design guidelines

Shneiderman's Eight Golden Rules of Dialogue Design

1. Strive for consistency
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialogues 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

These golden rules are paramount in the UI design process

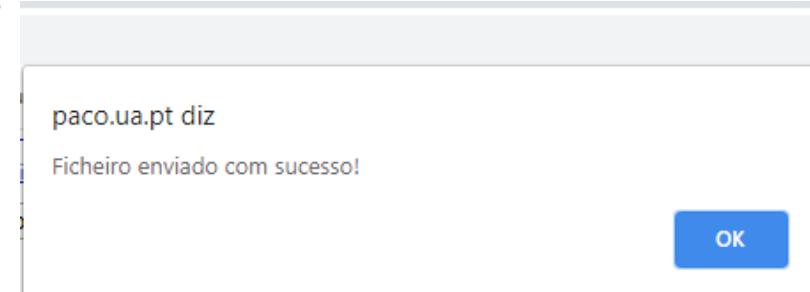
<https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces>

Support internal locus of control

- Allow users to be the initiators of actions
- Give users the sense that they are in control of events

Design dialogue to yield closure

- Don't keep users guessing.
- Tell them what their action has led them to

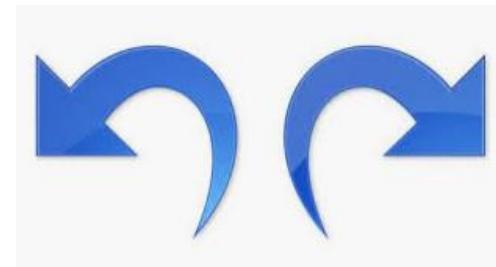


Allow easy reversal of actions



encouraging exploration of unfamiliar options

undo – CTRL z



Allow frequent users to use shortcuts

Common examples:

save – CTRL s

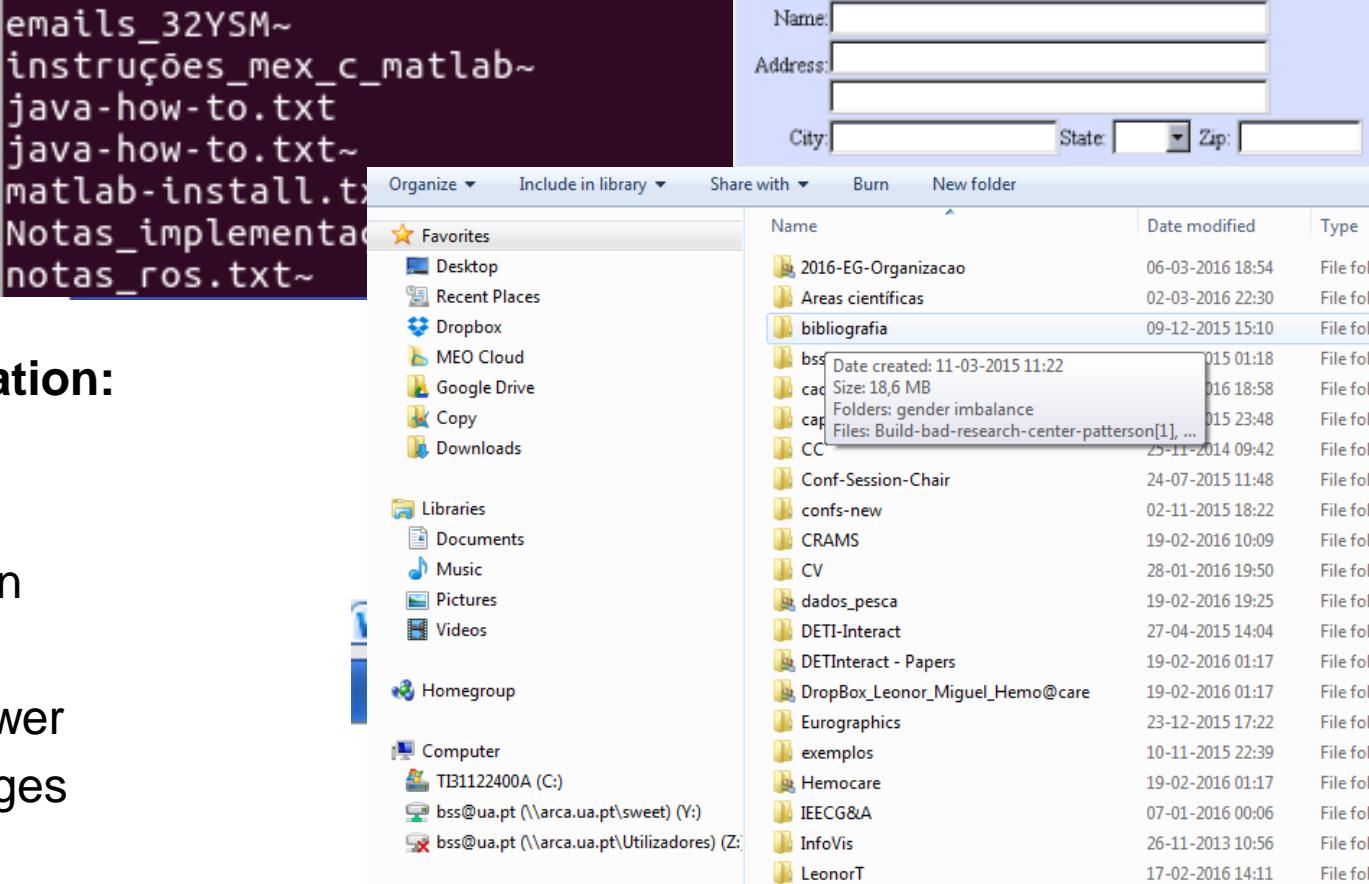
copy – CTRL c



Interaction/ Dialog styles

A possible classification:

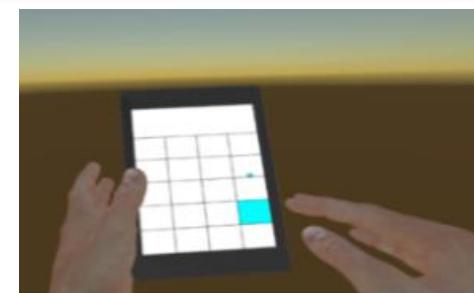
- Menus
- Fill-in-forms
- Direct manipulation
- Function keys
- Question and answer
- Command languages
- Natural languages



Often two or more styles are used simultaneously

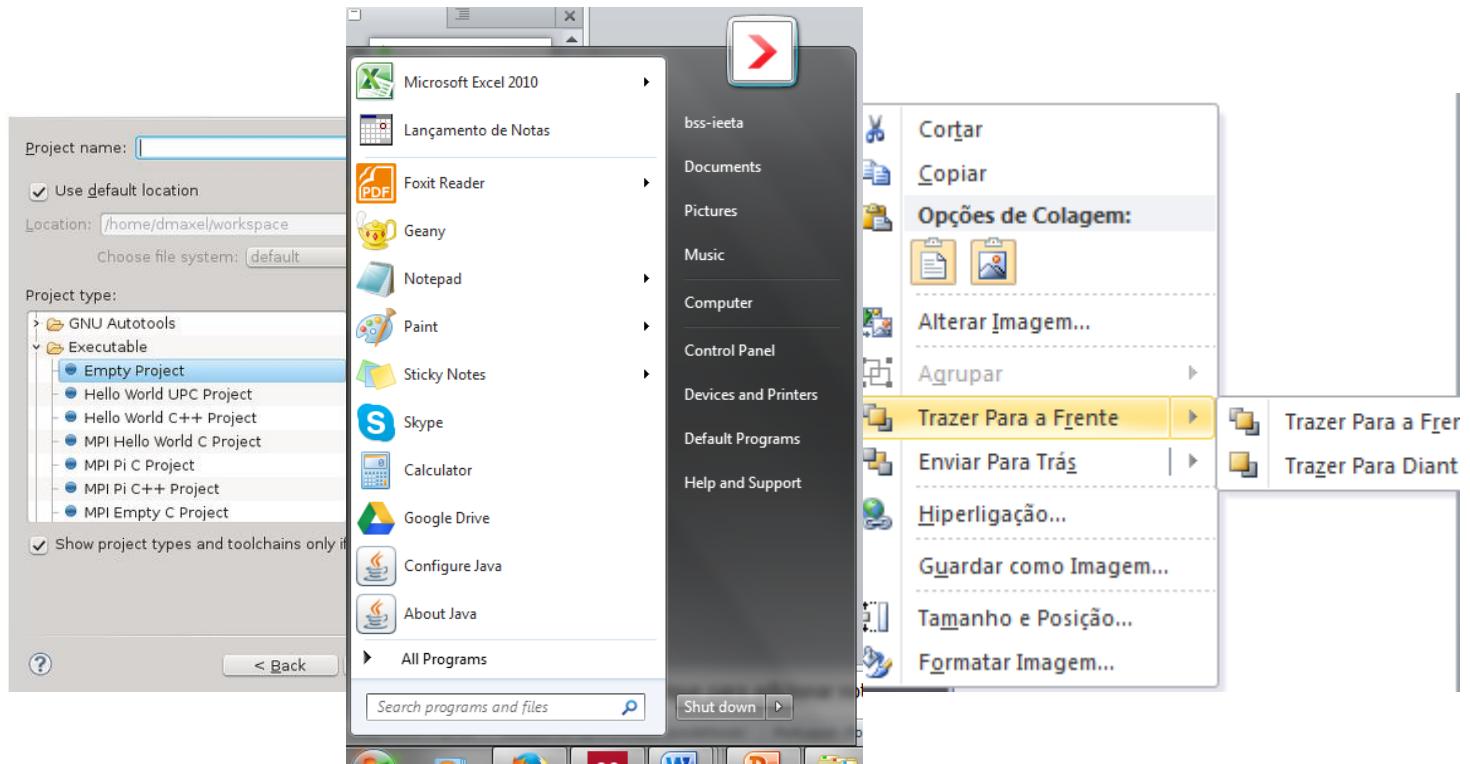
Less traditional user interfaces (UIs):

- 3D user interfaces
- Conversational user interfaces
- Tangible user interfaces
- etc....





Menus





LIVE SEAFOOD, SHELLFISHES EGGS & FISHES		
ENTREES		
Steak	\$10.50	
	\$9.75	
	\$13.95	
Cold	\$6.95	
cucumber sauce in pita bread		
Salad	\$13.95	
	\$7.25	
Hot		
Greek salad	\$14.25	
PASTA DISHES		
choice of Spaghetti, Linguine, Capellini or Ziti		
Oil, Garlic & Oil	\$8.95	
Oil or Sausage	\$8.95	
Sauce	\$7.95	
Feta Sauce or Fra Diavolo (spicy)	\$8.50	
above served with a tossed salad		
AMERICAN ENTREES		
New York Sirloin Steak*	\$11.95	
Center Cut Pork Chops*	\$10.95	
Broiled Filet of Sole	\$10.95	
Fried Sea Scallops	\$10.95	
Fried Clams	\$10.50	
Fried Shrimp	\$11.95	
above served with either a tossed salad or french fries		
GRINDERS		
(footlong)		
Combination (Ham & Swiss)	\$7.25	
Pastrami or Roast Beef	\$6.95	
Tuna or Turkey	\$6.95	
Veal Cutlets Parmigiana	\$7.95	
Above served hot or cold with lettuce & tomatoes		
Eggplant Parmigiana	\$6.95	
above served hot with lettuce sauce or mortadella cheese		
Extra on grinder: Cheese + \$1.00, Bacon or Ham + \$1.00,		
Mushrooms + \$1.00, Peppers or onions + 75¢		
CALZONES		
Plain (Stuffed w/Pepperoni & Mozzarella Cheese)	\$8.95	
Extra Toppings	\$2.00	
Items: Bacon, Broccoli, Eggplant, Garlic, Olives, Onion, Green Peppers, Marshmallows, Ham, Sausage, Salami, Oregano, Pepperoni, Mushrooms, Spinach, Tomatoes		

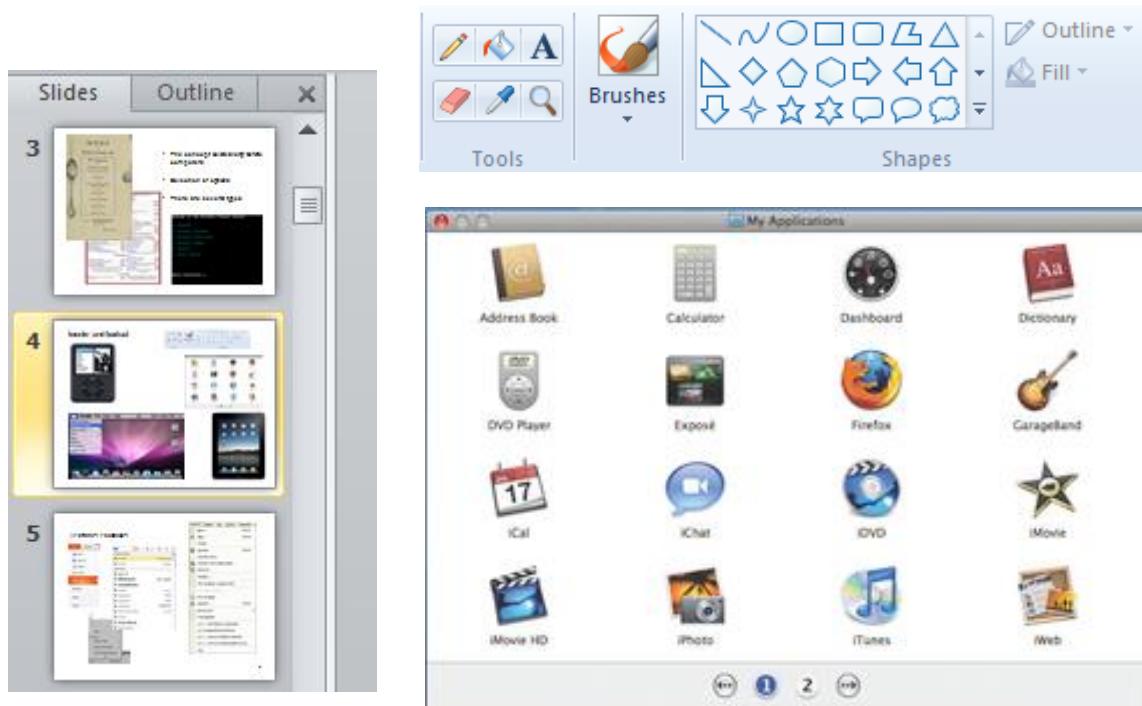
- The concept existed long before computers
- Selection of options
- There are **several types**

Welcome to the Viridian Finance System

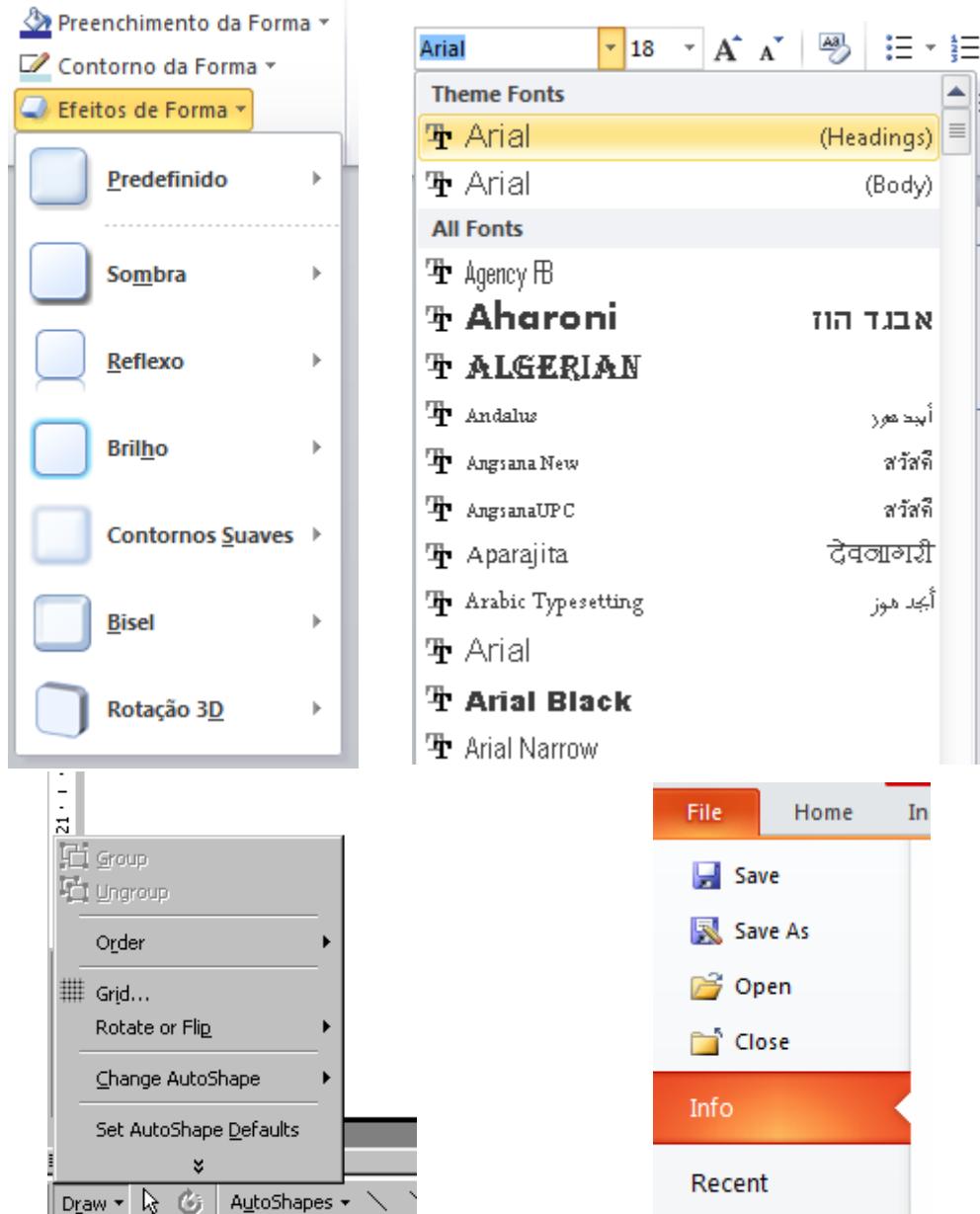
- Payroll
- Accounts Payable
- Accounts Receivable
- General Ledger
- Reports
- Write Checks

Enter Selection: __

Iconic

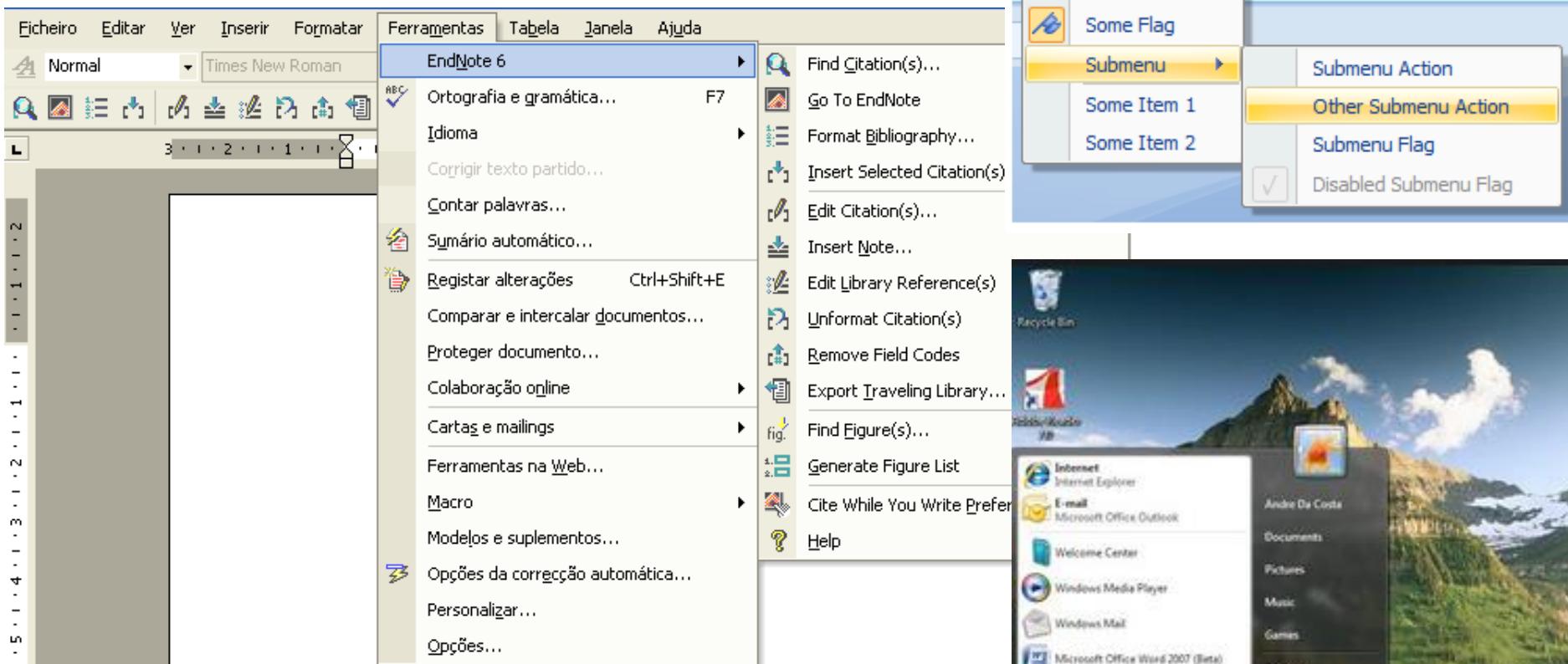


Textual



**Pull-down/ pull-up
menus**

Cascading menus



Main Heading

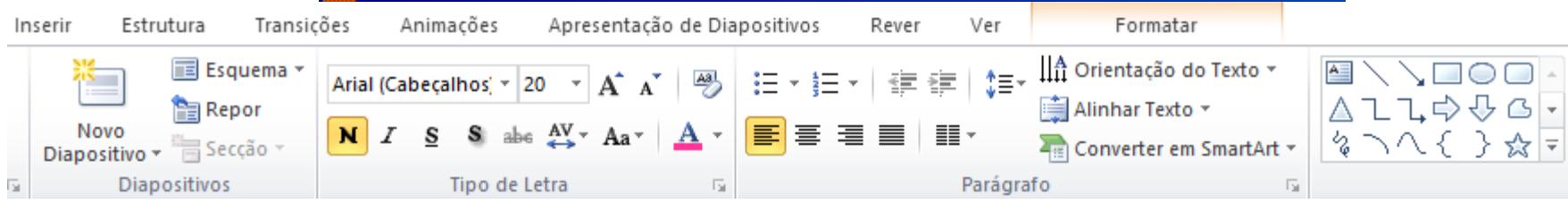
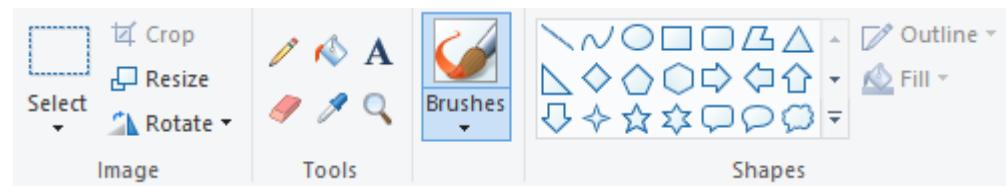
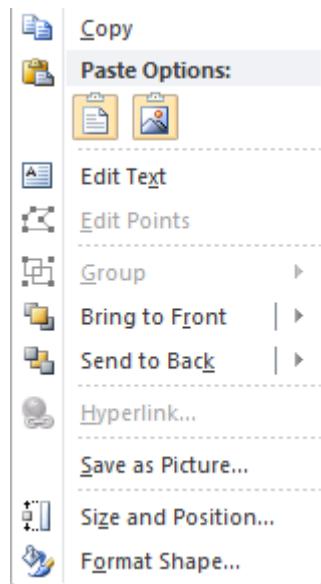
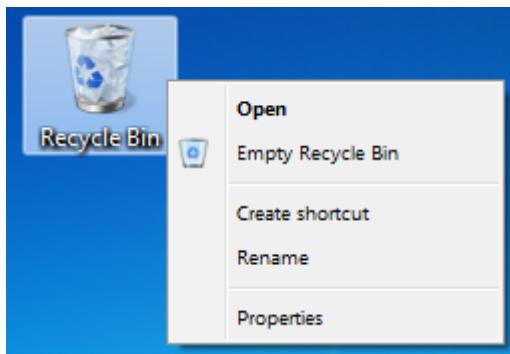
Menu 1

Menu 2

Menu 3

- Item 1
- Item 2 ... Subitem 1
- Item 3 Subitem 2
- Item 4 Subitem 3
- Subitem 4
- Subitem 5

Always visible / Pop-ups



Menus: main advantages and disadvantages

Advantages (potential, i.e. **if properly designed**)

- Auto-explanatory
- Do not load memory (recognition rather than recall)
- Prevent syntactic errors
- Visible improvements

Disadvantages

- Not efficient
- Not flexible
- Not practical for many options

User profile to whom menus are adequate:

Knowledge and experience:

- Low system and task experience
- Frequent usage of other systems
- Low computational literacy

Work and task:

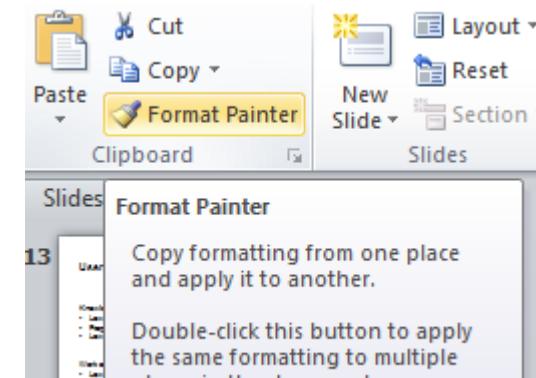
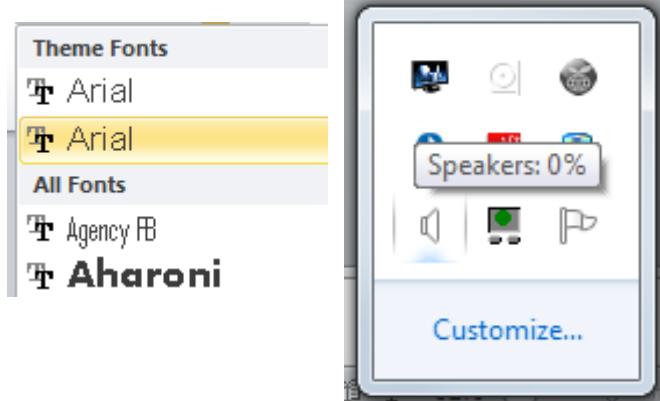
- Low frequency of use
- No training
- Optional usage
- Highly structured tasks

Menu design: relevant aspects

- Menu structure
- Option ordering
- Option selection
- Menu invocation
- Navigation

Menu design: guidelines

- Adequate the menu structure to the task structure
- Minimize depth increasing breadth (within reasonable limits)
- Use an adequate ordering method
- Be coherent (design, option names, etc.)
- Give selection feedback to the user
- Include tooltips if names or icons are not auto-explanatory
- Indicate currently unavailable options
- ...
- Etc. etc.



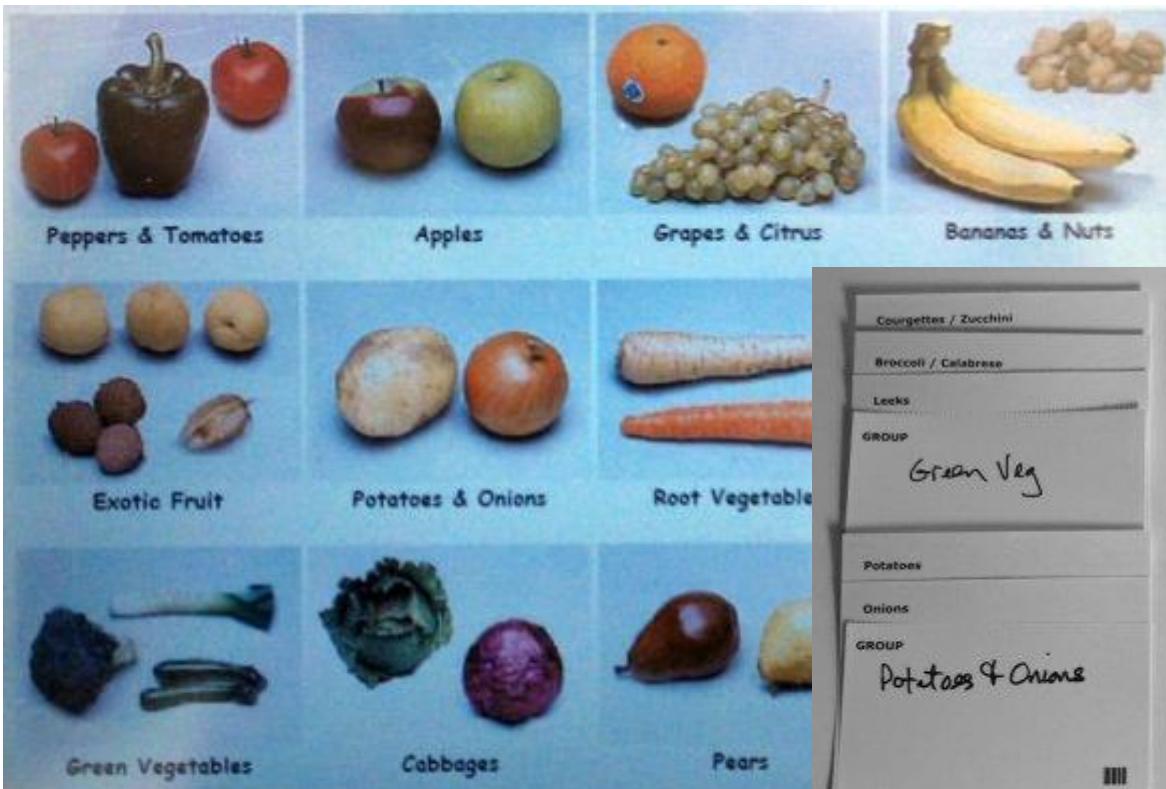
Menu design: guidelines

- Find the adequate structure using card sorting:
a low-cost method that helps understanding how users expect to find content or functionality



- Card sorting (usually performed by potential users of an interactive solution) provides information on:
 - Terminology (what people call things)
 - Relationships (proximity, similarity)
 - Categories (groups and their names)
- that can be used to decide upon:
 - which items should be grouped together in displays
 - how menu contents should be organized and labeled
 - what words should be employed to describe the objects of our users' attention

Card sorting example: think about how to sort the fruits and vegetables sold in a supermarket (may be it is not as easy as it seems...)



Three separate card sorting results are shown, each with handwritten group names:

- Group 1 (Left):** Labeled "Green Veg". It includes items from the first row: Peppers & Tomatoes, Apples, Grapes & Citrus, and Exotic Fruit.
- Group 2 (Middle):** Labeled "Root Veg". It includes items from the second row: Potatoes & Onions, Root Vegetable, and Green Vegetables.
- Group 3 (Right):** Labeled "Grapes & Citrus". It includes items from the third row: Cabbages and Pears.

Below these groups are lists of individual items:

- Group 1 (Left):** Courgettes / Zucchini, Broccoli / Calabrese, Leeks, GROUP, Green Veg.
- Group 2 (Middle):** Carrots, Turnips, Parsnips, Swede / Rutabaga, GROUP, Root Veg.
- Group 3 (Right):** Grapefruit, Lemons, Oranges, Grapes, GROUP, Grapes & Citrus.
- Group 4 (Bottom Left):** Potatoes, Onions, GROUP, Potatoes & Onions.
- Group 5 (Bottom Middle):** Kiwi Fruit, Lychees, GROUP, Exotic Fruit.
- Group 6 (Bottom Right):** Garlic, Chillies, Ginger, Fennel (bulb), GROUP, Exotic Veg.
- Group 7 (Bottom Far Right):** Mushrooms, Squash / Marrows, Pumpkin, GROUP, Squash & Mushrooms.

http://www.interaction-design.org/encyclopedia/card_sorting.html

Select adequate option ordering

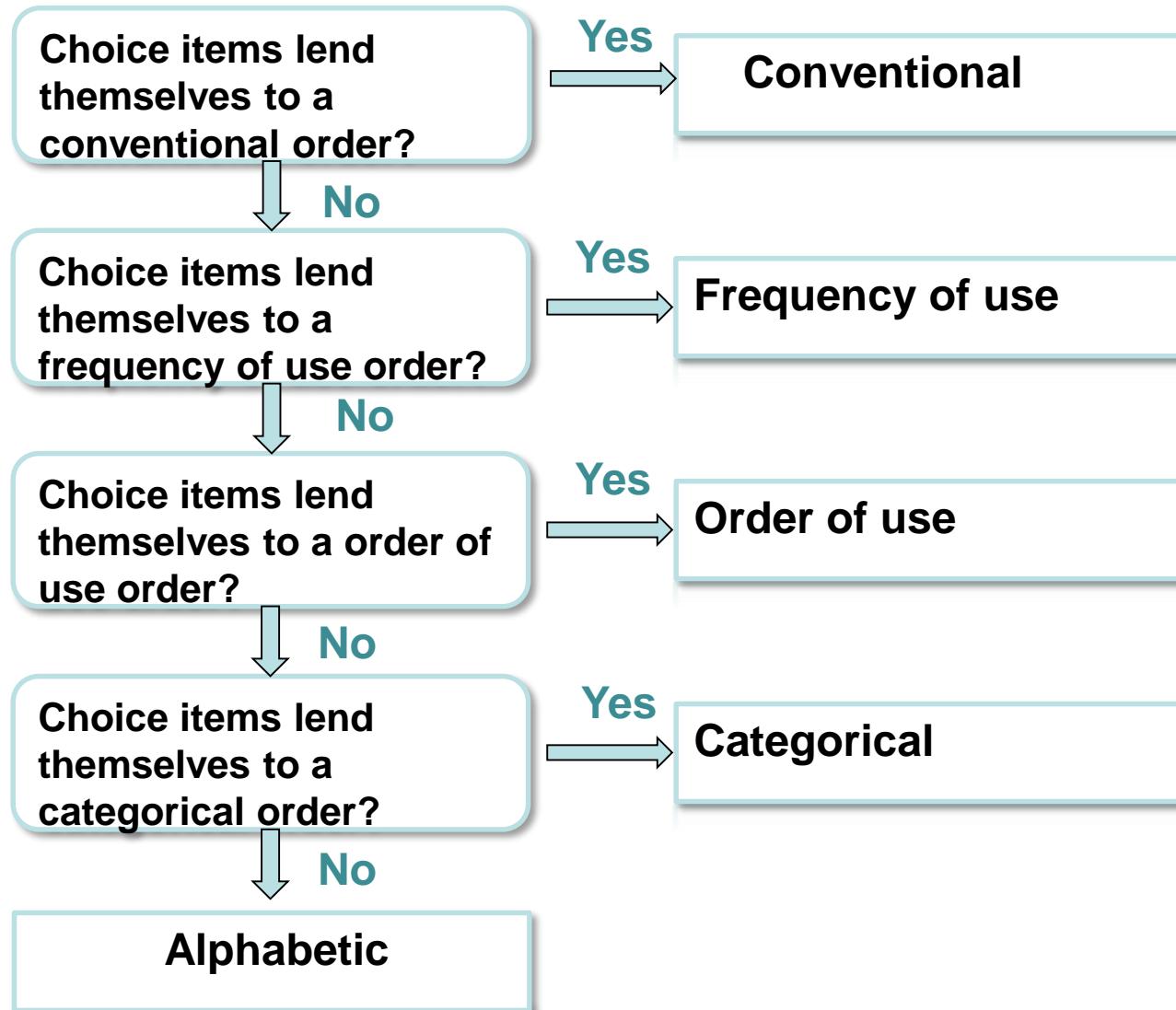
Which ordering scheme would you select?

April
August
February
January
June

January
February
March
April
May

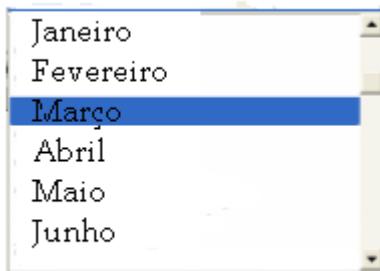
Why?

Select adequate option ordering

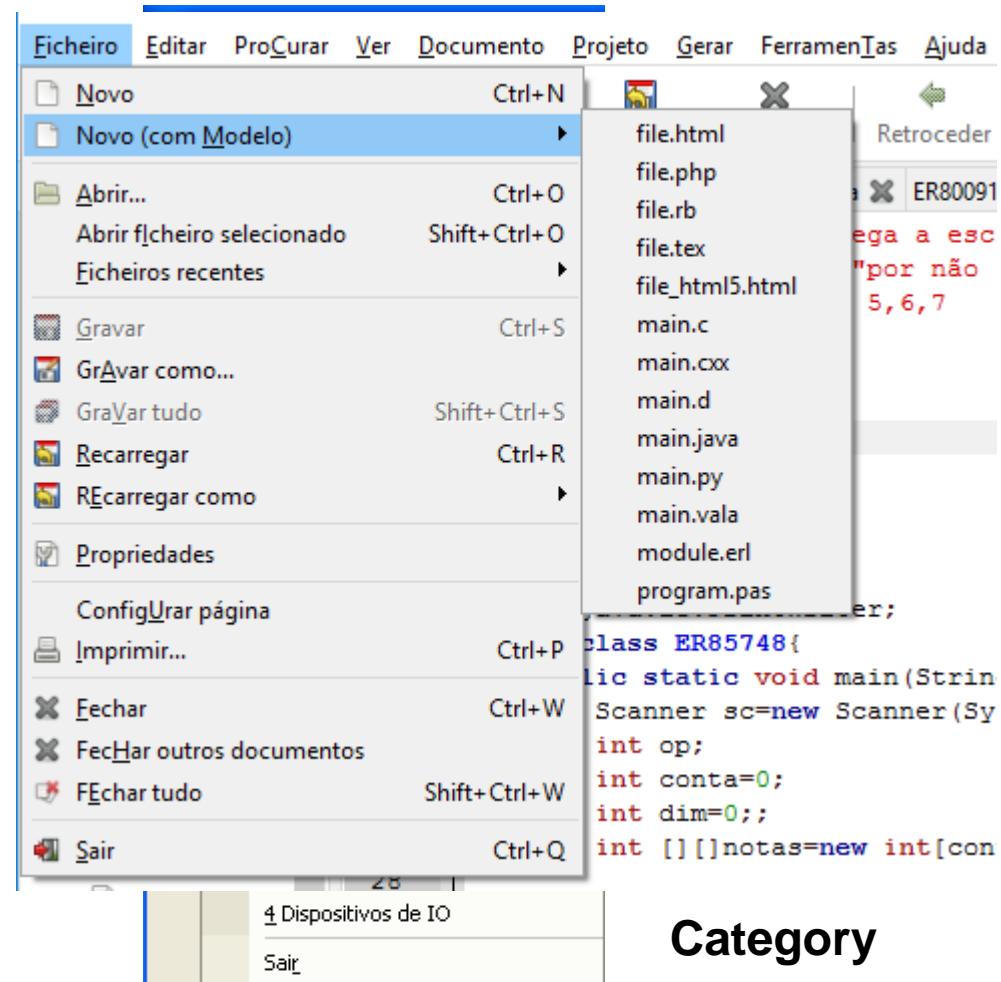
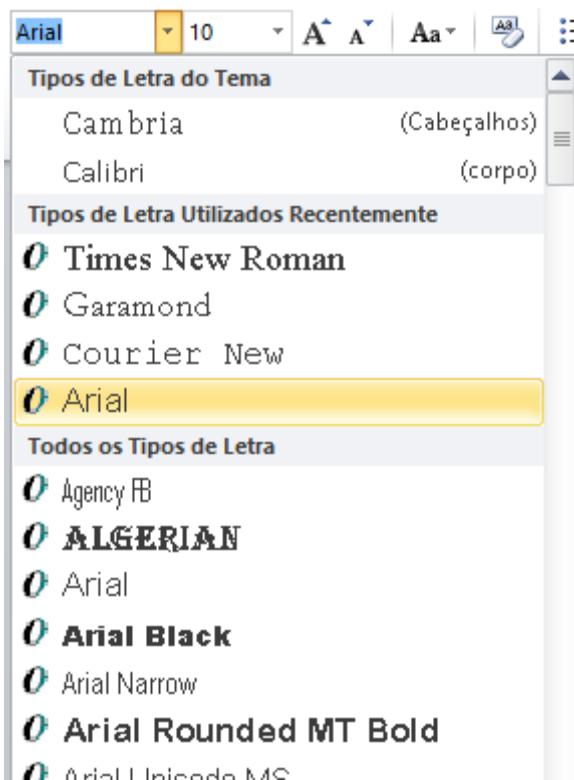


Option ordering

Conventional



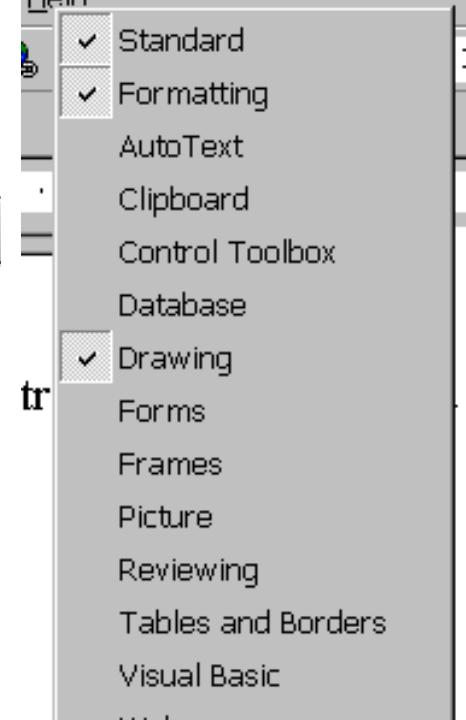
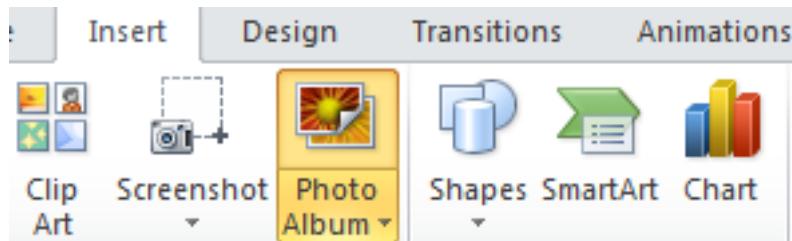
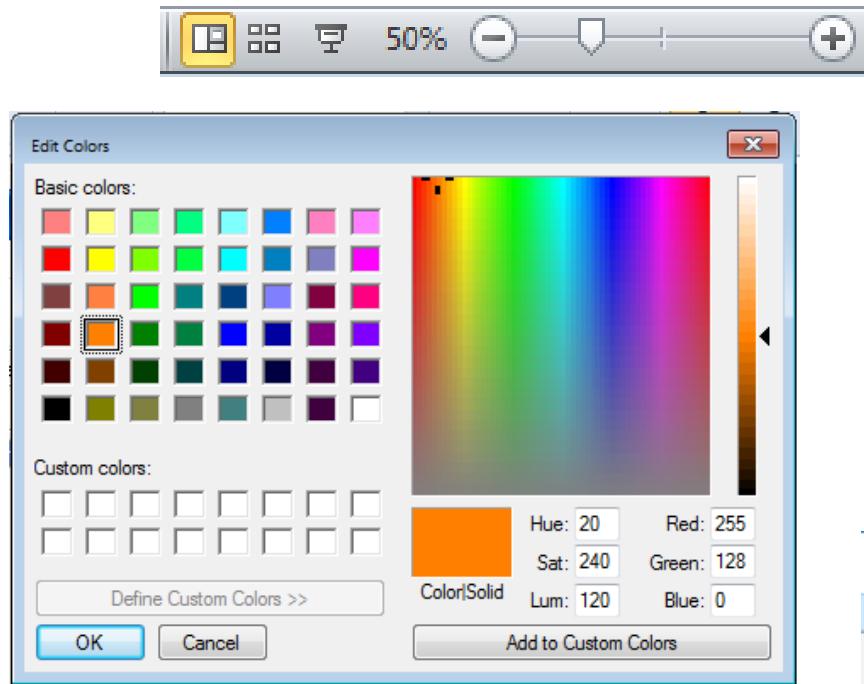
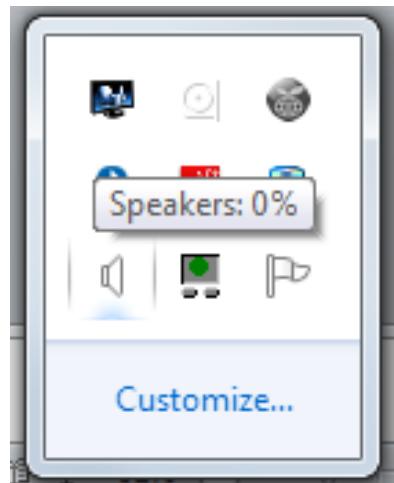
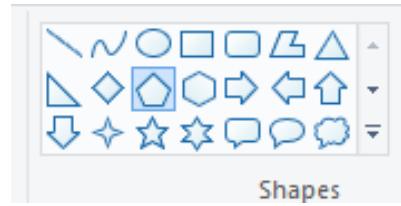
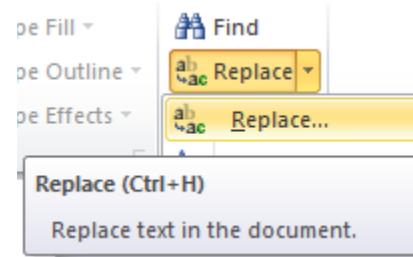
Alphabetic + frequency



Order of use

Category

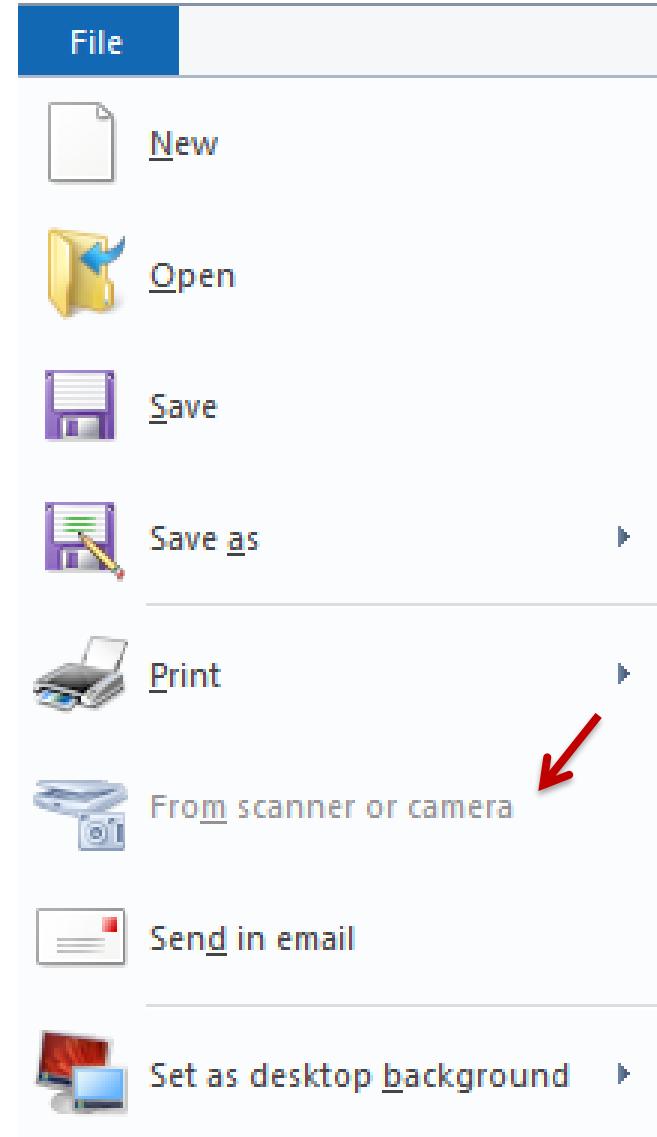
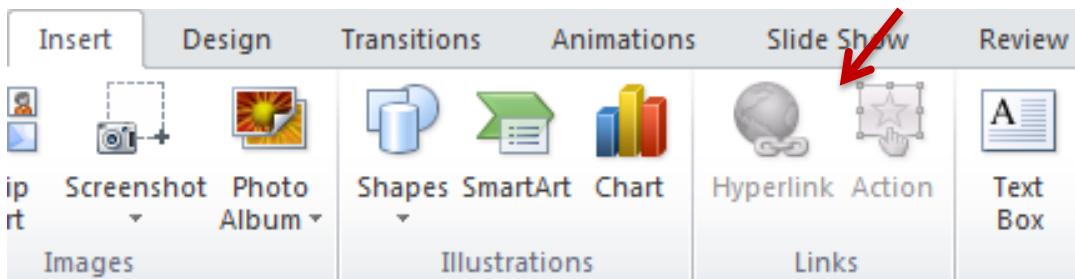
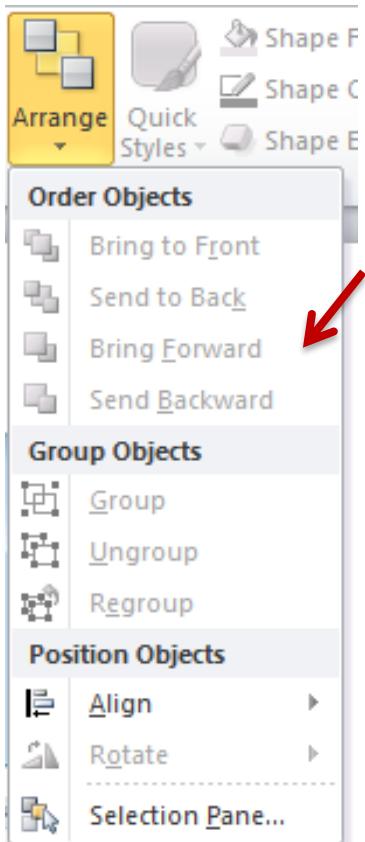
Give selection feedback



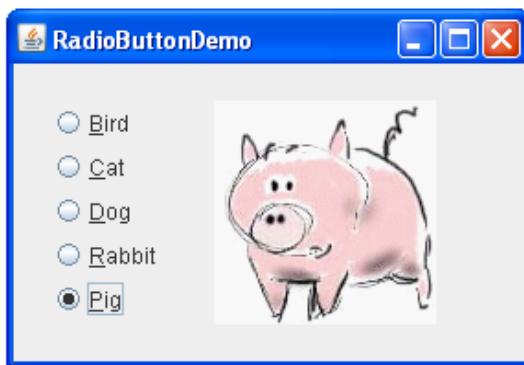
Indicate currently unavailable options

In grey to let users know they exist, but are unavailable

- Preventing errors
- And showing existing options (functionality)



Make clear the difference between choices of only one or several



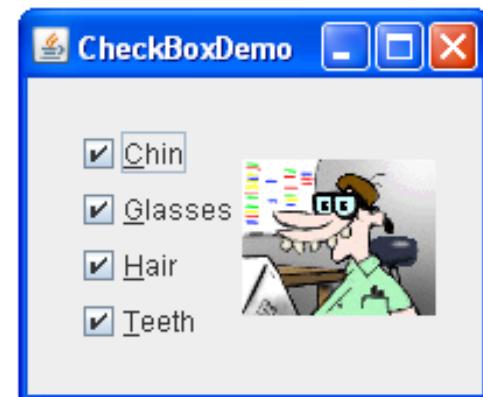
“Radio button”:
Only a single option can be selected from several mutually exclusive options

https://en.wikipedia.org/wiki/Radio_button

Effects

<input checked="" type="radio"/> No strikethrough	<input checked="" type="radio"/> No effect
<input type="radio"/> Strikethrough	<input type="radio"/> Shadow or outline
<input type="radio"/> Double strikethrough	<input type="checkbox"/> Shadow
<input checked="" type="radio"/> No super or subscript	<input type="checkbox"/> Outline
<input type="radio"/> Superscript	<input type="radio"/> Emboss
<input type="radio"/> Subscript	<input type="radio"/> Engrave

“Check box”:
Permits to make a binary choice.
A series of checkboxes may be presented
The user may select several of the choices



Select only one alternative:

	Service	From	To
<input type="radio"/>	AP No. 180	06:21	08:22
<input checked="" type="radio"/>	AP No. 130	07:21	09:22
<input type="radio"/>	IC No. 520	07:31	09:52

Possible to select more than one alternative:

 Additional seat options

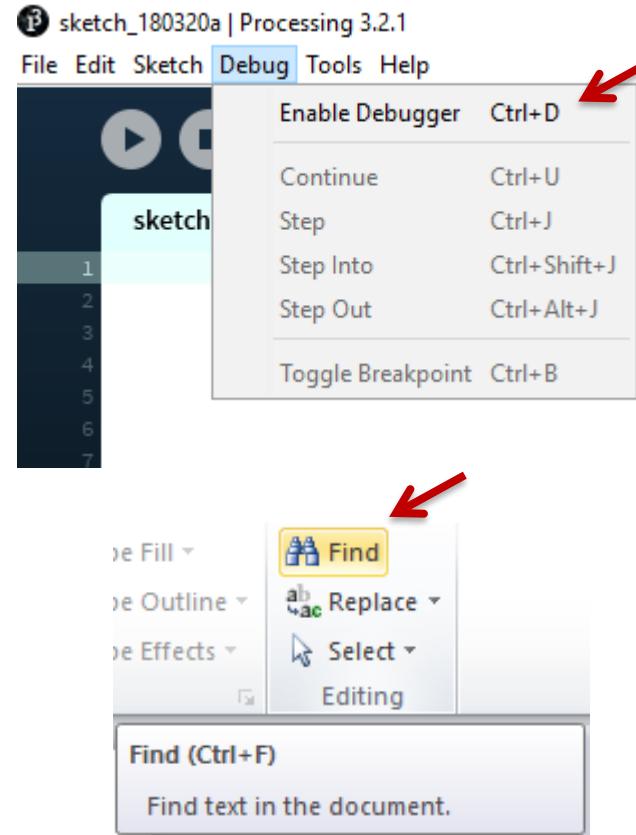
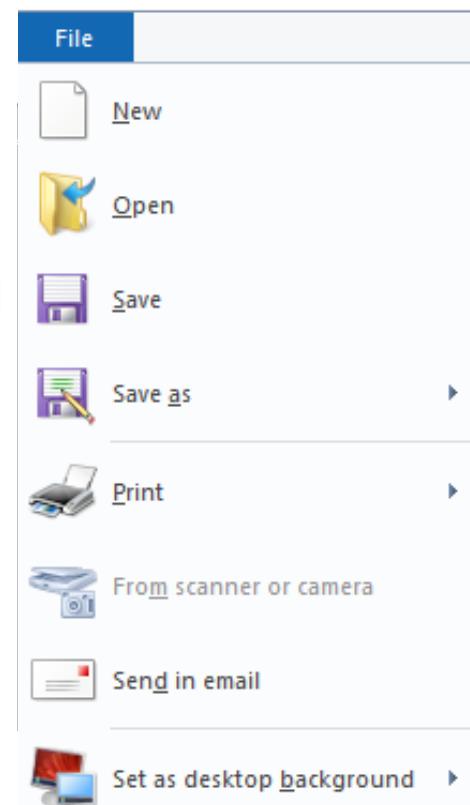
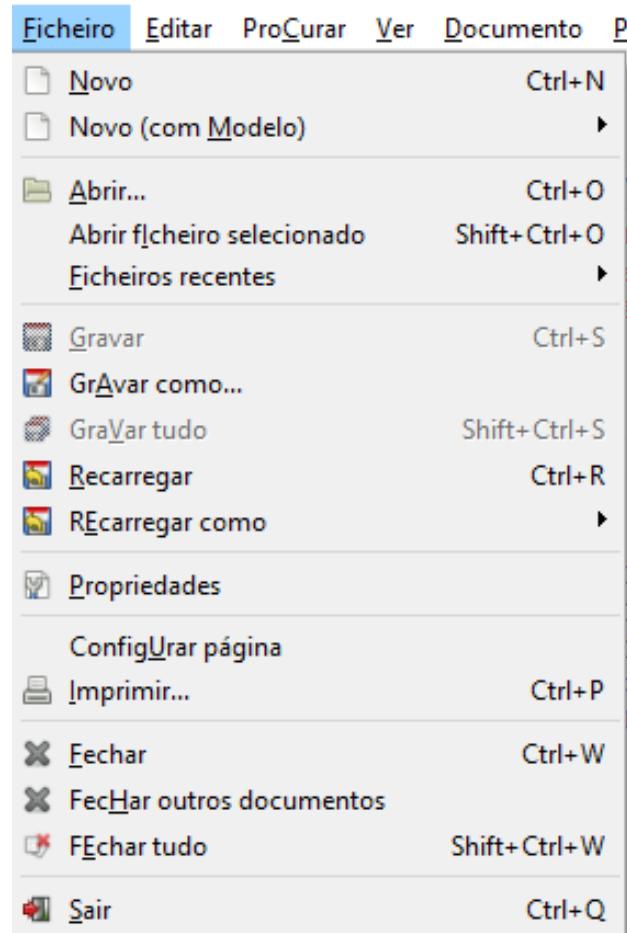
Special needs seat  Bikes 

May we send you updates using e-mail?

- Yes, please use e-mail to send me information about other offerings.

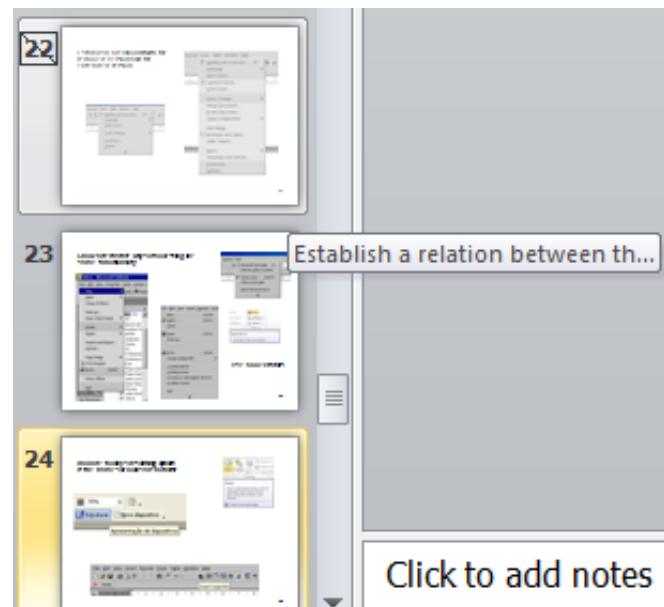
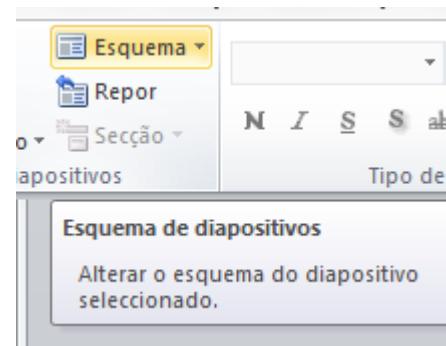
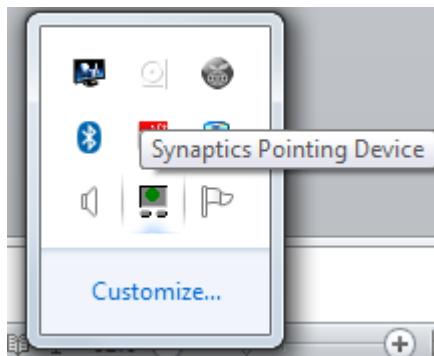
<https://www.nngroup.com/articles/checkboxes-vs-radio-buttons/>

Show alternative ways of accessing the same functionality

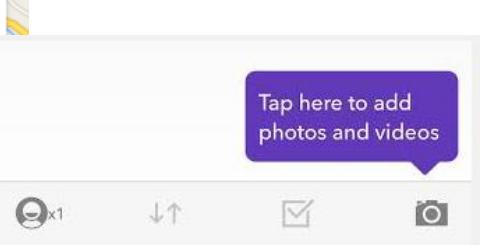
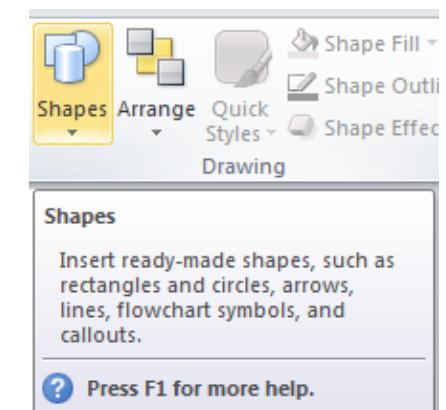
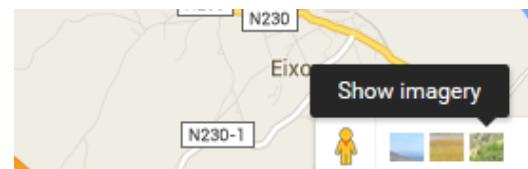


Use accelerators
(Flexibility and efficiency of use)

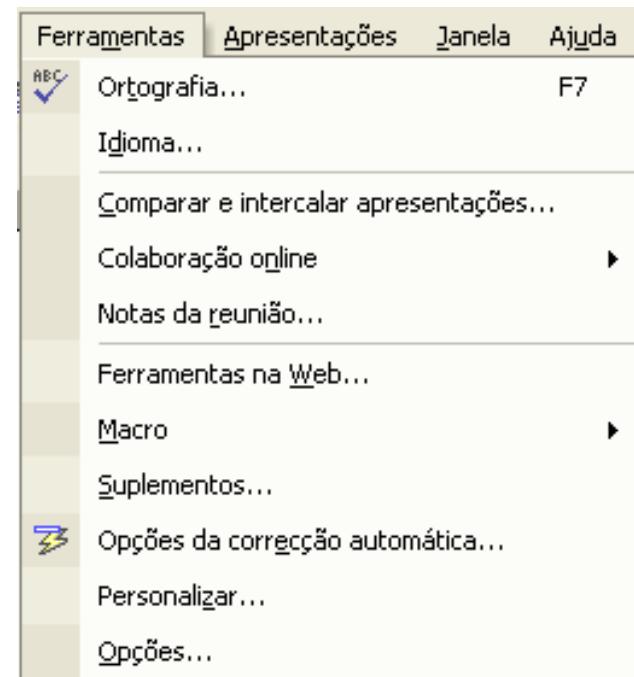
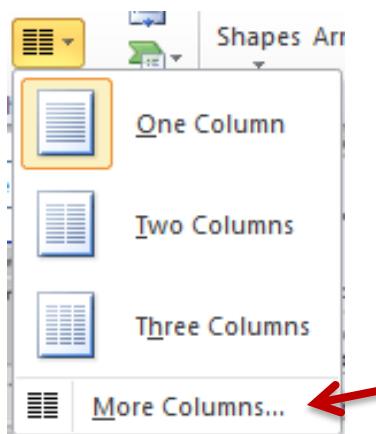
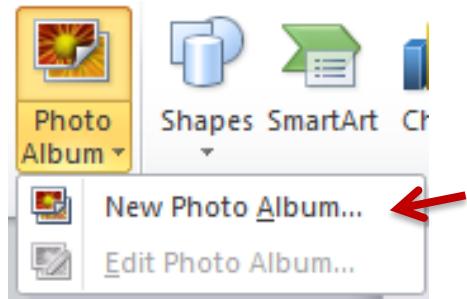
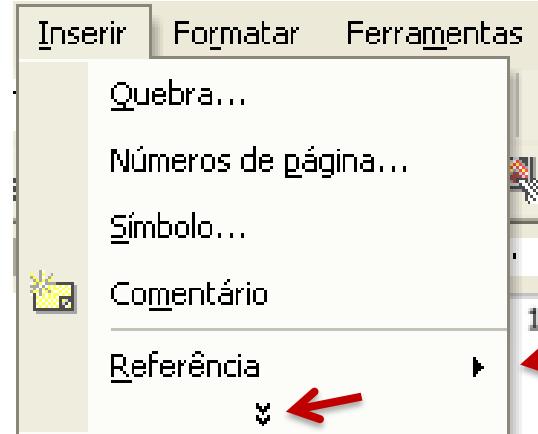
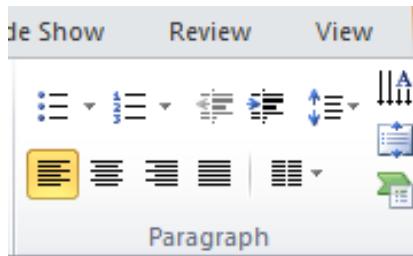
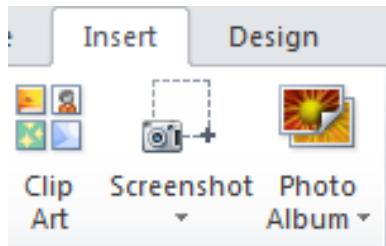
Include tooltips describing options if the names or icons are not clear



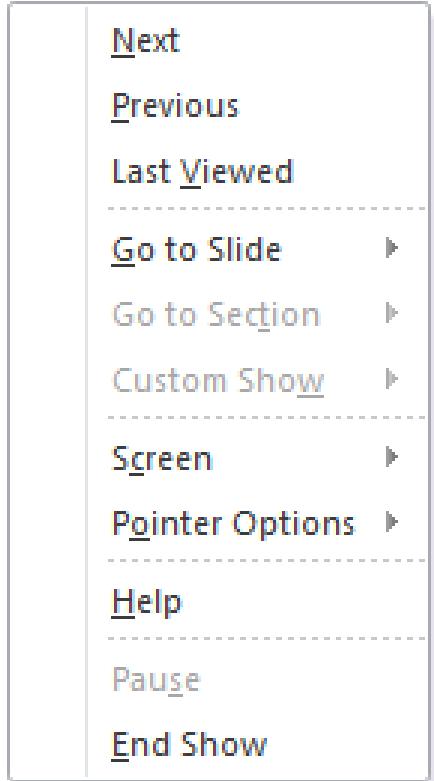
A screenshot of the Microsoft PowerPoint ribbon. A tooltip is shown for the 'Reset' button under the 'Layout' tab, stating 'Reset the position, size, and formatting of the slide placeholders to their default settings.'



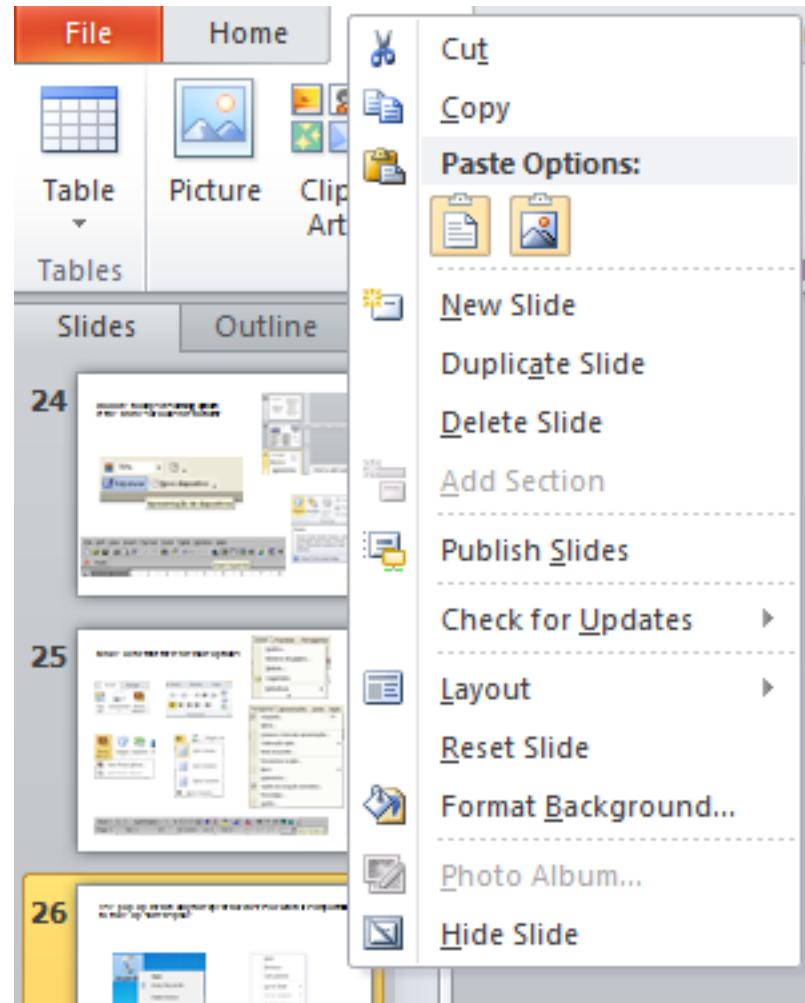
Make clear that there are more options



Use pop-up menus (context menus) only for experienced users or when it is very important not to take up screen space



(e.g.:
during a Power Point presentation)



Main Bibliography

- Ben Shneiderman, C. Plaisant, M. Cohen, et al., *Designing the User Interface-Strategies for Effective Human–Computer Interaction*, Pearson, 6th edition, 2016
- Mads Soegaard, Interaction Styles, In: Soegaard, Mads and Dam, Rikke Friis (eds.). *The Encyclopedia of Human-Computer Interaction*, 2nd Ed. Aarhus, Denmark: The Interaction Design Foundation
http://www.interactiondesign.org/encyclopedia/interaction_styles.html
- William Hudson, Card Sorting. In: Soegaard, Mads and Dam, Rikke Friis (eds.). *The Encyclopedia of Human-Computer Interaction*, 2nd Ed.". Aarhus, Denmark: The Interaction Design Foundation http://www.interaction-design.org/encyclopedia/card_sorting.html



Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Direct Manipulation

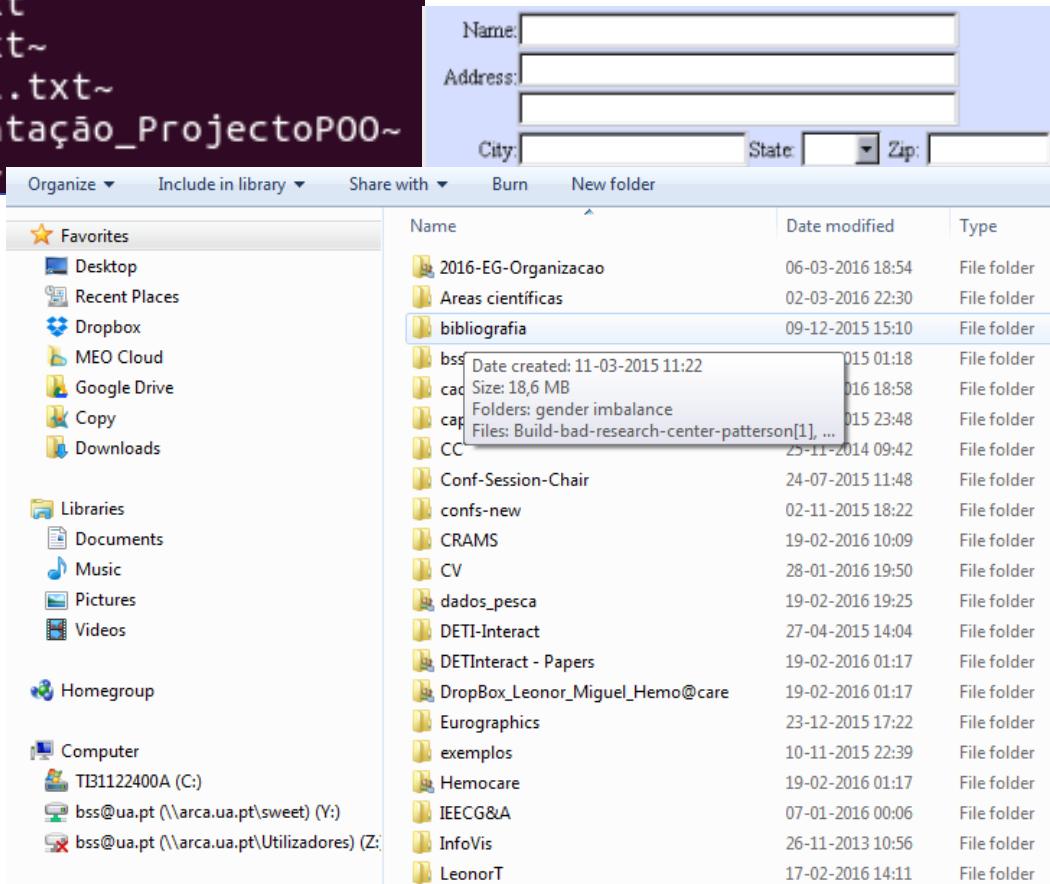


Interaction/ Dialog styles

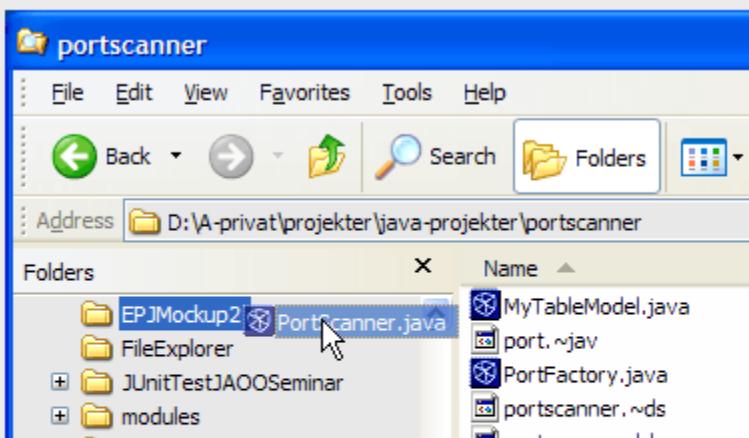
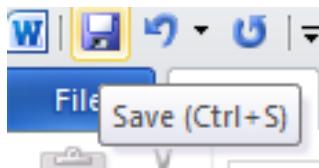
A possible classification:

- Menus
- Fill-in-forms
- Direct manipulation
- Function keys
- Question and answer
- Command languages
- Natural languages

emails_32YSM~
instruções_mex_c_matlab~
java-how-to.txt
java-how-to.txt~
matlab-install.txt~
Notas_implementação_ProjetoPOO~
notas_ros.txt~



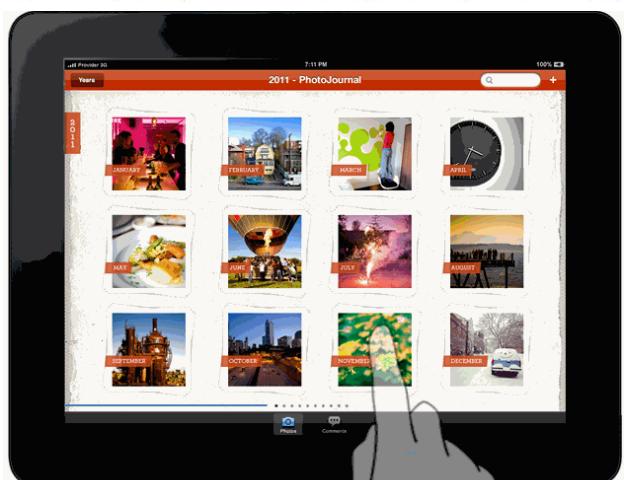
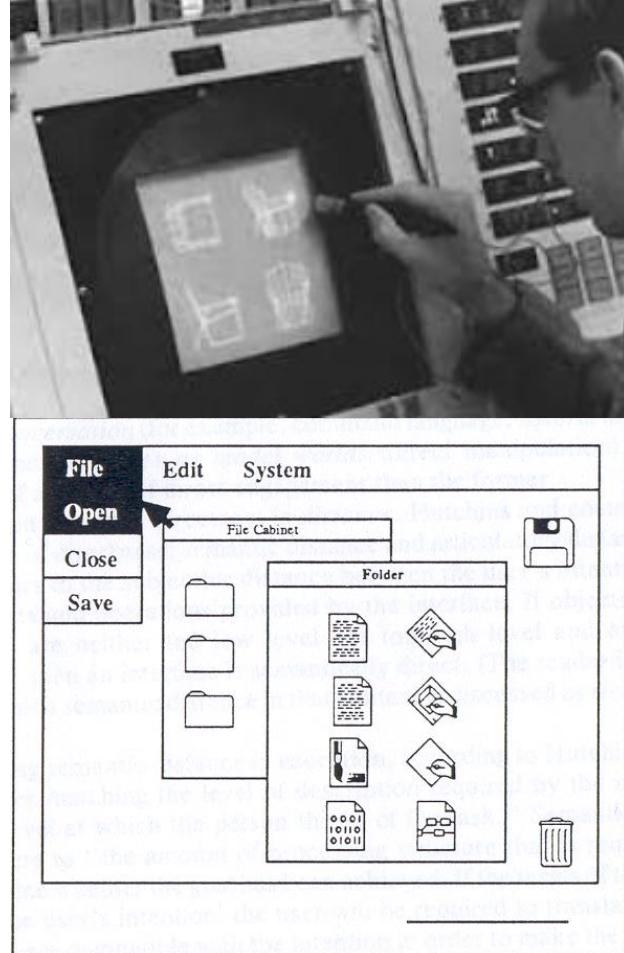
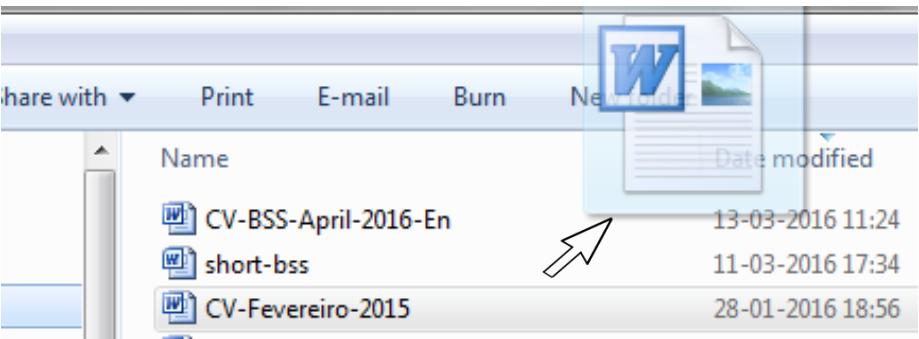
Often two or more styles are used simultaneously



Direct manipulation

(name coined by Shneiderman, 1982)

- Can be traced down to Sketchpad
<https://www.youtube.com/watch?v=495nCzxM9PI>
- **Actions are performed directly on visual representations** of the objects
- It is characterized by:
 - 1- Continuous representation of objects
 - 2- Physical actions instead of command languages
 - 3- Fast, incremental, reversible actions with visible results



Direct manipulation **does not necessarily imply icons**; however, in most situations they are involved

2 RELATED WORK

Despite the growing interest in usability related research in the VE community, not as many papers concerning usability evaluation exist, as compared to papers proposing new methods, techniques or systems. For instance, in a research recently conducted, we were able to find only a few studies directly comparing user performance while using VEs in desktops and systems including a HMD [Sousa Santos,2008] [Sousa Santos,2009]. Analyzing these studies, it can be observed that controlled experiments involving users have been the most used evaluation method, complemented in some cases with a questionnaire. We can also observe that most studies were performed in a general context (as opposed to applied to a specific situation), and that search and navigation were the chosen tasks in a significant part of them.

User studies have been considered an important method in other contexts, as Scientific Visualizations and Augmented Reality [Kosara,2008][Gabbard,2008]. We believe that they can, likewise, contribute to optimize VEs informing their design within a usability engineering approach; however, they can also be used to compare alternatives, validate solutions, and more fundamentally help seeking insight into why a particular solution is effective, thus allowing establish design guidelines.

Example: When a section of a text is selected and dragged elsewhere icons are not used, yet an action is performed on a visual representation of an object (text section)

2 RELATED WORK

User studies have been considered an important method in other contexts, as Scientific Visualizations and Augmented Reality [Kosara,2008][Gabbard,2008]. We believe that they can, likewise, contribute to optimize VEs informing their design within a usability engineering approach; however, they can also be used to compare alternatives, validate solutions, and more fundamentally help seeking insight into why a particular solution is effective, thus allowing establish design guidelines.

Despite the growing interest in usability related research in the VE community, not as many papers concerning usability evaluation exist, as compared to papers proposing new methods, techniques or systems. For instance, in a research recently conducted, we were able to find only a few studies directly comparing user performance while using VEs in desktops and systems including a HMD [Sousa Santos,2008] [Sousa Santos,2009]. Analyzing these studies, it can be observed that controlled experiments involving users have been the most used evaluation method, complemented in some cases with a questionnaire. We can also observe that most studies were performed in a general context (as opposed to applied to a specific situation), and that search and navigation were the chosen tasks in a significant part of them.

To study and compare usability issues concerning our low cost platforms we had to choose a context of use since usability cannot be defined in abstract. In fact, it is associated to users performing certain tasks [Nielsen,1993] (page 27) [Dix,2004] (page 192). Given that we had not a

Direct manipulation does not necessarily imply icons; however, in most situations they are involved

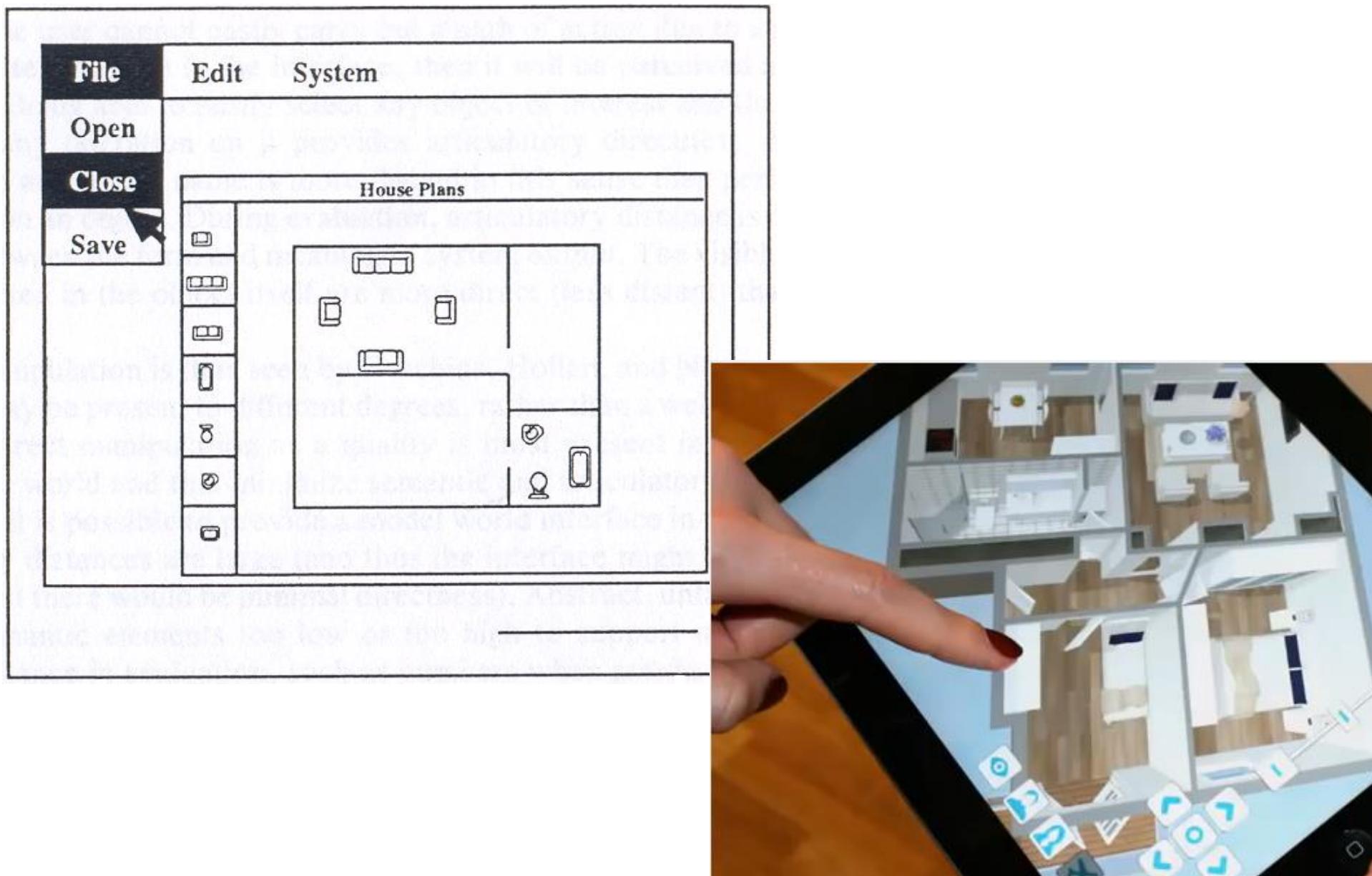


Another example:

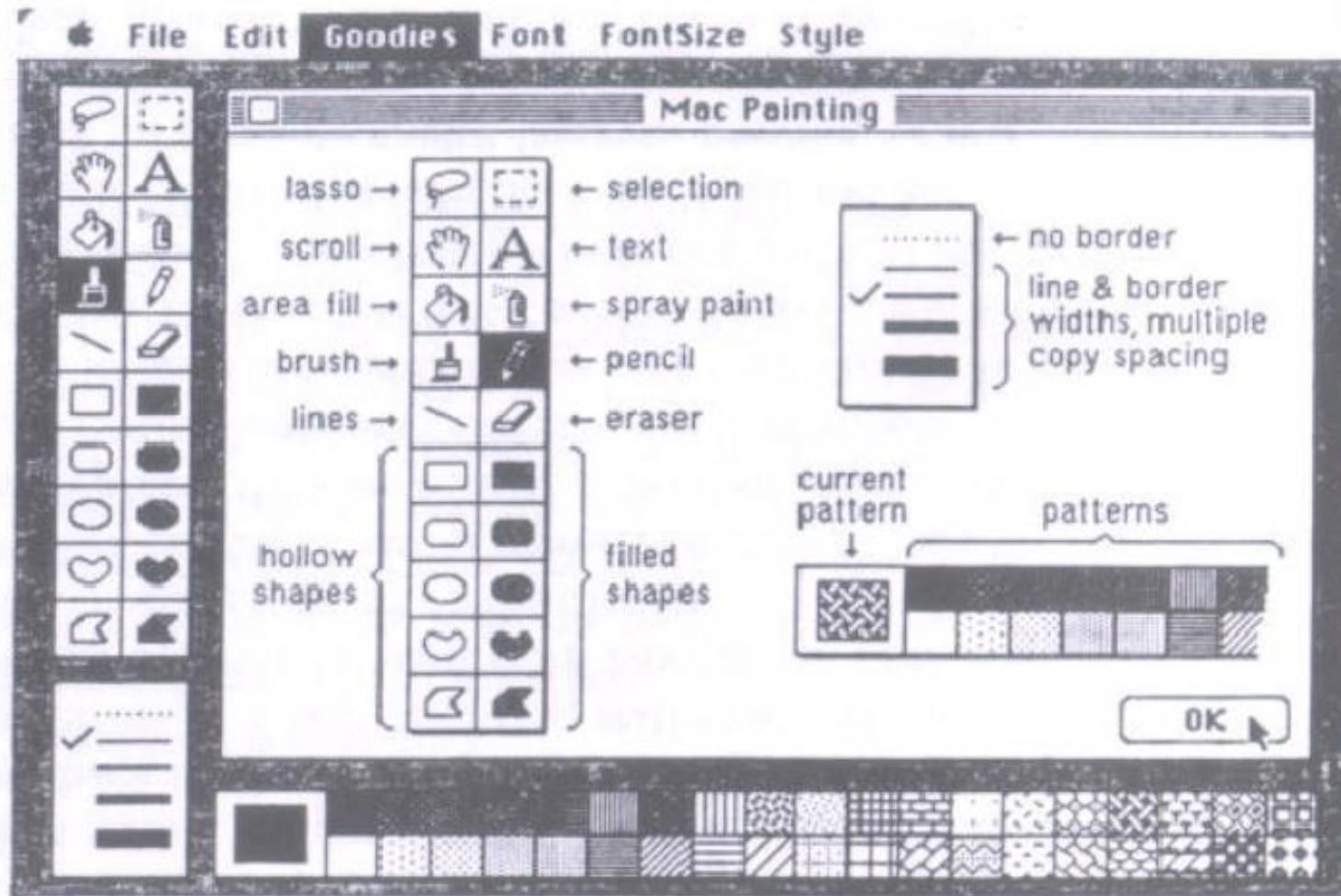
On a mobile phone you can pinch out/in to zoom into an image or to zoom out

<https://www.nngroup.com/articles/direct-manipulation/>

Some applications are adequate to use direct manipulation:



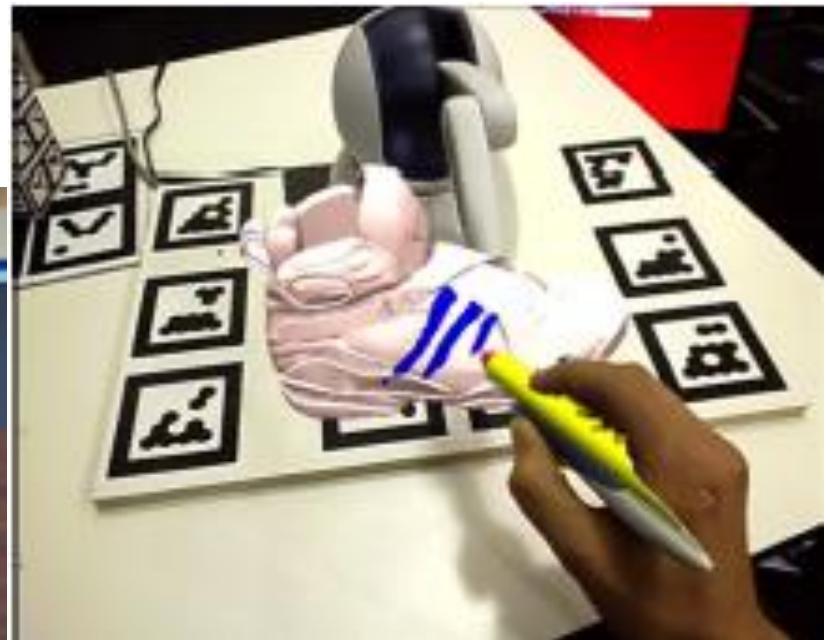
One of the earliest commercially available UI using Direct Manipulation (MacPaint)



http://www.magicvisionlab.com/pub/eck_i_eeevr13/paper.pdf

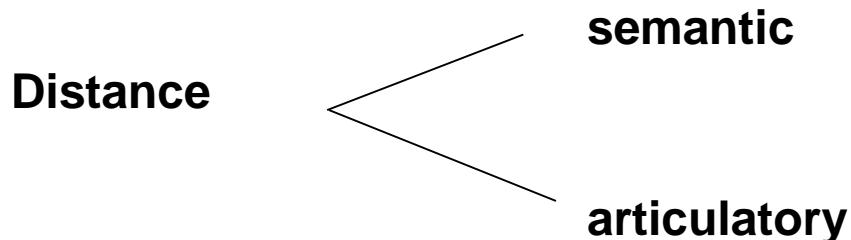
Virtual and augmented reality

Take direct manipulation to another level



<http://www.cyberglovesystems.com/cad-evaluator>

- It does not exist a “pure” direct manipulation User Interface (UI)
- Direct manipulation is a **quality which may be present in different degrees**
- According to Hutchins, Hollan e Norman (1986) a UI has the following aspects:

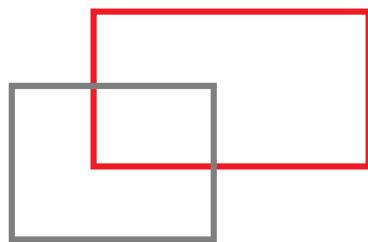
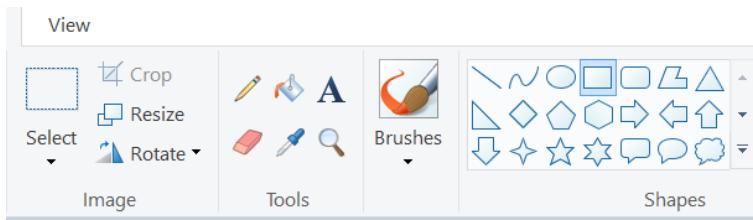


Semantic and articulatory distance

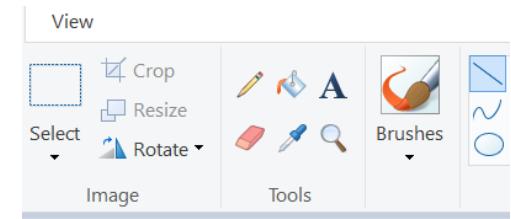
- Semantic Distance – **subjective distance** between the **user's goal** and **interface semantics**
- Articulatory distance – distance between the **meaning of the actions** and their **physical form**

Semantic Distance

If the objects and actions do not support the users' goals, semantic distance is high



If the user wants to draw rectangles this application has a smaller semantic distance
,

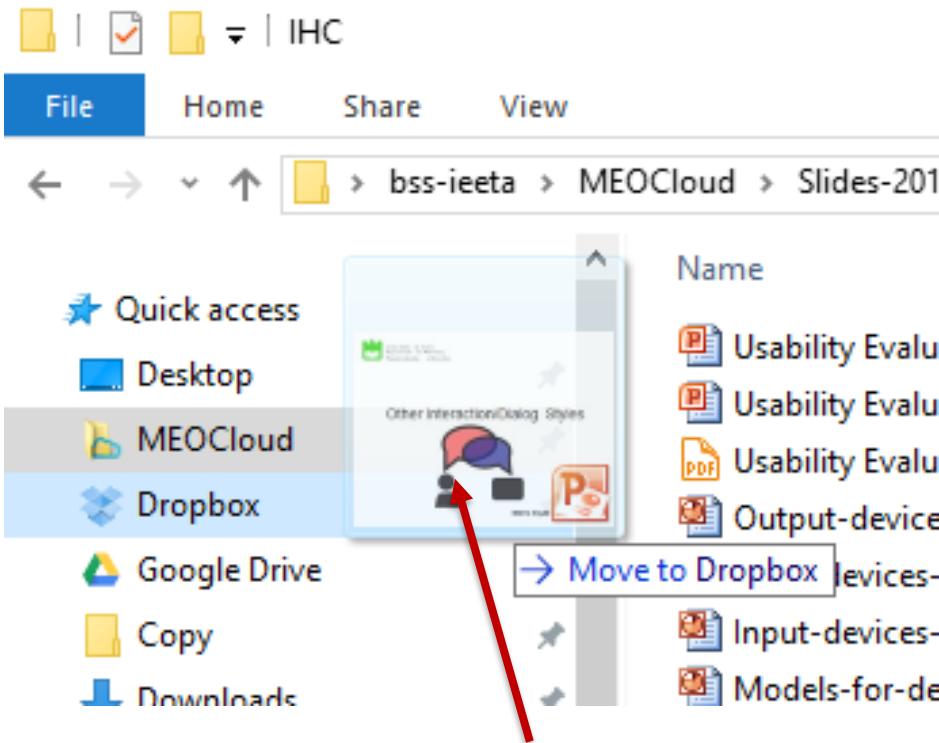
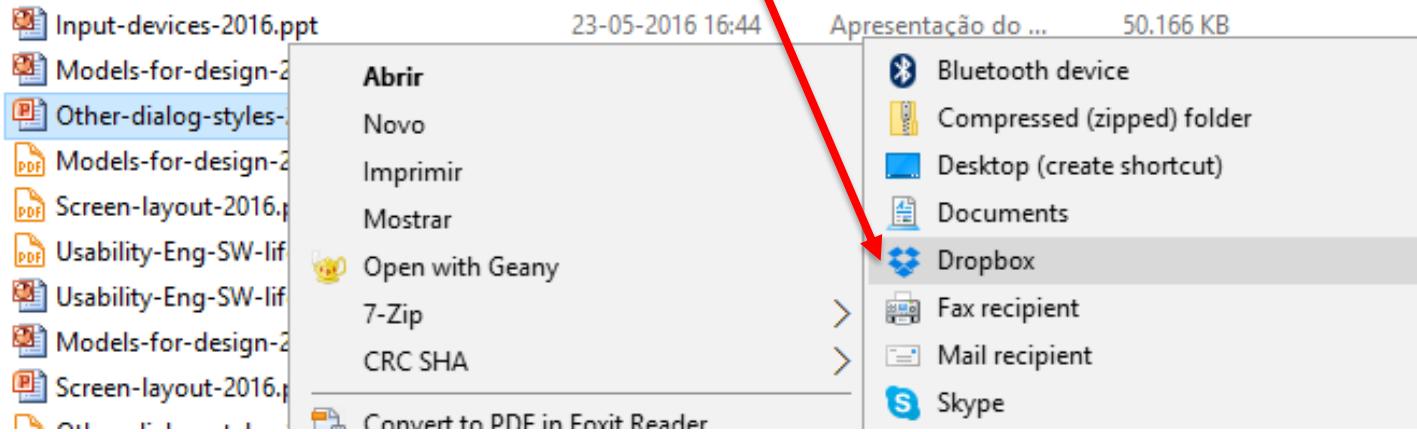


In this case the user is still able to draw rectangles, yet the application has a greater semantic distance (feels less direct)
,

Articulatory Distance

When the physical way actions are performed is more similar to their meaning, articulatory distance is smaller

Selecting an option corresponds to a greater articulatory distance



- According to Wolf e Rhyne (1987) there are two relevant aspects in any user interaction:

Object specification

name generation

visual correlation

Action specification

name generation (write a name)

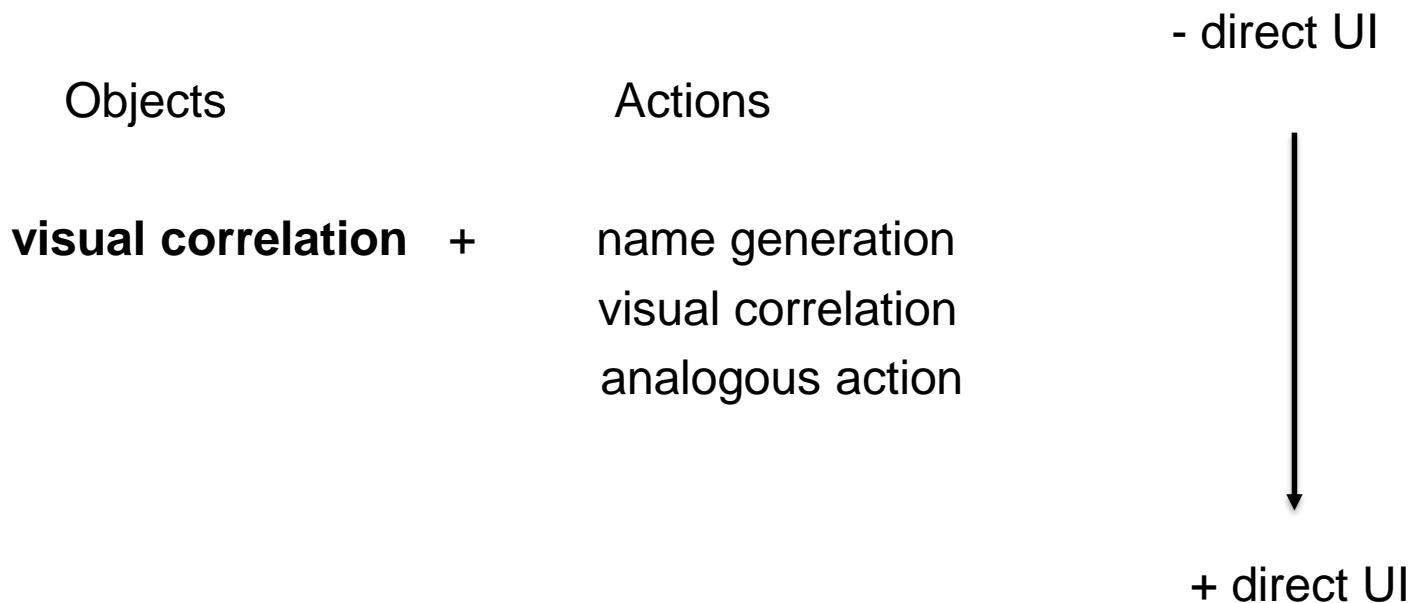
visual correlation (select)

gesture generation (draw a symbol)

analogous action

coded selection (write a command)

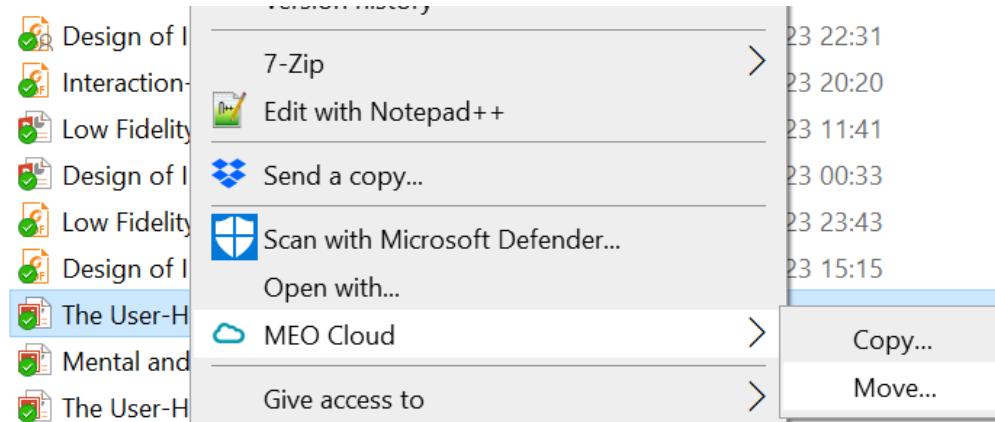
- Specifying objects by visual correlation implies the presence of direct manipulation
- How actions are specified defines the degree of direct manipulation



Examples

```
bi@ub:~/Desktop$  
bi@ub:~/Desktop$ mv java-how-to.txt smartbike_paper/  
bi@ub:~/Desktop$  
bi@ub:~/Desktop$
```

Not direct manipulation UI: name generation + name generation



Direct manipulation UI: visual correlation + visual correlation

+ Direct manipulation UI: visual correlation + analogous action



Main advantages and disadvantages of direct manipulation UIs

Advantages (potential)

- Easy to learn and remember (are great for novices with good design)
- Direct, WYSIWYG (What you see is what you get)
- Flexible, easily reversible actions
- Immediate visual and context feedback
- May be less prone to errors

Disadvantages

- Not auto-explanatory
- May be inefficient
- Repetitive tasks are not well supported
- Some gestures can be more error-prone than typing
- Difficult to draw recognizable icons (particularly for actions)
- Icons occupy more screen real estate than text



User profile to whom direct manipulation is adequate:

Knowledge and experience:

- Moderate system experience
- Moderate to high task experience
- Frequent usage of other systems
- Low computational literacy

Work and task

- Low frequency of use
- Moderate training
- Optional usage
- Low structured tasks

Direct Manipulation design: some guidelines

Minimize articulatory and semantic distance

Use general guidelines to design a usable UI:

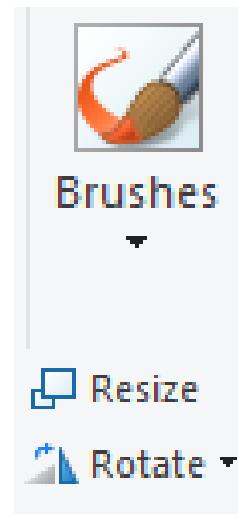
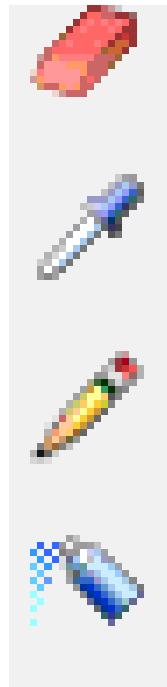
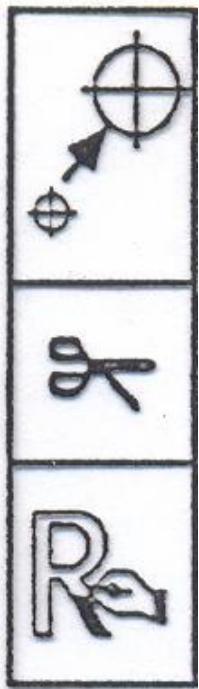
- Coherence
- Good conceptual model
- Feedback
- Adequate organization of functionality
- Adequate screen layout
- Adequate colour usage
- Adequate error handling
- Etc.

<https://www.interaction-design.org/literature/article/a-brief-history-of-the-origin-of-the-computer-icon>

Use a coherent Icon scheme

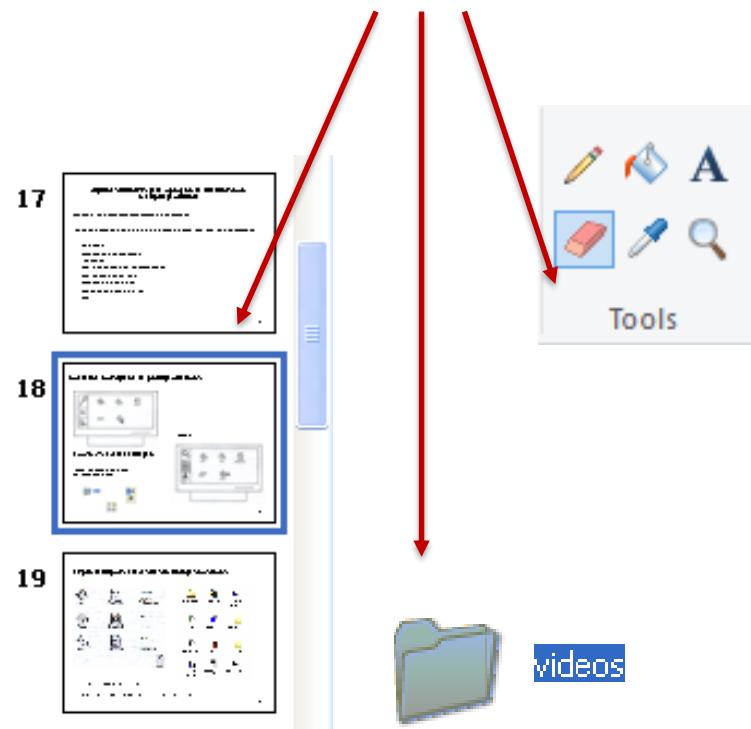
Different schemes:

better: same scheme

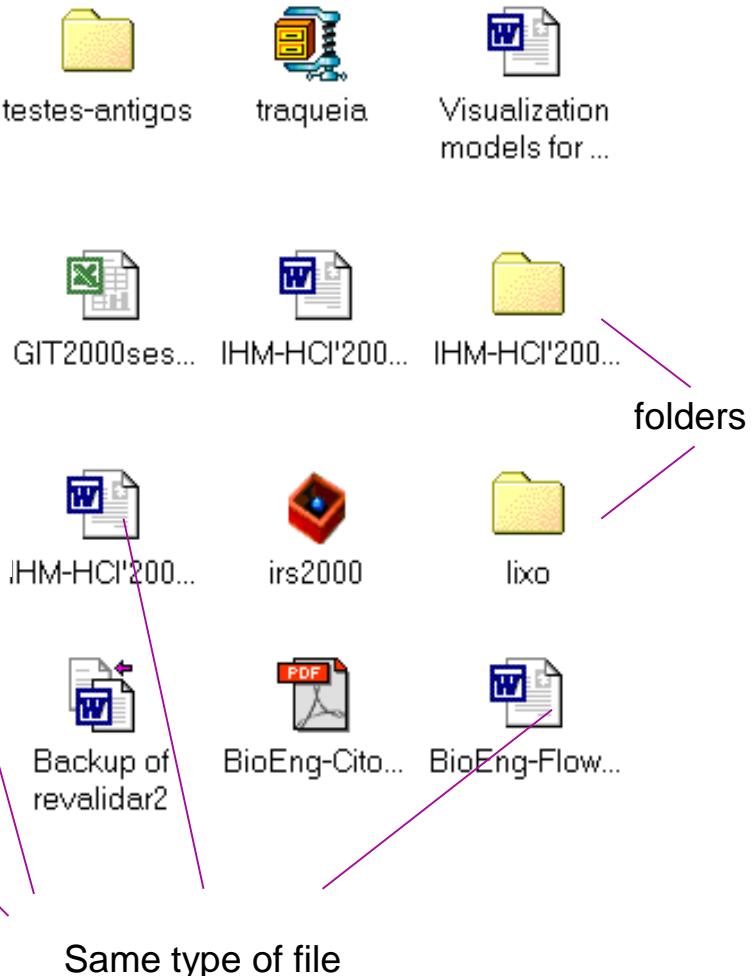


Adding names
(+ recognizable)

Visual selection feedback

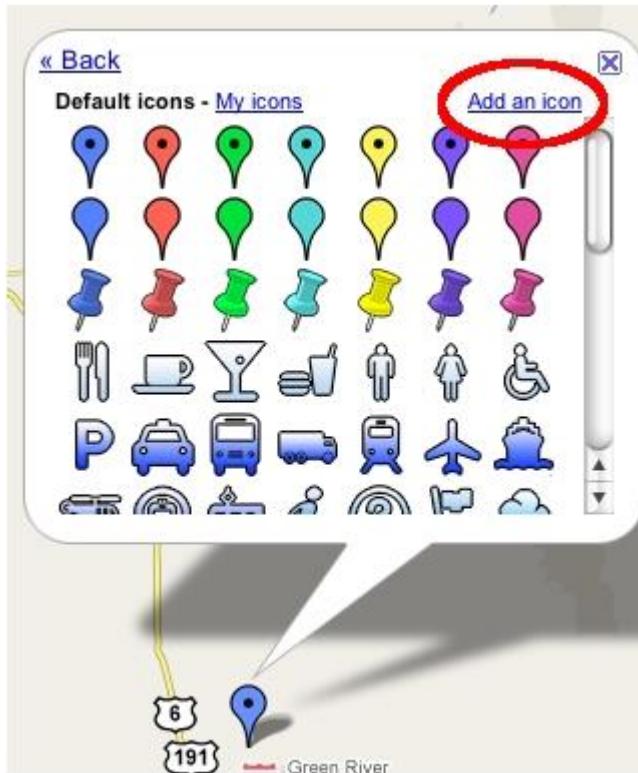


Express relation through icon similarity



Coherence in the icons production scheme

Add names to icons to make them more recognizable
(recognition rather than recall)



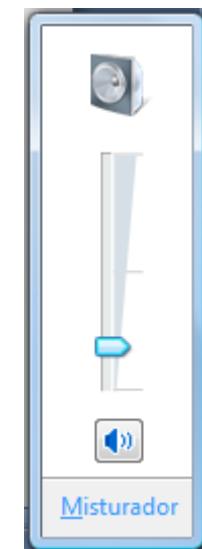
Allow name definition



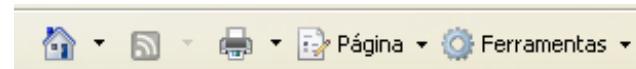
contactos



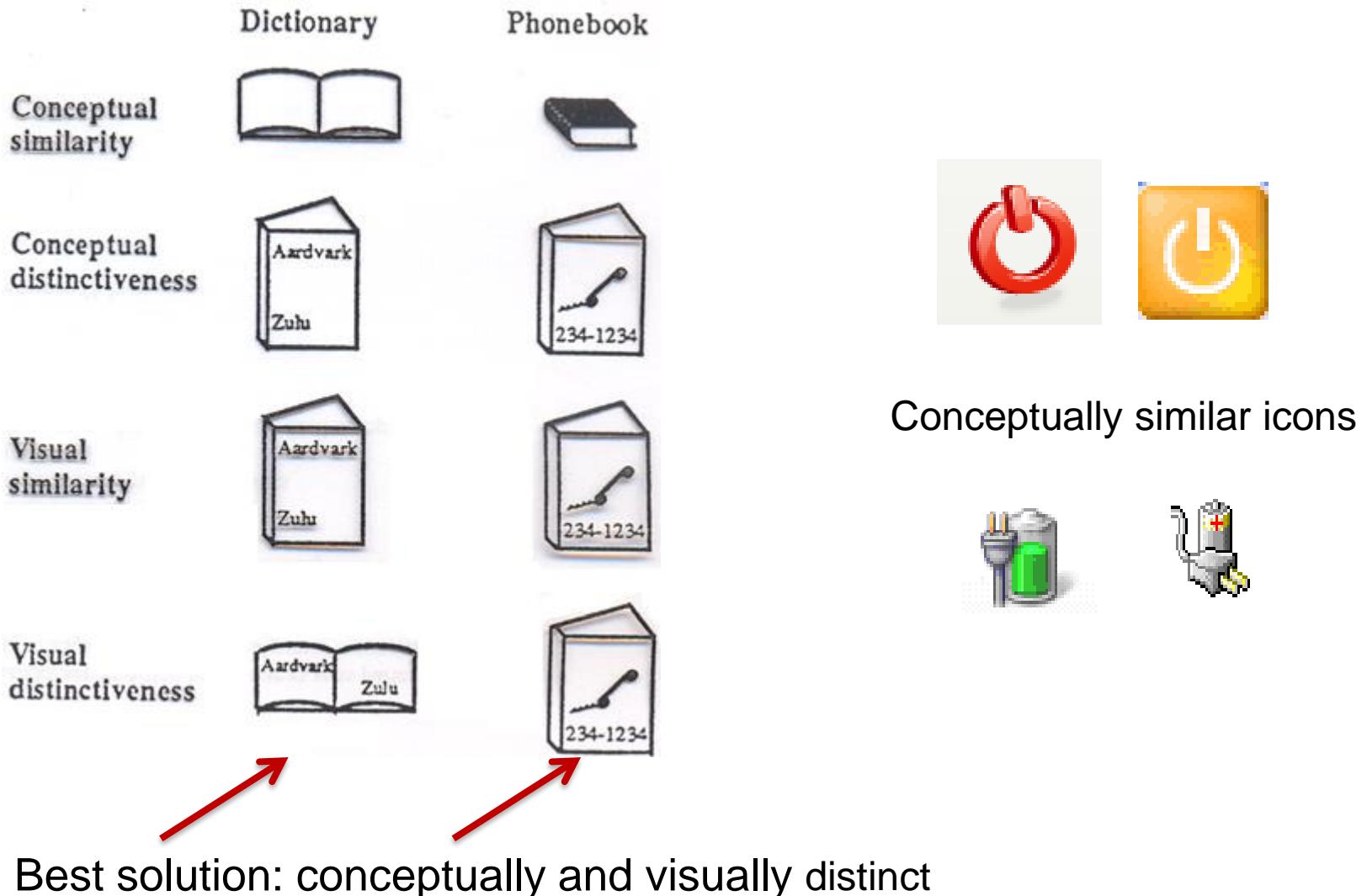
directório



Misturador



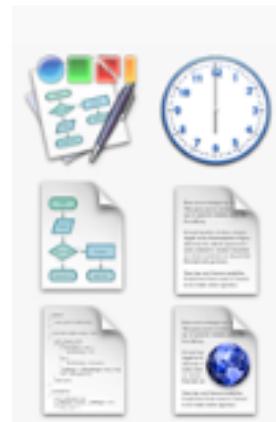
Icons must be conceptually and visually distinctive
(recognition rather than recall)



Icons should be specific/familiar not abstract/non-familiar
(familiarity)



- ⌄ This PC
- > 3D Objects
- > Desktop
- > Documents
- > Downloads
- > Music
- > Pictures



Familiar Icons?

Express objects' attributes through icons (visibility of the system status)



synced

empty



In progress

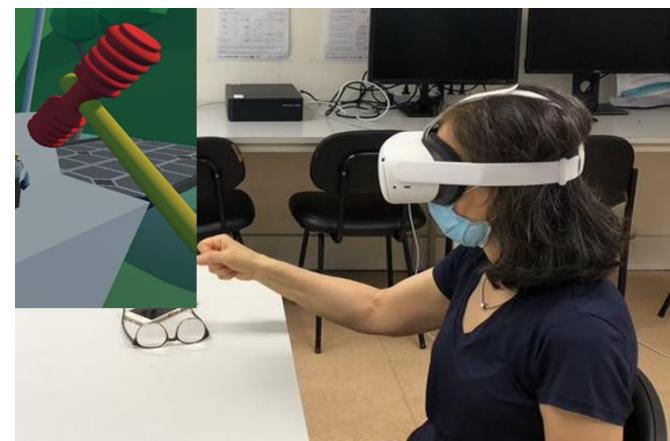
Direct manipulation - Concluding remarks

“It’s hard to imagine modern interfaces without direct manipulation ...

Augmented-reality and virtual-reality systems will push DM to even newer limits ...

Despite the many downsides, we still recommend a heavy dose of direct manipulation for most UIs”

<https://www.nngroup.com/articles/direct-manipulation/>

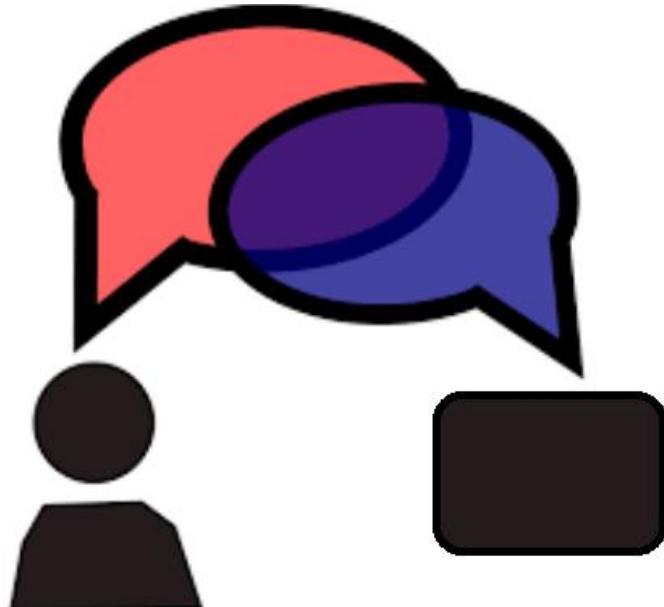


Main Bibliography

- B. Schneiderman, C. Plaisant, M. Cohen, S. Jacobs, *Designing the User Interface- Strategies for Effective Human–Computer Interaction*, 5th ed., Addison Wesley, 2010
- H. Sharp, J. Preece, and Y. Rogers, *Interaction Design: Beyond Human-Computer Interaction*, 5th Edition Wiley, 2019
- M. Soegaard, Interaction Styles, *Interaction Design Foundation Encyclopedia*, 2nd edition,
http://www.interactiondesign.org/encyclopedia/interaction_styles.html



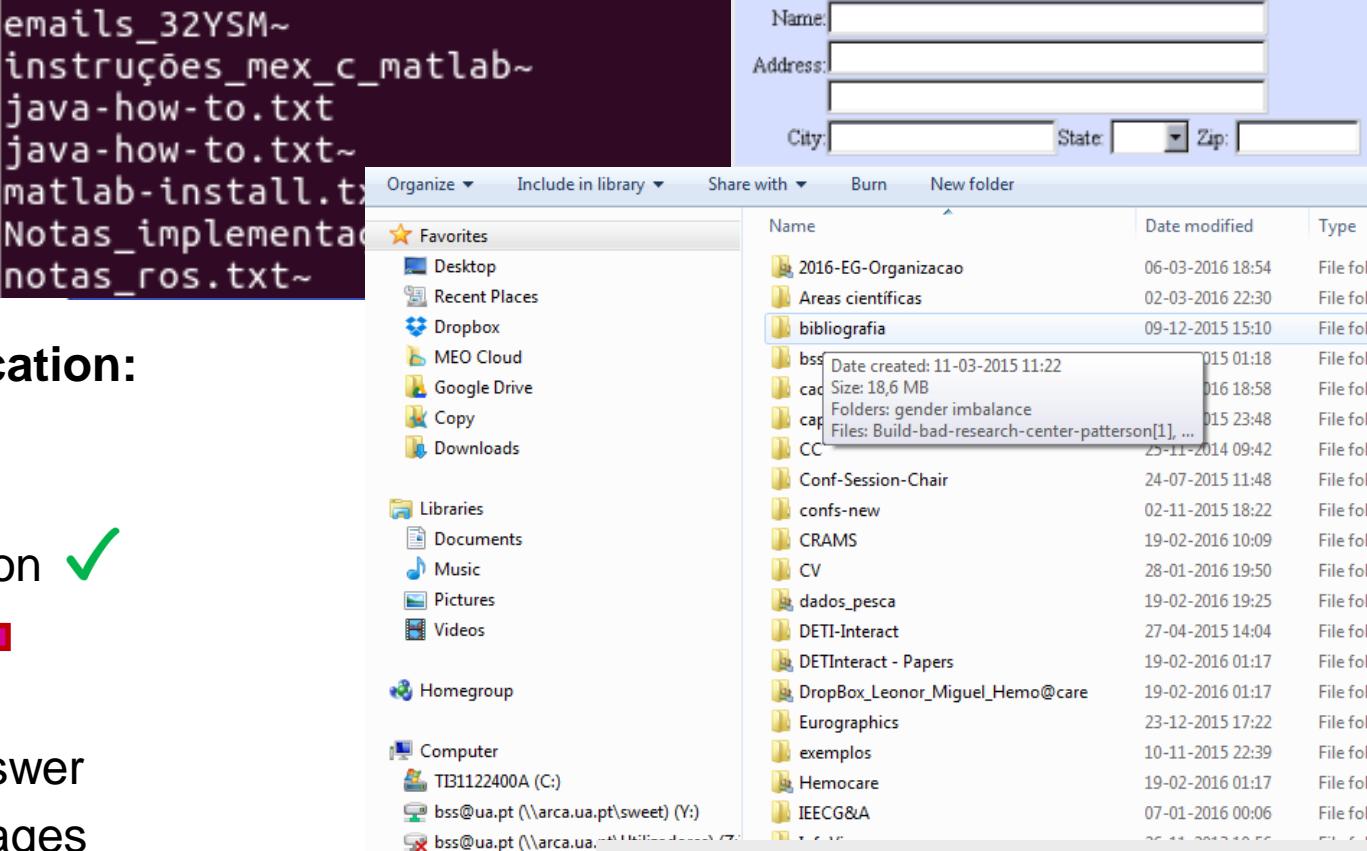
Other Interaction Styles



Interaction styles

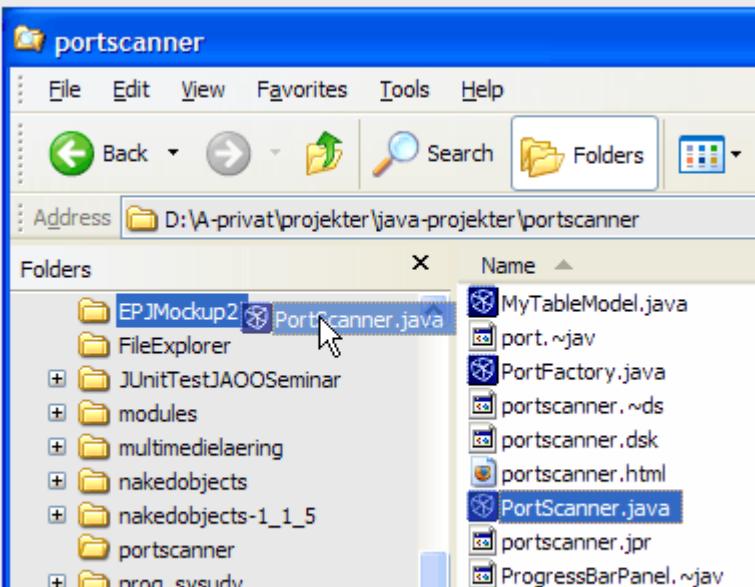
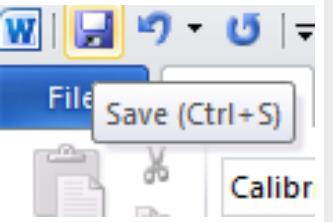
A possible classification:

- Menus ✓
- Direct manipulation ✓
- Fill-in-forms ↙
- Function keys
- Question and answer
- Command languages
- Natural languages
- ...



Often two or more styles are used simultaneously;

Why?



Fill in forms

Endereço <http://www.omega.com/cgi-win/cgw.cgi?ADD>

BUSINESS ADDRESS (Required)
denotes a required field in this business address block.

First Name
Last Name
Title
Company
Street Address
Department/Mail
Stop
City
State/Province 
Zip/Postal Code
USA/U.S. Military: Enter Zip +4 code without the hyphen
CANADA: Enter postal code per usual (e.g. A1B 2C)
E-mail Address
You may receive renewal reminders and other communications from Computer Graphics World magazine via e-mail. If you would like to receive correspondence from other PennWell publications, please check here.
You may receive subscription renewal notices via e-mail. If you would like to receive other business related third-party offers, please check here.

IDA

Origem	Estações	Tipo de Serviço	
<input type="text" value="Aveiro"/>	<input checked="" type="radio" value="Todos"/>	Todos	
Destino	Estações	<input type="radio" value="Alfa Pendular"/>	Alfa Pendular
<input type="text" value="Oriente"/>	<input type="radio" value="Intercidades"/>	Intercidades	
	<input type="radio" value="InterRegional"/>	InterRegional	
	<input type="radio" value="Regional"/>	Regional	
	<input type="radio" value="Urbano"/>	Urbano	

VOLTA

Data	<input type="text" value="2014-03-17"/> 	<input type="text"/> 		
Partida		pelas	<input type="checkbox"/>	Horas

OK

- Fill in forms are particularly useful for routine, clerical work or for tasks that require much data entry
- The concept already existed long ago
- Currently they are often used with other styles



```
PINE 3.96 ADDRESS BOOK (Edit)

Nickname : NBA
Fullname : Players in the NBA
Fcc :
Comment :
Addresses : mjordan@nba.com,
kmalone@nba.com,
drobinson@aol.com

^G Get Help ^X eXit/Save ^R RichView ^Y PrvPg/Top
^C Cancel ^U NxtPg/End
```

Main advantages and disadvantages

Advantages (potential)

- Self-explanatory
- Recognition instead of recall
- Allow many different inputs (unlike menus)
- Give context and guide the user
- New functionality is visible (unlike command languages)

Disadvantages

- Imply knowledge of valid inputs
- Error prone
- Not very flexible

Fill in form design: relevant aspects in design

- Organization and layout
- Titles and fields
- Input formats
- Instructions and help
- Navigation
- Error handling

Fill in form design: guidelines

Which is preferable?

Example:

Zip code:
Name:
Country:
Address:
City:

Better:

Name:
Address:
Zip code:
City:
Country:

Avoid unfamiliar layouts!

Provide a menu when possible inputs are known
(combining two interaction styles...)

Timetables and Prices

The image consists of four separate screenshots arranged in a grid:

- Top Left:** A search field for "Aveiro" with a location pin icon.
- Top Right:** A search field for "Lis" with a location pin icon, displaying a dropdown menu of Lisbon train stations: Lisboa - Cais do Sodré, Lisboa - Entrecampos, Lisboa - Oriente, Lisboa - Rossio, Lisboa - Santa Apolonia, and Lisboa - Sete Rios.
- Bottom Left:** A payment method selection field showing "Mastercard" selected from a dropdown menu, accompanied by the Mastercard logo.
- Bottom Right:** A detailed payment options form with sections for "Payment options", "Billing currency", "Card number", "Card type", "Card expiration date", "CVV2/CVC2 code", and "Card holder name". The "Payment options" dropdown is expanded to show Visa/MasterCard/Eurocard, PayPal, American Express, Bank/Wire transfer, Discover/Novus, Diners Club, JCB, and Fax.

Provide a format for fields that may be ambiguous

Show which fields are mandatory

Mbit.pt > Registo de Clientes

Área Cliente

Informação

Pesquisa

Top Vendas

Usualmente indicados por *

Username*

Password*

Repassword*

Name*

Email*

N.º de Contribuinte*

Morada*

Código Postal* -

Telefone*

Fax

Telemóvel

Data de Nascimento*

Registrar

••• Voltar

Nome do utilizador:

Password:

OK

Registrar

Recuperar Password

13 Anos de Experiência, 14 Lojas para o servir!

Loja 1 - Porto Torrinha

OK

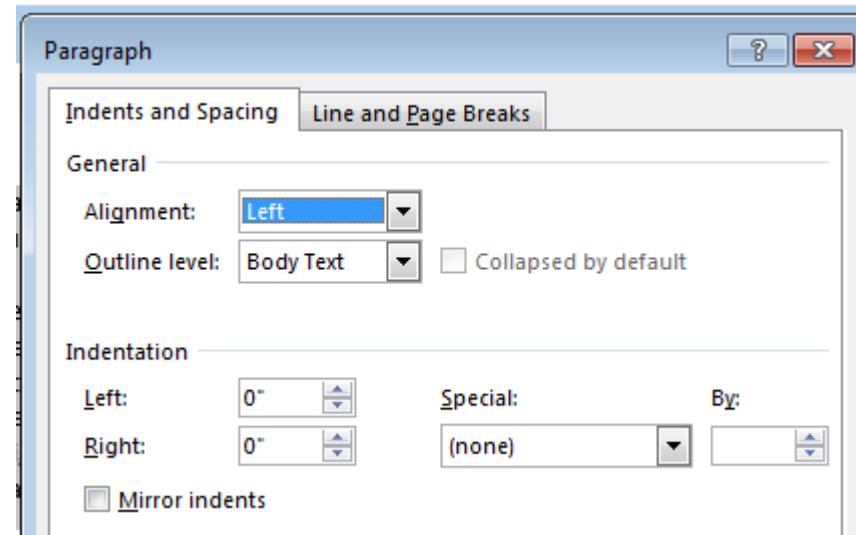
Usually indicated by *

It should be possible for the user to choose the type of input (it prevents errors) or adapt to the context

Portuguese version (cm):



English version (inches):



Instructions to fill the fields should be clear as well as messages

Messages

Headers: Show brief headers on incoming messages (recommended) Show all headers on incoming messages

Font Size:

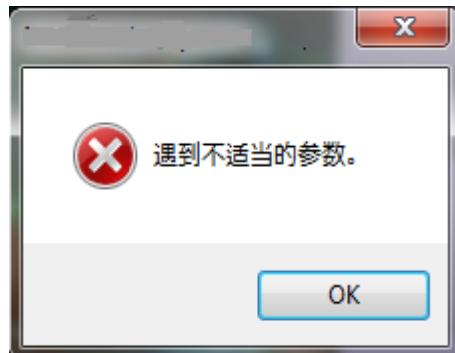
(plain text only)

Screen Width: characters (range: 50 - 99 chars.)
(viewing plain text mail)
This is the maximum line length of your incoming messages.
The default value is 72.

Screen Width: characters (range: 50 - 99 chars.)
(composing plain text mail)
This is the maximum line length of your outgoing messages. The default value is 55.

Security: Block HTML graphics in email messages from being downloaded [What's This?] Warn me about sending information outside Yahoo!

This message did not help me much...

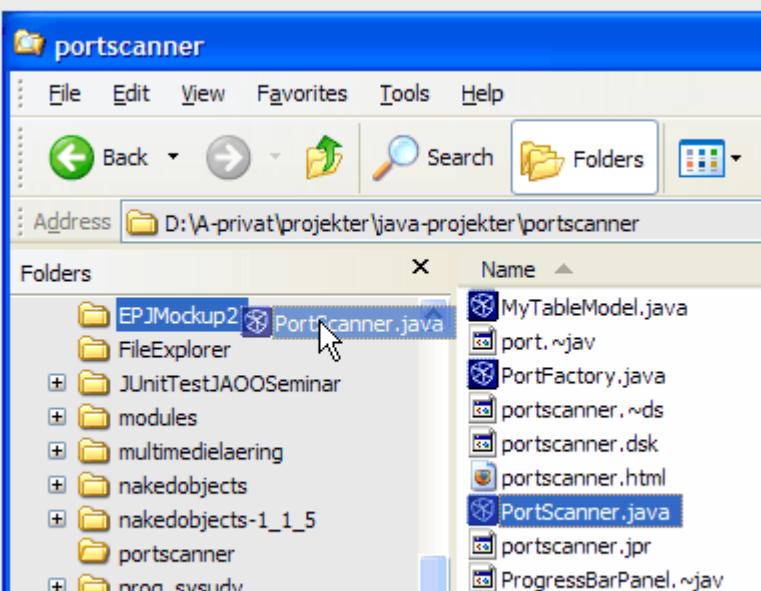
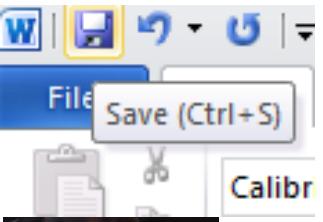
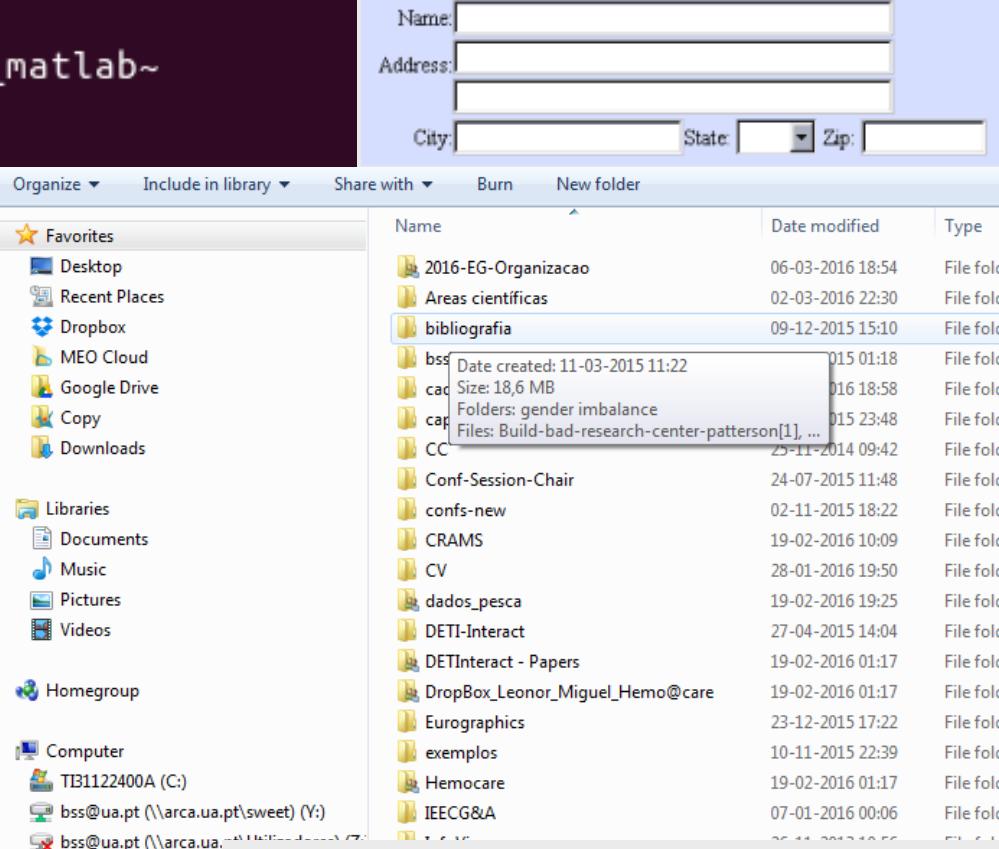
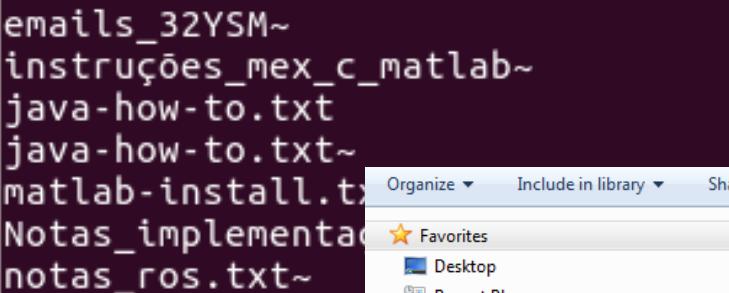


Interaction styles

A possible classification:

- Menus
- Direct manipulation
- Fill-in-forms
- Function keys ←
- Question and answer
- Command languages
- Natural languages

Often two or more styles are used simultaneously



Function keys

- Two types:
 - **Hard Keys** – Always invoke the same functionality (as the keys of a calculator and some specific keys of PCs)
 - **Soft Keys** – invoke different functionality according the context of use (as the keys (F1...Fn) and the generic keys of an Automated Telling Machine, e.g. Multibanco)
- PCs have 12 generic Keys (F1 a F12) and a few other specific keys



Keys that invoke specific functionality in PCs and MACs

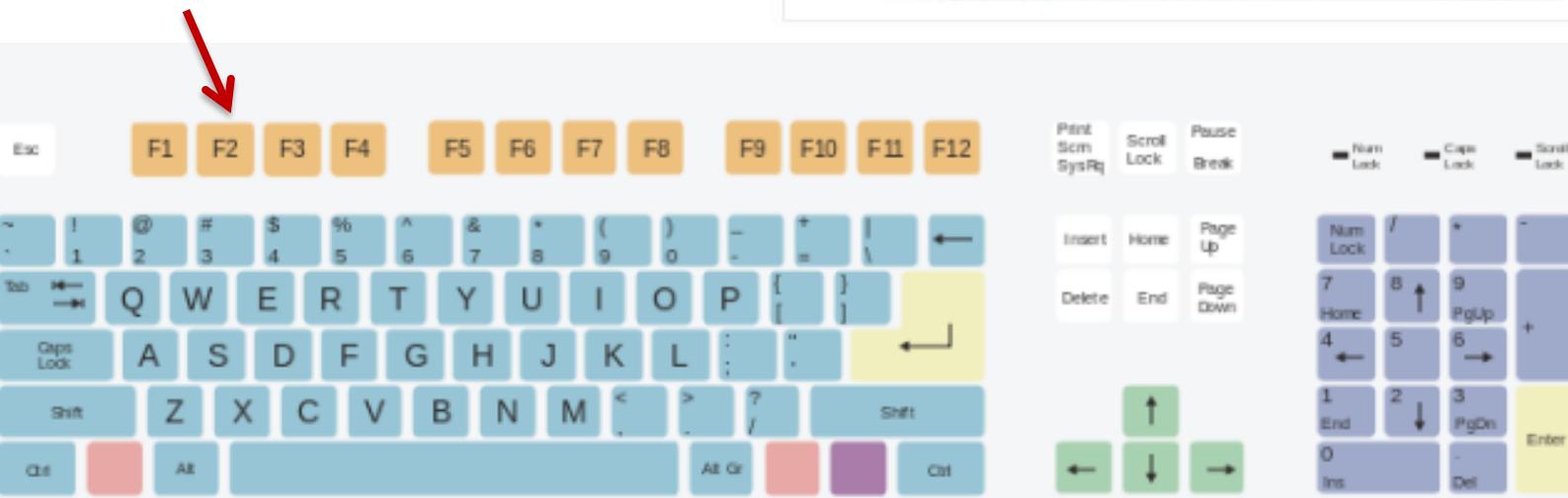


Soft Keys

Soft function keys don't have abbreviations of default actions printed on/besides them, they may have "F-number" designations.



Function keys (generic)



https://en.wikipedia.org/wiki/Function_key

Main advantages and disadvantages

Advantages (potential)

- Self-explanatory
- Recognition instead of recall
- Easy to use
- Flexible
- Require little or no screen real estate

Disadvantages

- Limited number of keys
- Hardware expansions are expensive

Function keys design: guidelines

Provide enough keys to call the functionality

But no too many as not to make it difficult to learn

Use:

- free space
- different size, color and shape to different groups
- category groups
- clear and distinctive names



TV remote control

Multi-media remote control keyboard



ATM keyboard



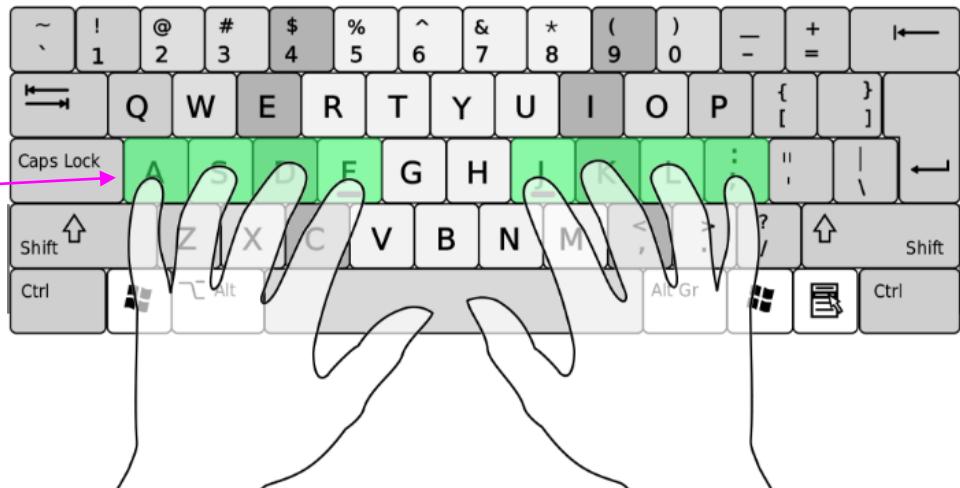
Industrial keyboard



Shop system keyboard



Frequently used keys
should be near the “home
row”



Keys with serious
consequences should not
be easy to activate

(e.g. ctrl Alt Del)

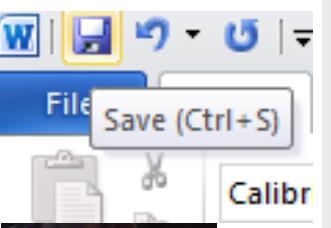
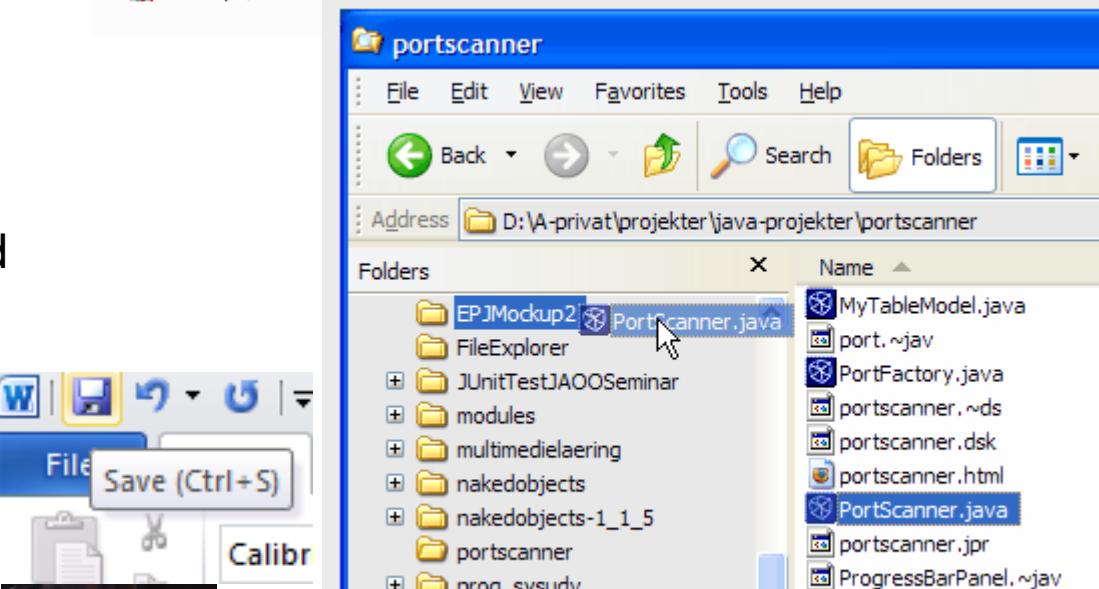
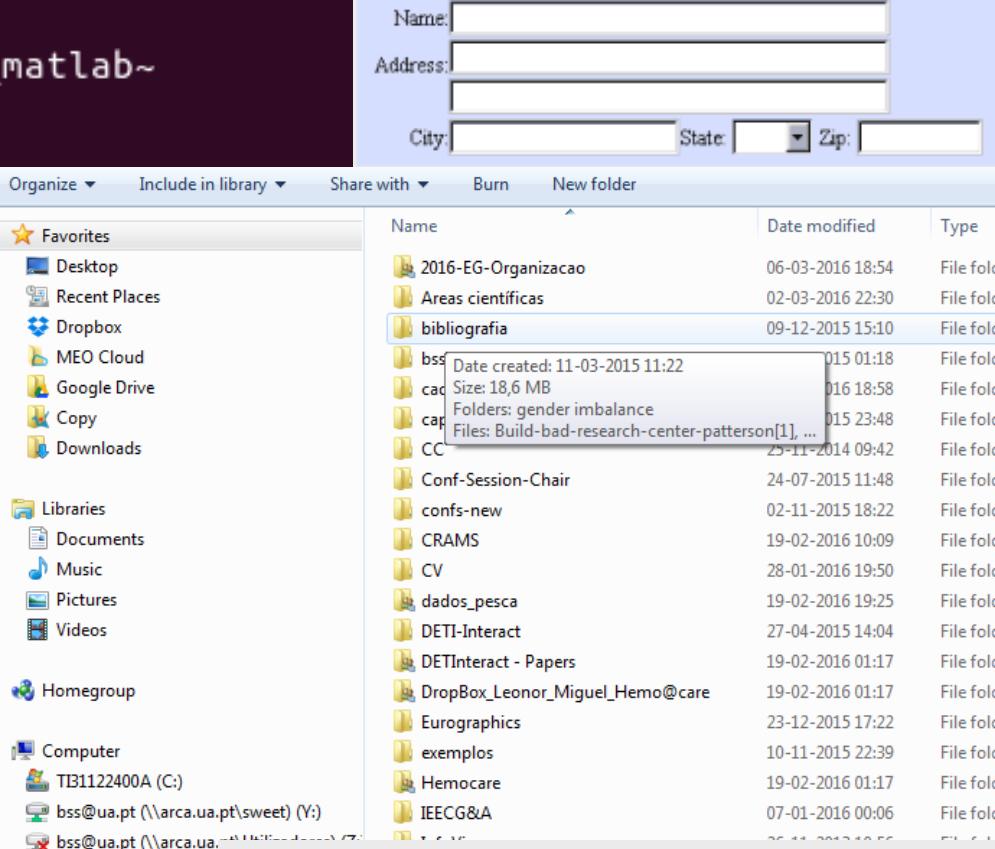
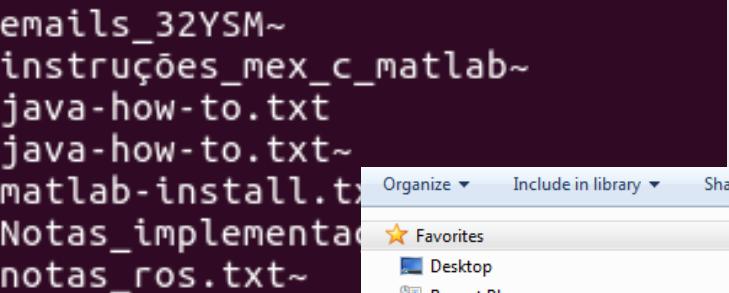


Interaction styles

A possible classification:

- Menus
- Direct manipulation
- Fill-in-forms
- Function keys
- Question and answer
- Command languages 
- Natural languages

...
Often two or more styles are used simultaneously



Command languages

```
cd /tmp
echo "line 1
line 2
line 4" > tmp1$$
echo "line 2
line 3" > tmp2$$
diff tmp1$$ tmp2$$
rm tmp1$$ tmp2$$
```

```
guru99@VirtualBox:~$ history
 1  cat > sample
 2  cat sample
 3  cat sample ^a
 4  cat sample a
 5  cat sample | grep a
 6  cat sample | grep ^a
 7  useradd home
 8  useradd mycomputer
 9  sudo useradd mycomputer
10  sudo adduser MyLinux
11  sudo adduser mylinux
12  vi scriptsample.sh
```

Command languages shall also be **designed as to be as usable as possible**

Basic Goals of Language Design

- Precision
- Compactness
- Ease in writing and reading
- Speed in learning
- Simplicity to reduce errors
- Ease of retention over time

Usability Questions concerning a command language

- Does the language support necessary functions?
- Is it fast to enter a command?
- Is it easy to recognize what the command might do?
- Is it easy to recall a command?
- Are there few errors when using the language?

Main advantages and disadvantages

Advantages (potential)

- Powerful
- Flexible
- Efficient
- Do not take much screen real estate

Disadvantages

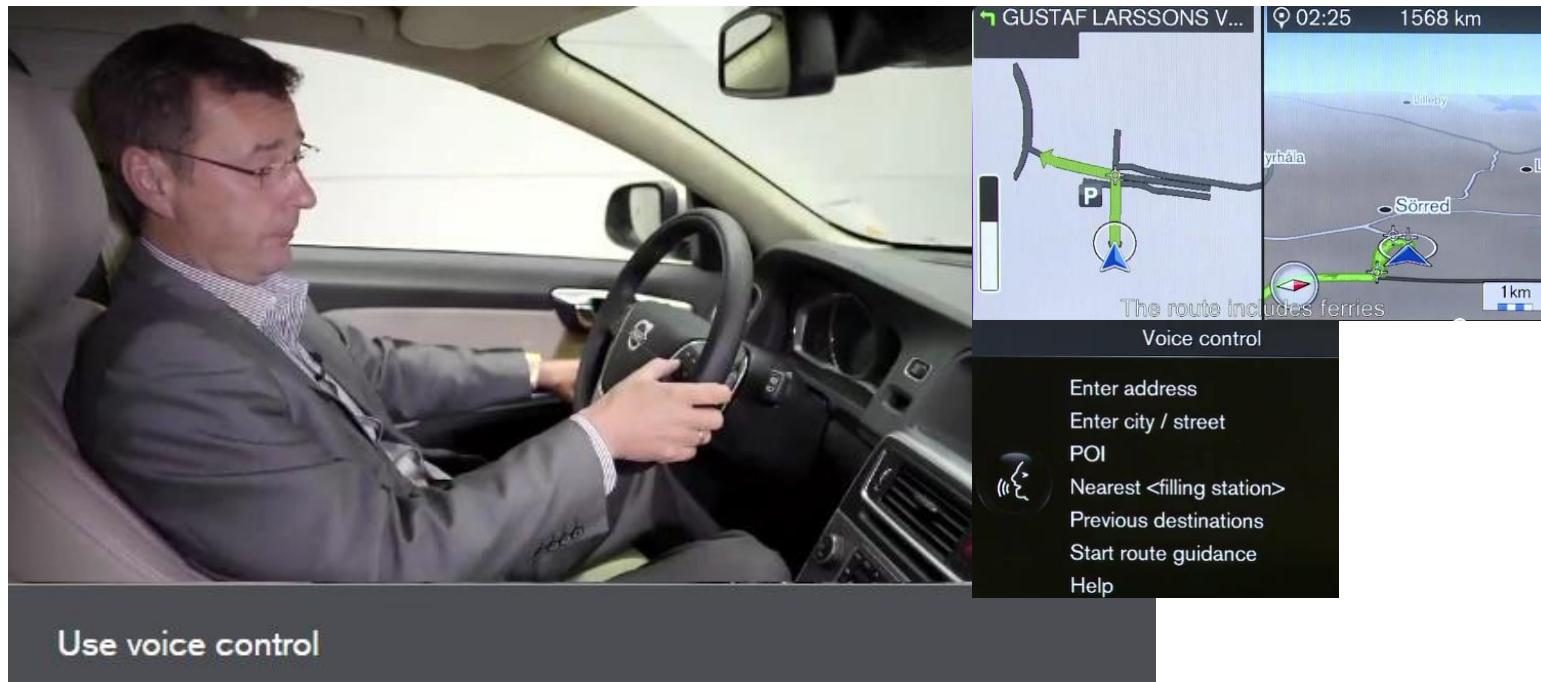
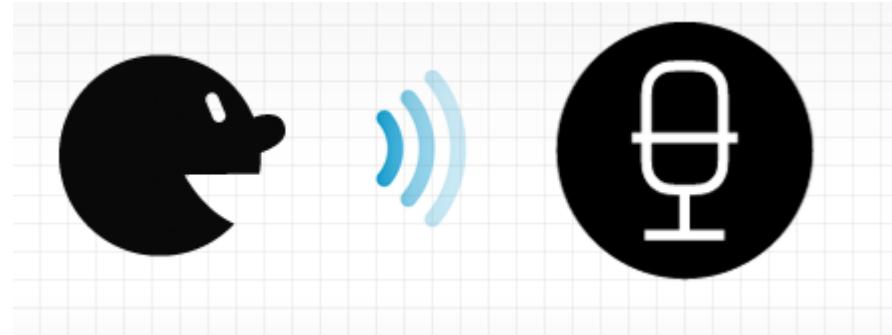
- Difficult to learn
- Not self-explainable
- Error prone
- Improvements are not visible

Note that:

Command languages may be used
not only through text but also via voice
But they must be very simple ...

e.g.

While driving a car to control the media, the phone or navigate



Interaction style: command language
interaction devices: speech recognition/synthesis

Relevant issues in Command Language design

- Semantics
- Syntax
- Lexicon
- Interaction

Use a coherent syntax

Use a natural and easy to remember action-object grammar

VolB!FileA!D\$\$
FileA!VolB!ER\$L!:KO!*\$\$

Uncoherent syntax and unfamiliar commands

search filea volb.
open filea volb.
list all lines with "KO".

or

s filea volb.
o filea volb.
lal "KO".

Command abbreviations should be simple and coherent
Easy to remember (not easy to recognize as for function keys)

Name	Abbreviations	
	Poor:	Improved:
Move forward	MovF	MovF
Move backward	Mvb	MovB
Insert	I	Ins
Delete	Dl	Del
Replace	Repl	Rep
Search	Srch	Sea
Delete	X	Del
Send	Sn	Sen
Print	Prt	Pri
Search	Srch	Sea
Send	Sn	Sen
Find	Fi	Fin
Choose	Ch	Cho

Allow the following interaction features:

- Defaults
- Command edition
- Intelligent interpretation
- Type-ahead
- Feedback
- Help and documentation
- Make the language “user tailorable”

Example of intelligent interpretation:

“delate”: did you mean “delete”? Y or N

Example of a (complex) command with defaults

ls - Linux man page

Name

ls - list directory contents

Synopsis

ls [*OPTION*]... [*FILE*]...

Description

List information about the *FILE*s (the current directory by default). Sort entries alphabetically if none of **-cftuvSUX** nor **--sort**.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
do not ignore entries starting with `.`

-A, --almost-all
do not list implied `.` and `..`

--author
with `-l`, print the author of each file

-b, --escape
print octal escapes for nongraphic characters

You don't need to use all arguments;
there are default values

-d, --directory

list directory entries instead of contents, and do not dereference symbols

-D, --dirent

generate output designed for Emacs' dirent mode

-f

do not sort, enable **-aU**, disable **-ls --color**

-F, --classify

append indicator (one of `*/=>@|`) to entries

--file-type

likewise, except do not append `**`

--format=WORD

across **-x**, commas **-m**, horizontal **-x**, long **-l**, single-column **-1**, verbose

--full-time

like **-l --time-style=full-iso**

-g

like **-l**, but do not list owner

--group-directories-first

group directories before files.

augment with a **--sort** option, but any

use of **--sort=none (-U)** disables grouping

-G, --no-group

in a long listing, don't print group names

-h, --human-readable

with `-l`, print sizes in human readable format (e.g., 1K 234M 2G)

--si

likewise, but use powers of 1000 not 1024

-H, --dereference-command-line

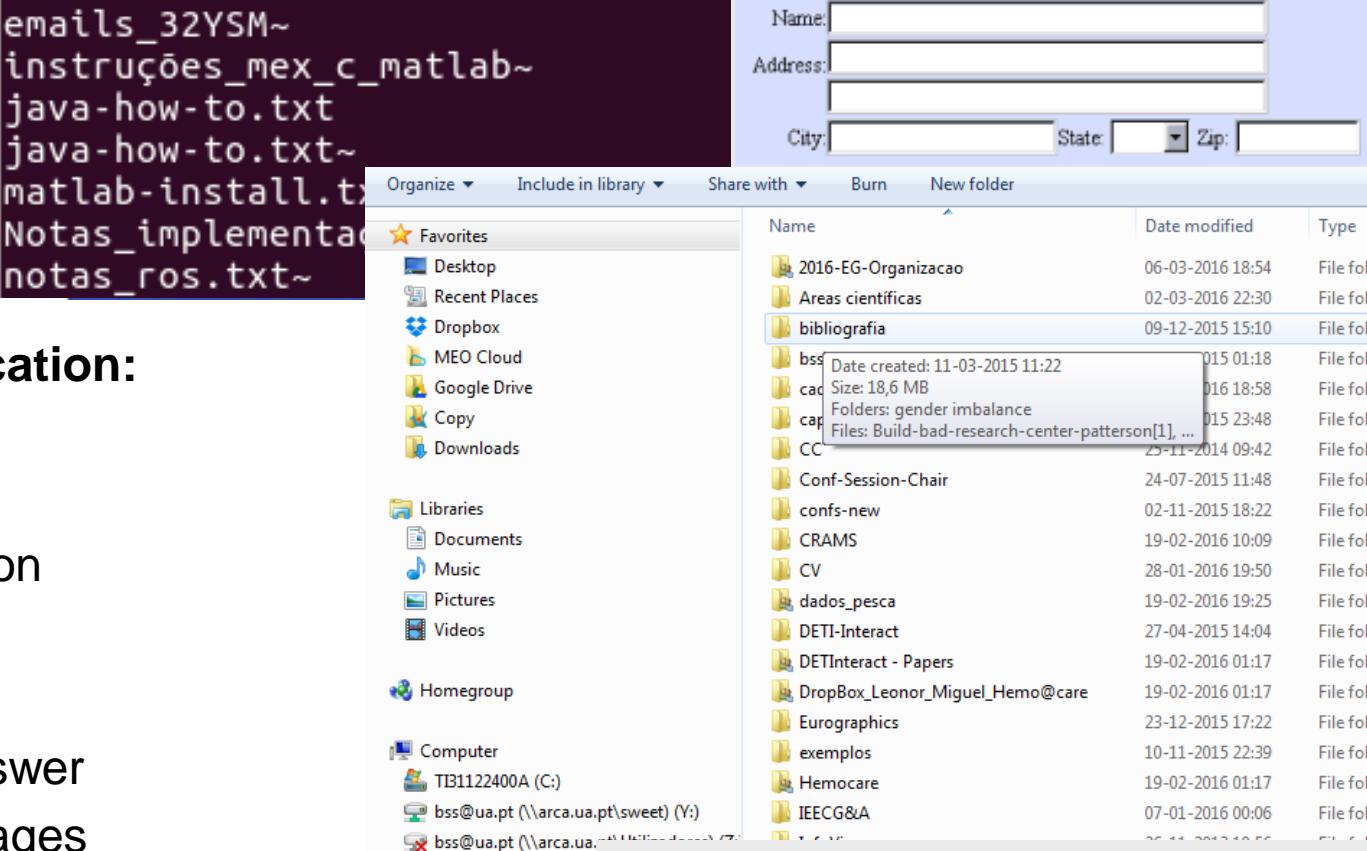
follow symbolic links listed on the command line

Etc., etc., etc.

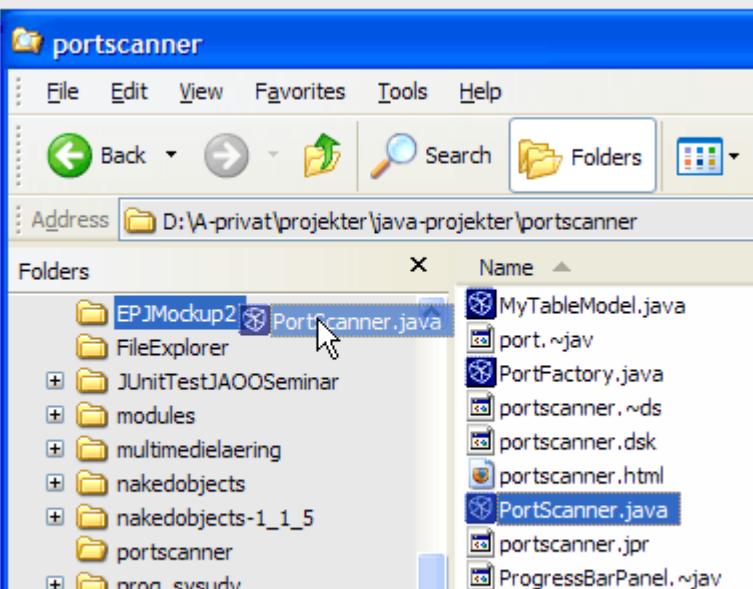
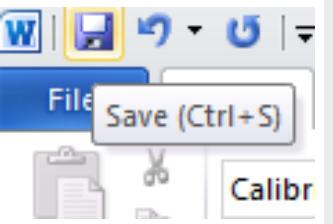
Interaction styles

A possible classification:

- Menus
- Direct manipulation
- Fill-in-forms
- Function keys
- Question and answer
- Command languages
- Natural languages 



Often two or more styles are used simultaneously



Natural language

- Communication between humans and computers through natural language involves:
 - recognition
 - generation
- Natural language processing (NLP) has been evolving a lot ...

Note:

**natural language as a interaction style and voice interaction
are different things!**

Conversational User interfaces (CUIs)

Based on natural language

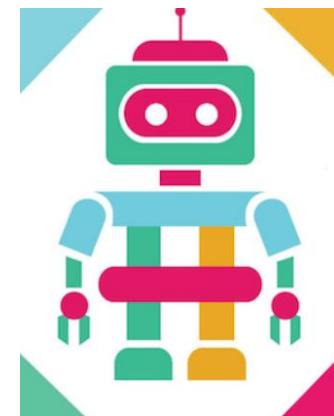
Think of the potential advantages and disadvantages of CUIs:

- Chatbots

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

- Voice assistants

“Just like the touch interface, not everything will become conversational”



What doesn't fit the principles of Conversational UI well?

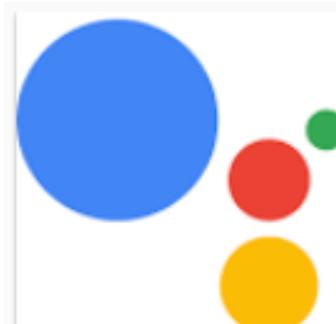
Products where the use case involves a technical user who wants fine grain control over the interface, e.g. CAD software, or a programming IDE....”

<https://uxdesign.cc/conversational-ui-its-not-just-chat-bots-and-voice-assistants-case-study-cb1865da306a> 39

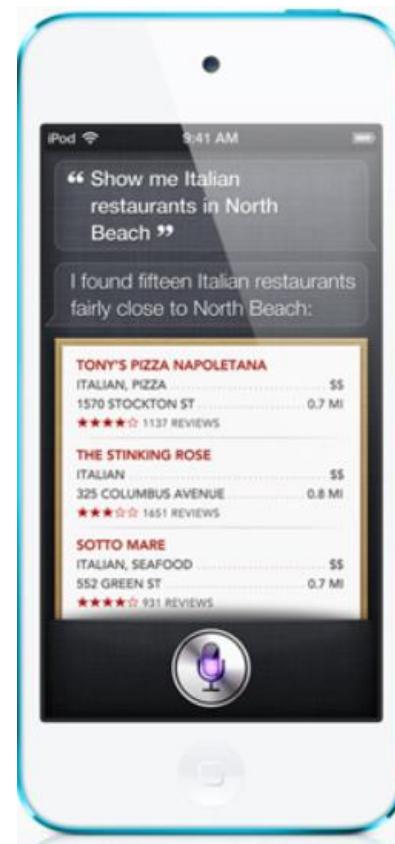
Current examples of Natural language interaction (mostly via voice)

Mobile phone personal assistants:

- Siri for Apple's iOS
- Google assistant

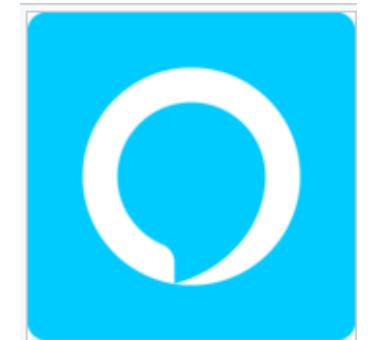


Google
Assistant



Another example (natural language via voice)

Interaction style: natural language;
interaction devices: speech recognition/synthesis

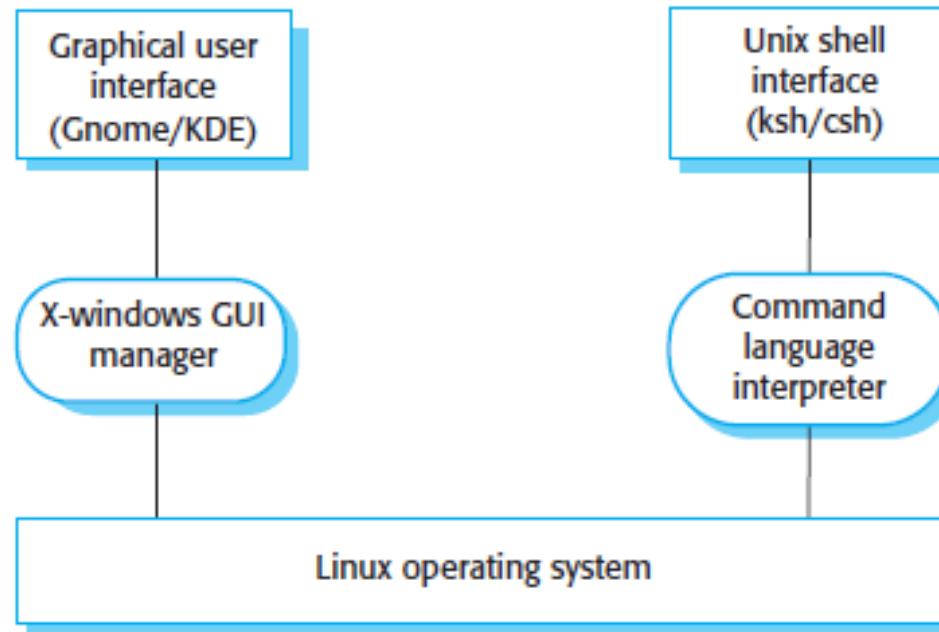


<https://www.nngroup.com/articles/voice-interaction-ux/>

Main advantages and disadvantages of interaction styles

Interaction style	Main advantages	Main disadvantages	Application examples
Direct manipulation	Fast and intuitive interaction Easy to learn	May be hard to implement Only suitable where there is a visual metaphor for tasks and objects	Video games CAD systems
Menu selection	Avoids user error Little typing required	Slow for experienced users Can become complex if many menu options	Most general-purpose systems
Form fill-in	Simple data entry Easy to learn Checkable	Takes up a lot of screen space Causes problems where user options do not match the form fields	Stock control Personal loan processing
Command language	Powerful and flexible	Hard to learn Poor error management	Operating systems Command and control systems
Natural language	Accessible to casual users Easily extended	Requires more typing Natural language understanding systems are unreliable	Information retrieval systems

Multiple user interfaces example



(Sommerville, 2010, chap.29)

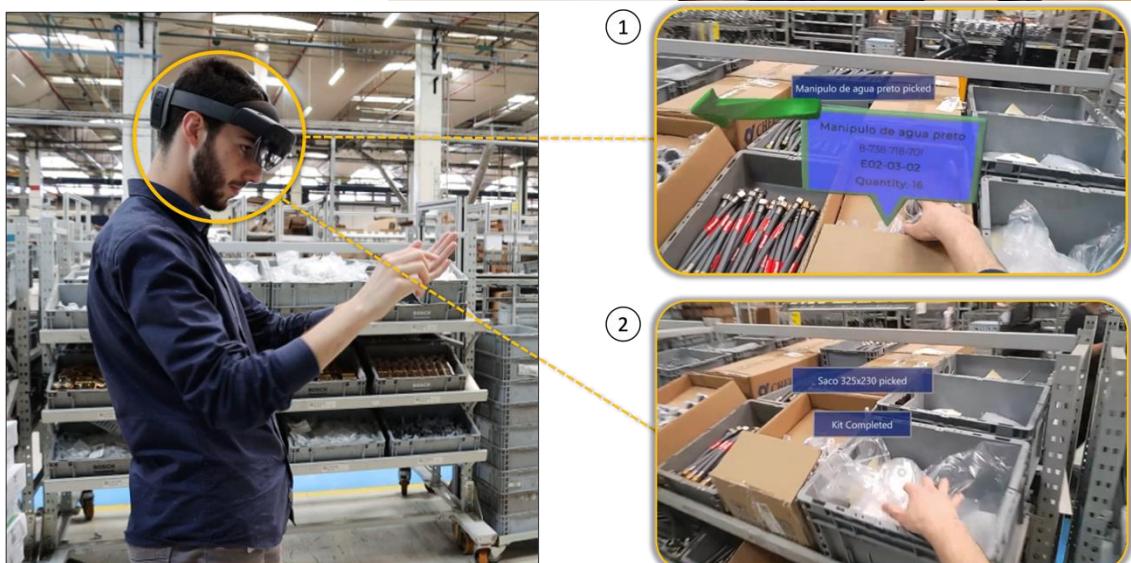
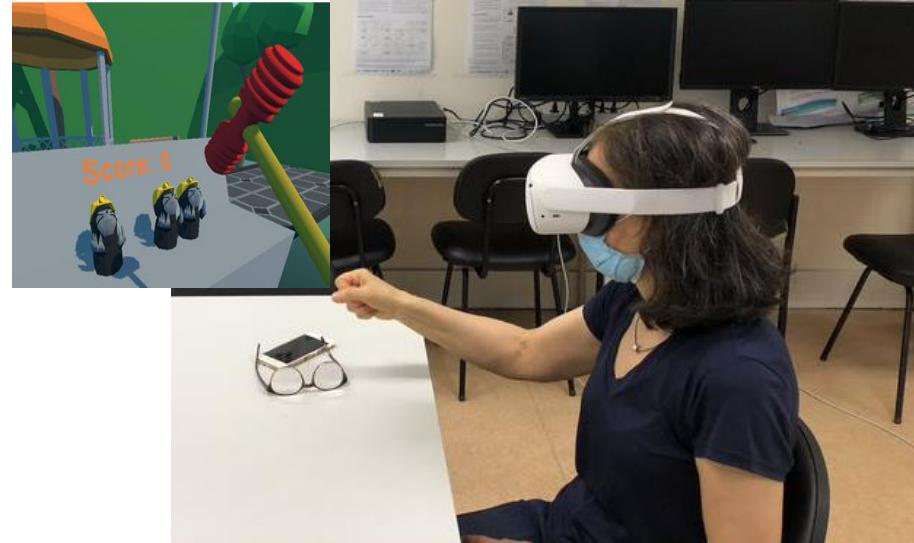
3D User Interfaces

- User interfaces involving 3D interaction (i.e. interaction in which the user's tasks are performed directly in a 3D spatial context).
- Are more and more used:
 - Virtual and augmented reality
 - 3D workspaces
 - Data Visualization ...
- But have some issues:
 - User disorientation, ...
(in the real world we have more information)



Applications of virtual and augmented reality (eXtended Reality):

- Training and simulation
- Assistance in tasks
- Project review
- Therapy
- Experiments
- Entertainment
- ...

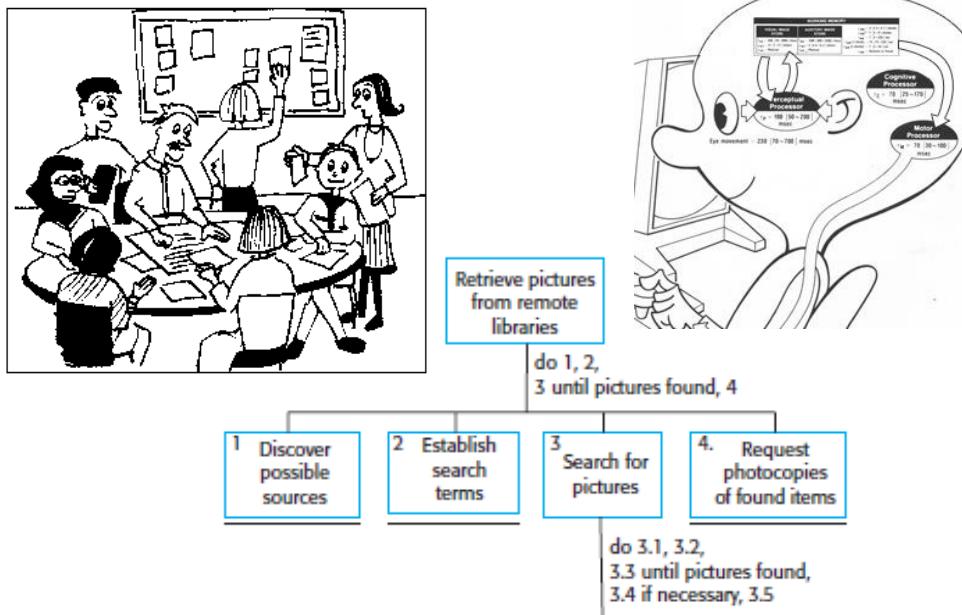


Main bibliography

- Soegaard, Mads. Interaction Styles
http://www.interactiondesign.org/encyclopedia/interaction_styles.html
<https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/3d-user-interfaces>
- Ian Sommerville, Software Engineering, 9 ed, Addison Wesley , 2010
https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/WebChapters/PDF/Ch_29%20Interaction_design.pdf



Other methods for design and evaluation

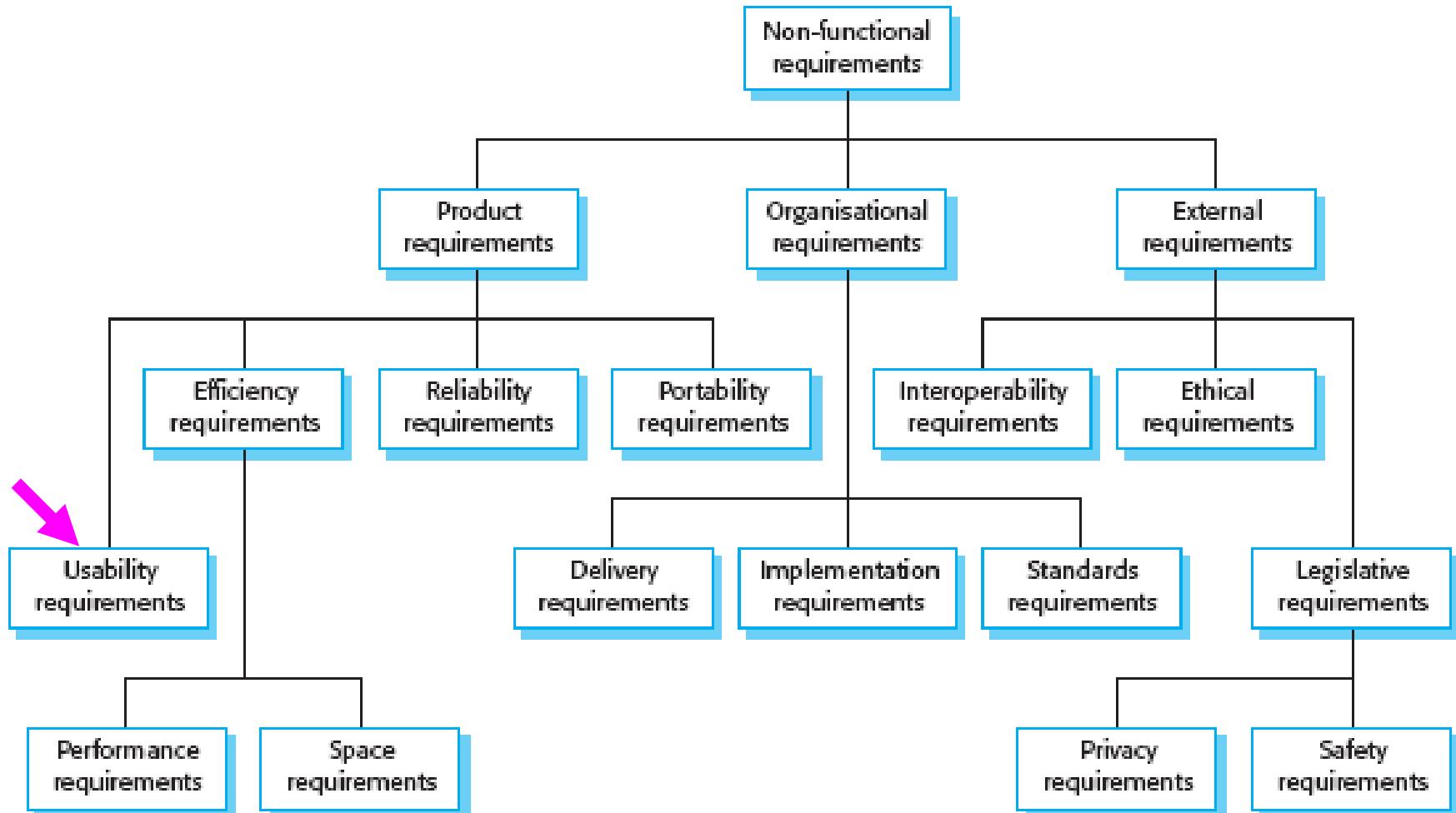


- All engineering fields use models:
 - To prescribe – directly contribute to the design
 - To evaluate – does the design have the needed characteristics?
- Models are needed also in Interactive systems design (you already know and have applied a few...)
- Several types of models may be used throughout the design of interactive systems and user interfaces:
 - User models
 - Task analysis
 - Dialog notation
 - ...

Models to obtain user requirements

- Obtaining user requirements is very important in S/W engineering
- But often focuses on functional requirements: what the system must do
- Overlooking non-functional requirements, as if it is:
 - acceptable
 - usable
- There are several models to capture user requirements

Non-functional requirements



(Sommerville, 2016)

Models to obtain user requirements

- *Socio-technical models* - acknowledge that technology is used within an organization
- *Soft System methodologies*- put the emphasis on the understanding of the situation and not on getting a solution
- *Participatory Design* – encompasses the complete development cycle and includes users as active project team members, not only as evaluation participants

- Palestra sobre React
- 16h30; Anf. IV

Participatory design

- Users are involved as domain experts (e.g. business representatives and users) along the complete process and work together with developers to design a solution
 - It is work oriented and not system oriented
 - It is collaborative- users contribute to all phases
 - It is iterative – design is evaluated and reviewed in every phase

Participatory design

- Uses a set of techniques (that can be used in other contexts) to help transfer information from users to designers:
 - Brainstorming
 - Scenarios
 - Story boarding
 - Workshops
 - Paper and pencil exercises ...

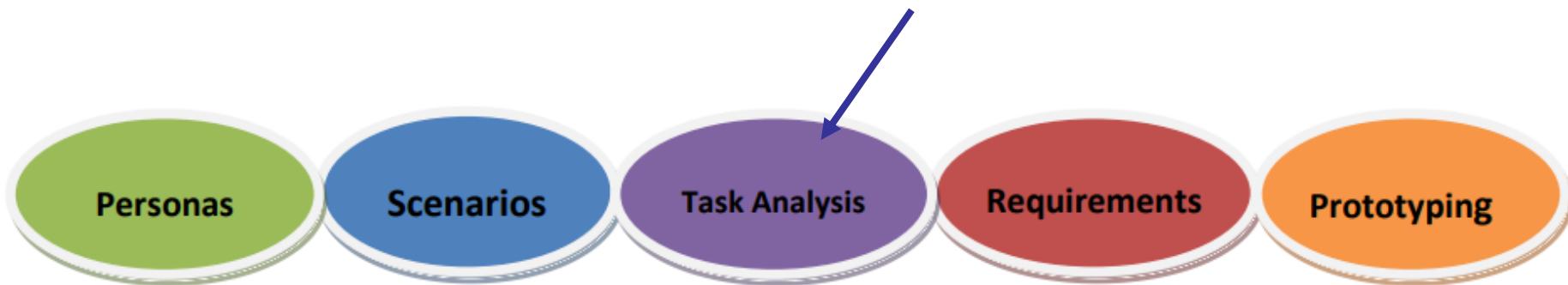


<https://www.usabilitybok.org/participatory-design>

<https://www.usabilityfirst.com/usability-methods/participatory-design/index.html>

Task analysis

- What it is and how it can be performed (more or less formally)
(Note that in Lab classes we started by doing it in an informal way...)
- Techniques
- Sources of information



<https://www.usabilitybok.org/hierarchical-task-analysis>

<https://digital.ahrq.gov/health-it-tools-and-resources/evaluation-resources/workflow-assessment-health-it-toolkit/all-workflow-tools/hierarchical-task-analysis>

Task analysis

- It is the analysis of how people perform their work
 - what they do
 - what they use
 - what they need to know
 - Daily-life example: vacuum cleaning a house
 - Get the vacuum cleaner
 - Choose the adequate attachment
 - Clean the rooms
 - Empty the bag when it is full
 - Put the vacuum cleaner and attachments away
- Users have to know about vacuum cleaners, rooms, ...

Task Analysis

- Observation (of various types) is a fundamental tool
- It can be used to:
 - produce documentation and training materials
(the observation of how existing systems are used is enough)
 - design new systems
(work-re-engineering is usually necessary)

Task Analysis approaches

- There are a lot of approaches and methods; we have used an informal way in the practical classes, but there are more formal methods
 - Task decomposition ✓
 - Knowledge based
 - Relation and entities based

- We have used an informal way in the practical classes, but there are more formal methods
 - Task decomposition - divides tasks into subtasks that must be performed in a specific sequence
 - Knowledge based – considers what users need to know about the objects and actions involved in performing the task and how knowledge is organized
 - Relation based - is focused on actors and objects, relations among them and the actions they perform

Task Decomposition

- Hierarchical Task Analysis (HTA) is one of the most used task analysis techniques and produces:
 - a task and sub-task hierarchy
 - plans with a sequence and execution conditions

Simple daily life example: vacuum cleaning the house:

0. in order to clean the house
 1. get the vacuum cleaner
 2. fix the appropriate attachment
 3. clean the rooms
 - 3.1. clean the hall
 - 3.2. clean the bedrooms
 4. empty the dust bag
 5. put the vacuum cleaner and attachments away

Plan 0: do 1 – 2 – 3 – 5 in that order
when the dust bag gets full do 4

Plan 3: do any of 3.1, 3.2, or 3.3 in any order depending on which rooms need cleaning

Plan 3 could be more specific; what if it were varnishing the house?

- Where should the decomposition stop?
- The decomposition detail depends on the goals of each task analysis
- Example: in a factory what should be done in an emergency
 - 0. in a emergency
 - 1. Read the alarms
 - 2. Determine the corrective actions
 - 3. Execute the correction actions
- If the goal is
 - installing a monitoring system → expand 1 e 3
 - produce operation manuals → expand 2

- Where should the decomposition stop?
- A stop decomposition rule:

Stop if $P \times C <$ a specific value

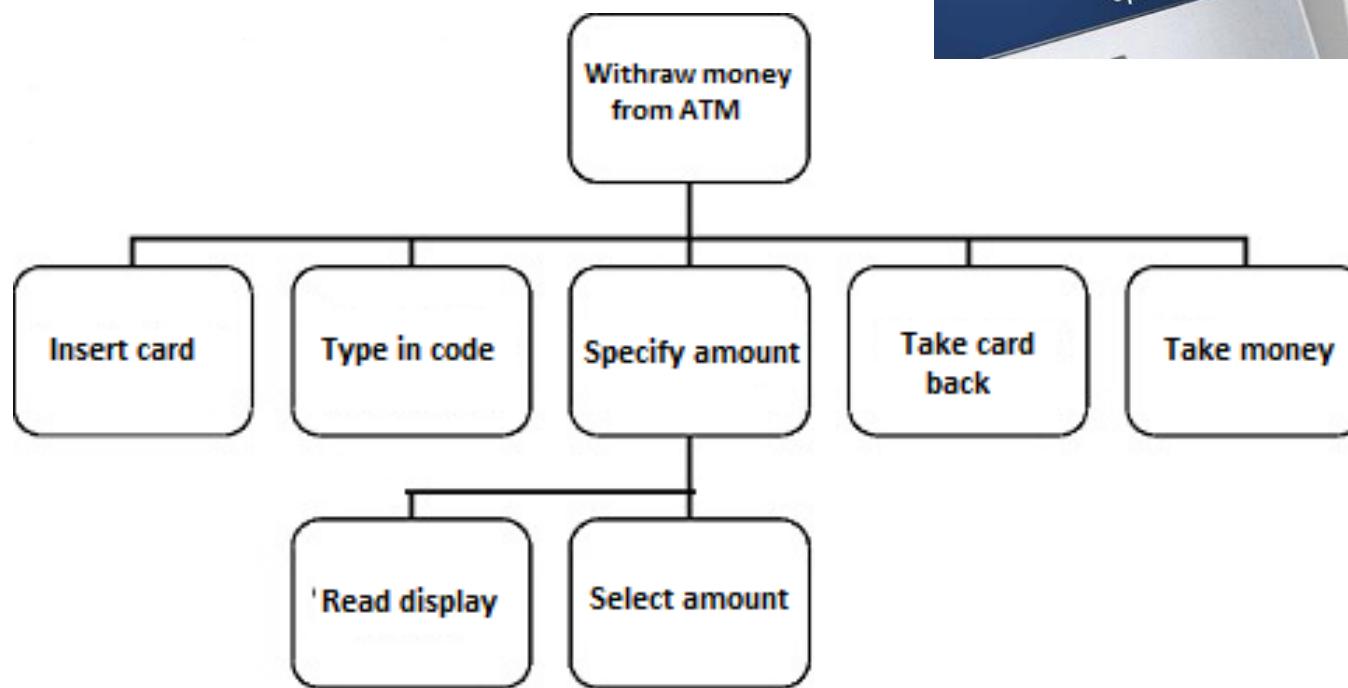
Probability of making an error

error cost

i.e.: simple tasks do not need decomposition unless they are critical!

Incomplete decomposition of the task:
Withdraw money from an ATM

Can you fix and complete it?



Another daily life example:

Preparing a cup of tea

Can you do a HTA describing this task?



HTA- Preparing a cup of tea

(graphical representation
of first approach)



0.
make a
cup of tea

Plan 0
do 1
At the same time, if the pot is full 2
Then 3 – 4
After four or five minutes do 6

I.
boil water

2.
empty pot

3.
put tea leaves
in pot

4.
pour in
boiling water

5.
wait 4 or 5
minutes

6.
pour tea

Plan I
I.1 – I.2 – I.3
When kettle boils I.4

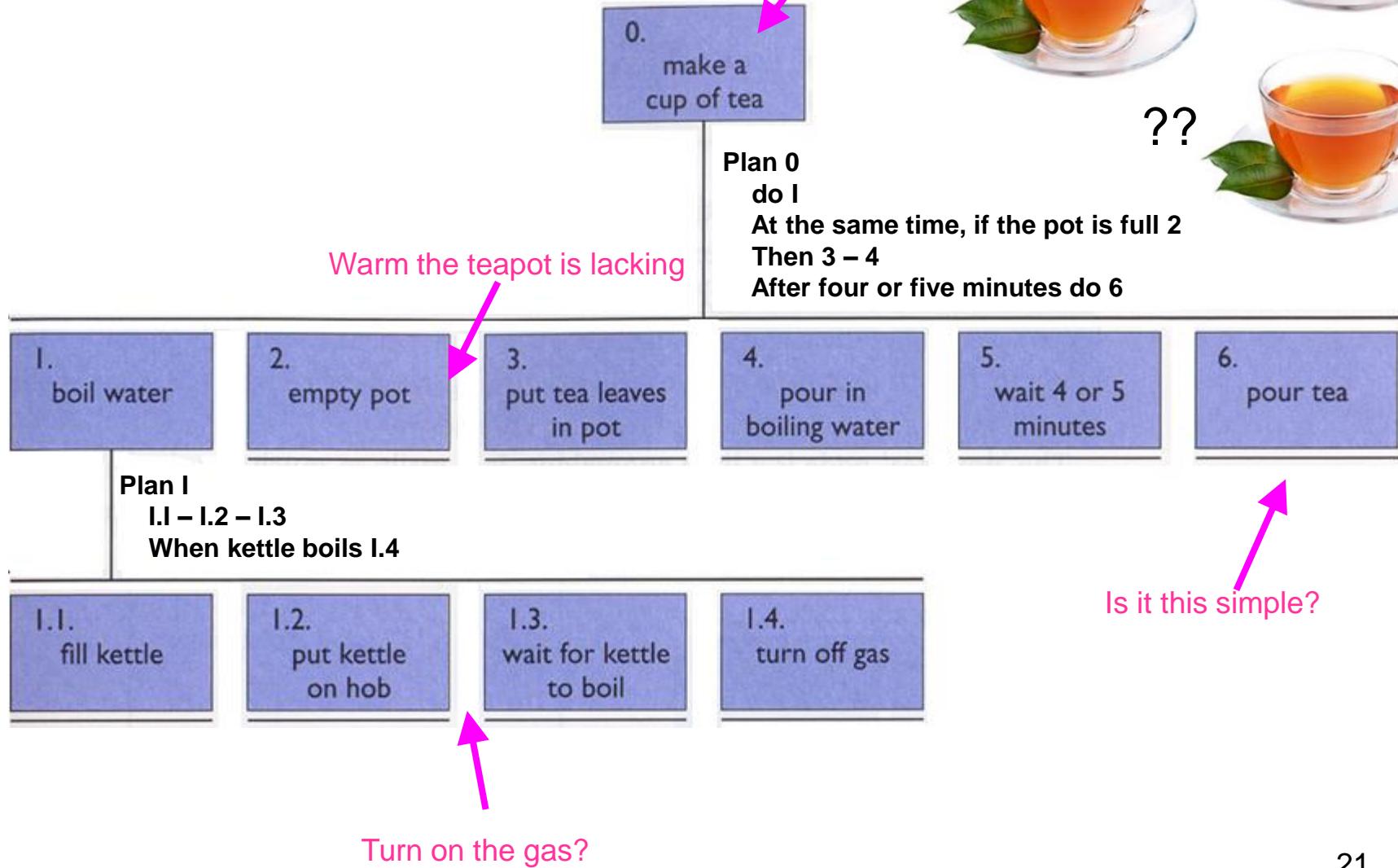
I.1.
fill kettle

I.2.
put kettle
on hob

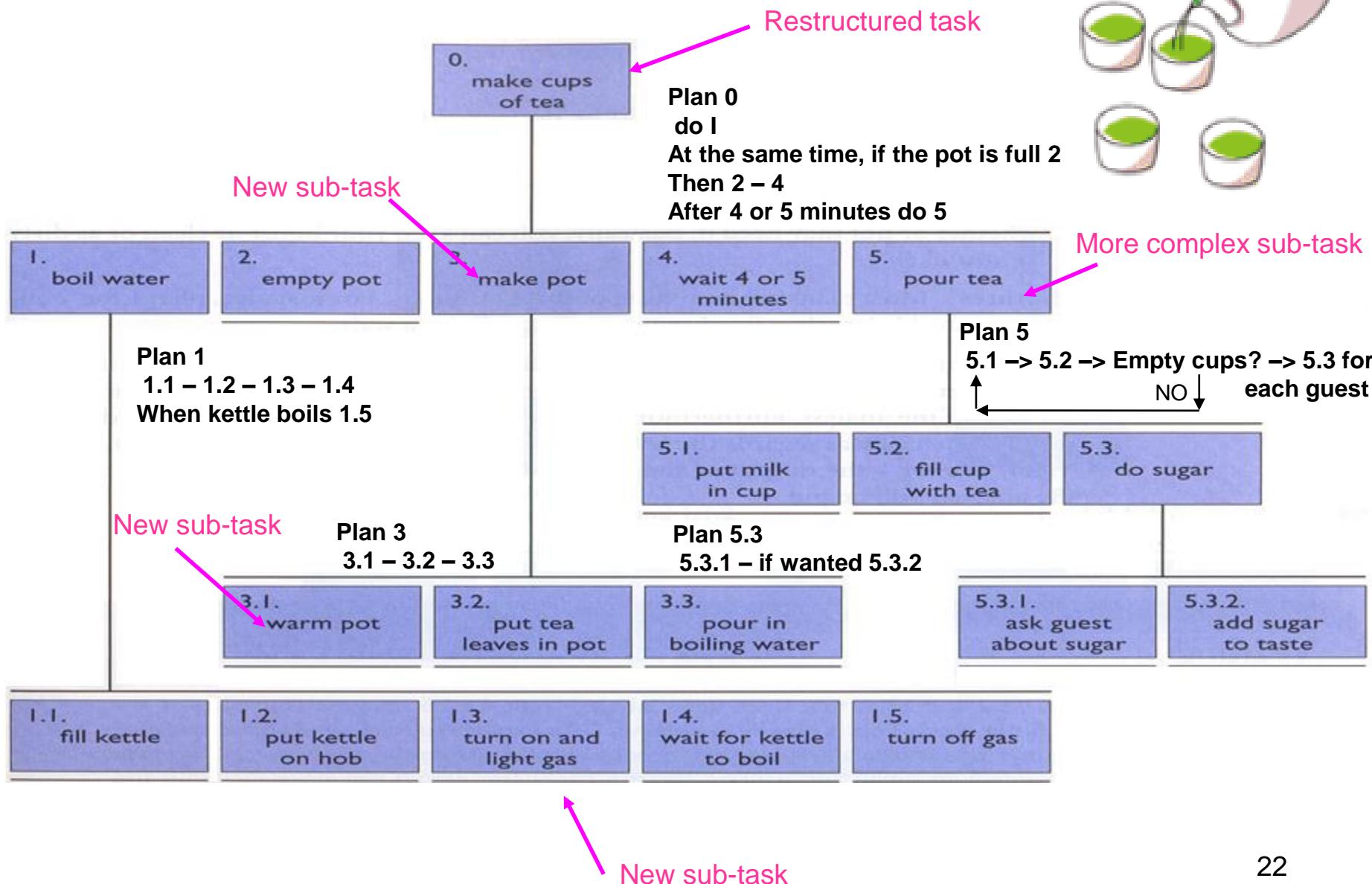
I.3.
wait for kettle
to boil

I.4.
turn off gas

HTA- Preparing a cup of tea (analysis of the first approach)



HTA- Prepare several cups of tea (a new hierarchy)



Plan types in HTA

- Fixed sequence (plan 3 – prepare the teapot)
- Optional tasks (5.3 sugar?)
- Waiting for events (4- wait 4 or 5 minutes)
- Cycles (plan 5 – serve tea)
- Time sharing (1 and 2 prepare teapot, boil water)
- Random (vacuum cleaning rooms)
- Mix of several types

- The result of the analysis depends a lot on the experience of the analyst
- **Different analysts usually produce different results** (mainly at the detail level) varying with the goal of the analyst

Task analysis information sources

- The quality of task analysis results cannot be better than the original data
“garbage in garbage out”
- The process of analysis in general triggers new questions, thus several phases of data collection and analysis are needed
- There are several types of information sources:
 - Documentation
 - Observation → (expensive)
 - Interviews



Documentation:

- Manuals, instruction books, training documentation ... are very good information sources
- But they describe what people are supposed to do, not what they actually do
- System's manuals usually describe functionality, not how they are used
- Observation and user interviews should be performed based on this information
- Be careful with user interviews!!

Interviews:

- Interviewing domain experts is a good way of getting information about the task; should include:
 - General questions (e.g. a typical day)
 - Specific questions (e.g. why did you do that?)
 - Task decomposition (~ HTA)

Observation:

- It is always necessary to perform (formal or informal) observation to understand the tasks
- Reading documentation and observing users is a good starting point
- More observation should follow:
 - In the lab
 - In the field
 - Passive (only observation)
 - Active (questions, post-task walkthrough)

Using Task Analysis:

- May be used in:
 - Manuals and teaching materials
 - High-level system design
 - Detailed design of the system user interface
- In the first case users are observed while performing tasks using the system
- In the other cases task analysis contributes to the design of the new system

Manuals and teaching materials

- The first task analysis techniques were developed to train people to perform a task
 - (e.g.: clean a gun)
- HTA may be used to structure manuals or tutorials
 - (e.g: how to prepare tea)
- May be used also to help users transfer from one system to another

Obtaining requirements

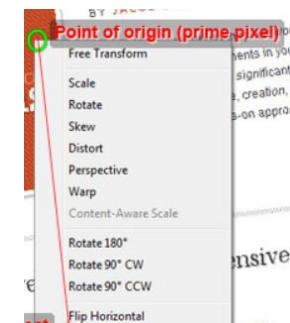
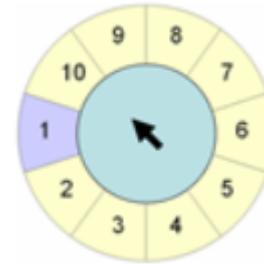
- TA is not a direct way of obtaining a system requirements as it refers to an already existing system and not to the new one
- However, it may give a good contribution
- In general the new system includes old and new functionality
- TA may help determine:
 - What objects, tasks, etc., should be maintained
 - What should be included

II- User Models

- User models of the users' mental, perceptual and motor processes (GOMS, KLM...)
- Personas- fictional characters based upon research in order to represent the different types of users
- ...

<https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/personas>

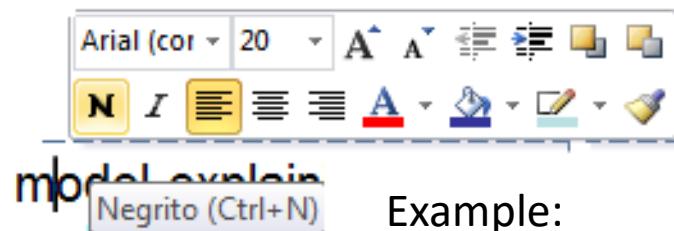
Fitts's law



- Empirical model explaining speed-accuracy tradeoff characteristics of human muscle movement with some analogy to Shannon's channel capacity theorem
- Estimates the average time a user takes to select a target considering the distance (D) from the cursor and the Width (W) of the target:

$$T = a + b \log_2 \left(2 \frac{D}{W} \right)$$

Time
Distance
Coefficients
Width

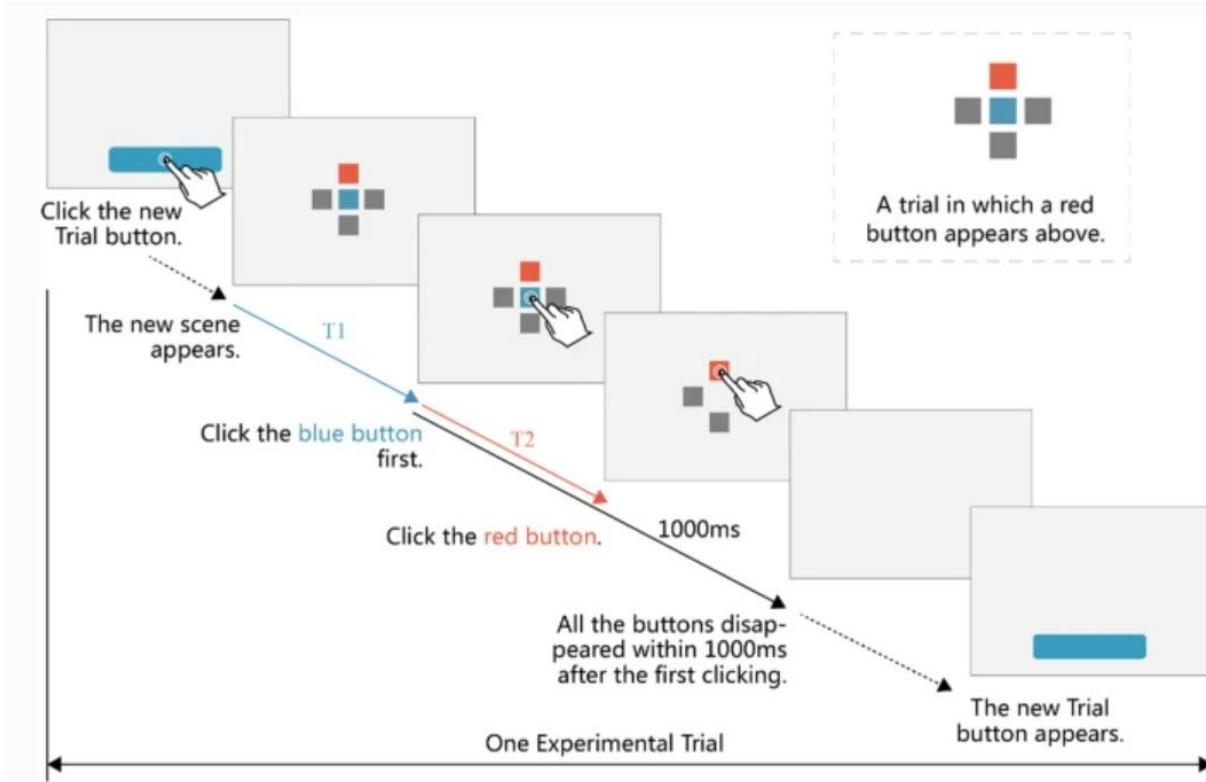


model explain
Negrito (Ctrl+N)

Example:
Minimizing D

- The larger the target the easier to select (no fine control needed)
- The farther the target from the cursor the longer it will take

Example: using Fitts's law in Virtual Environment research



Zhou, X., Guo, Y., Jia, L. et al. A study of button size for virtual hand interaction in virtual environments based on clicking performance. *Multimed Tools Appl* **82**, 15903–15918 (2023).
<https://doi.org/10.1007/s11042-022-14038-w>

GOMS- Goals, Operators, Methods and Selections

- Proposed by Card, Moran and Newell, 1983
- A GOMS decomposition has the following elements:
 - **Goals:** what the user wants to attain
 - **Operators:** basic operations that the user has to perform to use the system; may affect the system or not (press a key or read a message)
 - **Methods:** possible decompositions of the goal into sub-goals (e.g. Select an option “Save” or press “ctrl S”)
 - **Selections:** rules to select the possible methods (taking into account the type of user and the system status)

<https://digital.ahrq.gov/health-it-tools-and-resources/evaluation-resources/workflow-assessment-health-it-toolkit/all-workflow-tools/goms>

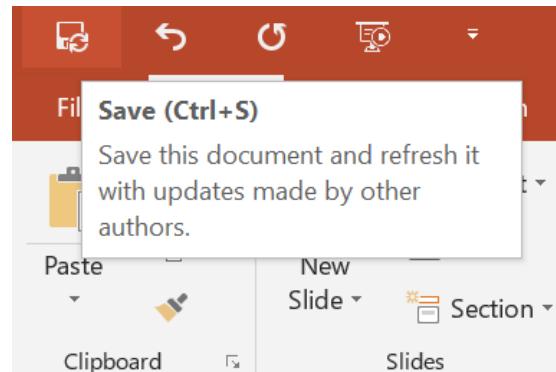
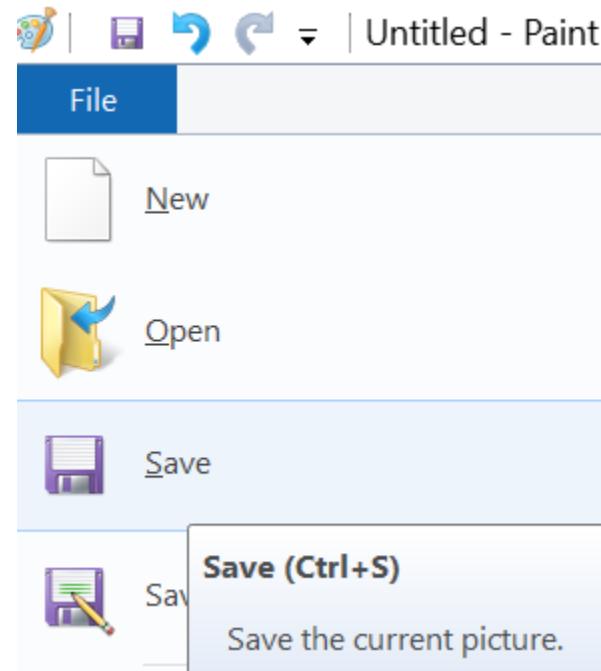
Example: ‘save’ a file: using two common ways

- **GOAL: SAVE-A-DOCUMENT**
 - . [select GOAL: USE-SAVE-OPTION-METHOD
 - . MOVE-POINTER-TO-MENU-BAR
 - . CLICK-OVER-FILE-MENU
 - . MOVE-POINTER-TO-SAVE-ICON
 - . CLICK-SAVE-ICON
 - . GOAL: USE-CTRLS-METHOD
 - . PRESS-‘CTRL’+‘S’-KEYS

User BSS:

Rule 1: **USE-CTRLS-METHOD** unless other rule applies

Rule 2: If has hand on mouse/touch pad **USE-SAVE-OPTION-METHOD**



GOMS- Goals, Operators, Methods and Selections

- A typical GOMS analysis consists in decomposing a high level goal in a sequence sub-goals
- Selection rules must be adjusted to the user profile
- Analyzing the structure of the GOMS decomposition may give an **approximate measure** of :
 - Short Term Memory load (depth of the goal structure)
 - Time needed (a time for each operator)

Example: Copy/scan an article from a journal

- Goal: Photocopy/scan-paper
- . Goal: Locate-article
- Goal: Photocopy/scan-page repeat
 - Goal: Orient-page
 - open cover
 - select-page
 - position-paper
 - close-cover
- Goal: Press-copy-button
- Goal: Verify-copy (only if copy)
 - locate-out-tray
 - examine-copy
- Goal: Collect/send-copy
 - locate-out-tray
 - remove-copy/send pdf (**outer goal satisfied**)
- Goal: Retrieve-journal
 - open-cover
 - remove-journal
 - close-cover

Closure problem
(the user attains the goal before the task is complete)

The “closure problem”



In earlier ATMs the money was given before returning the card

... many users left the card:
their goal was getting money!

This was changed.

The copies usually are available to the user before they remove the original from the photocopier and walk away!

To prevent this, the overall goal should be satisfied only after removing the original



The “closure problem” in MultiBanco



In stores usually these are the following steps:

- Insert the card
- Insert the pin code
- Transaction approval -> audio signal
- Remove the card
- Receipt is handed to the client



At the ATMs the money is given (goal satisfaction) only after the card is removed by the client

These procedures help not to forget the card!

GOMS- Goals, Operators, Methods and Selections

- Capacities:
 - It has been used in cognitive model research
 - It may describe adequately how **experienced users** perform **routine tasks**
 - Associated to a device model allows time estimates

Limitations:

- It does not give information concerning user knowledge to estimate training or transfer times
- It requires experience to apply correctly

Differences between GOMs and Task Analysis (TA)

- The scope of Task Analysis is very wide
- TA models also aspects of the real world not part of the system
(example: feeding paper into a printer; getting paper documents)
- TA describes the tasks users perform from an external point of view and has more detail
- GOMS aims at understanding the user's cognitive processes while performing the task
- TA is more used in early phases of the S/W lifecycle and GOMS for evaluation

Main bibliography

- Y. Rogers, H. Sharp, J. Preece, *Interaction Design: beyond human-computer interaction*, 5th ed., Wiley, 2019

<https://learning.oreilly.com/library/view/interaction-design-beyond/9780470665763/>

- Interaction Design Foundation – IxDF, *How to improve your UX designs with Task Analysis*. Interaction Design Foundation – IxDF, 2023

<https://www.interaction-design.org/literature/article/task-analysis-a-ux-designer-s-best-friend>

- D. Benyon, *Designing Interactive Systems*, 3rd ed., Pearson, 2014

https://www.academia.edu/40407620/Designing_Interactive_Systems_A_comprehensive_guide_to_HCI_UX_and_interaction_design

- Alan Dix, J. Finley, G. Abowd, R. Beale, *Human-Computer Interaction*, 3rd ed., Prentice Hall, 2004

- Ian Sommerville, *Software Engineering*, 10th ed., Pearson, 2016

Interesting links:

- <http://www.usabilitybok.org/goms>
- <http://www.usabilitybok.org/task-analysis>
- <http://web.mit.edu/6.813/www/sp17/classes/09-more-efficiency/>

“Look inside” the following books:

Y. Rogers, H. Sharp, J. Preece, *Interaction Design: beyond human-computer interaction*, 3rd Edition, Wiley, 2011,

<https://learning.oreilly.com/library/view/interaction-design-beyond/9780470665763/>

B. Weyers, J. Bowen, A. Dix, P. Palanque (eds), *The Handbook of Formal Methods in Human-Computer Interaction*, Springer, 2017

[https://books.google.pt/books?id=cGm8DgAAQBAJ&printsec=frontcover&source=gb\(summary_r&cad=0#v=onepage&q&f=false](https://books.google.pt/books?id=cGm8DgAAQBAJ&printsec=frontcover&source=gb(summary_r&cad=0#v=onepage&q&f=false)

D. Benyon, *Designing Interactive Systems*, 3rd ed., Pearson, 2014

https://www.academia.edu/40407620/Designing_Interactive_Systems_A_comprehensive_guide_to HCI UX and interaction design



Input Devices

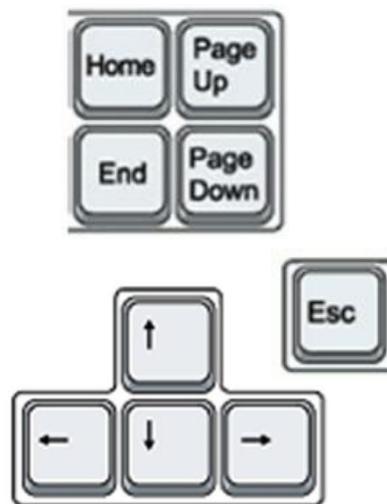


Main Input devices

- Keyboards
- Pointing devices
 - Mouse
 - Touch screen
 - Touch pad
 - Joy stick
 - Track ball, ...
- Voice recognizers
- Eye trackers
- Motion and position trackers
- 3D input devices
- ...

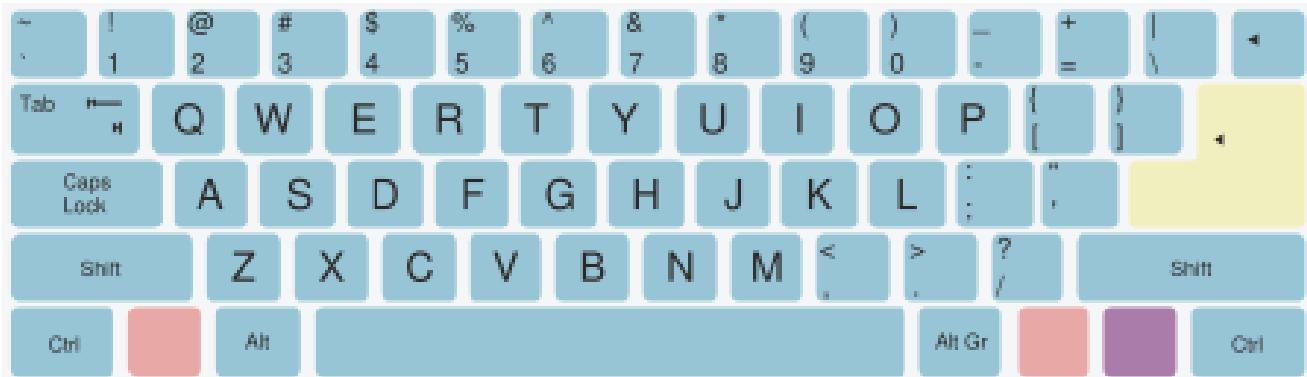
Keyboards

- Relevant issues in UI design:
 - Key layout
 - Operational characteristics:
 - Keyboard size
 - Keyboard angle
 - Hand resting area
 - Key spacing
 - Key activation force
 - Key surface and finishing
 - Key displacement
 - Activation feedback
 - Home row indicators



Keys layout

The Qwerty layout dates from the XIX century, and we still use it!

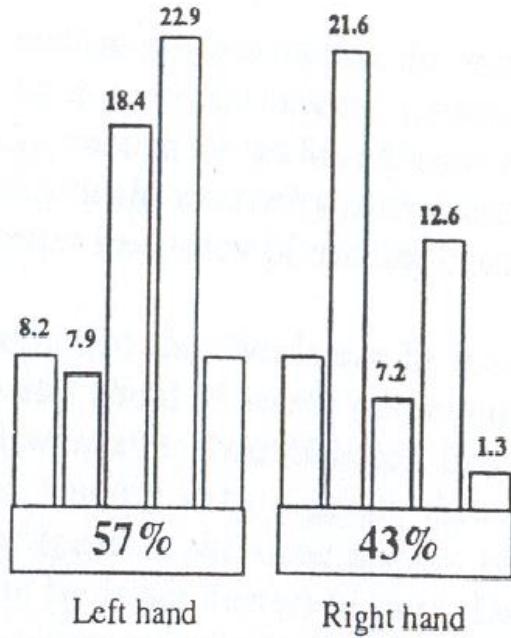


Dvorak

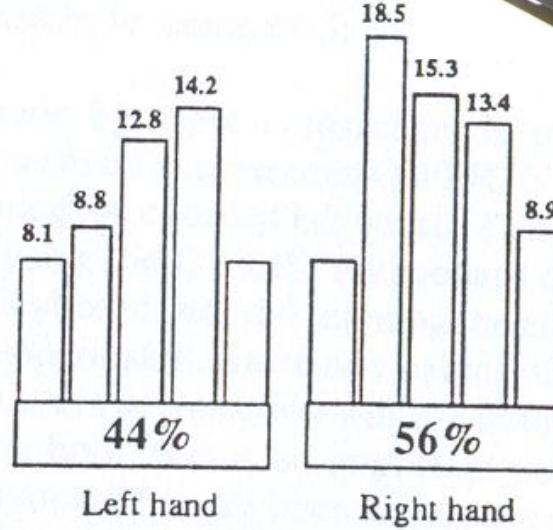


Combining both

Percentage of work performed by each hand (in English)



QWERTY



Dvorak

QWERTY was devised to prevent jams in early typewriters.

<http://www.dvorak-keyboard.com> https://en.wikipedia.org/wiki/Dvorak_keyboard_layout



Ergonomic keyboards

Help avoid RSI (Repetitive Strain Injury) WRULD (Work Related Upper Limb Disorder) and KRP (Keyboard Related Pain)



- ① Zoom
- ② Customizable Hot Keys
- ③ Improved Number Pad
- ④ Ergonomic Design

https://en.wikipedia.org/wiki/Ergonomic_keyboard

Keyboards for specific contexts of use



Chorded keyboard →
used in wearable computing

https://en.wikipedia.org/wiki/Chorded_keyboard



Pointing Devices

They are used to:

- Point a target
- Select a target
- Drawing
- Positioning objects
- Orient and rotate objects
- Define paths among objects
- Handle text
- etc.



- Their efficiency varies according to the tasks
- Shneiderman (98) divided them into:
 - Direct control —— touch screen
light pen (deprecated)



- Indirect control
- mouse
 - track ball
 - digitizing tablet
 - joystick (track point)
 - touch pad



Mice

Currently are optical

- Relative coordinates
- Different shapes, n. of buttons,...



Advantages:

- Direct relation between hand and cursor movement / distance
- Allow speed control speed
- Allow continuous movement in all directions direction

Disadvantages:

- Require hand movement between mouse and keyboard
- Additional space (footprint)
- Hand-eye coordination

<http://www.douengelbart.org/firsts/mouse.html>

<http://www.computerhistory.org/revolution/input-output/14/350>

Trackballs

- Relative coordinates
- Many different shapes

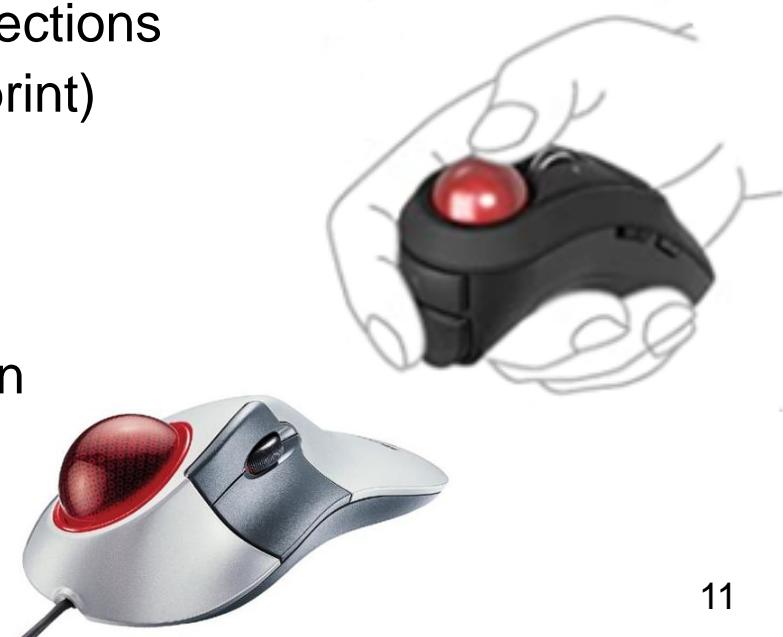


Advantages:

- Direct relation between hand and cursor movement (speed and direction)
- Allow speed control
- Allow continuous movement in all directions
- May not need additional space (footprint)

Disadvantages:

- Require hand-eye coordination
- May require hand movement between trackball and keyboard



Ergonomic Pointing Devices

Zero tension mouse



Whale mouse



Vertical mouse

Wireless Ergonomic Mouse



Standard mouse

Vertical mouse

For users with Repetitive Strain Injury,
Carpal Tunnel Syndrome or other problems
Or to avoid these problems

Touchscreens

- There are several technologies
- Usually are combined with a display

Advantages:

- Direct
- Do not need additional space

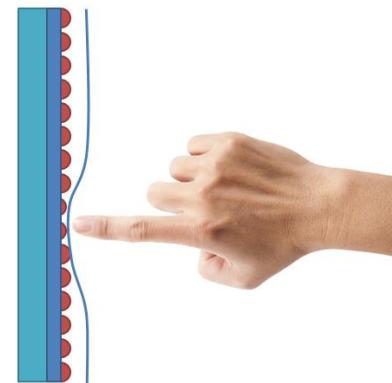
Disadvantages:

- May be tiring if used for long periods (“gorilla arm effect”)
- The finger may obstruct part of the screen
- Get dirty easily

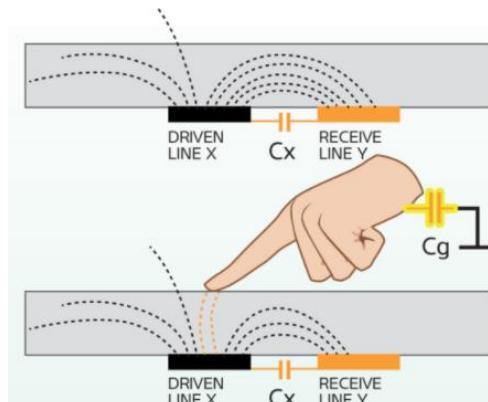
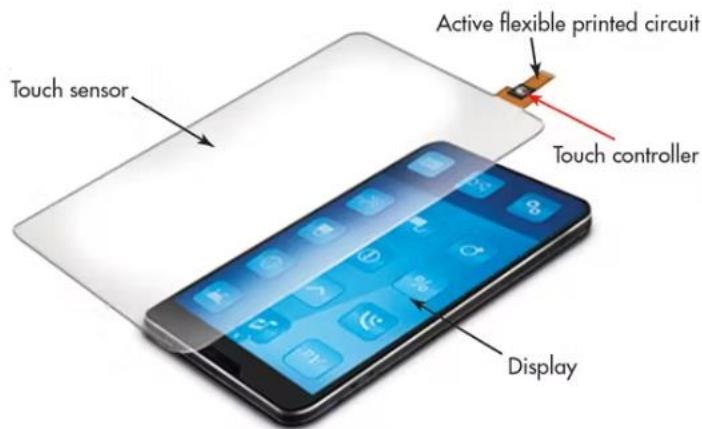


Touchscreens

Resistive (less expensive)



Capacitive (more used in smaller screens)



<https://en.wikipedia.org/wiki/Touchscreen>

<https://www.electronicdesign.com/technologies/displays/article/21800710/whats-the-difference-between-resistive-and-capacitive-touchscreens>

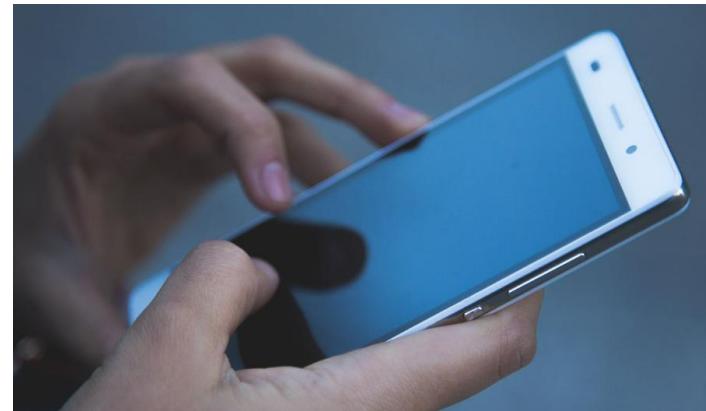
Resistive vs. Capacitive Touchscreens

Resistive touchscreen advantages include:

- Lower cost to manufacture
- Higher sensor resolution
- Fewer accidental touches
- Can sense any object touching the screen hard enough
- More resistant to the elements like heat and water

Capacitive touchscreen advantages include:

- More durable
- Sharper images with better contrast
- Provide multi-touch sensing
- More reliable
- More sensitive to light touch



More Input devices...

- cameras
- eye trackers
- trackers and sensors
- microphones
- controllers of different types
- custom made devices
- etc.



<https://www.manus-meta.com/vr-gloves>

Input/output
device



<https://www.tobii.com/products/eye-trackers/wearables/tobii-pro-glasses-3>

~



Some guidelines to select these interaction devices

- Choose a device **after a careful task analysis and test**
- **Minimize hand and eyes movements**
- Use touch screens when
 - There is no training
 - Targets are large, discrete and scattered
 - Space is important
 - No (or little) text entry
 - Are not used for a long time

Voice recognition systems

- The first system was developed in 1972 at Bell Lab
- It is becoming more used
- Has two types of challenges:
 - Technological (have improved a lot ...)
 - Human factors

Voice recognition as input

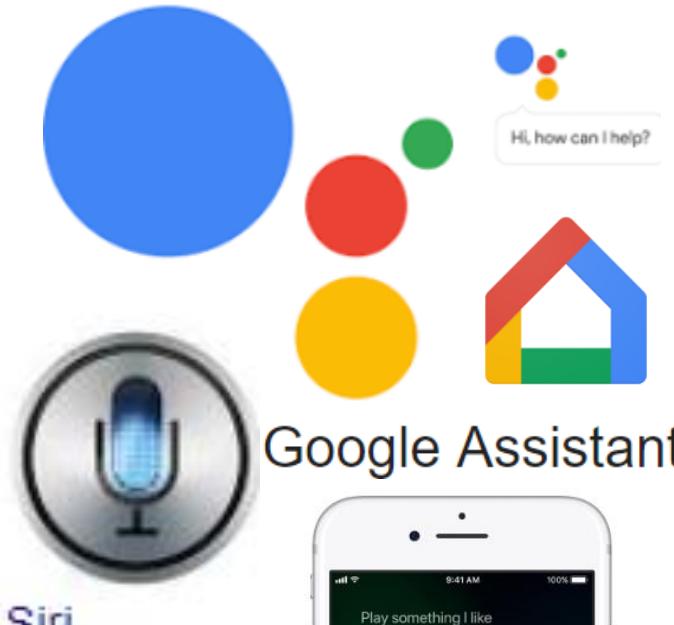
Independently of the technology state of the art,

- Has advantages when the user:
 - Has physical deficiency
 - Must move around
 - Has eyes busy
 - Is in a low visibility or cluttered environment
- Has inherent disadvantages:
 - Voice is transient
 - Does not have natural feedback
 - May disturb other people
 - May result in lack of privacy
 - May be slower and more tiresome (overloading STM)

- Consider voice input when:
 - The user has to move
 - Has eyes or hands busy



- Avoid voice input when:
 - Privacy is important
 - Error rates, even low, are not acceptable
 - Usage frequency is high
 - Speed is important
- Voice input/output has became more used



Siri



Some guidelines for voice interfaces

- Provide output dialog with structure to guide input
- Use a distinct and familiar vocabulary to avoid errors
- Consider voice input if technology constraints are acceptable considering:
 - Ambient noise
 - Privacy
 - Vocabulary extent
 - Error cost

“No matter how different the technology, **the people who are using it haven’t changed**. And most usability principles have more to do with human capabilities and limitations than with technology. (Examples of such eternal design principles include error prevention, flexibility, efficiency, visibility of system status, and recognition vs. recall.)”

<https://www.nngroup.com/articles/voice-interaction-ux/>



Input devices for 3D user interfaces (mostly used in Virtual Reality)

- Trackers:
 - Magnetic
 - Optical
 - Inertial, ...
- Navigation and manipulation interfaces:
 - Controllers, ...
- Gesture interfaces:
 - Gloves
 - Spatial gestures sensors, ...



What future?

It seems likely that we will use more often:

gestures

two hand input

voice

3D pointers

wearable devices

whole-body environments

tactile/force feedback

brain-computer interfaces ...



Conclusion

When choosing an input device, consider:

- Ergonomics / human factors
- Typical scenarios of use
- Cost
- Generality
- DOFs (Degrees Of Freedom)
- Output devices
- Interaction techniques
- ...



Output devices



The ultimate display?

"The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal." (Ivan Sutherland, 1965)

We are not yet there ...

**There are a lot output devices
for a lot of different applications**
Visual displays:



RATE / MIN 999999
TOTAL COUNT 999999



less conventional displays...



Graphics/visual Displays

Are computer interfaces that present images to one or several users
A possible taxonomy:

- Personal displays:
 - monitors
 - HMDs (VR/AR)
 - Monitor-based displays/active glasses
 - Autostereoscopic displays
- Large volume displays:
 - Caves
 - Walls
 - Domes
 - ...

Personal Displays



The images may be monoscopic or stereoscopic, monocular (for a single eye) or binocular (displayed on both eyes).

- Screens of various sizes
- Wearable Displays
- Hand-held
- Auto-stereoscopic displays
(desk supported)



Large-volume displays

- CAVE type displays
- Wall-type displays
- Domes
- ...

<https://steantycip.com/vr-cave/>



Main technologies:

- LED displays (several types)
 - LCD displays (older)
 - Autostereoscopic displays: lenticular/barrier
 - ...
-
- Other technologies:
electrophoretic,...



https://en.wikipedia.org/wiki/E_Ink



- **Images provided by computer monitors are poor when compared to the real world**
- **It is amazing what we get from such simple devices**

- Monitors have several limitations:
 - Small range of intensities and colors
 - Lack of focusing distance
 - Small field of view
 - ...



Stereoscopic displays



*Two images for the two eyes provided by a HMD
(Head-Mounted Device)*



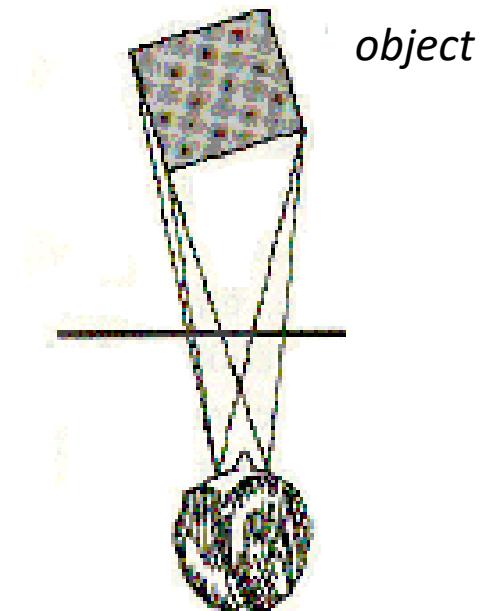
Right eye image



Left eye image

Projection plane

eyes



- Need to present **two images** of the same scene (one for the **right eye** and another for the **left eye**)
- The two images can be presented:
 - **at the same time** on two displays (HMD)
 - **time-sequenced** on one display (active glasses)
 - **spatially-sequenced** on one display (auto-stereoscopic displays)



Left eye, right eye images
(Burdea and Coiffet., 2003)



Curious about the future of visual displays?



SolidLight replaces physical things with software-controlled holograms

<https://www.lightfieldlab.com/watch-how-it-works>

Xiong, J., Hsiang, EL., He, Z. et al. Augmented reality and virtual reality displays: emerging technologies and future perspectives. *Light Sci Appl* **10**, 216 (2021).

<https://doi.org/10.1038/s41377-021-00658-8>

A glimpse of the future? Interactive live holography

<http://realviewimaging.com/technology/>



And not only to produce visual displays...

[https://www.3dsystems.com
/haptics-devices/touch](https://www.3dsystems.com/haptics-devices/touch)



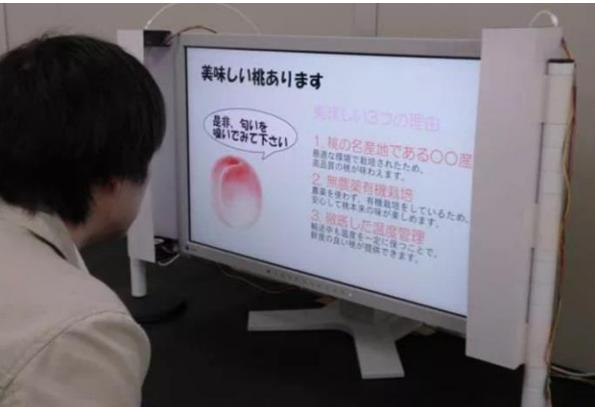
sound



smell



Touch and force feedback

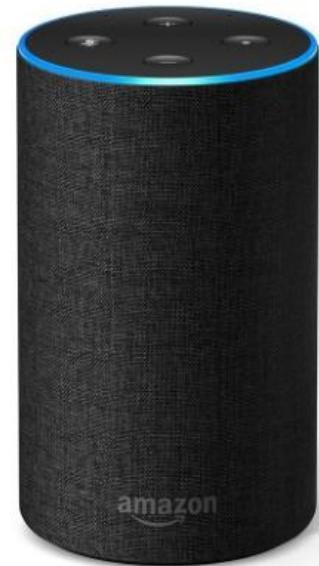


<https://vrscout.com/news/olfactory-engineering-scent-based-vr/>



Examples of using voice input/output and natural language interaction style:

- Siri
- Alexa
- Google Home
- Google Duplex



[https://en.wikipedia.org/
wiki/Amazon_Alexa](https://en.wikipedia.org/wiki/Amazon_Alexa)

Voice synthesizers

- There are several types:
 - Digitized - concatenates recorded basic sounds
 - Synthesised – concatenates sounds generated with models
- There are several technical challenges due to the nature of human voice:
 - different pronunciation rules
 - meaning may be changed by intonation
 - differences in intonation reflect different moods
- **The quality of a synthesizer implies much more than intelligibility**

Advantages of using voice output:

When the user has:

- physical deficiency
- to move around
- hands and eyes busy
- Adverse conditions: low visibility, low O₂, high Gs

Disadvantages:

- Is tiresome and uncomfortable for long periods
- Is transient (taxes STM)
- May have privacy issues
- May disturb other people

Some guidelines to use voice output

- Consider voice output as an alternative when the user must move around, has hands and eyes busy
- Avoid voice output in open environments, when the privacy and security are important issues and frequency of usage is high
- Use approx. 180 words per minute
- When messages are not expected, start with non-critical words that provide context
- Say first the goal and then the solutions
- Allow messages to be repeated

Every year new devices appear

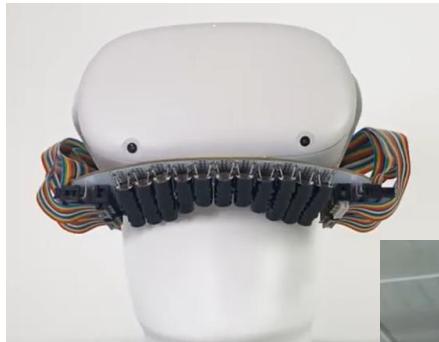
- Some are really weird ...



[Haptic System Creates Finger-Touch Sensations Hardware-Free - IEEE Spectrum](#)

- Some never go beyond experimental research
- But once in a while a few become wide-spread

Another example:



**Splashing into a puddle, which triggers a series
of short random impulses.**

Mouth Haptics in VR using a Headset Ultrasound Phased Array | Proceedings
of the 2022 CHI Conference on Human Factors in Computing Systems
(acm.org)

<https://www.youtube.com/watch?v=3q6dZQfV1x8>

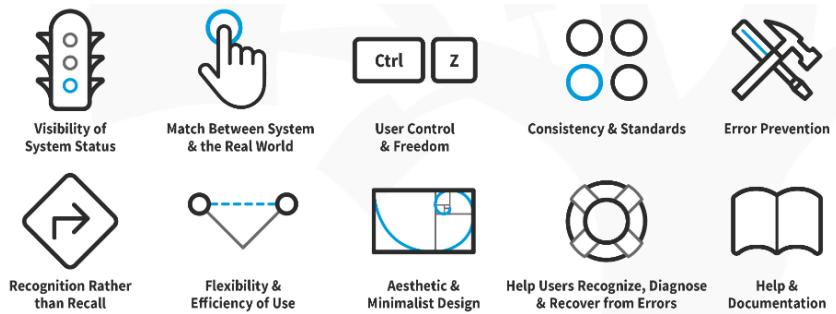
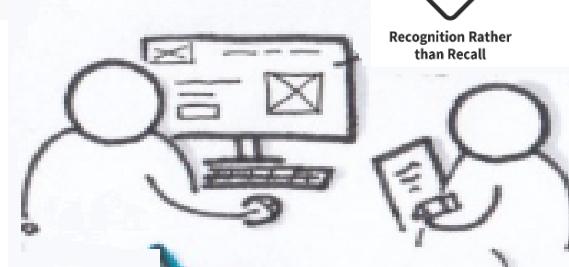
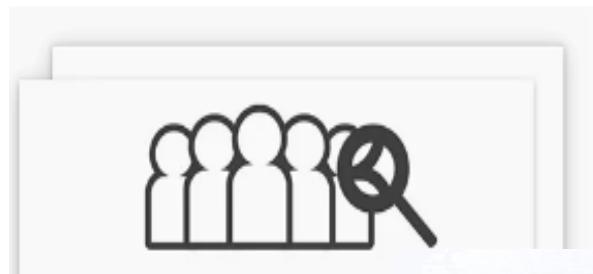
Conclusion

- **Technology shall not be used only because it is new!**
- **Independently from the type or state of the art of the input / output devices it is necessary to understand their usability for different types of users, tasks and context**



Usability

Evaluation Methods - overview



[What is Usability Evaluation?](#) |
[IxDF \(interaction-design.org\)](http://IxDF(interaction-design.org))

Beatriz Sousa Santos

- Usability is, according to ISO 9241-11:

“the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”

- How to measure it??

Standards evolve:

- ISO 9241-11's three factors of usability have become five in ISO 25010's quality in use factors:
- Effectiveness
- Efficiency
- Satisfaction
- Freedom from risk
- Context coverage

<https://www.iso.org/standard/35733.html>

<https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/usability-evaluation>

(Cokton, 2013):

“Put simply, usability evaluation assesses the extent to which an interactive system is easy and pleasant to use”.

Things aren't this simple at all though, but ...:

- Usability is a measurable property of all interactive digital technologies
- Evaluation methods determine if an interactive system or device is usable
- And the extent of its usability, through robust, and reliable metrics
- Evaluation methods and metrics are thoroughly documented ...

http://www.interaction-design.org/encyclopedia/usability_evaluation.html

<https://www.nngroup.com/articles/which-ux-research-methods/>

Evaluation Methods

(widely used classification)

- **Analytical** (without users)

- Heuristic Evaluation ✓
- Cognitive Walkthrough ✓
- Model based methods
- Review methods

- **Empirical** (involving users)

- Observation > usability tests ✓
- Query
- Controlled Experiments

(Dix et al.)

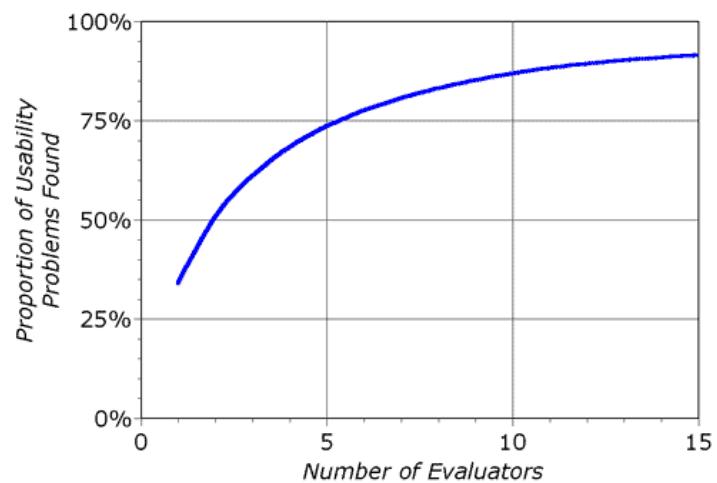
(✓ - have used in Lab classes)

Heuristic Evaluation (Nielsen and Molich 1990)

- A “**discount usability engineering method**” for quick, cheap, and easy evaluation of a UI design
- The most popular of the usability inspection methods
- It is a systematic inspection of a design for usability
- Meant to find the usability problems in the design so that they can be attended to as part of an iterative design process.
- Involves a small set of analysts judging the UI against a list of usability principles (“heuristics”).

- Is difficult for a single individual to do; one person will never be able to find all the problems
- Involving multiple evaluators improves the effectiveness of the method significantly
- Nielsen generally recommends to use three to five evaluators
- not much gain by using larger numbers

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



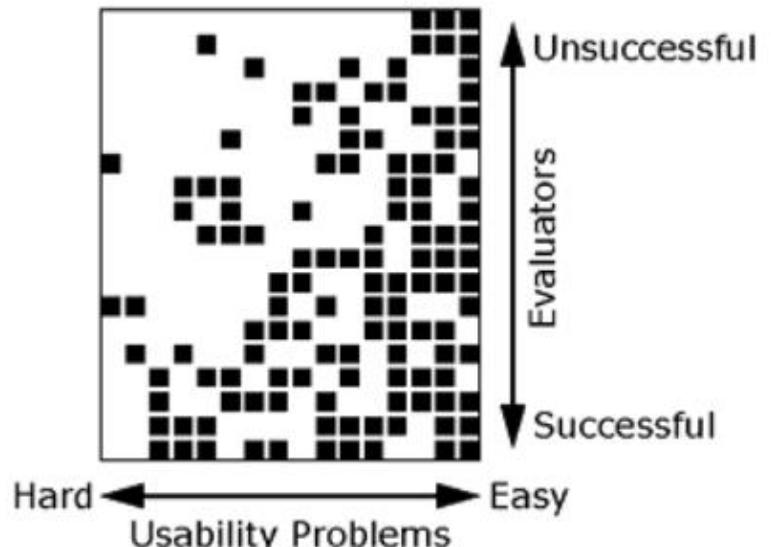
Example:

- Heuristic evaluation of a banking system:
 - 19 evaluators
 - 16 usability problems

black square - problem found

white square – not found

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



This suggests that in general 3 to 5 evaluators may be reasonable...

How to select the number of evaluators for a specific case?

- Consider the following criteria:
 - **Complexity** of the user interface
 - **Experience** of the evaluators
 - **Expected costs /benefits**
 - **Criticality** of the system (cost of user errors)
 - ...

How to perform HE

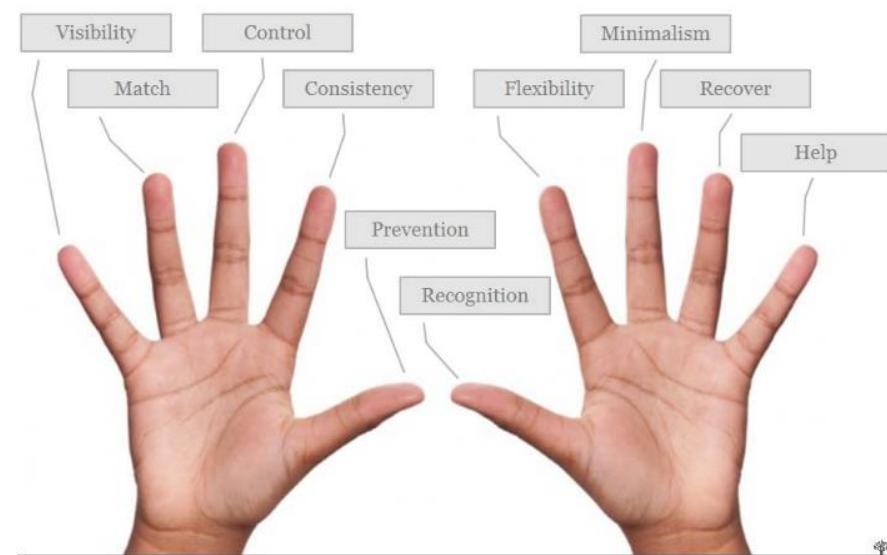
- Should be performed by **several evaluators**
(one person will never be able to find all the problems)
- **Evaluators should work independently:**
 - First get a **general idea** of the UI
 - Then perform a **detailed inspection using a set of heuristics**
 - List usability problems (**heuristics not followed and severity degree**)
- Findings of all evaluators should be **integrated in the same report**

The report should help the development team to prioritize problem fixing!

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

- Nielsen proposed **10 general usability heuristics**,
- yet there are other sets, e.g. for
 - different types of applications
(web, mobile, visualization ... applications)
 - different types of users
(for seniors, children...)

http://www.interaction-design.org/encyclopedia/usability_evaluation.html



How to perform Heuristic Evaluation

Each evaluator:

- First **make a general analysis** to get to know the UI
- Then, make a **systematic analysis** having in mind **the heuristics**
- **Take note of each potential problem**, the heuristic and the severity grade

Finally, compile all the potential problems and discuss with other evaluators

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

Ten Nielsen's heuristics

- **Visibility of system status**
- **Match between system and the real world**
- **User control and freedom**
- **Consistency and standards**
- **Error prevention**
- **Recognition rather than recall**
- **Flexibility and efficiency of use**
- **Aesthetic and minimalist design**
- **Help users recognize, diagnose, and recover from errors**
- **Help and documentation**

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

Example:

Heuristic #6 - Recognition rather than recall



Example of Usability Heuristic #6:

It's easier for most people to recognize the capitals of countries, instead of having to remember them. People are more likely to correctly answer the question Is Lisbon the capital of Portugal? rather than What's the capital of Portugal?

Tips

- Let people recognize information in the interface, rather than having to remember ("recall") it.
- Offer [help in context](#), instead of giving users a long tutorial to memorize.
- Reduce the information that users have to remember.

Learn more:

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

Severity rating of usability problems

Is a combination of **three factors**:

- The **frequency** with which the problem occurs
- The **impact** of the problem if it occurs
- The **persistence** of the problem

The following 0 to 4 **rating scale** can be used to rate the severity of usability problems:

0 = I don't agree that this is a usability problem at all (to be used in the discussion)

1 = Cosmetic problem

2 = Minor usability problem

3 = Major usability problem

4 = Usability catastrophe

- **Main advantages of heuristic evaluation:**
 - May produce **useful results with modest investment**
 - **Simple to apply** even by not very experienced evaluators
 - May be used **along the development process from early phases**
- **Main limitations:**
 - **Subjective** (partially overcome with more and more experienced evaluators)
 - **Tends to find many small problems** which may not be very important
 - **Can't find all usability problems**
 -> **evaluation involving users is needed!**

Cognitive Walkthrough (Wharton, et al., 1992)

- Usability inspection method **not involving users** (analytical)
- Based on the fact that users usually prefer to learn a system by using it (e.g., instead of studying a manual)
- **Focused on assessing learnability** (i.e., how easy it is for new users to accomplish tasks with the system)
- **Applicable at early phases**, before any coding

How to perform a cognitive walkthrough

- 1- Task analysis:** sequence of steps or actions required by a user to accomplish a task, and the system responses
- 2- Designers and developers walkthrough as a group,** asking themselves a set of questions at each step
- 3- Data gathering during the walkthrough: answering the questions** for each subtask usability problems are detected
- 4- Report of potential issues**
- 5- UI redesign to address the issues identified**

CW Four questions:

- **Will the user try to achieve the effect that the subtask has?**
(Does the user understand this subtask is needed to reach the goal?)
- **Will the user notice that the correct action is available?**
(E.g. is the button visible?)
- **Will the user understand that the wanted subtask can be achieved by the action?**
(E.g. the button is visible but the user doesn't understand the text and will not click on it)
- **Does the user get feedback?**
Will the user know that they have done the right thing?

Common issues

- The evaluator may not know the optimal way to perform the task; the method involves the optimal sequence of actions
- Involves an extensive analysis and documentation and often too many potential issues are detected, resulting very **time consuming**

Thus:

Lighter variants of Cognitive Walkthrough were proposed to make it **more applicable** in S/W development companies

Streamlined Cognitive Walkthrough (Spencer, 2000)

- Only two questions:
 - **Will the user know what to do at this step?**
 - **If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?**
 - And a set of rules to streamlining the walkthrough and trade-off granularity for coverage
- comprises the 3 first
questions of CW
- 

According to Spencer the method can be applied successfully if the usability specialist:

- takes care to prepare the team for the walkthrough,
- avoids design discussions during the walkthrough,
- explicitly neutralizes defensiveness among team members,
- streamlines the procedure by collapsing the first three questions into one question,
- and captures data selectively

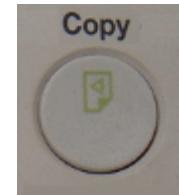
Example: Evaluation of a desktop photocopier UI

- Machine UI:
 - numeric keypad,
 - "Copy" button,
 - push button on the back to turn on the power



The machine automatically turns itself off after 5 min inactivity

- Task: copy a single page
- User: any office worker
- Actions needed: turn on the power,
put the original on the machine,
press the "Copy" button



<http://hcibib.org/tcuid/chap-4.html#4-1>

- Story for action number one:
“the user wants to make a copy and knows that the machine has to be turned on. So she pushes the power button. Then she goes on to the next action”

Not convincing!

- why shouldn't the user assume that the machine is already on?
That is often the case
- Will the user figure out that the machine is off, and find the power switch?
etc. etc.

Another example: Look for a person's phone number and email address at the University of Aveiro Web site
User: any student from the University



Task analysis:

- find the icon  (search);
- input part of the person's name and search in “Pessoas”
- get the phone number

But the defined user profile (any student from the University) includes foreign students, thus a previous action is needed:

- select the English version 

For each action we need to ask the two questions and put ourselves in the shoes of the user!



Previous action for foreign students: Select the English version seems easy (it is a “standard” way to do it in sites)

First action in the Portuguese version: find the icon



Q1 - Will the user know what to do at this step?

Even without tooltip the correct icon seems recognizable (it is “standard”)



Q2 - If the user does the right thing (selects the icon), will they know that they did the right thing, and are making progress towards their goal?



Probably yes; while it may not look a search bar, it is adequately labeled (Pesquisa em páginas, ...)

Second action: input part of the person's name and search in “Pessoas”

 Beatriz Sousa

[Todo o Portal](#)

[Pessoas](#)

[Notícias](#)

[Locais](#)

Aproximadamente 3,590 resultados (0.15 segundos)

Q1 - Will the user know what to do at this step?

Probably yes; it is easy to recognize that s/he should input the person's name and select “Pessoas”

 Beatriz Sousa

[Todo o Portal](#)

[Pessoas](#)

[Notícias](#)

[Locais](#)

Aproximadamente 3,590 resultados (0.15 segundos)

 Beatriz Sousa

[Todo o Portal](#)

[Pessoas](#)

[Notícias](#)

[Locais](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Q2 - If the user does the right thing (inputs the name and selects “Pessoas”), will they know that they did the right thing, and are making progress towards their goal?

The screenshot shows a search interface. At the top, there is a search bar with a magnifying glass icon containing the text "Beatriz Sousa". Below the search bar is a navigation menu with four items: "Todo o Portal" (in blue), "Pessoas" (underlined in red, indicating the active category), "Notícias" (in blue), and "Locais" (in blue). Below the menu is a horizontal row of letters from A to Z, each in a small blue font. Underneath this row, the search results are displayed. The first result is a link to a profile page for "Maria Beatriz Alves de Sousa Santos". The profile information includes her name, affiliation ("Departamento de Eletrónica, Telecomunicações e Informática"), and contact details ("24117 | bss@ua.pt").

Maria Beatriz Alves de Sousa Santos
Departamento de Eletrónica, Telecomunicações e Informática
24117 | bss@ua.pt

Probably yes; however, some users may not recognize 24117 as a phone number (it only has 5 digits, as it is internal, and not 9 as possibly expected)

In conclusion:

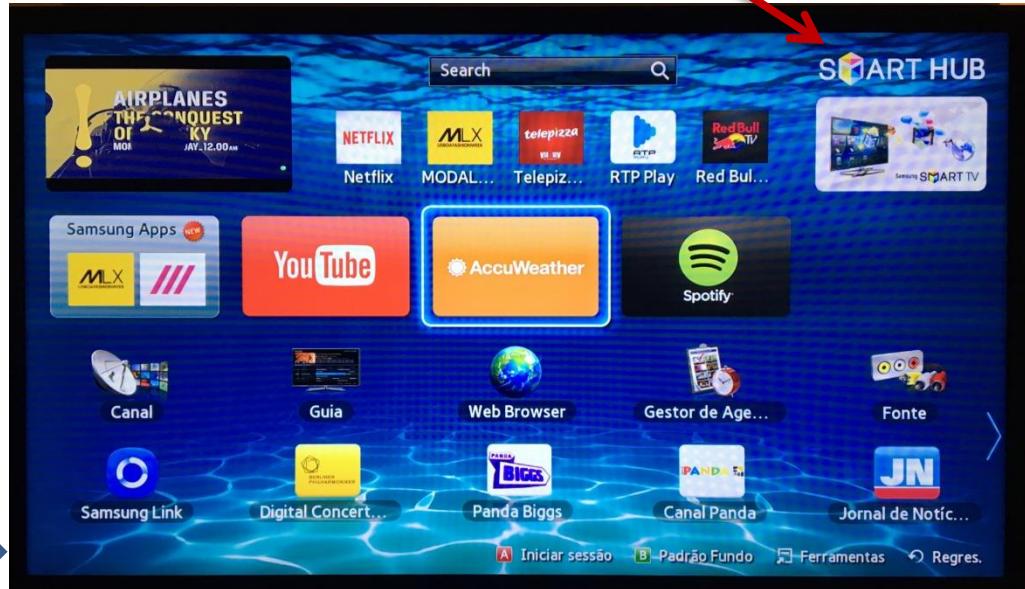
- it seems easy for the target users to reach the phone number and email address;
- however, the phone number may be not recognized as such

Another example: Smart TV

How to access
the Internet?

(before reading
the manual?)

(we see the
symbol at the
screen only
after pressing it
on the control!)



Practice the Streamlined Cognitive Walkthrough:

Analyzing interactive systems/applications that should be very intuitive (e.g. consumer electronics):

- Turn on and off the video projector in your Lab using the remote control or directly on the projector
user: any student from the University
- Change the Channel using the box of your TV service (not the remote control)
user: anyone having a TV box

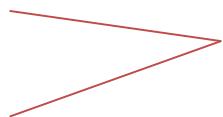


Limitations of Analytical Methods

- Are subjective
- Involve several usability experts
- Cannot find all usability problems

Thus, empirical methods (involving users) are needed !!

observation
query



Usability test (engineering approach)

controlled experiments (scientific approach)

Evaluation Methods

- Analytical (without users)

- Heuristic Evaluation ✓
- Cognitive Walkthrough ✓
- Model based methods
- Review methods

- Empirical (involving users)

- Observation > usability tests ✓
- Query
- Controlled Experiments ←

(✓ - have used in Lab classes
→ - have seen in papers)

Ethics in applying empirical methods

Involving users implies **specific cautions**:

- Asking for explicit consent
- Confidentiality
- Security (avoid any risk)
- Freedom (users may give up at any time)
- Limit stress

It's the system that is under evaluation not the user!

Anyone doing research with people should have a good grasp of research ethics!

<https://www.nngroup.com/articles/user-research-ethics/>

Empirical evaluation styles

These methods may be performed:

- In the laboratory (more controlled)
- In the field (more realistic)

They produce complementary information;

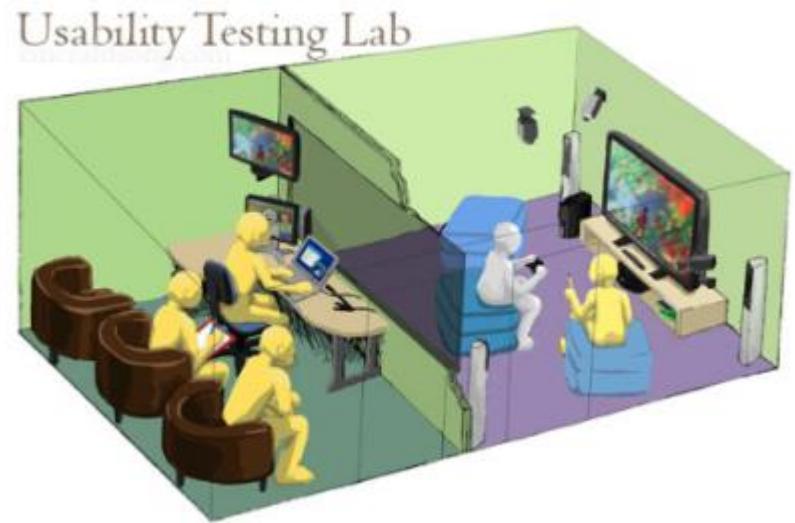
if possible use both!

<https://www.nngroup.com/articles/field-studies/>

Observation

Has many variants from very simple to very complex and expensive:

- Direct: observer takes notes
- Undirect: through audio/ video – more complex and time consuming
- Think Aloud: users are asked to explain what they are doing
- Logging: users activity is logged by the system
- Combinations of the previous, etc



<https://www.usabilitybok.org/usability-testing-methods>

Think aloud Observation

Participants are asked to use the system while continuously thinking out loud (verbalizing their thoughts as they use the system)

Benefits:

- Inexpensive
- Flexible
- Easy to learn and apply

Limitations:

- Unnatural situation
- Filtered statements
- Changing user behavior

<https://www.usabilitybok.org/usability-testing-methods>

<https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/>

Query

- Two main variants:
 - Questionnaire
(reach more people; less flexible)
 - Interview
- **Should always be carefully prepared and tested**
- Collected data should be carefully analyzed



<https://www.interaction-design.org/literature/article/useful-survey-questions-for-user-feedback-surveys>

<https://www.interaction-design.org/literature/article/how-to-conduct-user-interviews>

Well-known usability questionnaires

- **System Usability Scale (SUS)**

- Questionnaire for User Interface Satisfaction (QUIS)

- SUS provides a “quick and dirty”, reliable tool for measuring the usability
- It includes 10 questions with five response options
- QUIS is a measurement tool designed to assess a computer user's subjective satisfaction with the UI
- It is designed to be configured according to the needs of each UI analysis by including only the sections that are of interest to the user
- It includes questions with ten response options
- Both questionnaires should be completed following use of the UI in question

	Strongly Disagree	Strongly Agree
1. I think that I would like to use this product frequently.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
2. I found the product unnecessarily complex.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
3. I thought the product was easy to use.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
4. I think that I would need the support of a technical person to be able to use this product.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
5. I found the various functions in the product were well integrated.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
6. I thought there was too much inconsistency in this product.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
7. I imagine that most people would learn to use this product very quickly.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
8. I found the product very awkward to use.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
9. I felt very confident using the product.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
10. I needed to learn a lot of things before I could get going with this product.	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	

System Usability Scale (SUS)

- Provides a “**quick and dirty**”, reliable tool for measuring the usability
- It includes 10 questions with five response options
- It allows to **evaluate a wide variety of products and services** (H/W, S/W, mobile devices, websites and applications)
- Has become an **industry standard**, with references in many publications

Benefits of using a SUS

- Is a **very easy** scale to administer to participants
- Can be used on **small sample sizes** with reliable results
- **Is valid** – it can differentiate between usable and unusable systems

<https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

SUS Questions

- I think that I would like to use this system frequently.
- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

<https://www.usability.gov/how-to-and-tools/resources/templates/system-usability-scale-sus.html>

Scoring SUS

- SUS provides a value in **[0-100]**
- To obtain the value:
 - Add the scores of all questions:
 - odd numbered questions - subtracting 1 from the score
 - even numbered questions - subtracting their value from 5
 - Multiply the sum by 2.5.

SUS > 68 would be considered above average

Scoring SUS

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="text"/>				
	1	2	3	4	5
2. I found the system unnecessarily complex	<input type="text"/>				
	1	2	3	4	5

Step 1: Convert the user ratings from the 10 questions into points:

Strongly Disagree: 1 point

Disagree: 2 points

Neutral: 3 points

Agree: 4 points

Strongly Agree: 5 points

Step 2:

$x = \text{Sum of points of all odd-numbered} - 5$

$y = 25 - \text{Sum of points from all even numbered questions}$

Step 3: SUS Score = $(x+y) * 2.5$ **SUS [0... 100]**

SUS > 68 would be considered above average

QUIS - Questionnaire for User Interface Satisfaction

- The QUIS contains:
 - a demographic questionnaire,
 - a measure of overall system satisfaction,
 - a measure of specific UI factors (e.g. screen visibility, terminology and system information, learning factors, and system capabilities)
- QUIS has pen and paper and PC software versions for administration
- Uses a 10-point scale to rate 21 items relating to the system's usability
- These ratings produce data for the overall reaction to a system's usability on 6 factors.
- It is easy to use and analyse.

Example questions of QUIS

OVERALL REACTIONS TO THE SOFTWARE

terrible 0 1 2 3 4 5 6 7 8 9 wonderful

difficult 0 1 2 3 4 5 6 7 8 9 easy

frustrating 0 1 2 3 4 5 6 7 8 9 satisfying

inadequate power 0 1 2 3 4 5 6 7 8 9 adequate power

dull 0 1 2 3 4 5 6 7 8 9 stimulating

rigid 0 1 2 3 4 5 6 7 8 9 flexible

Characters on the computer screen

hard to read 0 1 2 3 4 5 6 7 8 9 easy to read

Highlighting on the screen simplifies task

not at all 0 1 2 3 4 5 6 7 8 9 very much

Organization of information on screen

confusing 0 1 2 3 4 5 6 7 8 9 very clear

Sequence of screens

confusing 0 1 2 3 4 5 6 7 8 9 very clear

USABILITY AND USER INTERFACE

Use of colors and sounds

poor 0 1 2 3 4 5 6 7 8 9 good

System feedback

poor 0 1 2 3 4 5 6 7 8 9 good

System response to errors

awkward 0 1 2 3 4 5 6 7 8 9 gracious

System messages and reports

poor 0 1 2 3 4 5 6 7 8 9 good

System clutter and UI “noise”

poor 0 1 2 3 4 5 6 7 8 9 good

Usability tests

- Involve **observation and query**
- Main aspects:
 - Participants
 - Tasks
 - Test facilities and systems
 - Protocol
 - Usability measures
 - Data analysis
- May have a **complex logistics**
- Standard: **Common Industry Format (CIF)** for usability test reports

<https://www.interaction-design.org/literature/topics/usability-testing>

<https://www.usability.gov/how-to-and-tools/methods/planning-usability-testing.html>

Participants

- The total number of participants to be tested
(a valid statistical analysis implies a sufficient number of subjects)
- Segmentation of user groups tested, if more than one
- Key characteristics and capabilities of user group
(user profile: age, gender, profession, computing experience, product experience, etc.)
- How to select participants
- Differences between the participant sample and the user population
(e.g. actual users might have training whereas test subjects were untrained)

Tasks

- The task scenarios for testing
- Why these tasks were selected
 - (e.g. the most frequent tasks, the most troublesome tasks)
- The source of these tasks
 - (e.g. observation of users using similar products, product specifications)
- Any task data given to the participants
- Completion or performance criteria established for each task
 - (e.g. n. of clicks < N, time limit)

Test Facilities and equipment

- The setting and type of space in which the evaluation will be done
(e.g. usability lab, cubicle office, meeting room, home office, home family room, manufacturing floor, etc.)
- Any relevant features or circumstances that can affect the results
(e.g. video and audio recording equipment, one-way mirrors, or automatic data collection equipment)
- Participant's computing environment
(e.g. computer configuration, including model, OS version, required libraries or settings, browser name and version; relevant plug-in, etc.)
- Display and input devices characteristics
- Any questionnaires to be used

Protocol

- Procedure: the logical design of the test
- Participant general instructions and task instructions
- The usability measures to be used:
 - a) for **effectiveness** (completeness rate, errors, assists)
 - b) for **efficiency** (times)
 - c) for **satisfaction**

Common Industry Format (CIF) for usability test reports

ISO/IEC 25062:2006

- Specifies the format for reporting the results of a **summative** evaluation
- The most common type of usability evaluation is **formative**, (i.e. designed to identify problems that can be fixed)
- A summative evaluation produces usability metrics that describe how usable a product is when used in a particular context of use
- The CIF report format and metrics are consistent with the ISO 9241-11

<https://www.iso.org/standard/43046.html>

<https://www.userfocus.co.uk/articles/cif.html>

Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- **Common Industry Format (CIF) for usability test reports**



This standard was last reviewed and confirmed in 2019.

The format includes the following elements:

- the description of the product,
- the goals of the test,
- the test participants,
- the tasks the users were asked to perform,
- the experimental design of the test,
- the method or process by which the test was conducted,
- the usability measures and data collection methods, and
- the numerical results.

Evaluation Methods

- Analytical (without users)

Heuristic Evaluation ✓
Cognitive Walkthrough ✓
Model based methods
Review methods

- Empirical (involving users)

Observation > usability tests ✓
Query
Controlled Experiments ←

(✓ - have used in Lab classes
→ - have seen in papers/participated)

Controlled experiments

- The “**work horse**”
of experimental science ...
- Important issues to consider:
 - Hypothesis
 - Variables (input or independent; output or dependent, secondary)
 - Experimental design (within-groups; between-groups)
 - Protocol
 - Participants (number, profile)
 - Statistics



<https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/experimental-methods-in-human-computer-interaction>

Controlled experiment

- Define **hypotheses**
- Define input (independent), output (dependent) and secondary **variables**
- Define **experimental design** (within-groups / between groups)
- Define **protocol**
- Select the **participants**
- Prepare all the **documentation and data gathering mechanisms**:
 - list of tasks and perceived difficulty
 - final questionnaire
 - list of tasks for the experimenter to take notes

To the user

To the experimenter
- Run a **pilot test**
- Take care of the **logistics** ... and after the experiment **analyze data**

Controlled experiment

Variables:

- **Independent or input variables** – what is controlled
(e.g. interaction method)
- **Dependent or output variables** – what is measured
(e.g. times and errors)
- **Secondary variables** – not controlled but may influence the result
(e.g. age, previous experience)

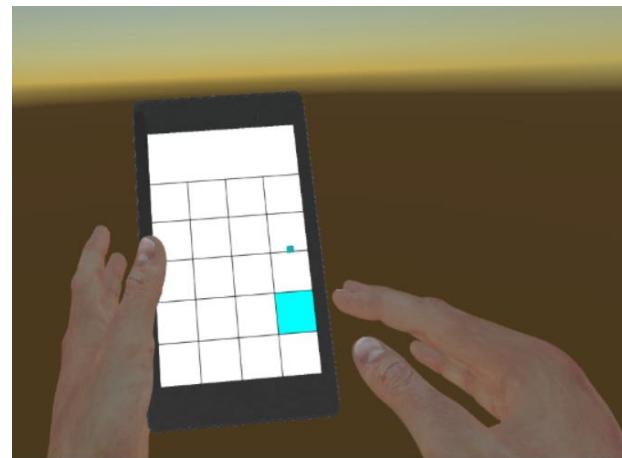
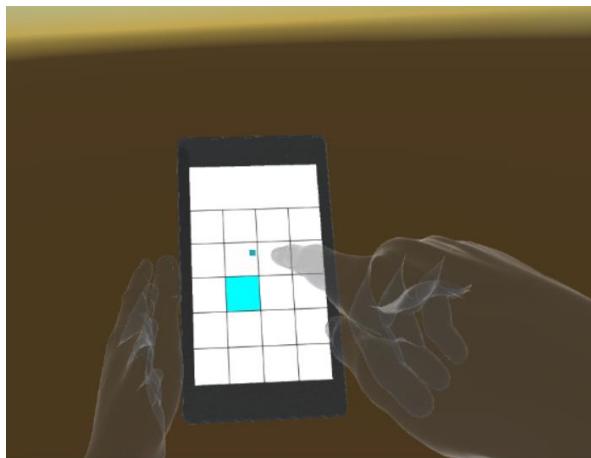
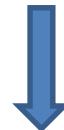
Controlled experiment

Experimental design:

- **Within-groups or within-subjects** – all participants use the same conditions (usually in randomized order to avoid bias)
 - advantages – a smaller number of participants
same profile
 - disadvantages – prone to fatigue or learning bias
- **Between-groups or between-subjects** – each participant uses only one condition
 - advantages – less fatigue or learning bias
 - disadvantages – higher number of participants needed
different participants' profile

Examples of Controlled Experiments performed @ HCI - DETI

- How should the user's hands be represented in a specific Virtual Environment?
- Study of the Effect of Hand-Avatar in a Selection Task using a Tablet as Input Device in an Immersive Virtual Environment



“Effect of Hand-Avatar in a Selection Task using a Tablet as Input Device in an Immersive Virtual Environment”

L. Afonso, P. Dias, C. Ferreira, B. Sousa Santos

IEEE 3D UI, Los Angeles, March 2017



- **Research question:** How does the virtual representation of the user's hands influence the performance on a button selection task performed in a tablet-based interaction within an immersive virtual environment?
- **Method:** Controlled experiment
- 55 participants used **three conditions:**
 - no-hand avatar,
 - realistic avatar,
 - translucent avatar.
- Participants were slightly faster but made more errors with no-avatar
- Considered easier to perform the task with the translucent avatar

Experimental Design

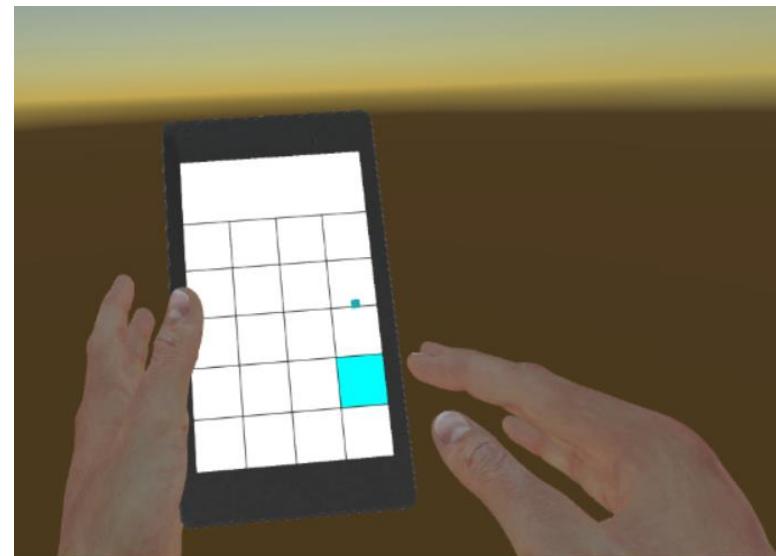
Null Hypothesis: usability is independent of the hands representation

Independent (input) variable (with 3 levels): representation of the hands

Dependent (output) variable: usability (performance + satisfaction)

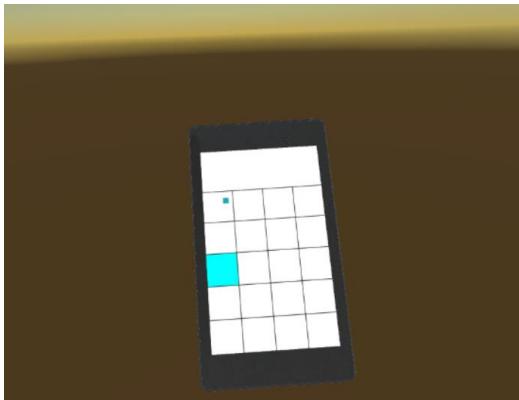
Within-groups: all participants used all experimental conditions (in different sequences to avoid learning or fatigue bias)

Task: selecting as fast as possible a highlighted button from a group of twenty buttons (repeated measures)

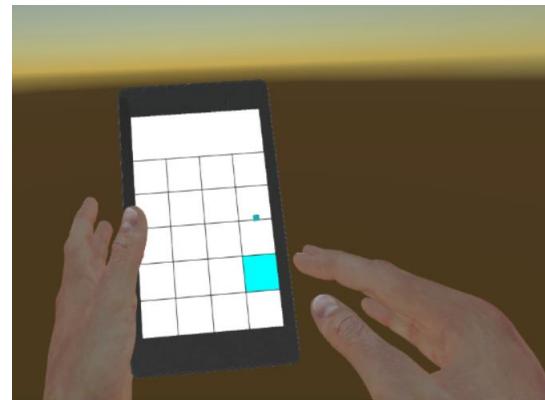


Experimental Conditions

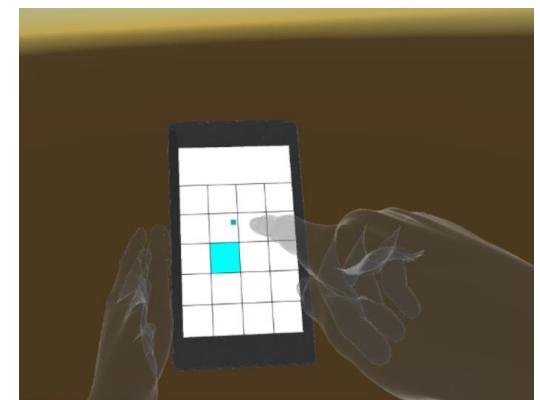
- 1- **No avatar:** the user only sees the virtual tablet;
- 2- **Realistic avatar:** a realistic representation of the hands is shown
- 3- **Translucent avatar:** a translucent hand model is used (to alleviate occlusion)



No-avatar



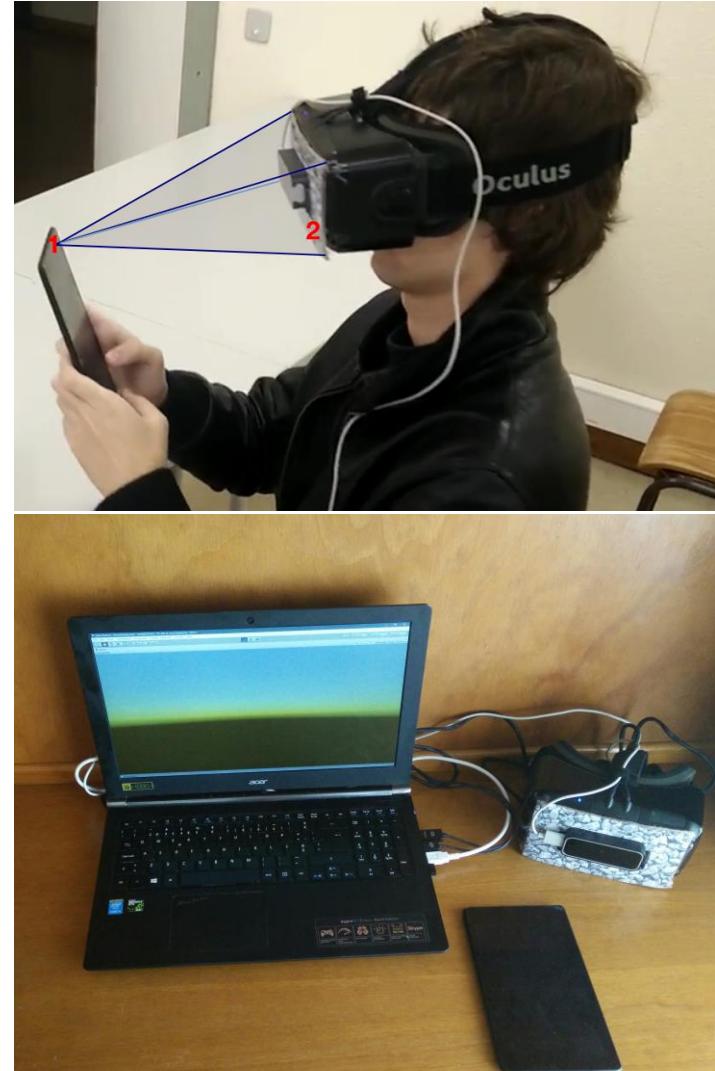
Realistic avatar



Translucent-avatar

Experimental Set-up

- Laptop running the main application (in Unity)
- HMD (Oculus Rift DK2) providing head tracking
- Tablet (Google Nexus 7) as input device running the controller application (in Unity)
- Leap Motion (mounted on the HMD) to track the user's hands
- Tablet camera tracking the position and orientation of an AR marker on the HMD to map tablet position in the virtual world (using Vuforia)



Main Results

Selection time:

Participants completed the button selections in average **faster with no-avatar (statistically significant)**

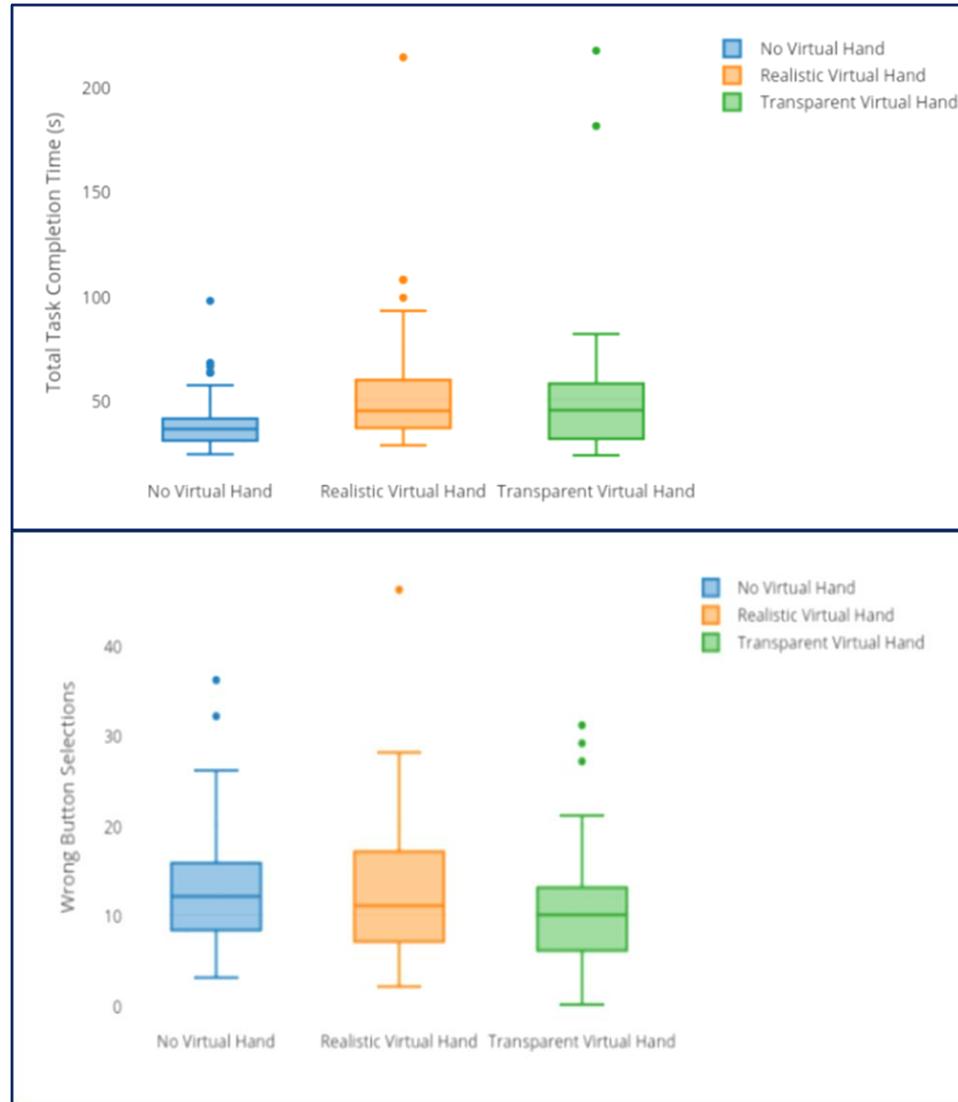
Selection errors:

Participants made slightly less errors with avatar - realistic or translucent- (statistically significant)

Participants' opinion:

The translucent avatar:

- was more often preferred
- was considered as better than the realistic avatar (statistically significant)



Bibliography for Usability evaluation – Books and links

- Gilbert Cockton, *Usability Evaluation*, Interaction Design Foundation-IxDF, 2014.
http://www.interaction-design.org/encyclopedia/usability_evaluation.html
- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, *Human-Computer Interaction*, 3rd edition, Prentice Hall, 2004
- Peter Mitchell, *A Step-by-step Guide to Usability Testing*, iUniverse, 2007

- Norman/ Nielsen Group site - <http://www.nngroup.com/articles/>
- Usability.gov site - <https://www.usability.gov/index.html>
- User focus site - <https://www.userfocus.co.uk/articles/>

Epilogue of this course

- We will have the Human in the loop for long in many situations
...
- And even when/if they are no longer in the loop

Technology shall serve the Human
(and not the other way around...)





Screen Layout Design and Color

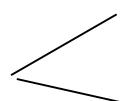


- The screen design is an important part of the UI development
- A poor screen design may degrade user performance
- Screen layout must be carefully designed
- There are numerous guidelines (we have seen already some of them)

Screen Layout Guidelines

- Several types:

General layout of information

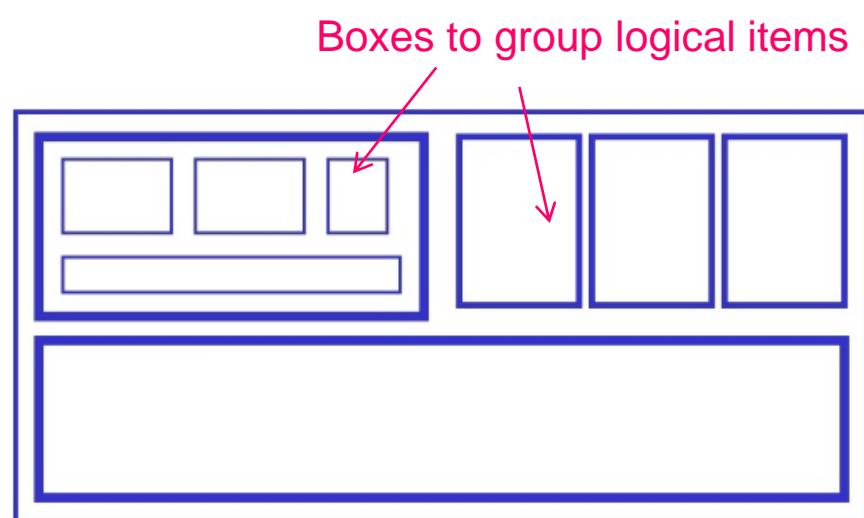
Text 
 messages
 instructions

Numbers

Coding techniques (color and others)

Information layout

- Include only the needed information
- Include all needed information
- Begin at the top left corner and align left (in Western culture)
- Group items according to type
- Leave plenty of white space
- Use leaders in multiple columns
- This is related to how humans analyse an image



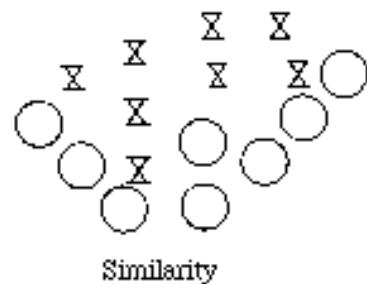
Gestalt Laws

Help understand how visual stimuli in a scene are perceived

XX XXX X XX

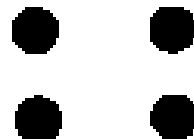
Proximity

Near stimuli are perceived as a group



Similarity

Similar stimuli tend to be grouped
(may override proximity)



Closure

Stimuli tend to be grouped
in complete figures



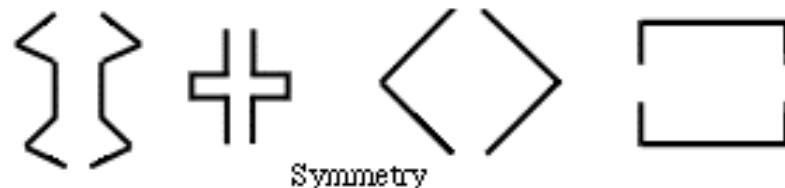
Simplicity

Ambiguous stimuli tend to be resolved
Using the simplest explanation



Good continuation

Stimuli tend to be grouped as to minimize
variations or discontinuities

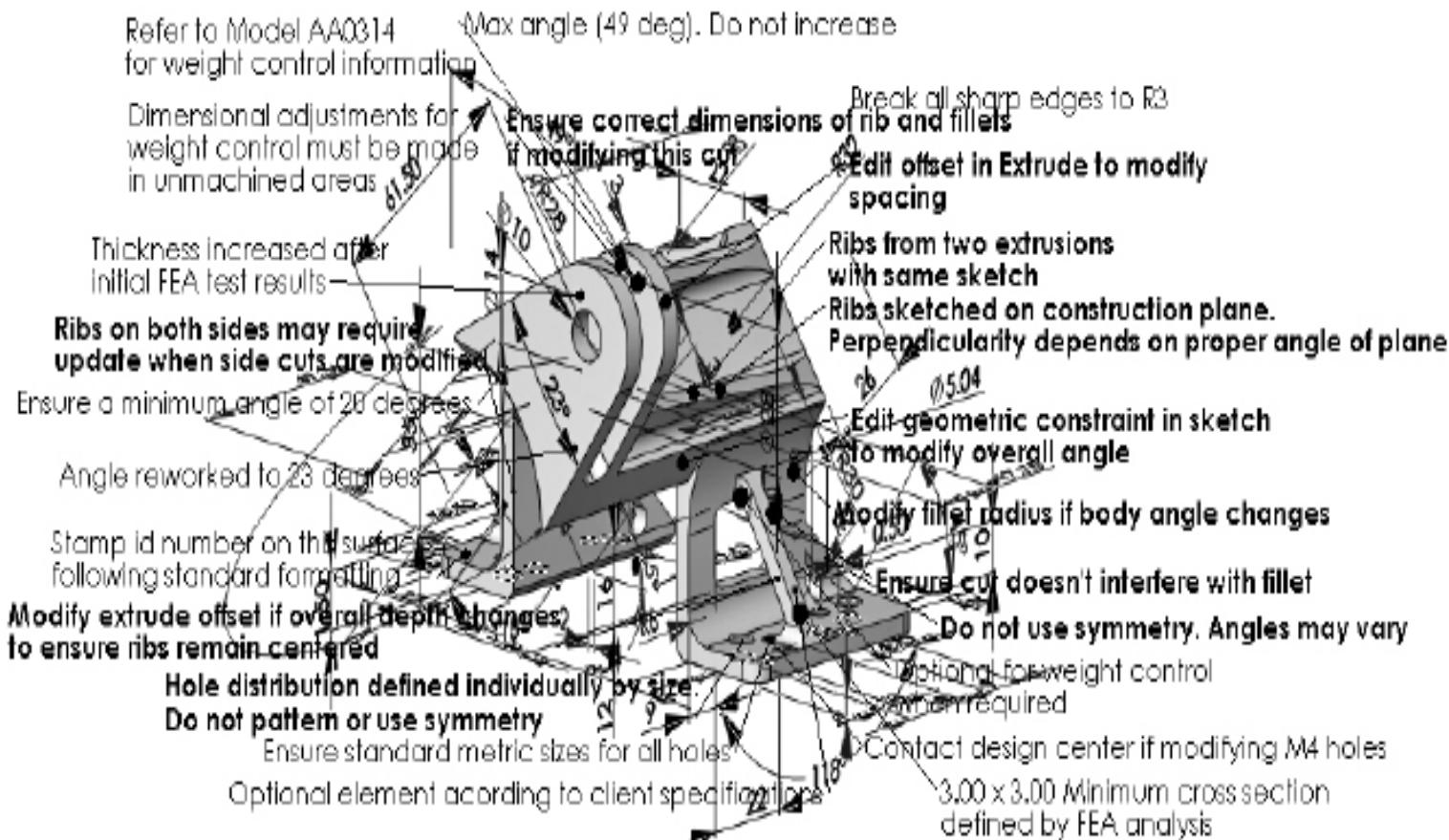


Symmetry

Regions delimited by symmetric tend
to be perceived as coherent figures

Include only the needed information

Avoid Visual Clutter



Text

- Avoid using only capital letters (are more difficult to read)
- Avoid text with many capital letters
- Do not use too many fonts for emphasis
- In multiple columns use leaders or greying

use fonts for emphasis
(but not too many)

A**C**DEF H**I**J**K**L**M**
N**O**P**Q**RSTUVWXYZ

Alcântara - Terra			12:36			13:06			13:36		
Campolide		12:15		12:41	12:45		13:11	13:15		13:41	13:45
Rossio	■	12:19			12:49			13:19			13:49
Sete Rios	■		12:19	12:43		12:49	13:13		13:19	13:43	
Entrecampos	■		12:22	12:47		12:52	13:17		13:22	13:47	
Roma - Areeiro	■		12:24	12:49		12:54	13:19		13:24	13:49	

Use greying

In multiple columns it is difficult to read across gaps:

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

use leaders

sherbert	_____	75
toffee	_____	120
chocolate	_____	35
fruit gums	_____	27
coconut dreams	_____	85

or greying

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

- Messages shall:
 - Have a detail level adequate to user knowledge and experience
 - Be specific and understandable
 - Be brief and concise
 - Be positive
 - Be helpful

Error messages

Too verbose

better

The processing of the text editor yielded 23 pages of output

Output 23 pages

Error in SIZE field

Error: SIZE range is 4 to 16

Too vague

Cannot exit before saving file

Save file before exiting

Bad/illegal file name

Maximum file name length is 8 chars

Negative

Syntax error 1542

Unmatched left parenthesis in line 210

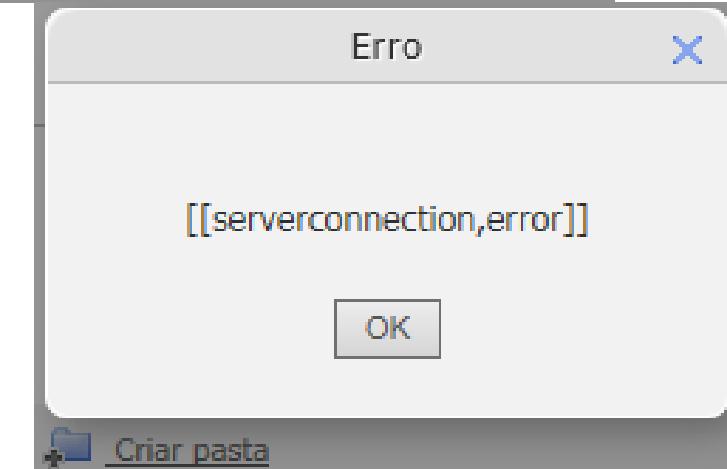
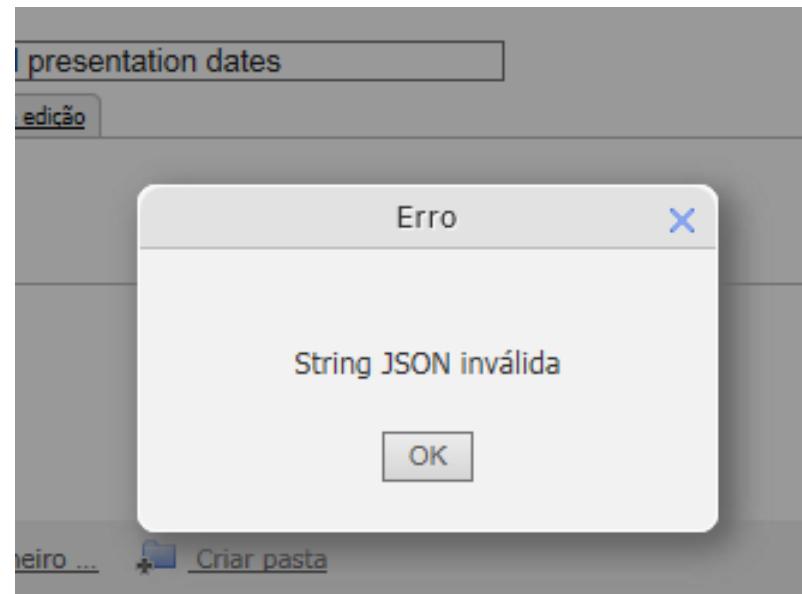
Not helpful

Examples of useless messages for users



Except (maybe) for Chinese people!

Moodle:

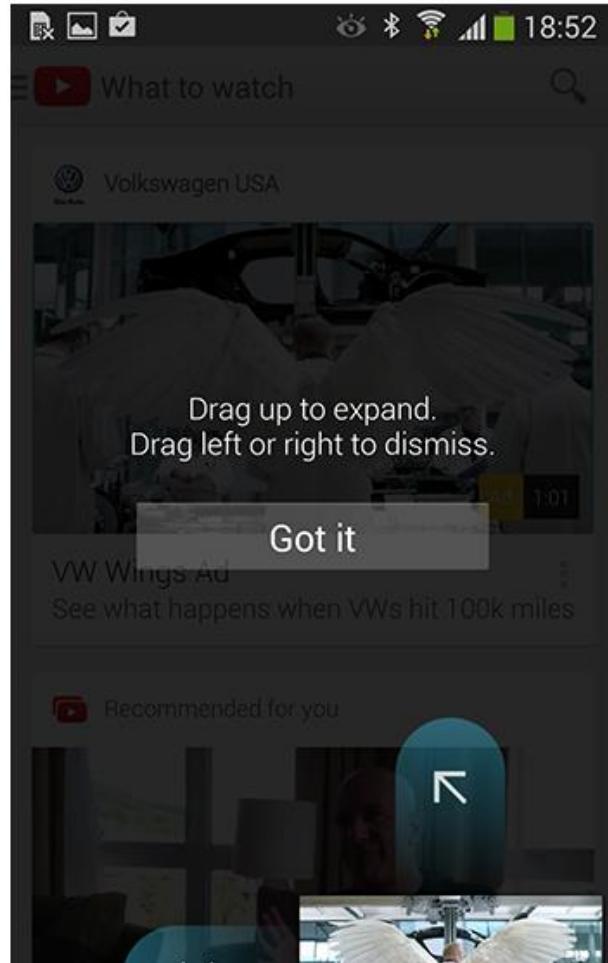


Instructional Overlays and Coach Marks for Mobile Apps

- Instructions in mobile applications must be designed for optimal scannability, as users tend to dismiss them quickly and do not read thoroughly

Main guidelines:

- Short, Focused Tips
- Avoid Chains of Tips
- Use Visuals When Possible
- Keep Tips Sparse



Numbers

- Integers shall be right justified
- Real numbers shall be aligned by the decimal point
- Avoid unnecessary zeros (at left)
- Long numbers shall be divided in groups of 3 or 4



Which is the largest?

532.56	627.865	
179.3	1.005763	75
256.317	382.583	120
15	2502.56	35
73.948	432.935	27
1035	2.0175	85
3.142	652.87	
497.6256	56.34	Right align integers

Align decimal points

Numbers

Better

10 100 1000 10000	10 100 1000 10000
100.00 25.365 5432.01 1.45591	100.00 25.365 5432.01 1.45591
10:1 p.m. 002	10:02 p.m. 2
6173954686	617-395-4686

Coding techniques

Blinking

Bold

Size

Font

Underlining

Shape

Special characters and icons

Proximity

Borders

Sound

Colour

Main guideline: use parsimoniously any coding technique!

Specific problems for different platforms: mobile

- Many guidelines are similar for mobile and desktop design, but their mobile interpretation is much more unforgiving
 - Context of use
 - Size of screen
 - Platform limitations



<http://www.nngroup.com/articles/mobile-sharpens-usability-guidelines/>

<https://developer.android.com/design/index.html>

Links on tablet and mobile usability

Raluca Budiu, The State of Mobile User Experience, NNGroup, March, 2015

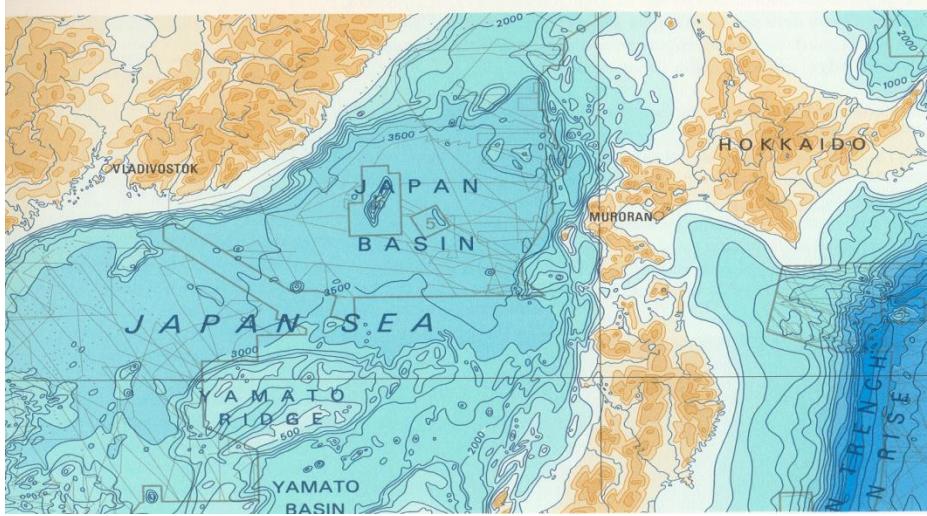
<http://www.nngroup.com/articles/mobile-usability-update/>

<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/>

<https://developer.android.com/design/index.html>

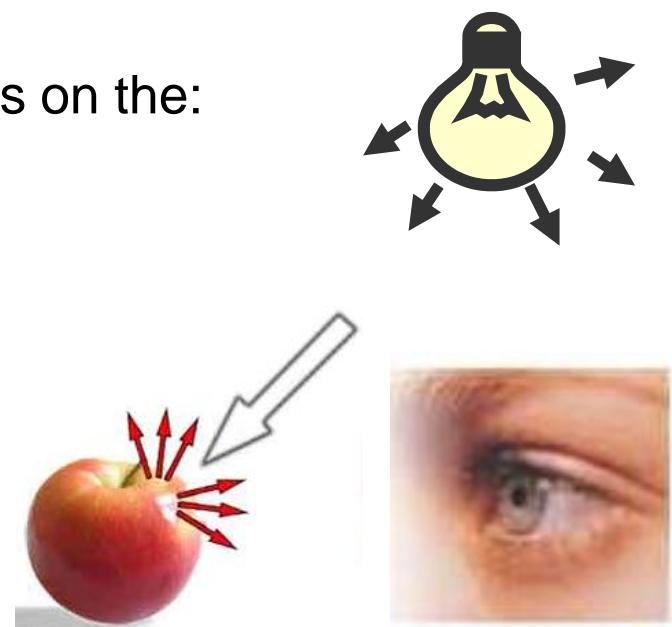


Color usage

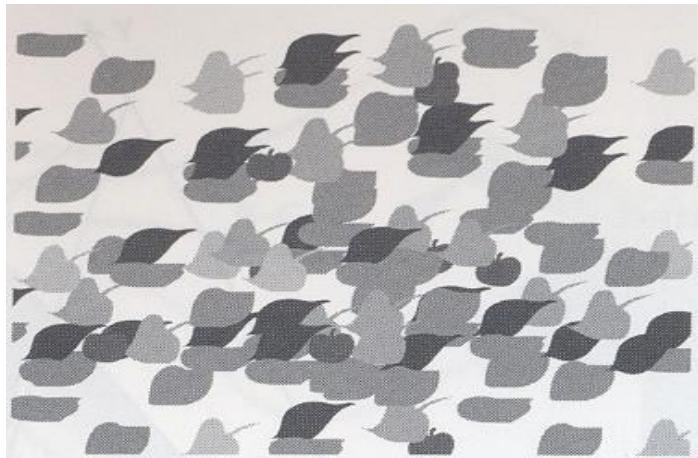


Color

- Color is a complex and multidisciplinary subject:
 - Physics
 - Physiology and psychology
 - Art and graphic design
 - Interactive systems design
- The perceived color of an object depends on the:
 - Material characteristics
 - Illumination
 - Ambient color
 - Human visual system



How many cherries?



(Ware, 2004)

How many cherries?



Color may support users in many tasks!
(yet, if not properly used may make them more difficult!)

Using color

Besides increasing realism, it may have the following **advantages**:

It may:

- Show the logical organization of the information displayed
- Represent approximate values
- Catch the attention
- Increase satisfaction
- Ease the search in complex displays
- Trigger emotions

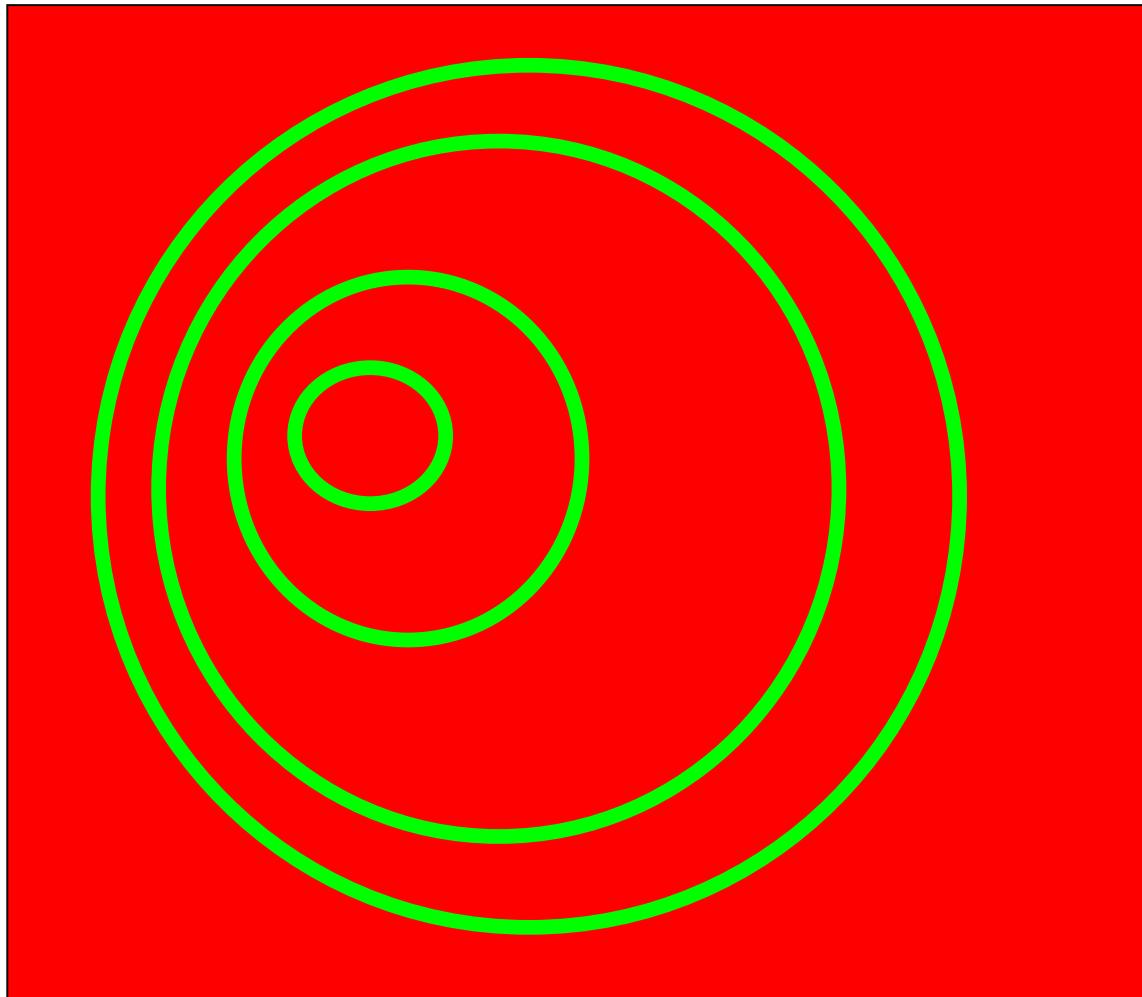
...

However, **it may degrade user's performance** if not used properly

Guidelines for using color

- **Use color parsimoniously**
- Use a limited number of colors
- Firstly make it work without color
- Use color coherently
- Avoid using simultaneously several saturated colors
- Do not convey information solely through color
- Make color coding support the user task
- Make the color coding as obvious as possible
- Allow the user to control the color code
- Take into account the cultural meaning of colors ...





Saturated complementary colors should not be used simultaneously

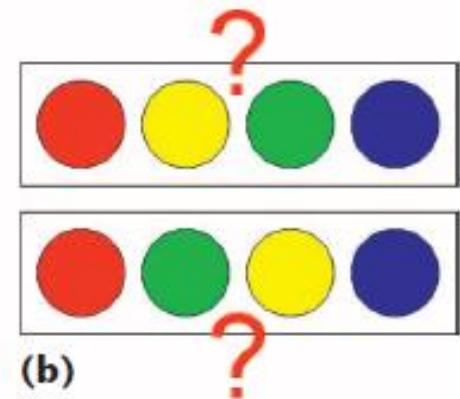
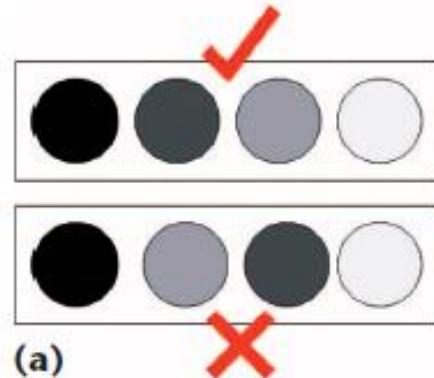


Small spots of color on a neutral background enhance relevant information

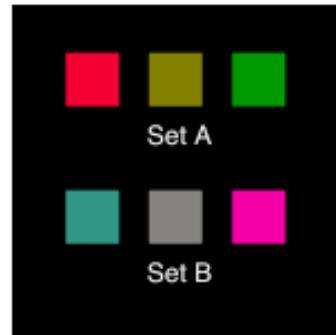
(Tufte, 1990) 31

- Do not expect to easily perceive order from color

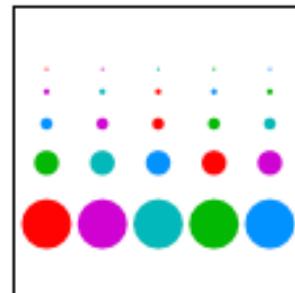
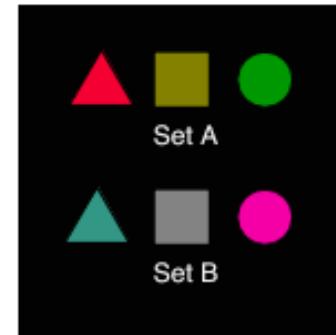
(Borland, Taylor II, 2007)



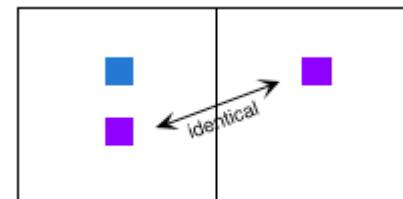
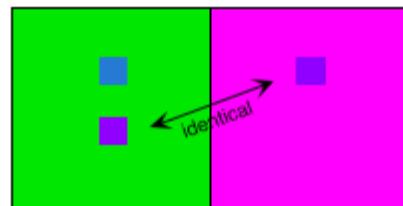
The elements within these sets look identical to deuteranopes, the most common kind of dichromat:



These can be discriminated on the basis of non-color differences:



Don't use colour coding on small elements



Use neutral gray surrounds where color judgments are critical.

Color Vision deficiencies

- $\approx 8\%$ of men and 1% of women have some type of color vision deficiency
- Generally it is genetic (associated to the X chromosome)
- Common deficiencies are explained by the lack of cones (color sensor cells in the retina) sensitive to the long and medium λ (dicromacies):
 - Protanopia (LW – “Red” cone)
 - Deutanopia (MW – “Green” cone) (Daltonism)
- There are three types of inherited deficiencies:
 - Monocromacy (disorder or lack of all color sensitivity)
 - Dicromacy (disorder or lack of one type of cone)
 - Anomalous Tricromacy (disorder in cones)

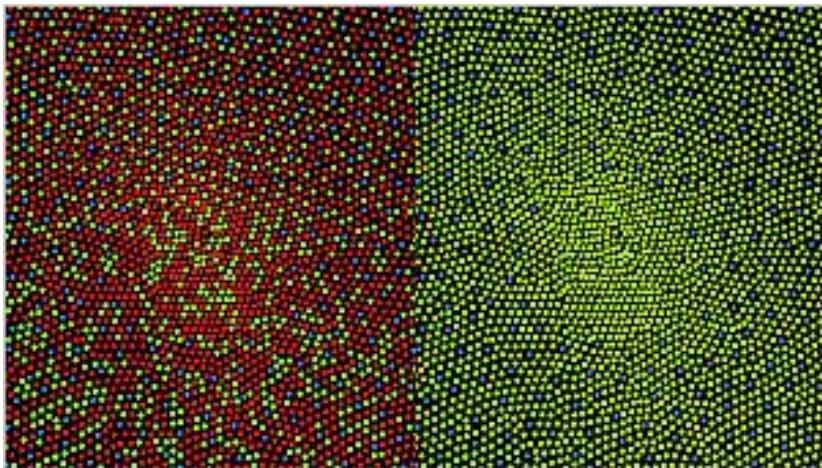


Illustration of the distribution of cone cells in the fovea of an individual with normal color vision (left), and a color blind (protanopic) retina. Note that the center of the fovea holds very few blue-sensitive cones.

http://en.wikipedia.org/wiki/Photoreceptor_cell

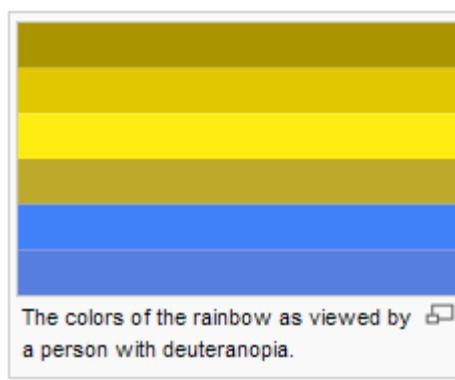
Rainbow colors as viewed by people suffering from color vision deficiencies



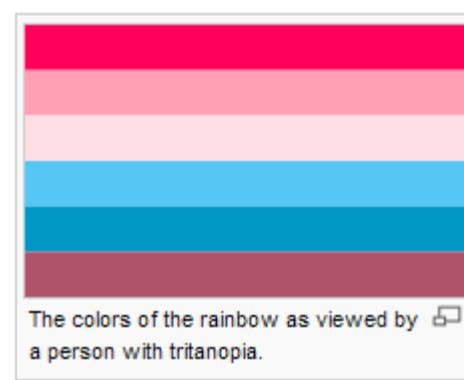
The colors of the rainbow as viewed by a person with no color vision deficiencies.



The colors of the rainbow as viewed by a person with protanopia.



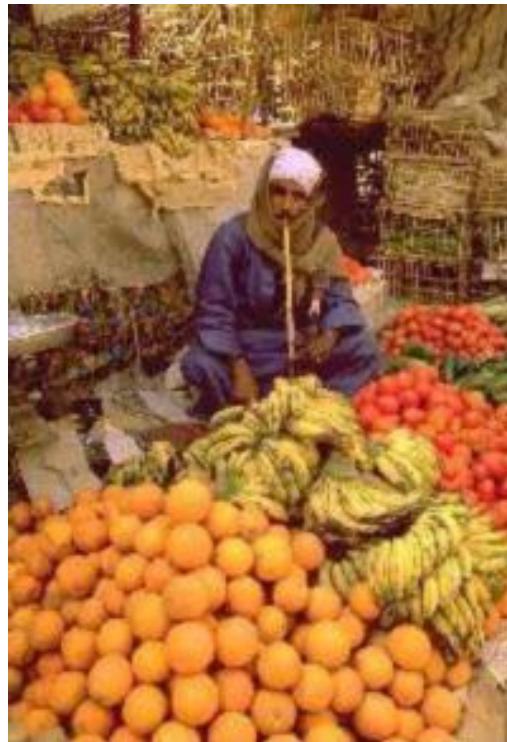
The colors of the rainbow as viewed by a person with deutanopia.



The colors of the rainbow as viewed by a person with tritanopia.

http://en.wikipedia.org/wiki/Color_blindness

Simulating color vision deficiencies



Original image as seen
by a normal observer

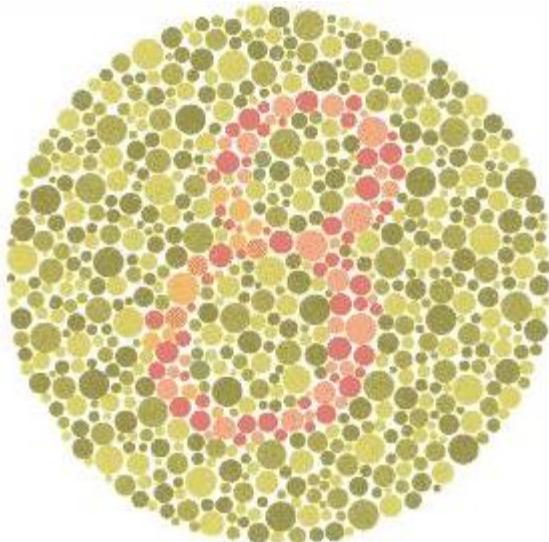


As seen by a deuteranope
(daltonic)

<http://www.daltonize.org/>

Simulating color vision deficiencies

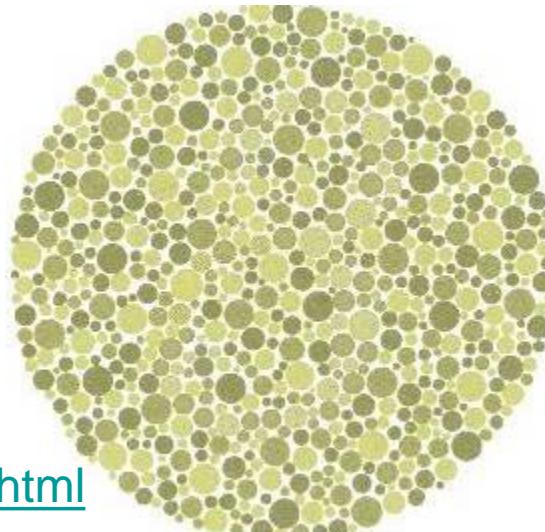
Ishihara-2



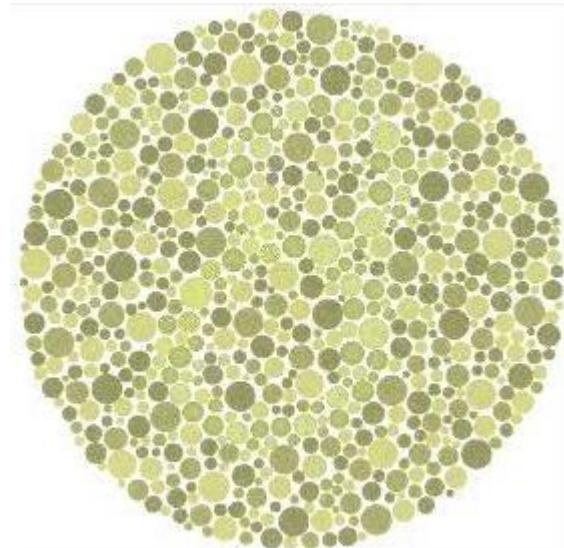
Original image as seen
by a normal observer

As seen by an observer
with a color vision
deficiency:

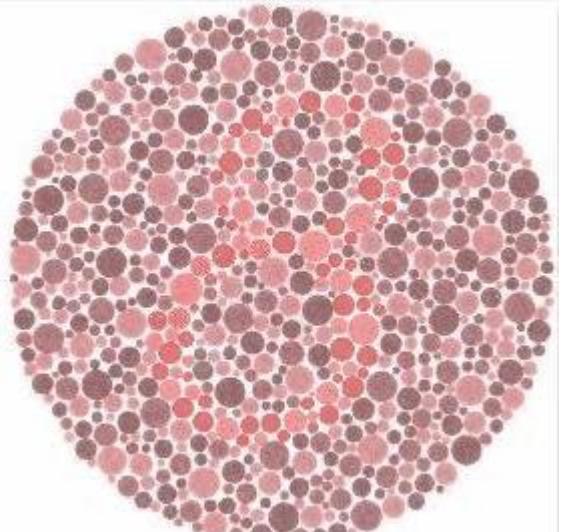
Red-Blind/Protanopia



Green-Blind/Deuteranopia

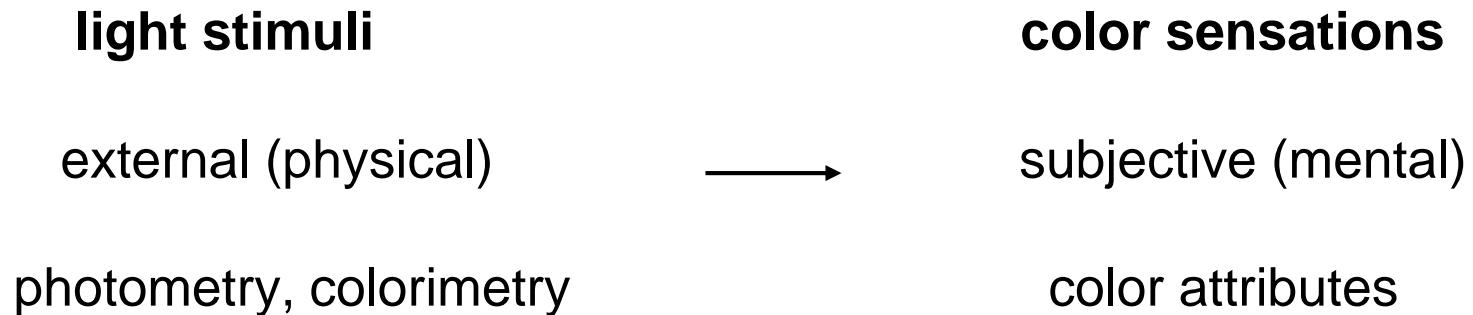


Blue-
Blind/Tritanopia



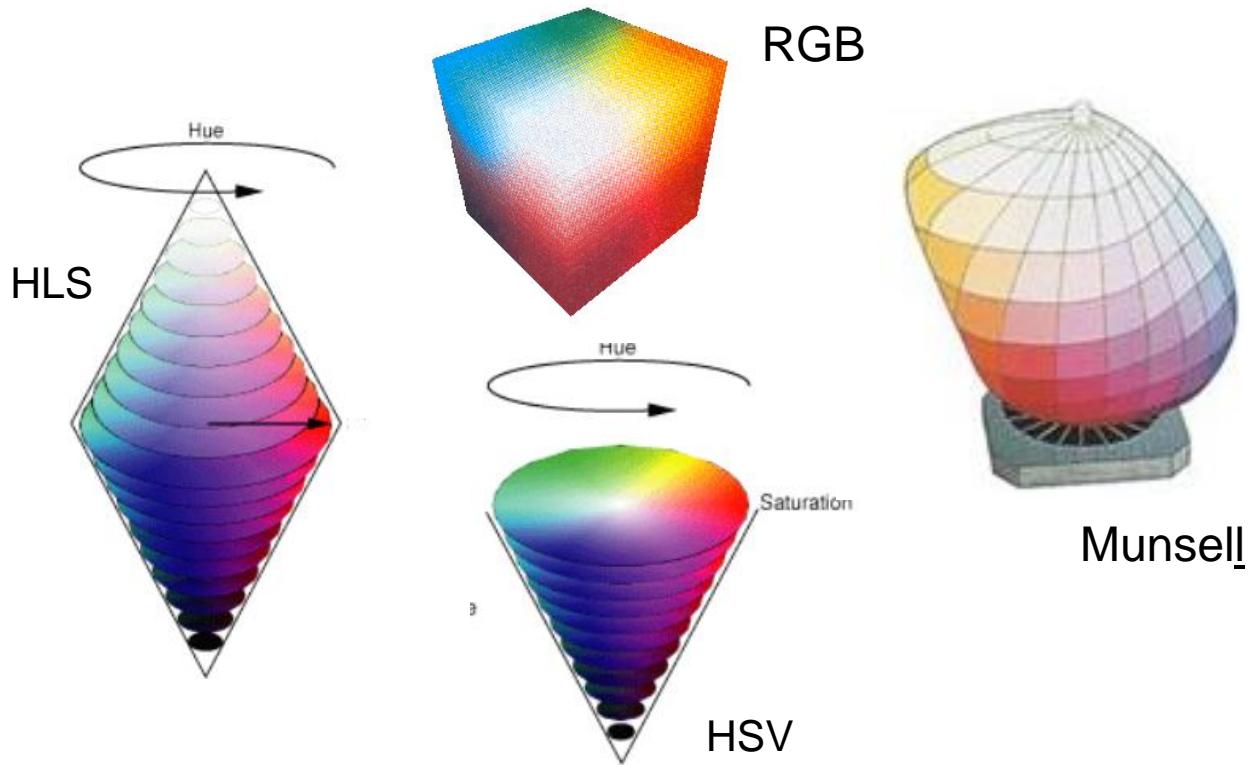
How can we describe color experience?

- Color perception happens in the mind due to light properties
- Different color descriptions are necessary for:



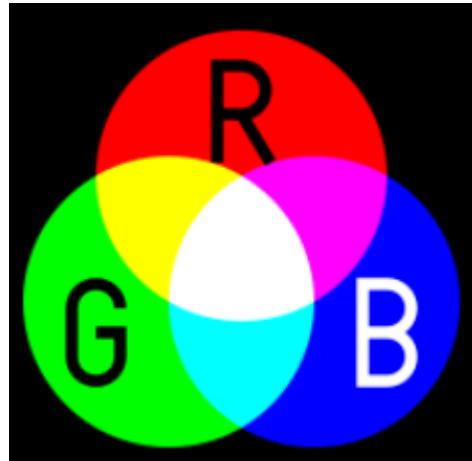


Color Models



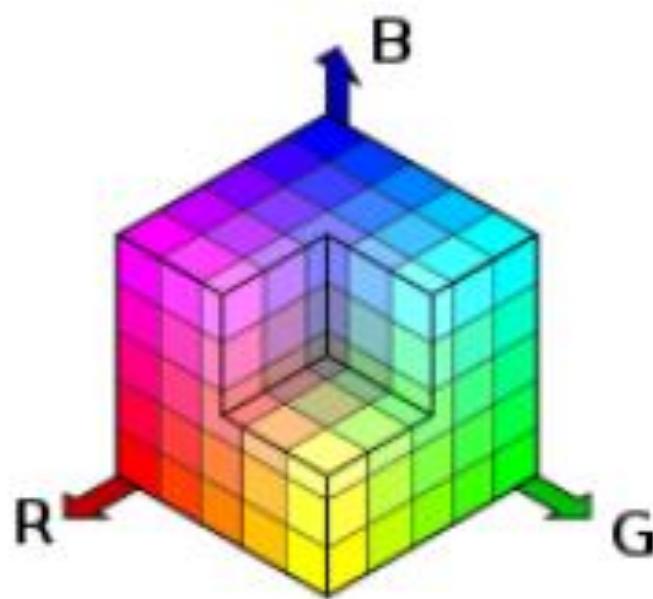
- Objects are perceived as having a color depending on the spectrum of the reflected light (or emitted)
- But different spectra may induce similar color sensations
- It is important to be able to describe color objectively
- There are two types of color production systems:
 - Additive (e.g.: monitors, TV sets, projectors) → RGB
 - Subtractive (e.g.: printers) → CMY
- RGB and CMY are H/W oriented color models not adequate for users
- There are more color models ...

The RGB color model:



The **RGB color model** is an additive color model in which red, green, and blue light (the primary colors) are added to reproduce a broad array of colors.

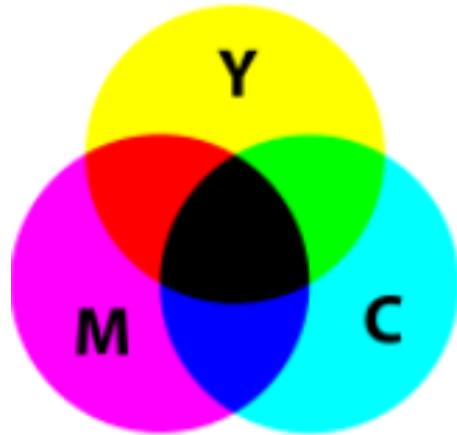
The color space is a cube in a Cartesian coordinate system



White -> 1, 1, 1

Black -> 0, 0, 0

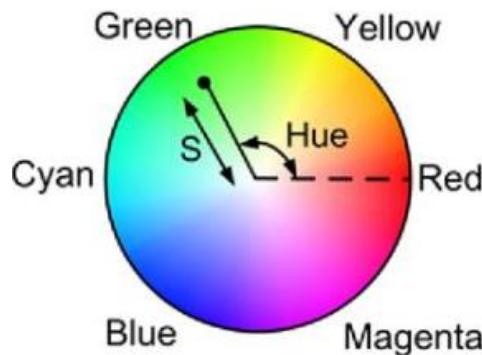
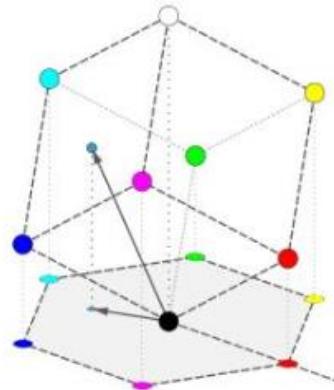
https://en.wikipedia.org/wiki/RGB_color_model



The **CMY color model** is a subtractive color model in which cyan, magenta, and yellow (the primary colors) are subtracted from white to reproduce a broad array of colors.

The color space is also a cube in a Cartesian coordinate system

White $\rightarrow 0, 0, 0$
 Black $\rightarrow 1, 1, 1$



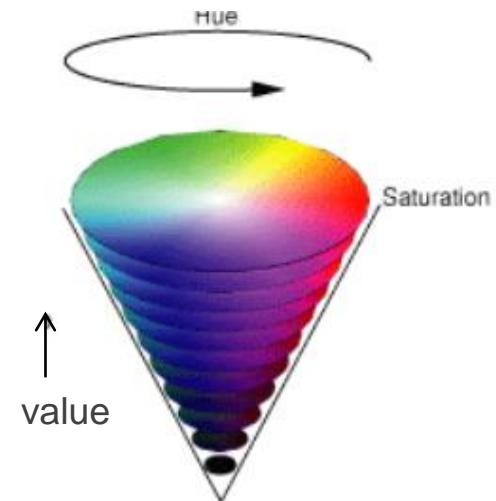
There are other models more adequate to color specification by the users:

- HSV
- HLS

- Humans describe color based on 4 psychophysical variables related to physical variables:
 - Hue – the degree to which is similar to or different from stimuli that are described as red, green, blue, and yellow
 - Saturation – related to the amount of achromatic light
 - Lightness – related to the objects reflectance (for reflecting objects)
 - Brightness – for light emitting objects



<https://en.wikipedia.org/wiki/Hue>



- HSV color model:

- Hue

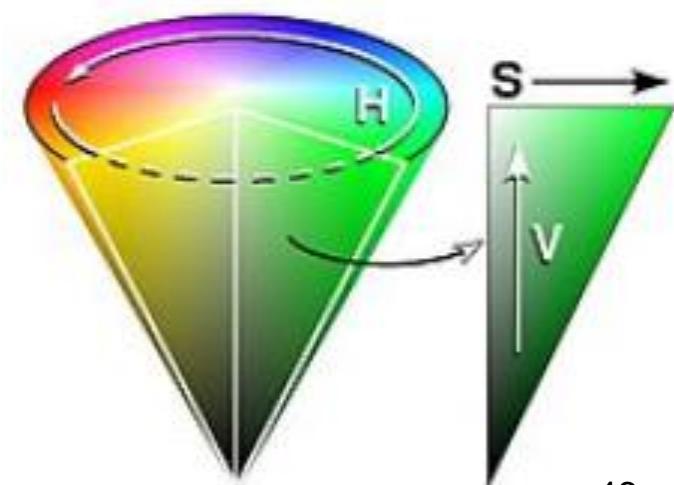


- Saturation – related to the amount of achromatic light
 - Value - controls the brightness: 0% - pure black 100% - pure white

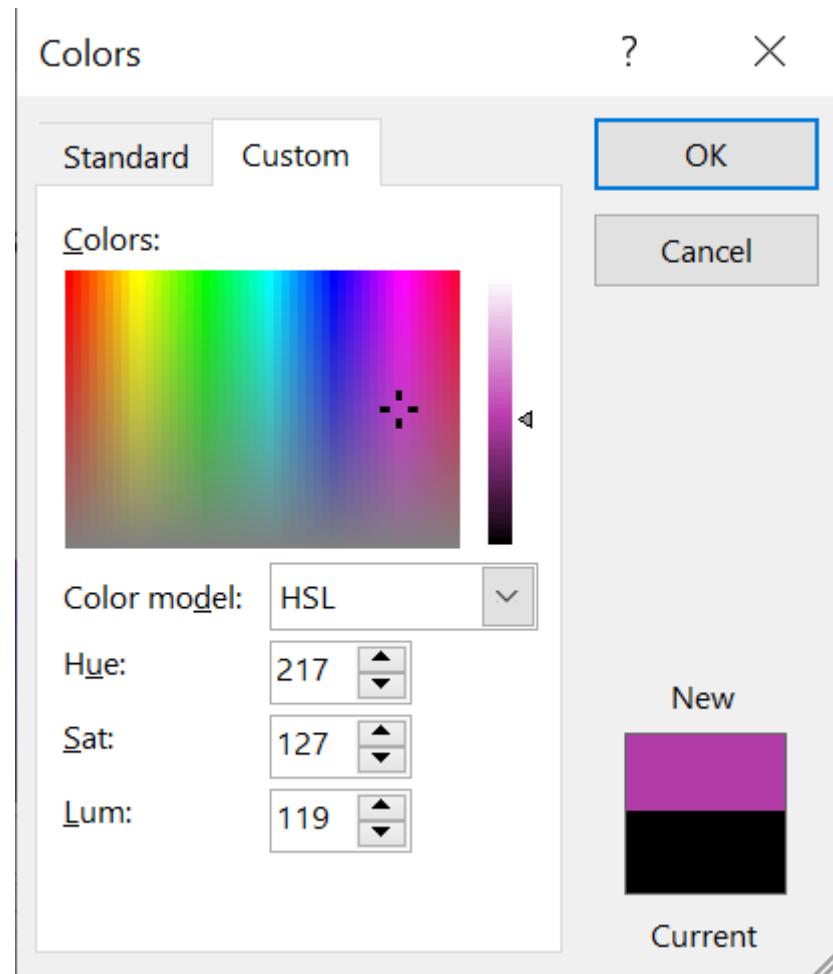
Uses cylindrical coordinates

<https://www.khanacademy.org/partner-content/pixar/color/color-101/v/color-3>

<https://programmingdesignsystems.com/color/color-models-and-color-spaces/index.html>



- Let the user select a color:



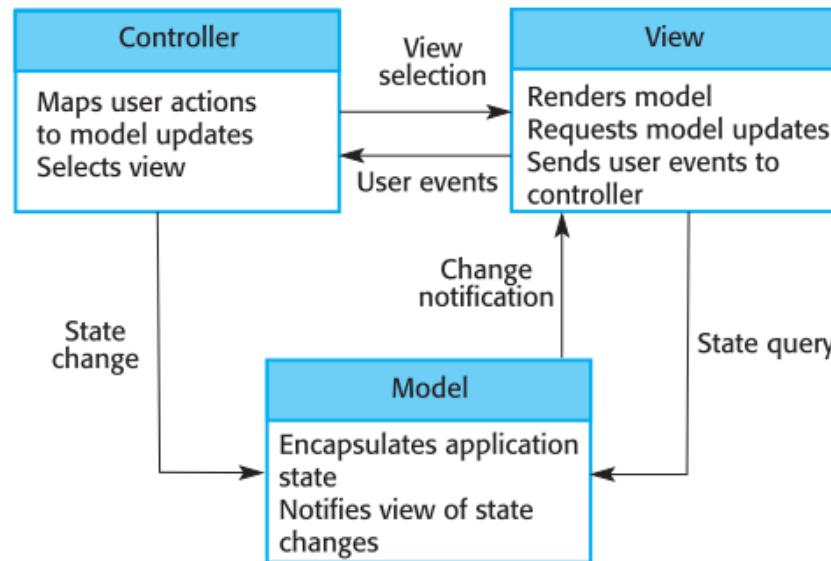
Interesting Links

- Introduction to color guidelines and standards (NASA)

http://colorusage.arc.nasa.gov/guidelines_0.php



S/W Patterns for Interactive Systems



- A complete application consists of:
 - User interface (UI)
 - Core functions of the application
- How can we effectively develop interactive applications with these two parts?
- Follow an established development methodology suited for interactive systems:

Architectural patterns

S/W Patterns – historical perspective

- 1987
 - First small pattern language for **designing user interfaces** by K. Beck and W. Cunningham (inspired by building and urban architect C. Alexander)
- 1993
 - Pattern Languages of Programming (PLoP) conference series by The Hillside Group
<https://www.hillside.net/conferences>
- 1994
 - "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma et al. (aka the Gang of Four (GoF))
<https://hillside.net/patterns/about-patterns>

S/W Patterns

- Proven **solution to a problem in a context** (Gamma et al., 1994)
- Each pattern documents a reusable solution, encapsulates knowledge about successful practices, and provides information about its usefulness and tradeoffs
- Many companies have written pattern collections:
 - Amazon,
 - Google,
 - IBM,
 - Microsoft,
 - Oracle,
 - Siemens, etc.

<https://hillside.net/patterns/about-patterns>

Content of a Pattern

- Different formats are used for describing patterns, generally including:
 - **Name**
 - **Problem**
 - **Context**
 - **Solution**
 - Forces
 - Resulting Context
 - Examples
 - Rationale
 - Related Patterns
 - Known Uses
- May include an Abstract providing an overview of the pattern and indicating the types of problems it addresses and the target audience
<https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap22.html>

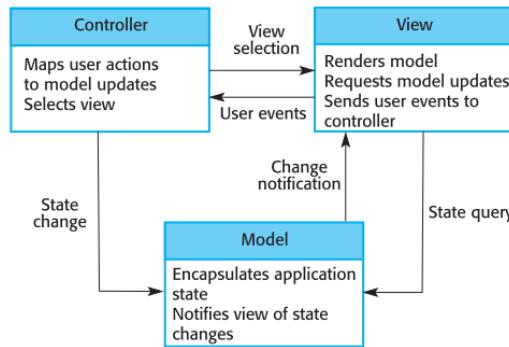
Forces of a Pattern

- Security, robustness, reliability, fault-tolerance
- Efficiency, performance, throughput, bandwidth requirements, space utilization
- Ease-of-use
- Ease-of-construction
- Completeness and correctness
- Scalability (incremental growth on-demand)
- Extensibility, evolvability, maintainability
- Modularity, independence, re-usability, openness, composability (plug-and-play), portability
- etc.

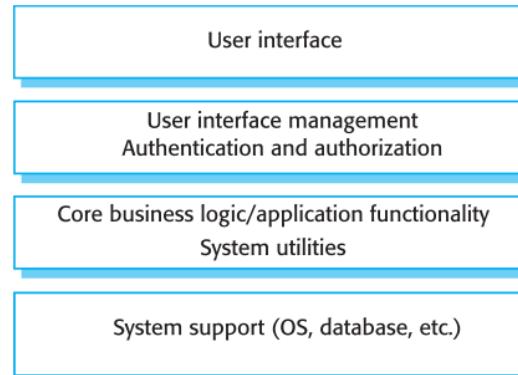
<https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap22.html>

Architectural patterns often used in interactive s/w

- Model View Control (MVC)



- Layered architecture



(Sommerville, 2010)

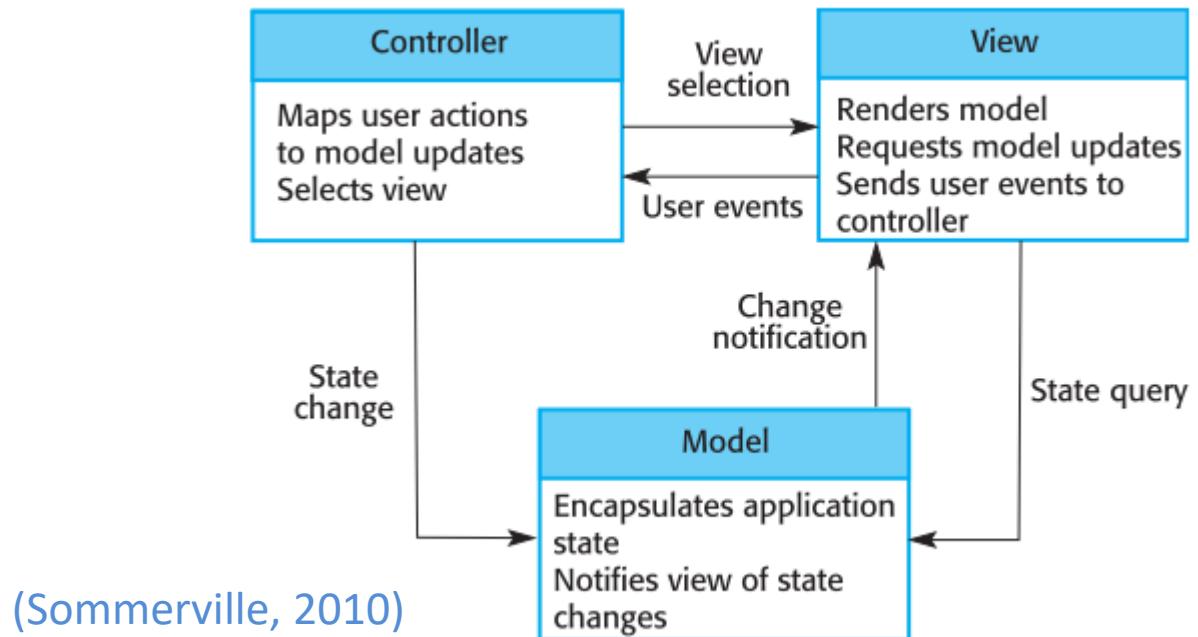
Model–view–controller (MVC)

- MVC is an architectural pattern/paradigm commonly used for interactive systems
- Was proposed in the 70's as a computational architecture for interactive programs by the designers of SmallTalk (one of the first object-oriented and modular languages)
- It expresses the "core of the solution" to a problem while allowing it to be adapted for each system

Model–view–controller (MVC)

- The application is divided in three parts:

- Model
- View
- Controller



- Separating internal representations of information from the ways it is presented to and accepted from the user
- It decouples these major components allowing for efficient code reuse and parallel development.

- Originally developed for desktop computing, MVC has been widely adopted as an architecture for Web applications (and others...)
- Frameworks vary in their interpretations in the way that the MVC responsibilities are divided between the client and server
- Some web MVC frameworks take a thin client approach that places almost the entire model, view and controller logic on the server: the model exists entirely on the server
- Other frameworks allow the MVC components to execute partly on the client

MVC Components

- The ***model*** is the central component of the pattern:
It expresses the application's behavior in terms of the problem domain, independent of the UI. It manages the data, logic and rules of the application
- A ***view*** can be any output representation of information:
Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants
- The ***controller*** accepts input and converts it to commands for the model or view

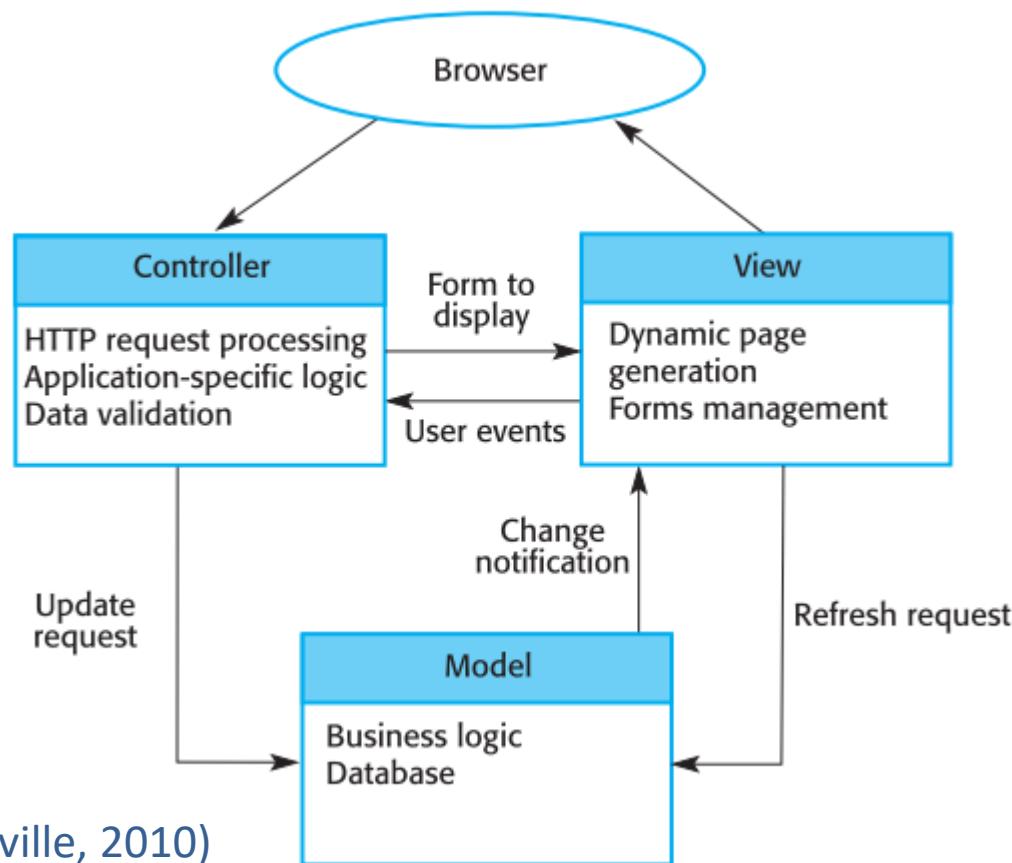
MVC Interactions

- This design defines the interactions among the three components
- The model is responsible for managing the data of the application. It receives user input from the controller
- The view means presentation of the model in a particular format
- The controller is responsible for responding to the user input and perform interactions on the data model objects. The controller receives the input, optionally validates the input and then passes it to the model

Name	MVC (Model-View-Controller)
Description	Separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction (e.g., key presses, mouse clicks, etc.) and passes these interactions to the View and the Model. See Figure 6.5.
Example	Figure 6.6 shows the architecture of a web-based application system organized using the MVC pattern.
When used	Used when there are multiple ways to view and interact with data. Also used when the future requirements for interaction and presentation of data are unknown.
Advantages	Allows the data to change independently of its representation and vice versa. Supports presentation of the same data in different ways, with changes made in one representation shown in all of them.
Disadvantages	May involve additional code and code complexity when the data model and interactions are simple.

Example:

Web application architecture using MVC



(Sommerville, 2010)

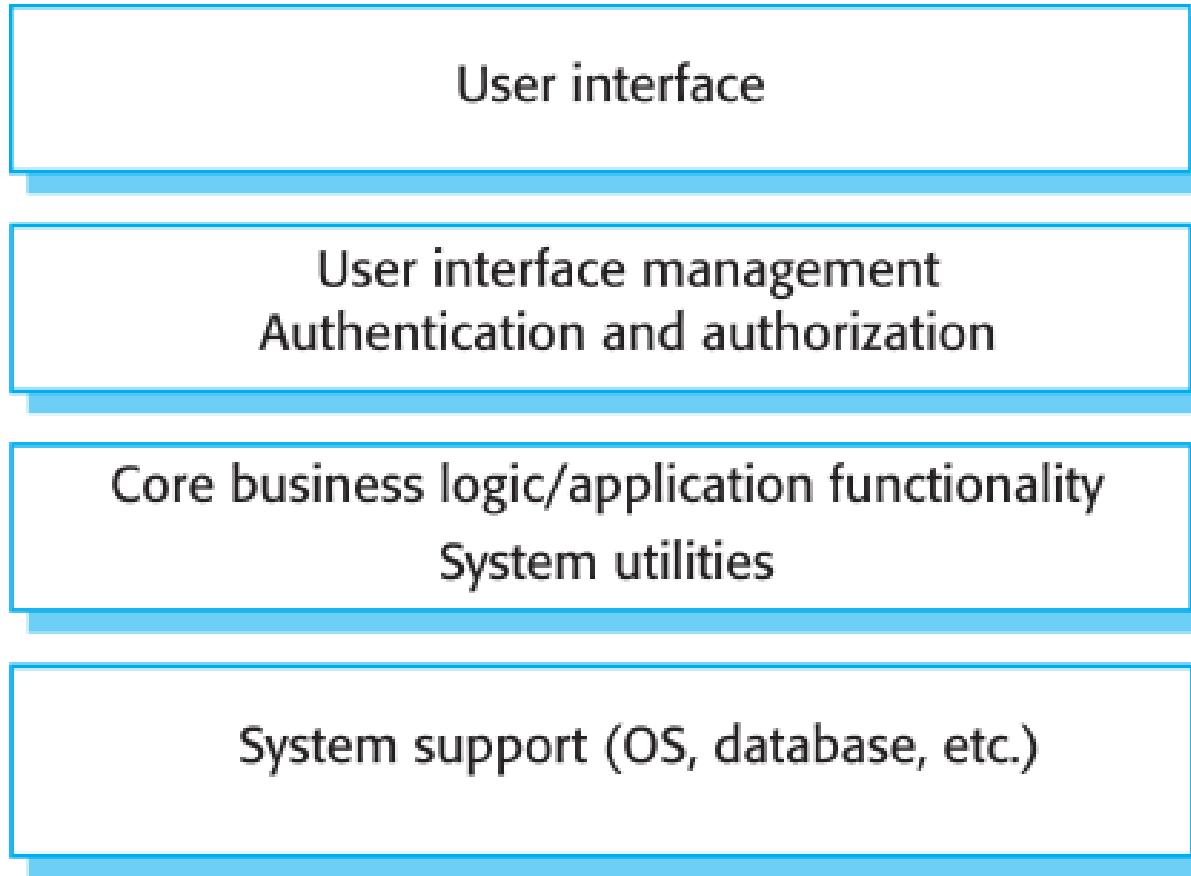
Layered Architecture

- Separation and independence are fundamental to S/W architectural design because they allow changes to be localized
- The Layered Architecture pattern is another way of achieving separation and independence
- The system functionality is organized into separate layers
- Each layer only relies on the facilities and services offered by the layer immediately beneath it
- It supports the incremental development of systems.
- The architecture is also changeable and portable.

Name	Layered architecture
Description	Organizes the system into layers, with related functionality associated with each layer. A layer provides services to the layer above it, so the lowest level layers represent core services that are likely to be used throughout the system. See Figure 6.8.
Example	A layered model of a digital learning system to support learning of all subjects in schools (Figure 6.9).
When used	Used when building new facilities on top of existing systems; when the development is spread across several teams with each team responsibility for a layer of functionality; when there is a requirement for multilevel security.
Advantages	Allows replacement of entire layers as long as the interface is maintained. Redundant facilities (e.g., authentication) can be provided in each layer to increase the dependability of the system.
Disadvantages	In practice, providing a clean separation between layers is often difficult, and a high-level layer may have to interact directly with lower-level layers rather than through the layer immediately below it. Performance can be a problem because of multiple levels of interpretation of a service request as it is processed at each layer.

(Sommerville, 2010)

Example: A generic layered architecture



(Sommerville, 2010)

Main bibliography

- Gamma, R., Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1994
- <https://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612>
- Kim, G., *Human–Computer Interaction- Fundamentals and Practice*. CRC Press, 2015
- Sommerville, I., *Software Engineering*, 10th ed., Pearson Education, 2016
<https://www.amazon.com/Software-Engineering-10th-Ian-Sommerville/dp/0133943038>

Prototyping



Paulo Dias, Beatriz Sousa Santos

Low fidelity prototypes

- Why?
 - Get feedback earlier, cheaper
 - Obtain users feedback concerning general aspects
 - Easy to modify and throw away even during user tests
 - Valuable to test the UI conceptual model

1

universidade de aveiro



2

Low fidelity prototypes



- Specifically adequate to get feedback concerning
 - Concepts and terminology
 - Navigation
 - Contents
 - Functionality

3

Low fidelity prototypes



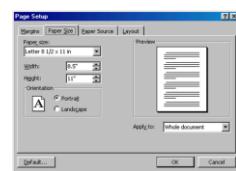
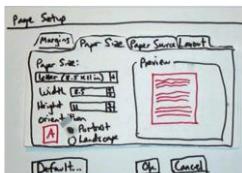
- It does not need to have much detail, nor to be very realistic, e.g.:
 - Text may be replaced by some lines
 - Images may be replaced by words
 - In general no colour is needed
 - Sizes of windows, fonts, etc. don't need to be final

4

3

4

Comparing Fidelity of Look & Feel



5

Paper Prototype



- Interactive paper mockup
 - Sketches of screen appearance
 - Paper pieces show windows, menus, dialog boxes
- Interaction is natural
 - Pointing with a finger = mouse click
 - Writing = typing
- A person simulates the computer's operation
 - Putting down & picking up pieces
 - Writing responses on the "screen"
 - Describing effects that are hard to show on paper
- Low fidelity in look & feel
- High fidelity in depth (person simulates the backend)

6

5

6

1

Why Paper Prototyping?



- Faster to build
 - Sketching is faster than programming
- Easier to change
 - Easy to make changes between user tests, or even *during* a user test
 - No code investment— everything will be thrown away (except the design)
- Focuses attention on big picture
 - Designer does not waste time on details
 - Customer makes more creative suggestions
- Nonprogrammers can help
 - Only simple skills are required

7

7

Tips for Good Paper Prototypes

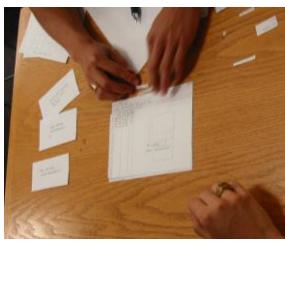


- Make it larger than life
- Make it monochrome
- Replace tricky visual feedback with audible descriptions
 - Tooltips, drag & drop, animation, progress bar
- Keep pieces organized
 - Use folders & open envelopes

9

9

Size Matters



11

11

Tools for Paper Prototyping

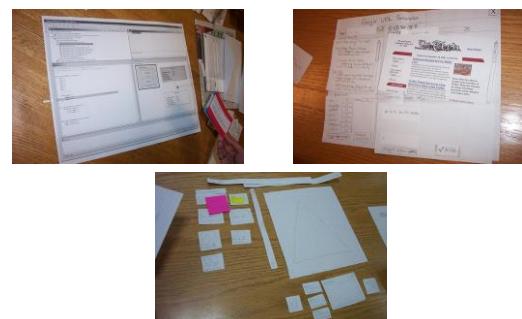


- White poster board
 - For background, window frame
- Big (unlined) index cards
 - For menus, window contents, and dialog boxes
- Restickable glue
 - For keeping pieces fixed
- Post Its
 - For text fields, checkboxes, short messages
- Overhead transparencies
 - For highlighting, user “typing”
- Photocopier
 - For making multiple blanks
- Pens & markers, scissors, tape

8

8

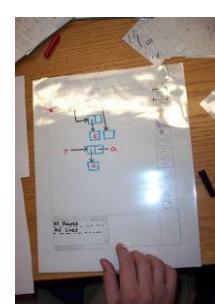
Hand-Drawn or Not?



10

10

Post-it Glue and Transparencies are Good



12

12

2

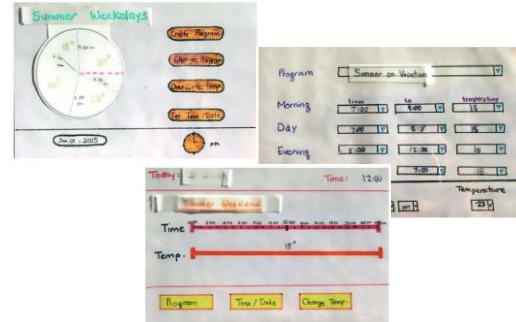
Low-Fidelity Prototypes Aren't Always Paper



<http://www.designinginteractions.com/interviews/JeffHawkins>

13

Multiple Alternatives Generate Better Feedback



"Getting the Right Design and the Design Right: Testing Many Is Better Than One."
CHI 2006.

14

How to prepare and use a paper prototype

- Draft the conceptual model
- Draw the screens, menus, dialog-boxes, messages etc. needed
- Prepare the test protocol
- Perform the user tests (see next slide)
- When needed change the prototype



15

15

How to prepare and use a paper prototype

- Perform the user tests – 3 Roles
 - Computer
 - Simulates system
 - No additional feedback from computer
 - Facilitator
 - Presents interface and tasks to the user
 - Encourages user to "think aloud" by asking questions
 - Keeps user test from getting off track
 - Observer
 - Don't talk
 - Takes copious notes



16

16

What You Can Learn from a Paper Prototype

- Conceptual model
 - Do users understand it?
- Functionality
 - Does it do what's needed? Missing features?
- Navigation & task flow
 - Can users find their way around?
 - Are information preconditions met?
- Terminology
 - Do users understand labels?
- Screen contents
 - What needs to go on the screen?

17

17

What You Can't Learn

- Look: color, font, whitespace, etc
- Feel: efficiency issues
- Response time
- Are small changes noticed?
 - Even the tiniest change to a paper prototype is clearly visible to user
- Exploration vs. deliberation
 - Users are more deliberate with a paper prototype; they don't explore or thrash as much

18

18

Change blindness

19



Example of change blindness
(Spence, 2007)

19

Change blindness

20



Example of change blindness
(Spence, 2007)

20

Change blindness example (Spence, 2007)

21



- Inattentional blindness (monkey business)
https://www.youtube.com/watch?v=lGQmdoK_Zhy
- Change blindness (Person swap)
http://www.youtube.com/watch?v=vBPG_OBgTWg&feature=related

21

Change blindness

22



Example of change blindness
(Spence, 2007)

22

Change blindness

23



Example of change blindness
(Spence, 2007)

23

Change blindness example

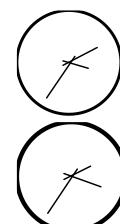
24

- A single number can be difficult to represent ensuring a user is made aware of it

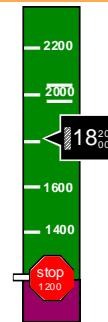
Example: the altimeter
(Spence, 2007)



The original aircraft altimeter,
responsible for many accidents



Two altimeter representations
easily assumed to be the same
due to change blindness



The modern
aircraft altimeter,

23

24

What You Can't Learn



- Look: color, font, whitespace, etc
- Feel: efficiency issues
- Response time
- Are small changes noticed?
 - Even the tiniest change to a paper prototype is clearly visible to user
- Exploration vs. deliberation
 - Users are more deliberate with a paper prototype; they don't explore or thrash as much

Several studies have shown that low-fidelity prototypes identify substantially the same usability problems as high-fidelity prototypes

25

25

Computer Prototype



- Interactive software simulation
- High-fidelity in look & feel
- Low-fidelity in depth
 - Paper prototype had a human simulating the backend; computer prototype doesn't
 - Computer prototype may be **horizontal**: covers most features, but no backend

26

26

What You Can Learn From Computer Prototypes



- Everything you learn from a paper prototype, plus:
- Screen layout
 - Is it clear, overwhelming, distracting, complicated?
 - Can users find important elements?
- Colors, fonts, icons, other elements
 - Well-chosen?
- Interactive feedback
 - Do users notice & respond to status bar messages, cursor changes, other feedback
- Efficiency issues
 - Controls big enough? Too close together? Scrolling list is too long?

27

27

Why Use Prototyping Tools?



- Faster than coding
- No debugging
- Easier to change or throw away
- Don't let your UI toolkit do your graphic design

28

28

Computer Prototyping Techniques



- Storyboard
 - Sequence of painted screenshots
 - Sometimes connected by hyperlinks ("hotspots")
- Form builder
 - Real windows assembled from a palette of widgets (buttons, text fields, labels, etc.)
- Wizard of Oz
 - Computer frontend, human backend

29

29

Storyboarding Tools



- Photoshop
- Balsamiq
- **Pencil**
- Mockingbird
- Proto IO
- Excel
- ...



30

30

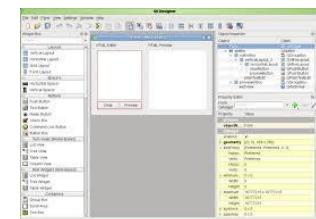
Pros & Cons of Storyboarding



- Pros
 - You can draw anything
- Cons
 - No text entry
 - Widgets aren't active
 - "Hunt for the hotspot"
 - Often useless for user testing
 - Better evaluated with other technique such as Heuristic Evaluation

31

31



32

Form Builders



- Silverlight
- Visual Studio
- Mac Interface Builder
- Qt Designer
- Android Studio
- ...

32

Pros & Cons of Form Builders



- Pros
 - Actual controls, not just pictures of them
 - Can hook in some backend if you need it
 - But then you won't want to throw it away
- Cons
 - **Limits** thinking to standard widgets
 - Less helpful for rich graphical interfaces

33

33



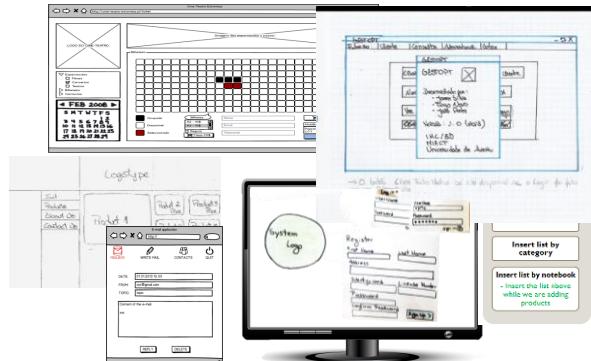
34

Wizard of Oz Prototype



- Software simulation with a human in the loop to help
- "Wizard of Oz" = "man behind the curtain"
 - Wizard is usually but not always hidden
- Often used to simulate future technology
 - Speech recognition
 - Learning
- Issues
 - Two UIs to worry about: user's and wizard's

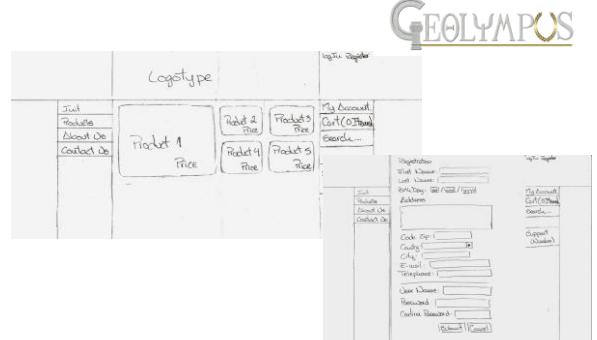
May be more or less "sophisticated"



35

35

Example: "on-line shop"



36

36

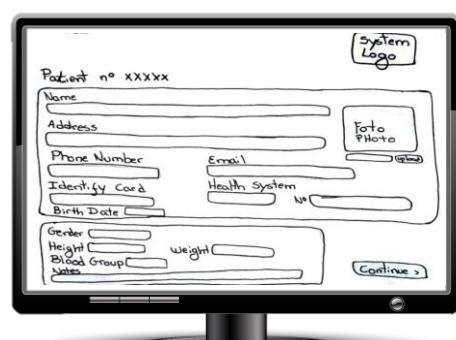
Example: “E-Doctor”



37

37

Example: “E-Doctor”



38

38

Example: “My shop” app

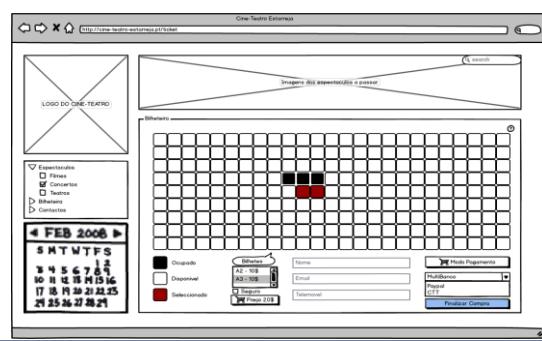


<ul style="list-style-type: none"> Participant #1 <ul style="list-style-type: none"> - Insert “Previous” and “Next” buttons 	<ul style="list-style-type: none"> Participant #2 <ul style="list-style-type: none"> - Scroll and swap options - Prev and Next should be arrows
<p>Search Localization</p> <p>New localization</p> <ul style="list-style-type: none"> - Change name “Mark” to “Supermarket Name” - Change order of “City” and “Supermarket” <p>Recent localizations</p> <p>Localization by GPS</p>	<p>Shopping List</p> <p>Recent lists</p> <p>Insert list by category</p> <p>Insert list by notebook</p> <ul style="list-style-type: none"> - Insert the list above while we are adding products <p>Modify list</p> <p>Interact with map</p> <ul style="list-style-type: none"> - Insert button “Next produit”

39

39

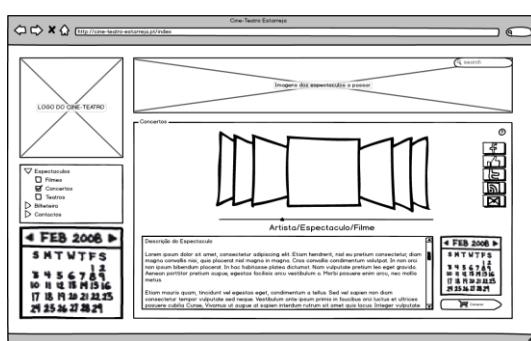
Example: a “ticketline” prototype



40

40

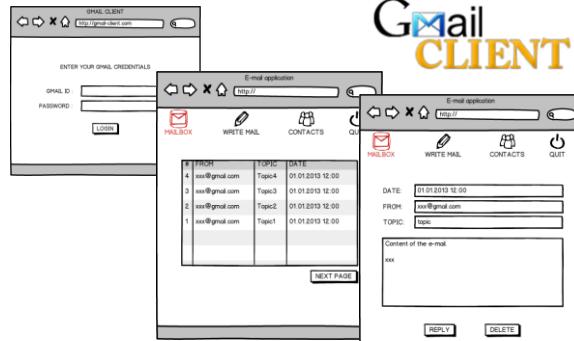
Example: a “ticketline” prototype



41

41

Example: “email client”



42

42

7

Example: DETImt



- The results for "check which brand is the most popular" had horrible results. People weren't able to find the dashboard button which was at the top-right of the application window. Also, users commented that "dashboard" is not an obvious name for that.



43

43

Even for less conventional applications



44

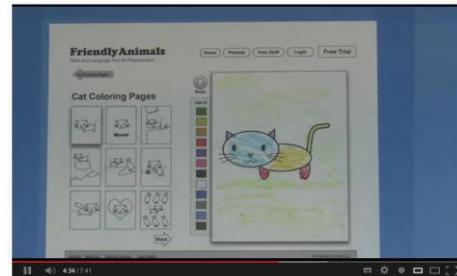
44

Example: gesture controlled “Pac-Man”



45

45



"Performing a usability test early in your website planning process can have huge returns - a paper prototype allows you to do this with a minimal time investment"

<http://www.youtube.com/watch?v=9wQkLthhHKA>

46

46

Summary



- Prototype fidelity
 - Depth, breadth, look, feel
- Kinds of prototypes
 - Paper
 - Computer: storyboard, forms
 - Wizard of Oz
- Don't get attached to a prototype
 - Because it may need to be thrown away

47

47

Useful links



- <https://www.userfocus.co.uk/articles/paperprototyping.html>
- <http://www.dreamscapedesign.co.uk/user-interface-design-the-use-of-paper-prototypes/>
- <https://balsamiq.com/>
- <https://prottapp.com/>
- <http://web.mit.edu/6.813/www/sp16/>

- Acknowledgment

To all students who have used paper prototyping in previous editions of the Human-Computer interaction course and colleagues who supervised them

48

48