Human Computer Interaction

Aula VIII

Departamento de Informática
UBI 2018/2019

João Cordeiro
jpaulo@di.ubi.pt

chapter 7

Design Rules

Design rules

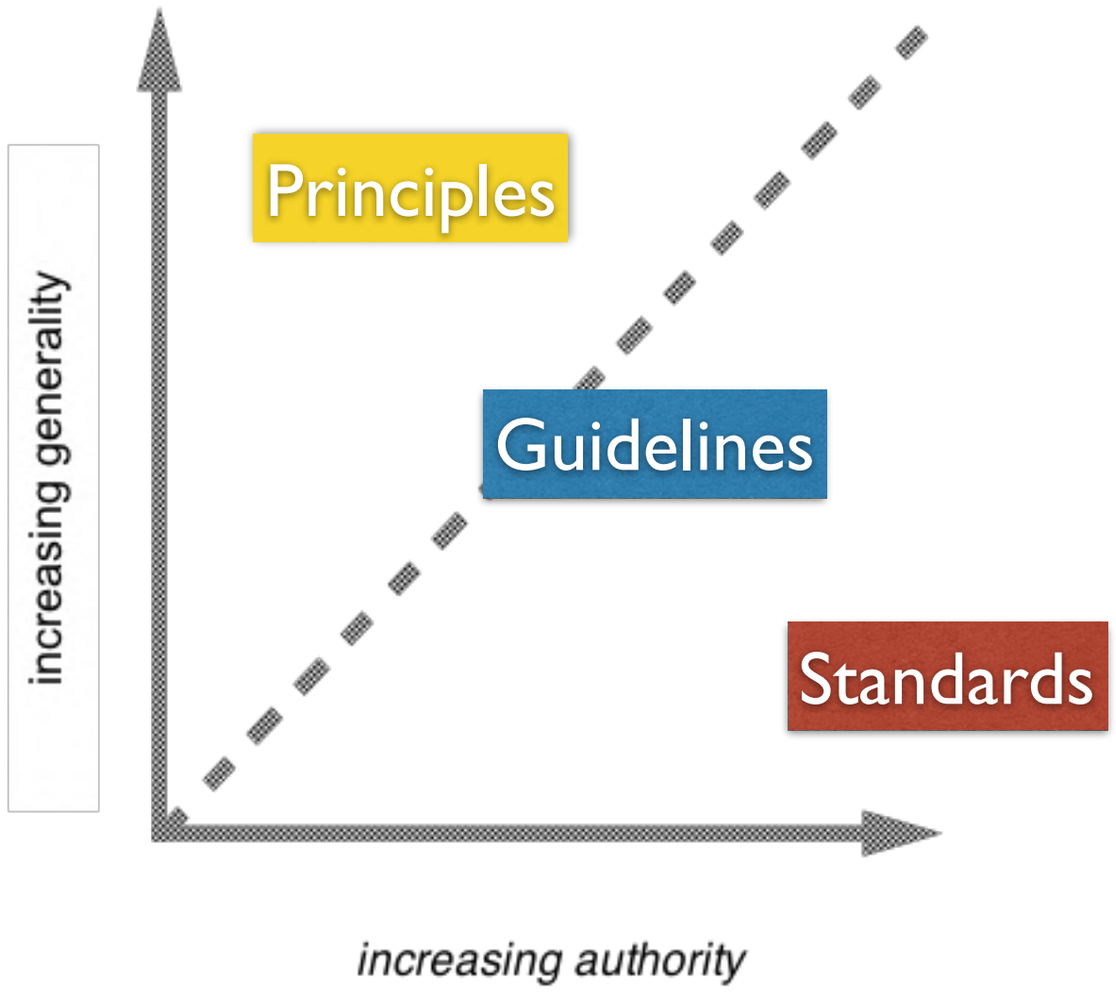
Designing for maximum usability – the goal of interaction design

- **Principles of usability**
  - general understanding

- **Guidelines and Standards**
  - direction for design

- **Design patterns**
  - capture and reuse design knowledge (gathered from the experience).

Using design rules



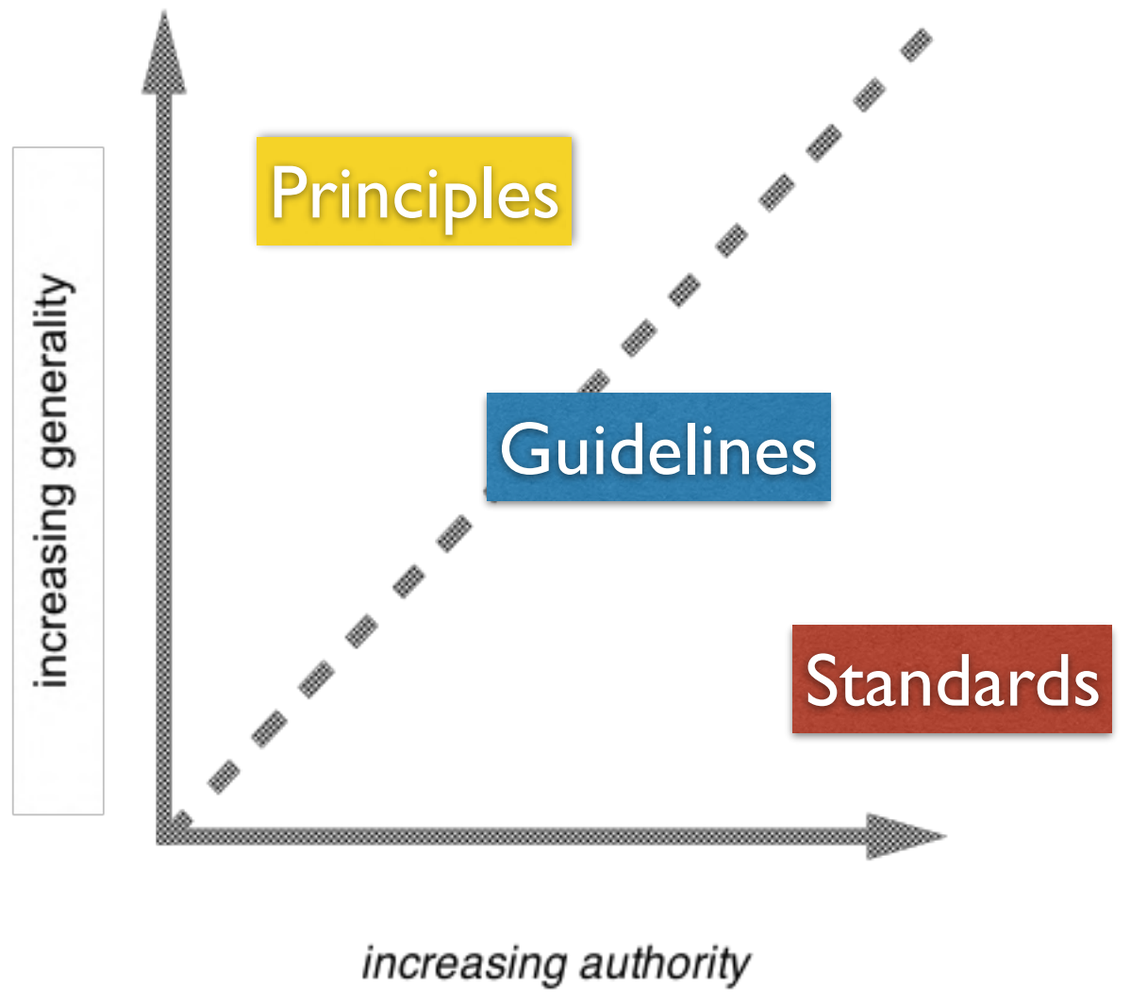
Design rules

- Suggest how to increase usability
- Different kinds, containing different levels of **generality** and **authority**.

Types of design rules

- **Principles**
  - **abstract** design rules
  - **low** authority
  - **high** generality
- **Guidelines**
  - claim **more** authority
  - **less** general / more tech.
- **Standards**
  - **specific** design rules
  - **high** authority
  - **limited** application

- Psychology
- Sociology
- Comp. Science



Principles to support usability

Learnability (Aprendizagem**)**

The ease with which new users can begin effective interaction and achieve maximal performance

Flexibility (Flexibilidade**)**

The multiplicity of ways the user and system exchange information (ex: OS multiple ways of doing ...)

Robustness (Robustez**)**

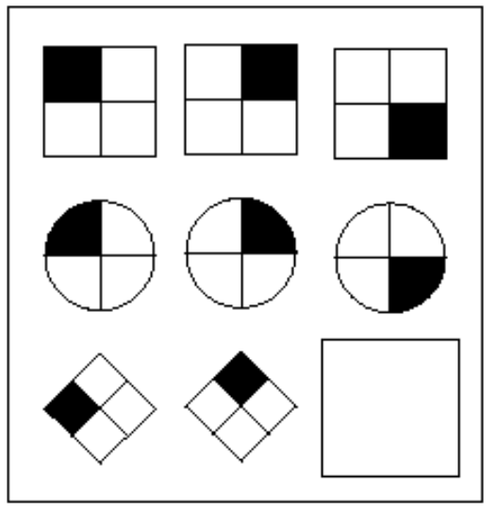
The level of support provided to the user to ensure successful achievement and assessment of goal-directed behavior.

Principles of learnability

Predictability (previsão**)**

- determining effect of future actions based on past interaction history (e.g. 123, 234, 345, ?)
- operation visibility (e.g. disabled fields)

F{recognition} > F{recall}



Synthesizability (síntese**)**

- assessing the effect of past actions
- honesty
  - immediate vs. eventual honesty
    - (e.g. Command vs GUI OS; MacOS Finder v7)
    - (e.g. "the the"

      "We will prove the theorem" ...)

Principles of learnability

Familiarity

- how prior knowledge applies to new system
- guessability; affordance (facilitadores)
  e.g. the typewriter metaphor.

Generalizability

- extending specific interaction knowledge to new situations (ex: square is constrained rectangle)
- across applications (ex: copy/paste)

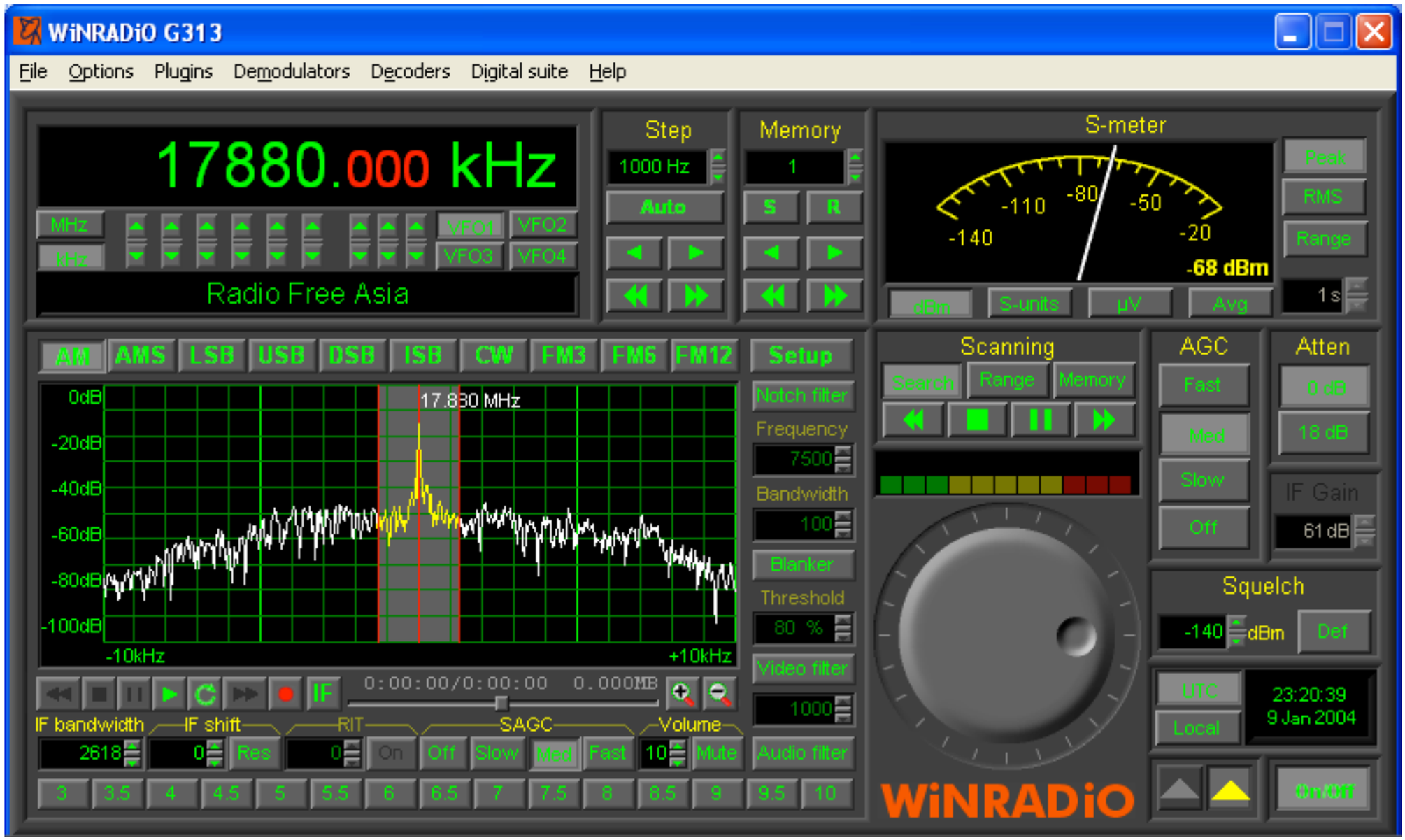
Consistency

- likeness in input/output behavior arising from similar situations or task objectives
- widely mentioned and related to other principles.
- (ex: ‘e’ ‘x’ ‘s’ ‘f’ in a directional keyboard)

Principles of learnability

Familiarity

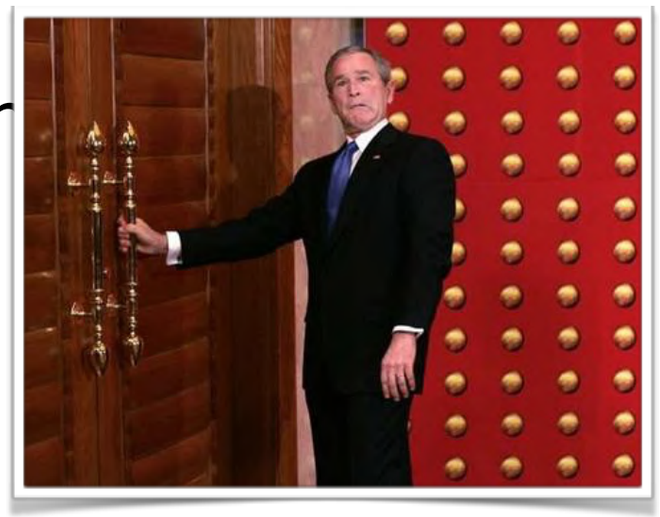
- how prior knowledge applies to new system
- guessability; **affordance**



Principles of learnability

Familiarity

- how prior knowledge applies to new system
- guessability; affordance (facilitadores)
  - (ex: typewriter metaphor)



Generalizability

- extending specific interaction knowledge to new situations (ex: square is constrained rectangle)
- across applications (ex: copy/paste)

Consistency

- likeness in input/output behavior arising from similar situations or task objectives
- widely mentioned and related to other principles.

  e.g. ‘e’, ‘x’, ‘s’ in a directional keyboard,

Principles of flexibility

Dialogue initiative

- freedom from system imposed constraints on input dialogue
- (**system** vs. **user**) **pre-emptiveness**
  - ex: modal dialog
  - ex: important for security (ex: cooperative editor)

Google docs

Multithreading (of a dialog)

- ability of system to support user interaction for more than one task at a time
- concurrent vs. interleaving

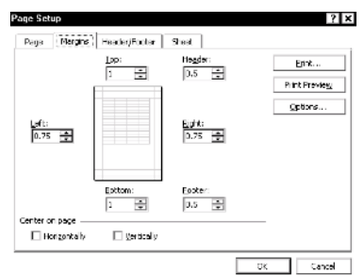
Task migratability

- passing responsibility for task execution between **user** and **system** (ex: spell checker; automation aviation)

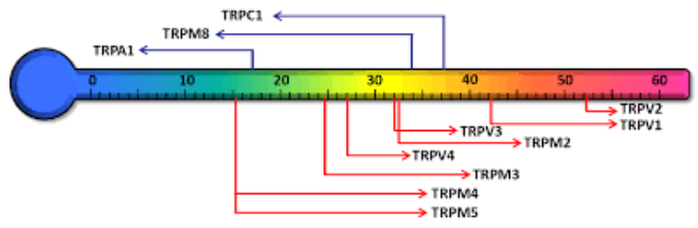
Principles of flexibility

Substitutivity (Equivalência**)**

- allowing equivalent values of input and output to be substituted for each other
  - e.g. margin definition. "=2/3 * (8.5-6.5)"
  - e.g. temperature readings.



- <u>With respect to output</u>: representation multiplicity; equal opportunity (ex: spreadsheet)



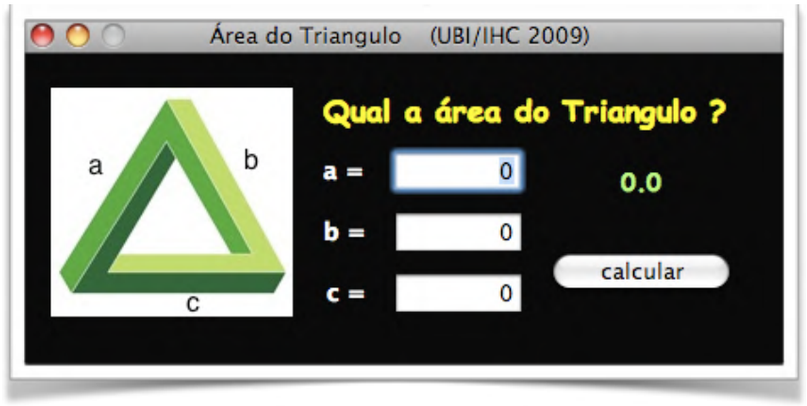
Customizability (Parametrização**)**

- User modifies/parametrizes the system: **adaptability**
- System automatically adjusts itself: **adaptivity**

Principles of robustness

Observability

- ability of user to evaluate the internal state of the system from its perceivable representation
- 5 other principles: browsability; defaults; reachability; persistence; operation visibility



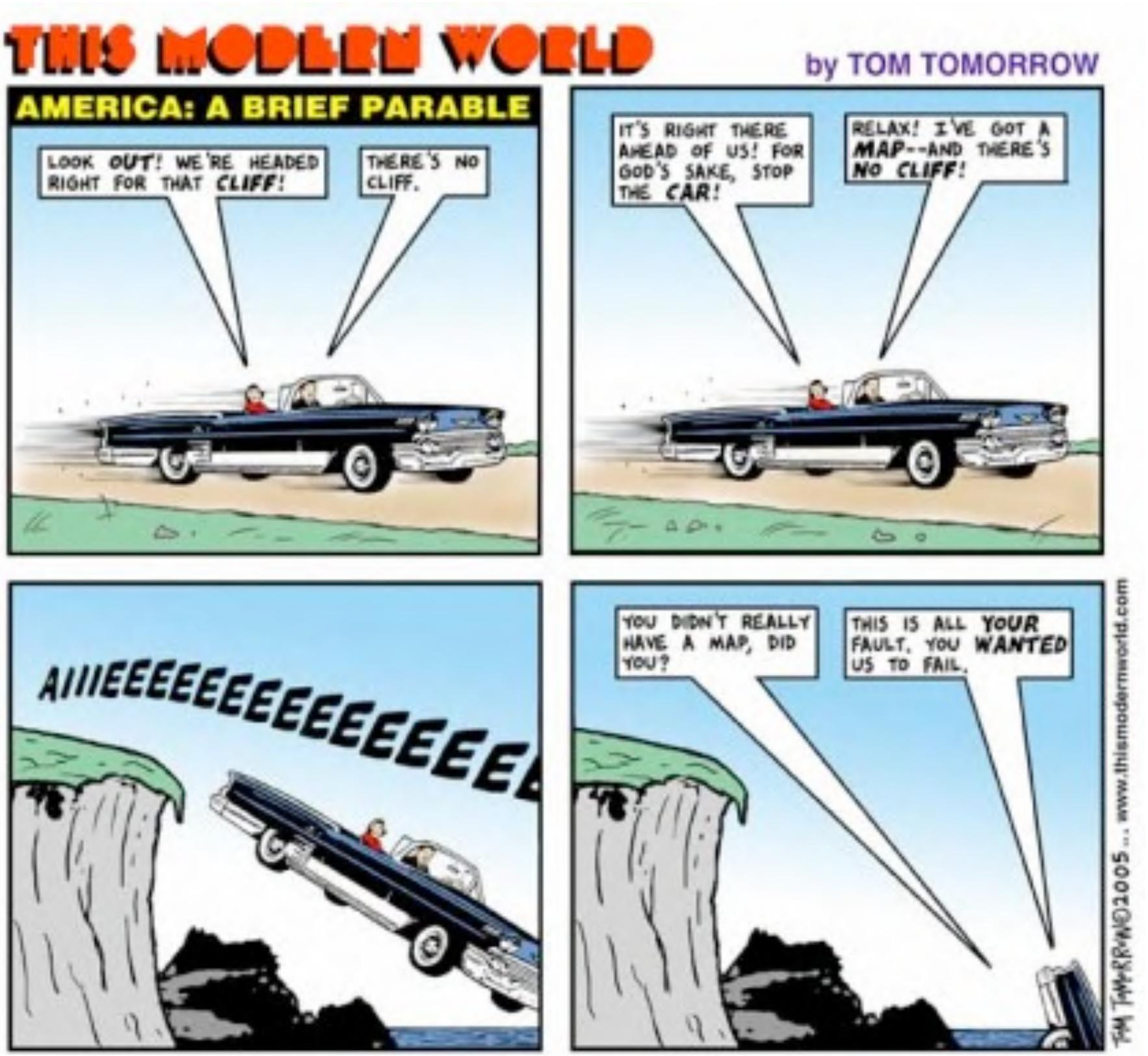
Vocal: email beep
Visual signal persists

Recoverability

- ability of user to take corrective action once an error has been recognized
- **forward**/**backward** recovery; commensurate effort

delete > rename

Principles of robustness



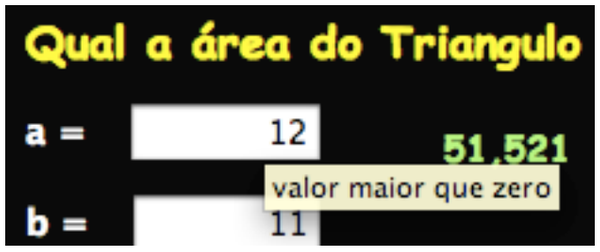
Principles of robustness

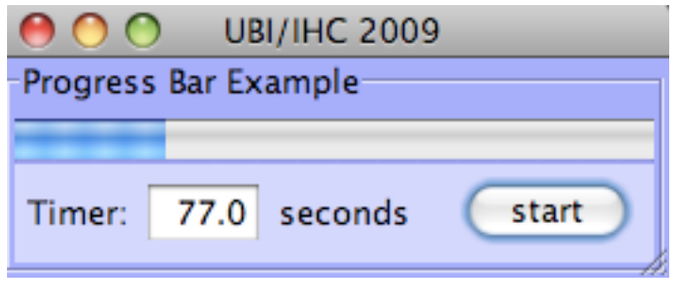
Responsiveness (Reatividade)

- How the user perceives the rate of communication with the system
- Time Stability

Response time (system)

Anticipation





Task conformance (Adequação)

- Degree to which system services support all of the user's tasks
- Task **completeness**; task **adequacy**

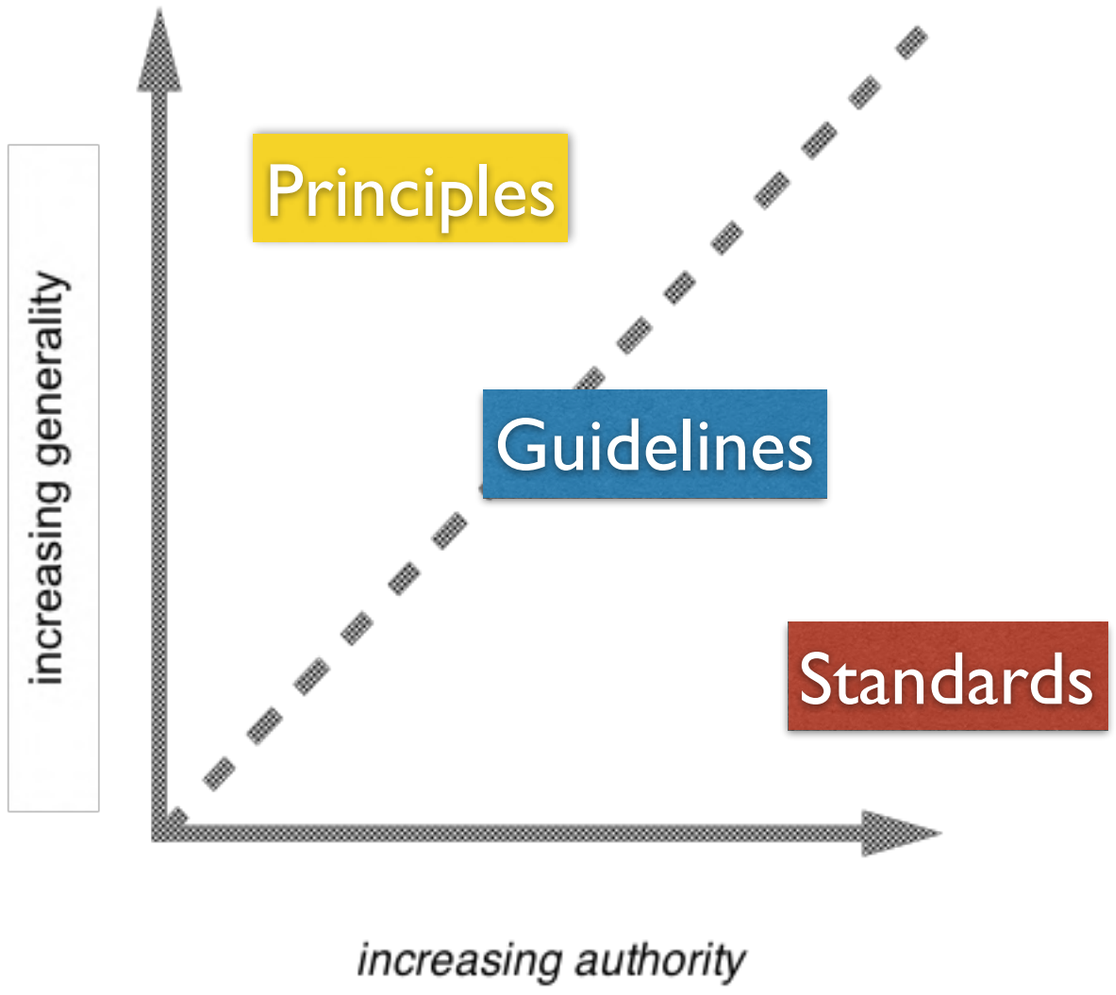
coverage

user understanding

Remember Norman's Gulfs !

the model world metaphor

Using design rules



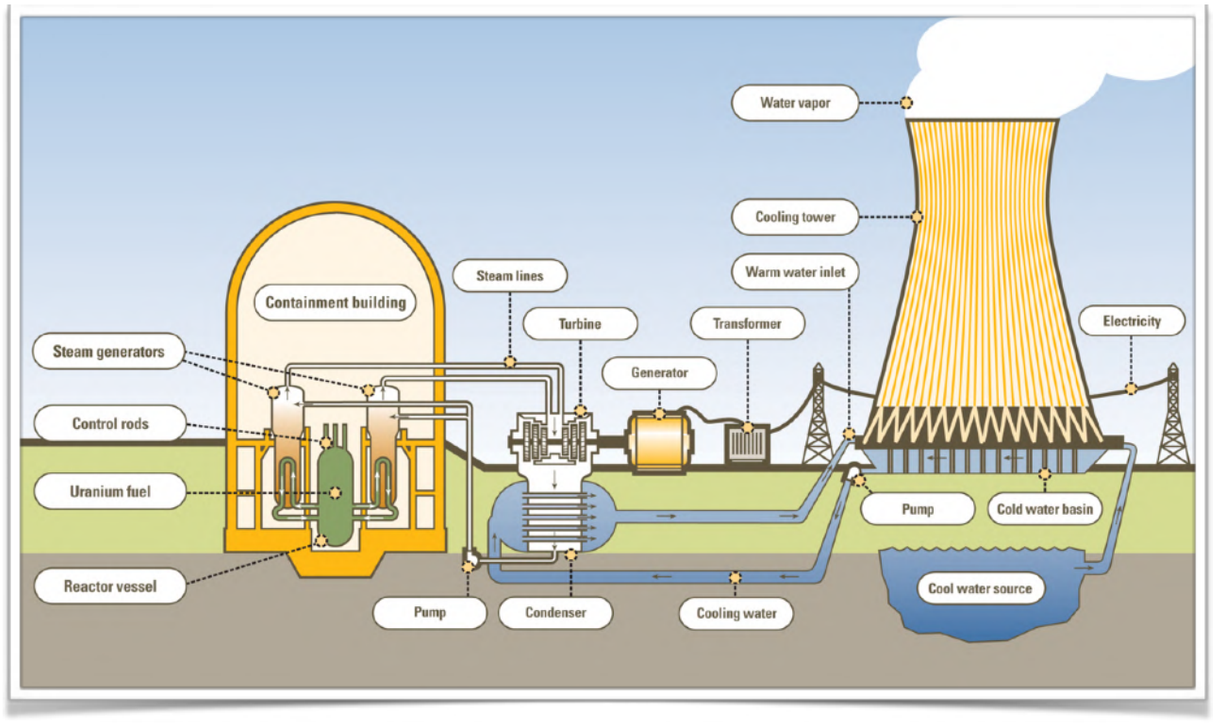
Design rules

- Suggest how to increase usability
- Different kinds, containing different levels of **generality** and **authority**.

Standards (ISO …)

- Set by **national or international bodies** to ensure compliance by a large community of designers standards require sound underlying theory and slowly changing technology.

- Longer history on safety-critical domains



- **Hardware** — More common

  Ergonomics and physiology

- **Software** — With authority but low level of detail

  Cognitive Sciences

Standards

- **Interim Defense Standard 00-25 on Human Factores for Designers of Equipment. (UK)**

Standard for the design of militar equipment

1. Introduction
2. Body Size
3. Body Strength and Stamina
4. Workplace Design
5. Stress and Hazards
6. Vision and Lighting
7. Visual Displays
8. Auditory Information
9. Voice Communication
10. Controls
11. Design Maintainability
12. Systems (software)

BSI: British Standard Institution.

Standards

- **Examples:**

11.3 *Arrangement of displays*
11.3.1 Vertical Grouping. The engine display parameters shall be arranged so that the primary or most important display for a particular engine and airplane (thrust, torque, RPM, etc.) be located at the top of the display group if a vertical grouping is provided. The next most important display parameter shall be positioned under the primary display progressing down the panel with the least important at the bottom.

(a) A typical example of a military standard

5.1 *Subdivision of the display area*
In consideration of a simple, fast and accurate visual acquisition, the display area shall be divided into different sub-areas.
Such a division should be:

- Input area
- Output area
- Area for operational indications (such as status and alarms)

(b) From German standard DIN 66 234 Part 3 (1984), adapted from Smith [324]

5.15.3.2.1 *Standardization*
The content of displays within a system shall be presented in a consistent manner.

(c) From US military standard MIL-STD-1472C, revised (1983), adapted from Smith [324]

Figure 7.1 Sample design standards for displays. Adapted from Smith [324].
Copyright © 1986 IEEE

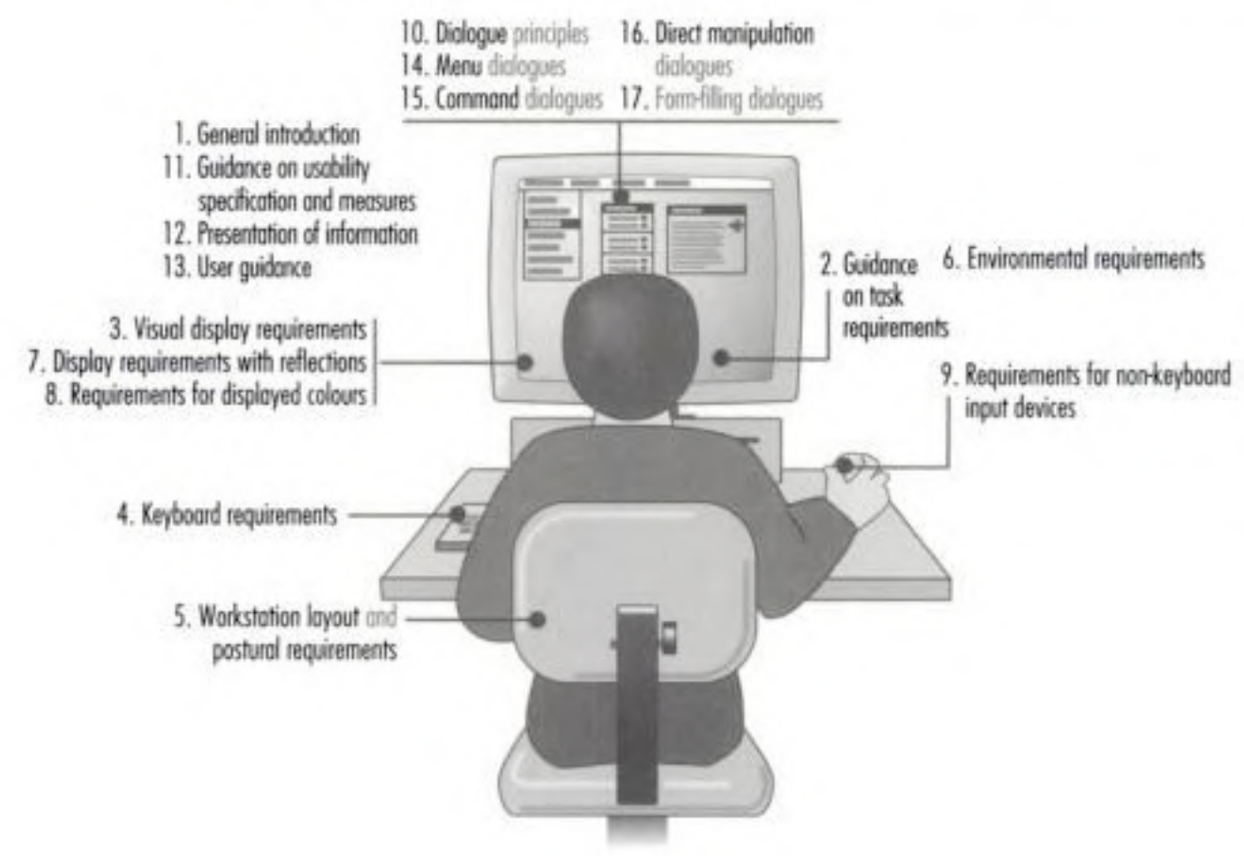
generality

Standards

- **ISO 9241** defines **usability** as **effectiveness**, **efficiency** and **satisfaction** with which users accomplish tasks

- **Effectiveness**
  - precision
  - completness
- **Efficiency**
  - resources spent
- **Satisfaction**
  - confort
  - acceptability

Figure 52.11 • Technical Working Groups of the Ergonomics of Human System Interaction Tecl (ISO TC 159 SC4). ISO 9241: Five working groups broke down the "parts" ol below. This illustration shows the correspondence between the parts of the stan of the workstation with which they are concerned.



Encyclopaedia of occupational health and safety, Volume 4
Jeanne Mager Stellman

Standards

- **ISO 9241** defines **usability** as **effectiveness**, **efficiency** and **satisfaction** with which users accomplish tasks

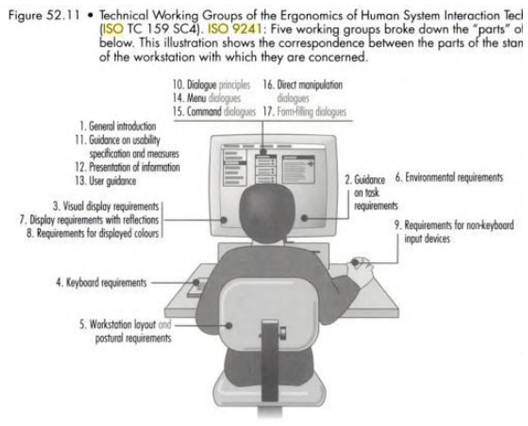


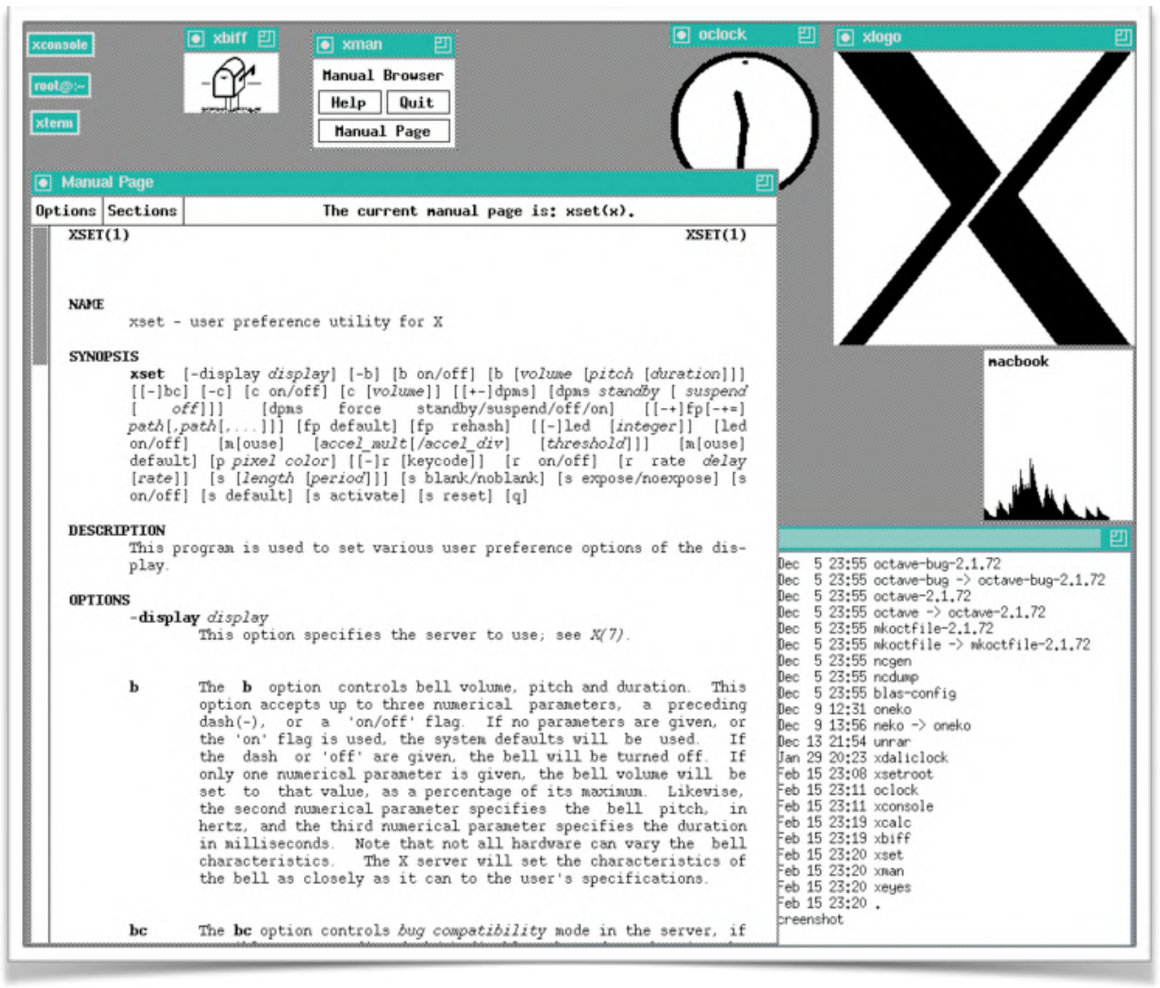
Figure 52.11 • Technical Working Groups of the Ergonomics of Human System Interaction Tecl (ISO TC 159 SC4). ISO 9241: Five working groups broke down the "parts" o below. This illustration shows the correspondence between the parts of the stan of the workstation with which they are concerned.

The published parts of ISO 9241

- Part 1: General introduction
- Part 2: Guidance on task requirements
- Part 4: Keyboard requirements
- Part 5: Workstation layout and postural requirements
- Part 6: Guidance on the work environment
- Part 9: Requirements for non-keyboard input devices
- Part 11: Guidance on usability
- Part 12: Presentation of information
- Part 13: User guidance
- Part 14: Menu dialogues
- Part 15: Command dialogues
- Part 16: Direct manipulation dialogues
- Part 17: Form filling dialogues
- ....

Standards in Software

- **Most of them are more suggestive than mandatory!** Usually become standards way before any formal standardization was set.



X-Windows

Massachusetts Institute of Technology

1984

The strength of a norm lays in its followed community.

Guidelines (Orientações)

- They range from the most **general** to the most **specific**.

- Lots of manuals and reports, full of guidelines, example: [Smith & Moiser 1986]

  - Data Entry
  - Data Display
  - Sequence Control
  - User Guidance
  - Data Transmission
  - Data Protection

1.4 DATA ENTRY: Data Forms
1.4/5 Data Field Labels

For each data field, display an associated label to help users understand what entries can be made.

Example:

(Good)
```
NAME: __ __ __ __ __ __ __ __ __ __ __ __

ORGANIZATION:  __ __/__ __

PHONE:  __ __ __-__ __ __ __
```

(Bad)
```
NAME, ORGANIZATION AND PHONE

__ __ __ __ __ __ __ __ __ __ __ __ __

__ __ __ __

__ __ __ __ __ __ __
```

Reference:
BB 2.1.7

See also: 1.0/24 4.0/11

Guidelines (Orientações)

- GUIDELINES FOR DESIGNING USER INTERFACE SOFTWARE: [Smith & Moiser 1986]

1.1 Position Designation

Position designation refers to user selection and entry of a position on a display, or of a displayed item.

1.1/1 Distinctive Cursor

For position designation on an electronic display, provide a movable cursor with distinctive visual features (shape, blink, etc.).

Exception

When position designation involves only selection among displayed alternatives, highlighting selected items might be used instead of a separately displayed cursor.

Comment

When choosing a cursor shape, consider the general content of the display. For instance, an underscore cursor would be difficult to see on a display of underscored text, or on a graphical display containing many other lines.

Comment

If the cursor is changed to denote different functions (e.g., to signal deletion rather than entry), then each different cursor should be distinguishable from the others.

Comment

If multiple cursors are used on the same display (e.g., one for alphanumeric entry and one for line drawing), then each cursor should be distinguishable from the others.

Reference

- Whitfield Ball Bird 1983

See also

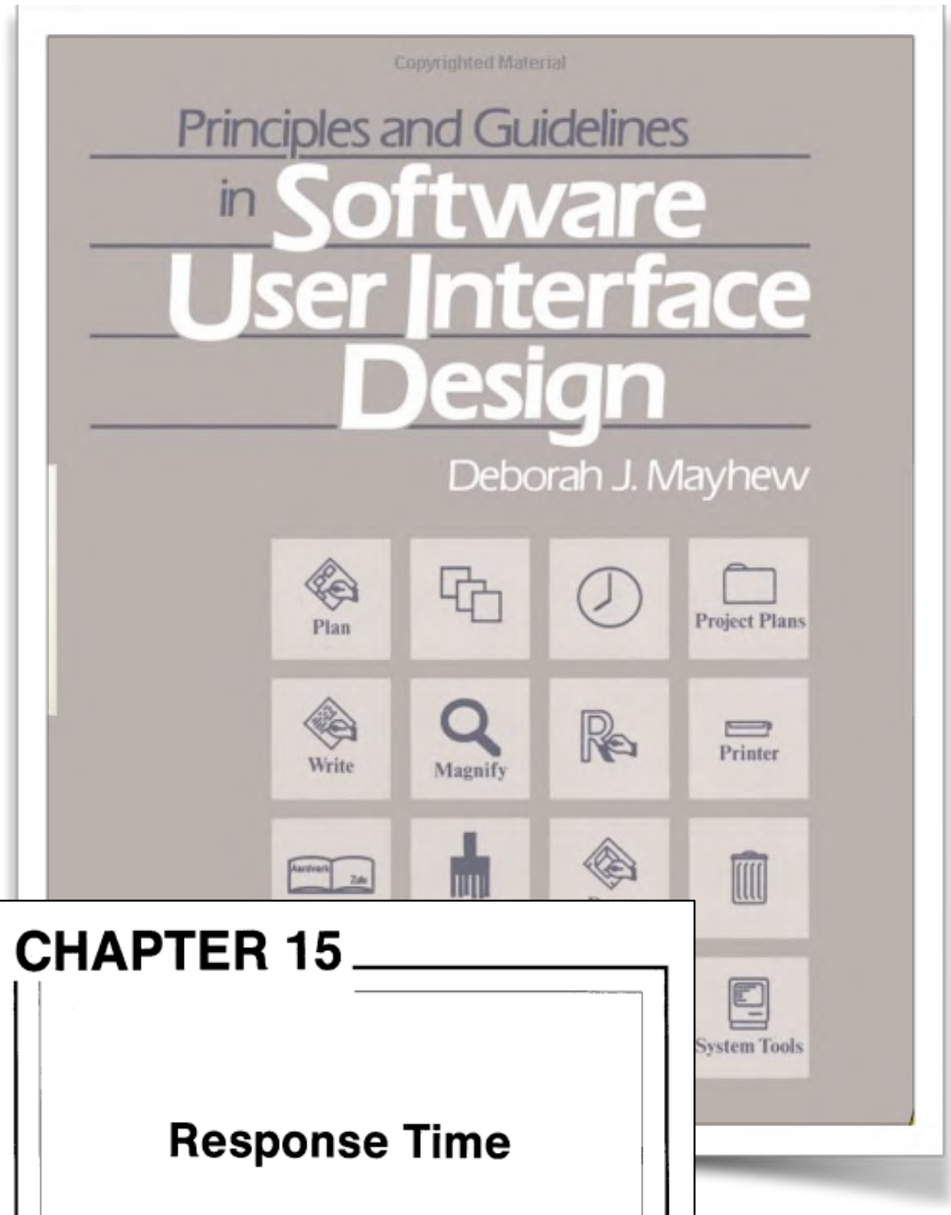
1.1/17 | 4.0/9

on-line:

http://hcibib.org/sam/

Guidelines

- Others also **suggestive** and **general**: [Mayhew 1992]



CHAPTER 15

Response Time

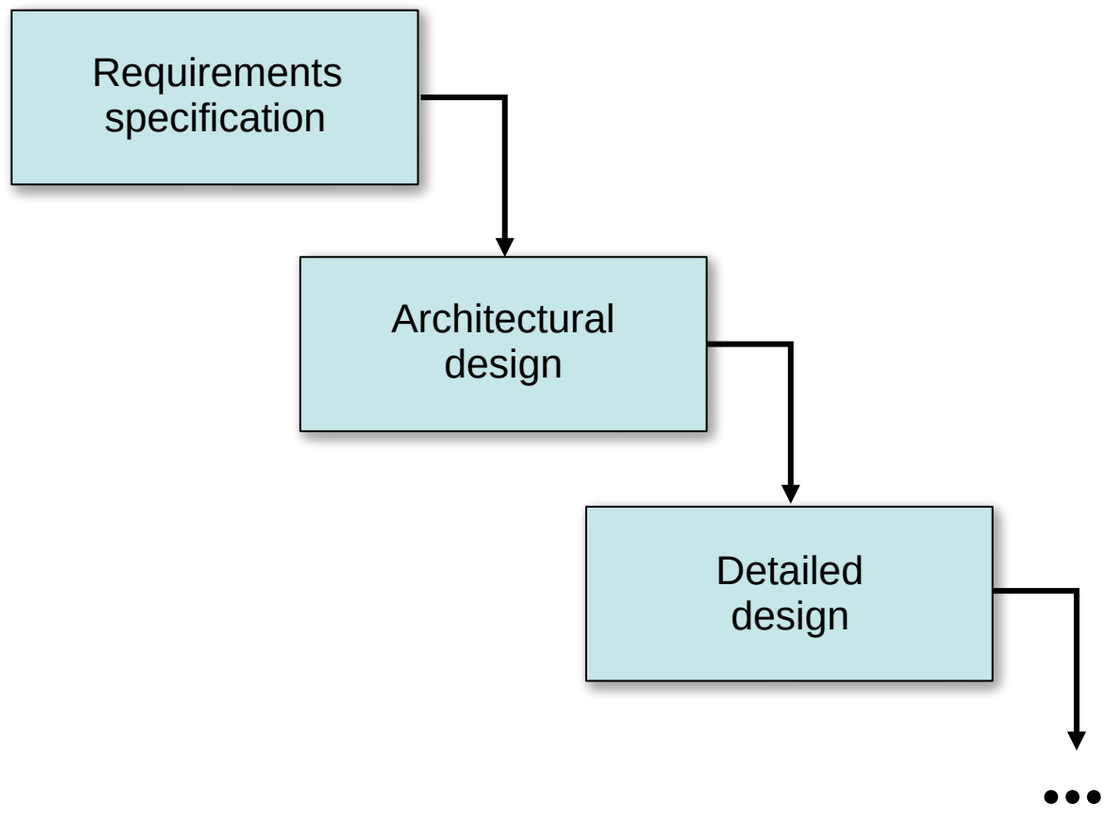
System response time is the one aspect of computer–human interfaces that has always been considered important. Vendor companies have long set up minimum acceptability criteria for *system performance* and tested this aspect of hardware and software design. It has always been an aspect of the competitive edge. This is in contrast to other aspects of software user interface design, such as dialog and screen design, which have only recently been considered aspects of the competitive edge and received any significant attention.

On the other hand, simple *computer* response time is not really the relevant measure, as far as usability is concerned. What will be considered acceptable computer performance by users will depend on a number of other interacting factors.

Guidelines

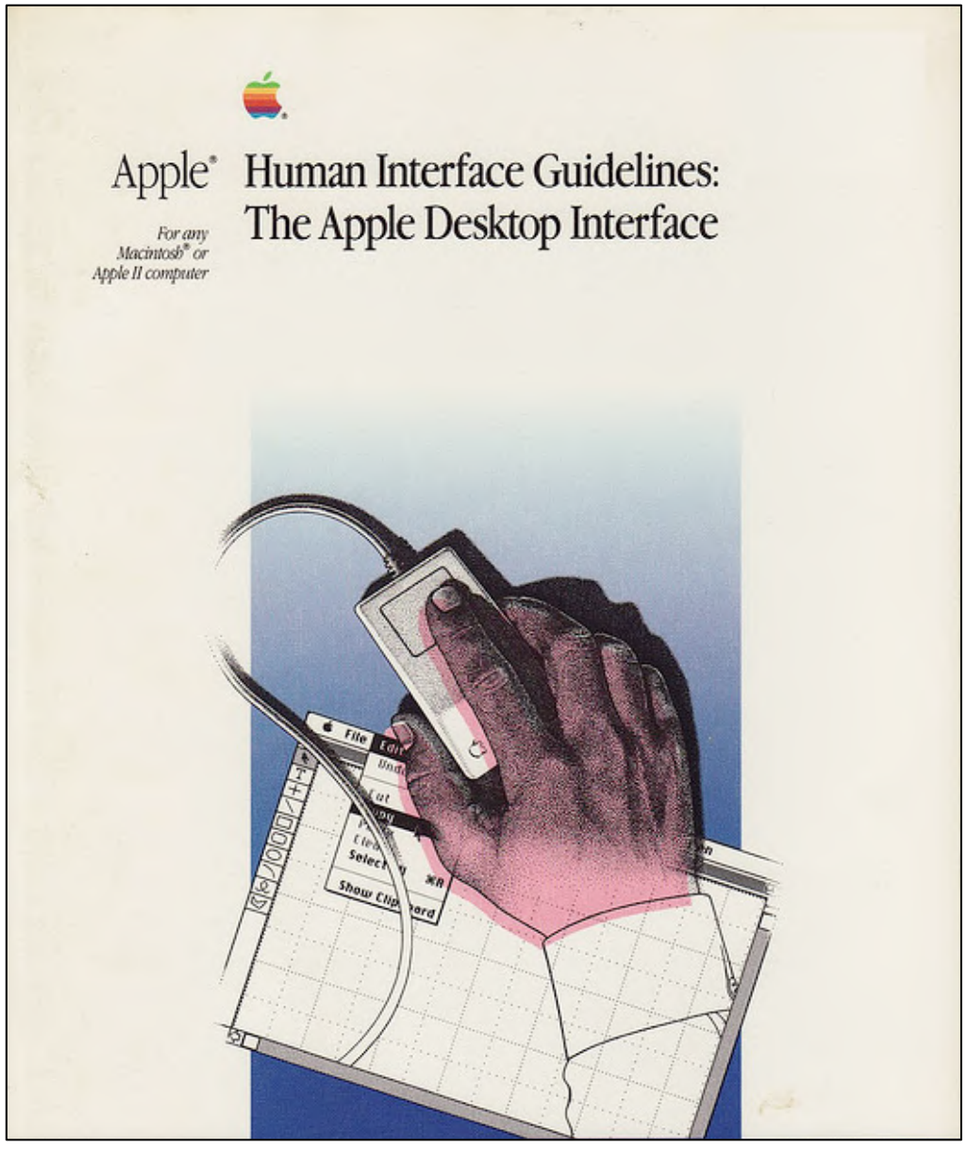
- The more **abstract** kind of guidelines (principles) — applicable during the initial activities of the life cycle.

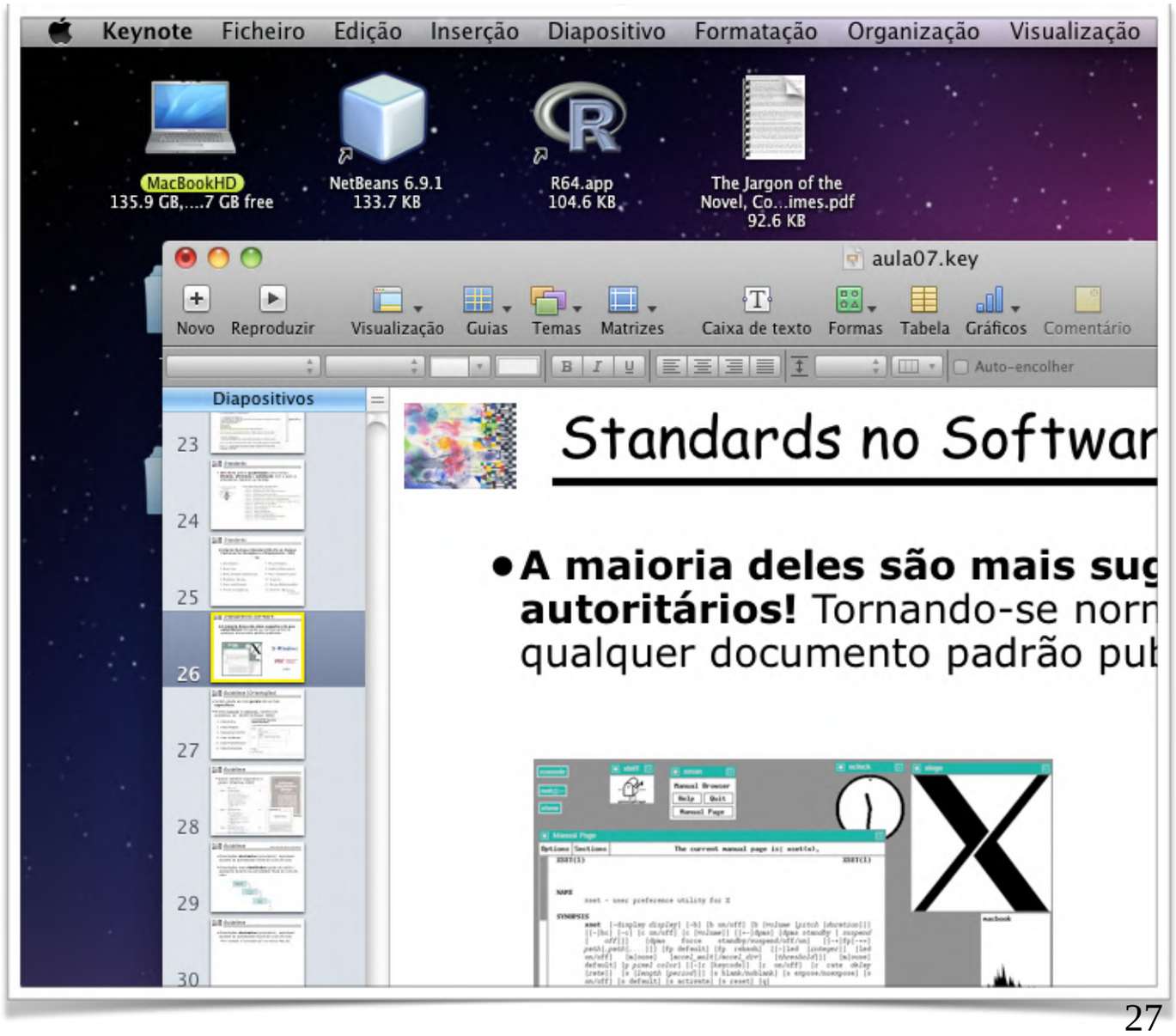
- There are also more **detailed** guidelines — more adequate to be observed during the final life-cycle activities.



Guidelines

- The more **abstract** kind of guidelines (principles) — applicable during the initial activities of the life cycle.
  - For example the “consistency” in the xOS menus





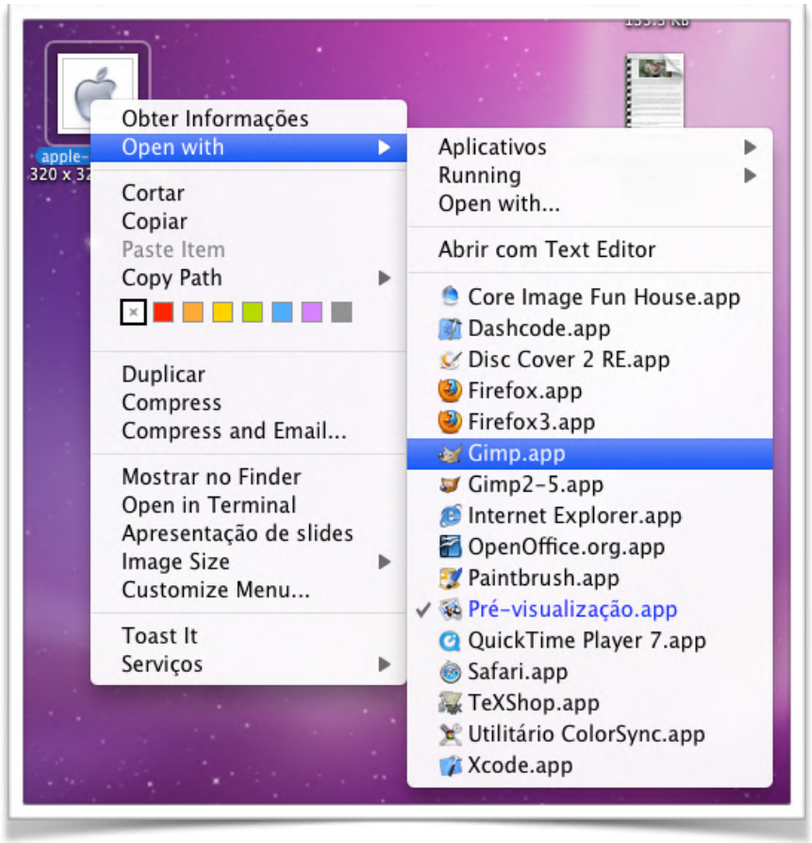
Guidelines

- The more **abstract** kind of guidelines (principles) — applicable during the initial activities of the life cycle.
  - For example the “consistency” in the xOS menus

Exemplo de falha de consistência, nos menus do *Gimp*.

Guidelines

- More detailed **guidelines** (style guides) - applicable during the final life cycle activities. The “Noun-Verb” directive from Apple.



- The dialog directive of “user-preemptiveness”:
  “*The user, not the computer, initiates and controls all actions*”.

Guidelines -- iOS

Developer Discover Design Develop Distribute Support Account

Human Interface Guidelines Resourc

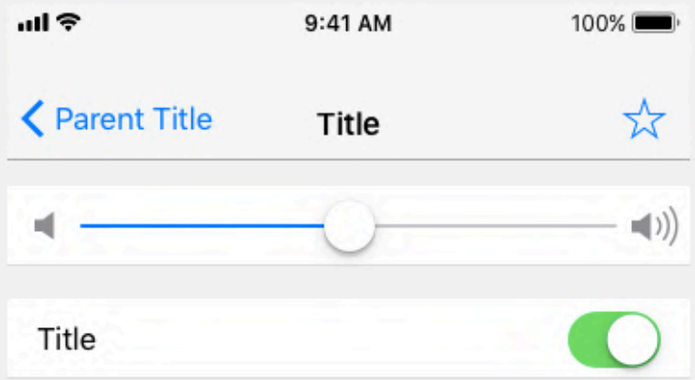
Apple UI Design Resources

Design apps quickly by using Photoshop and Sketch templates, plug-ins, and other preconfigured UI elements.

iOS Apps

Apple UI Design Resources include Photoshop, Sketch, and Adobe XD templates, along with comprehensive UI resources that depict the full range of controls, views, and glyphs available to developers using the iOS SDK. These resources help you design apps that match the iOS design language. Icon and glyph production files are preconfigured to automate asset production using Sketch slices or Adobe Generator for Photoshop CC. Color swatches, dynamic type tables, and fonts are also included.

For design guidance, see Human Interface Guidelines > iOS.



https://developer.apple.com/design/resources/

Guidelines -- iOS

Developer Discover Design Develop Distribute Support Account

Human Interface Guidelines iOS

Overview
- Themes
- iPhone X
- What's New in iOS 11
- Interface Essentials

App Architecture

User Interaction

System Capabilities

Visual Design

Icons and Images

Bars

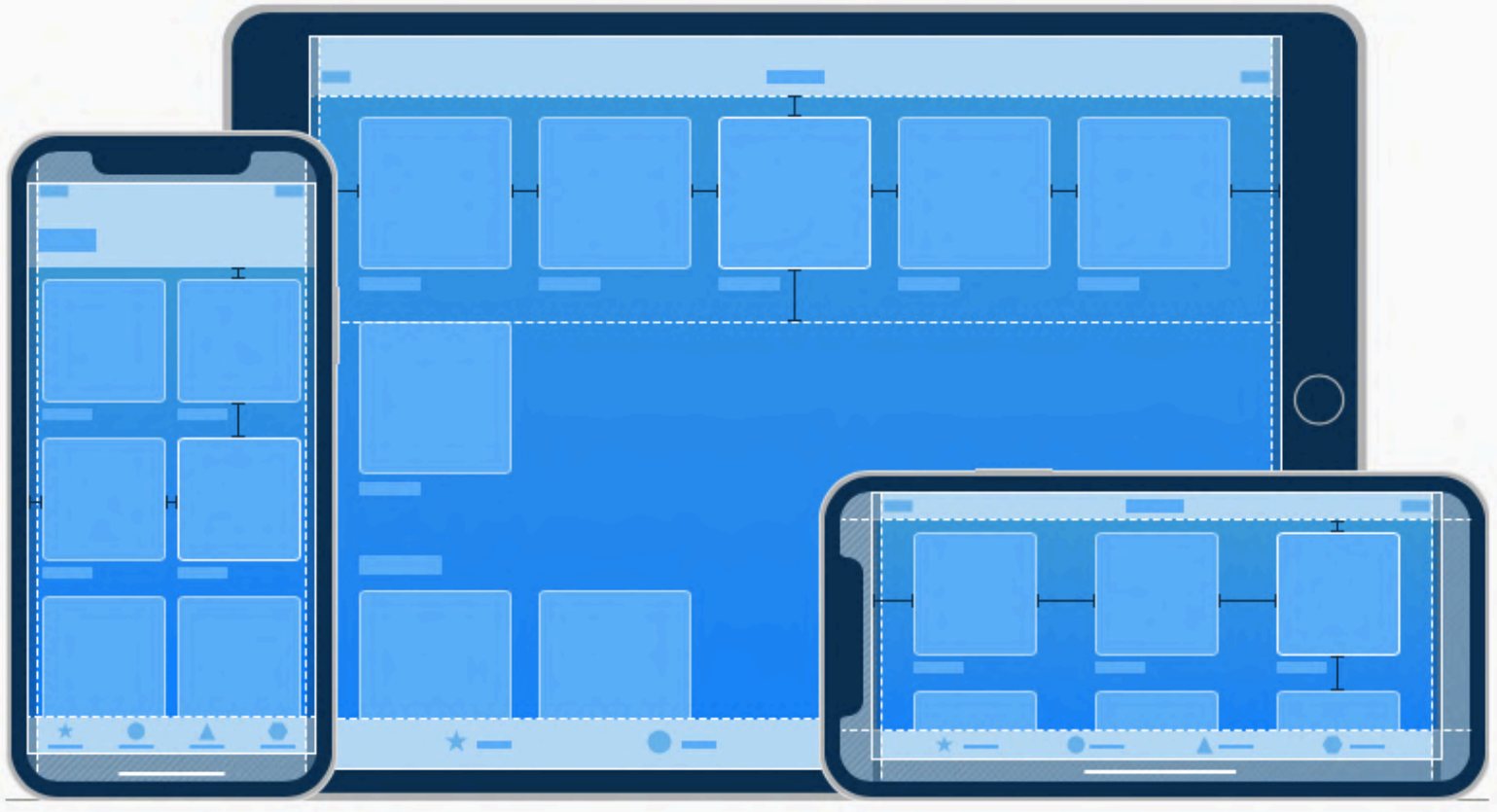
Views

Controls

Extensions

Technologies

Resources



iOS Design Themes

As an app designer, you have the opportunity to deliver an extraordinary product that rises to the top of the App Store charts. To do so, you'll need to meet high expectations for quality and functionality.

Guidelines - Java GUI Example

- For Java GUI there are several guidelines

Java Look and Feel Design Guidelines

Available online!

- VOLUME I. Java™ Look and Feel Design Guidelines, second edition

http://java.sun.com/products/jlf/ed2/book/index.html

- VOLUME II. Java™ Look and Feel Design Guidelines: Advanced Topics

http://java.sun.com/products/jlf/at/book/index.html

Guidelines - Java GUI Example

- Chapter 4: Visual Design

Themes

As a software developer, you can use the theme mechanism to control many of the fundamental attributes of the Java look and feel design, including colors and fonts. For instance, you might want to change the colors and fonts in your application to match your corporate identity. The theme mechanism enables you to specify alternative colors and fonts across an entire Java look and feel application. [**Link**].

Figure 22 Primary Colors in Default Color Theme

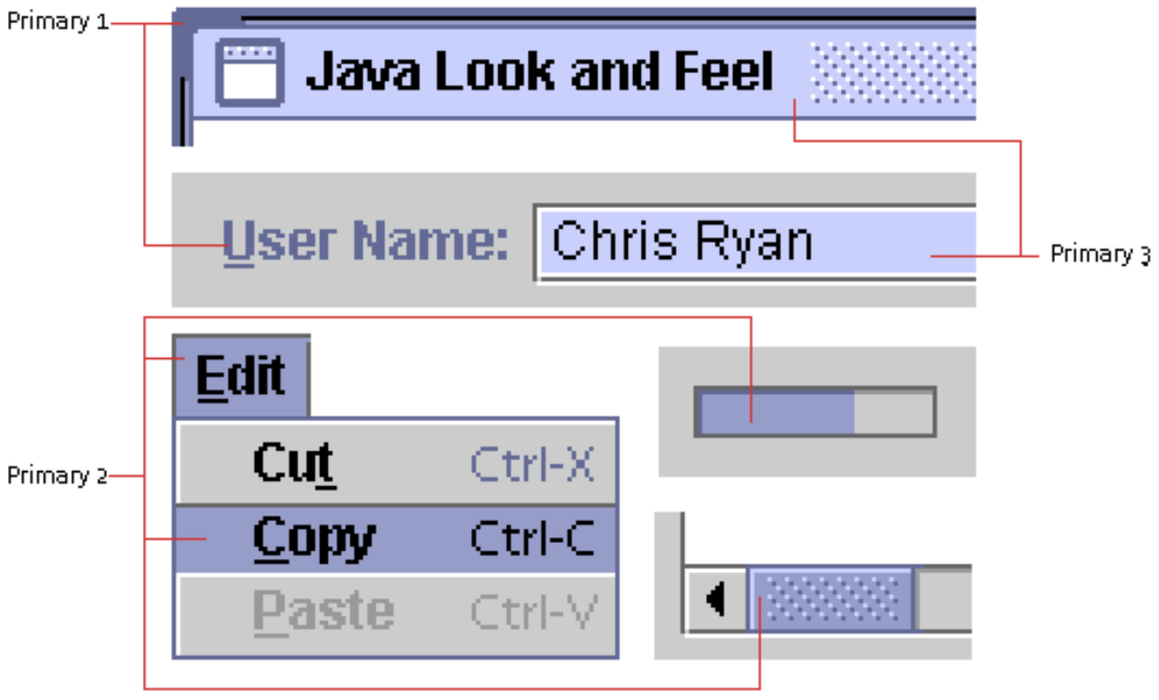
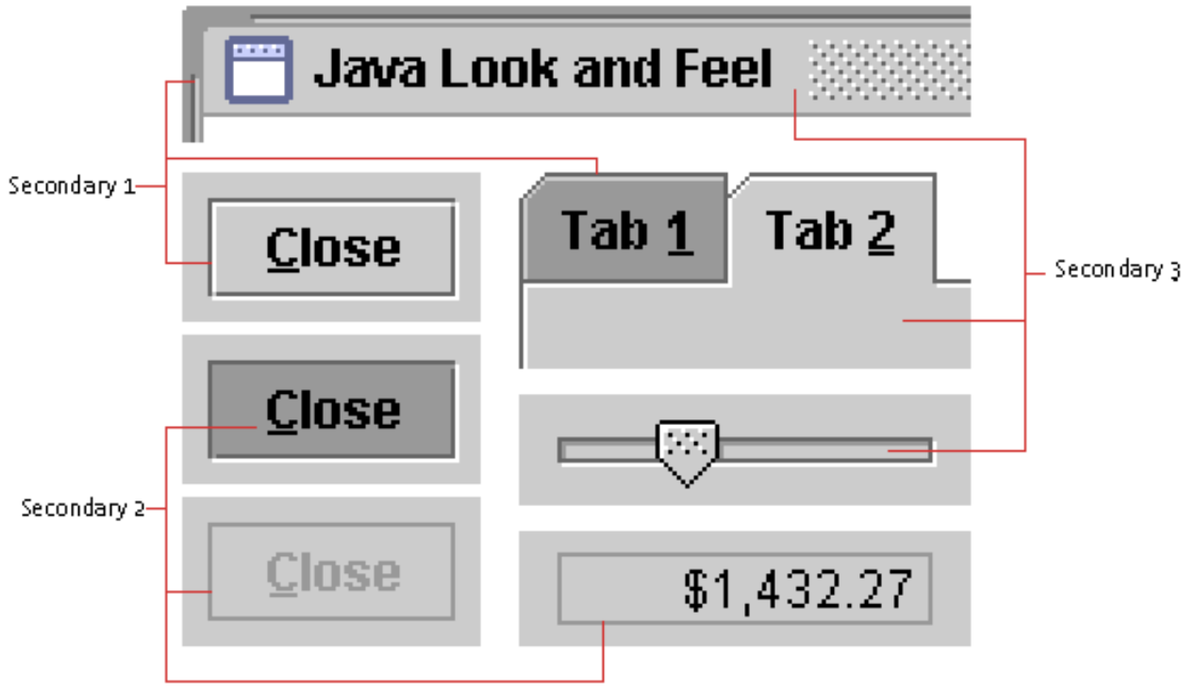


Figure 23 Secondary Colors in Default Color Theme

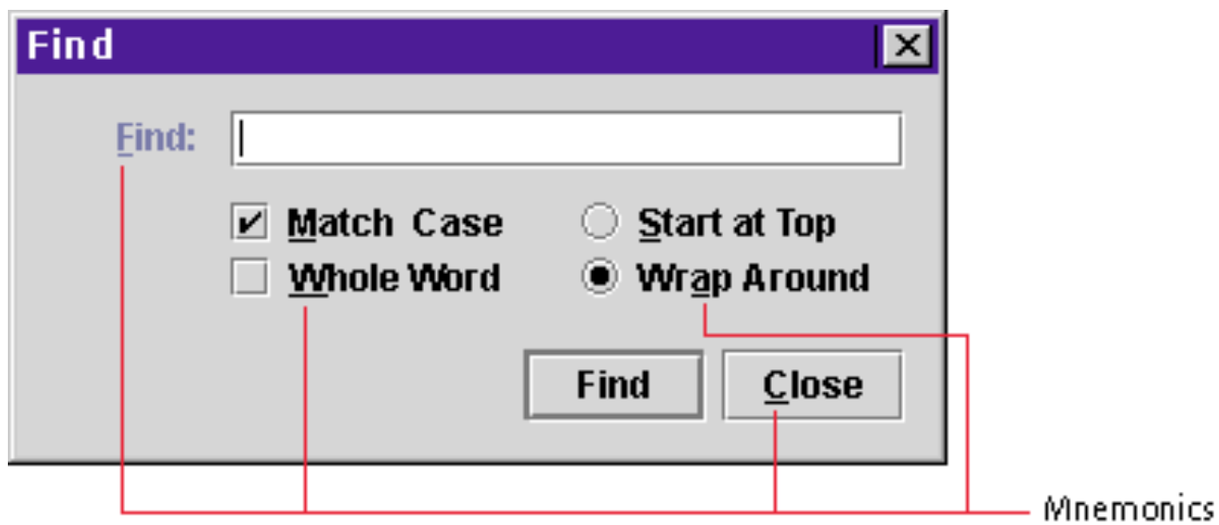


Guidelines - Java GUI Example

- For Java GUI there are several guidelines

Mnemonics and Keyboard Shortcuts

You should provide mnemonics and keyboard shortcuts throughout your application. A mnemonic is an underlined alphanumeric character that shows users which key to press (in conjunction with the Alt key) to activate a command or navigate to a component ... [http://java.sun.com/products/jlf/ed2/book/HIG.Issues2.html]



Guidelines - Java GUI Example

- Chapter 6: Responsiveness

Determining Acceptable Response Delays

The term **response delay** refers to how long an application takes to acknowledge or fulfill a particular user request. Providing responsiveness in an application depends on achieving response delays that are acceptable to users [**Link**].

Table 13 Maximum Acceptable Response Delays for Typical Events

User Interface Events	Maximum Acceptable Response Delay
Mouse click; pointer movement; window movement or resizing; key press; button press; drawing gesture; other user-input event involving hand-eye coordination	0.1 second (100 milliseconds)
Displaying progress indicators; completing ordinary user commands (for example, closing a dialog box); completing background tasks (for example, reformatting a table)	1.0 second
Displaying a graph or anything else that a typical user would expect to take time (for example, displaying a new list of all a company's financial transactions for an accounting period)	10.0 seconds
Accepting and processing all user input to any task	10.0 seconds

Guidelines - Java GUI Example

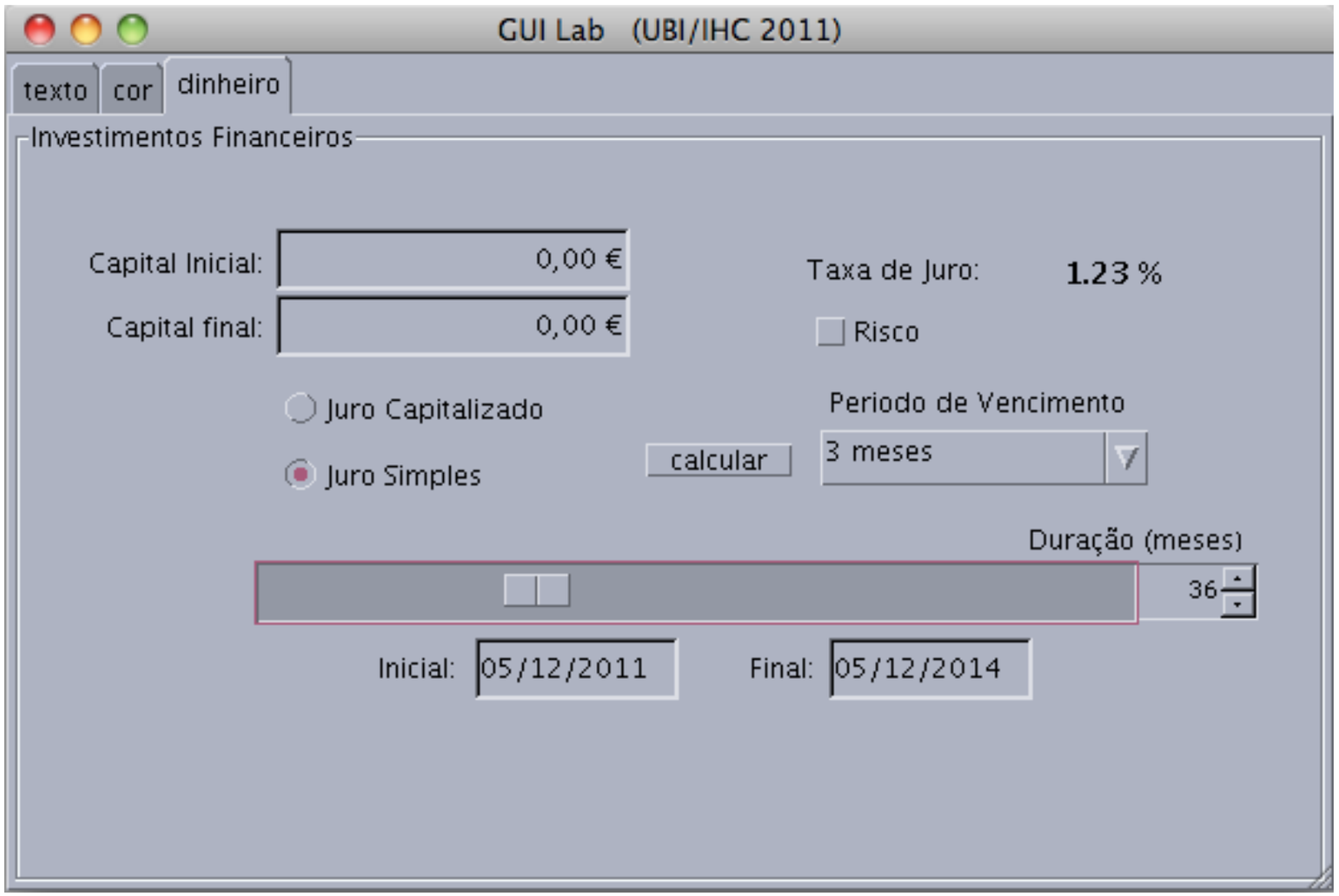
- The various "Look and Feel" themes are designed for aesthetic coherence, across multiple platforms

Platform	Look and Feel
Solaris, Linux with GTK+ 2.2 or later	GTK+
Other Solaris, Linux	Motif
IBM UNIX	IBM*
HP UX	HP*
Classic Windows	Windows
Windows XP	Windows XP
Windows Vista	Windows Vista
Macintosh	Macintosh*

Guidelines - Java GUI Example

- For Java GUI there are several guidelines

"Motif"



`com.sun.java.swing.plaf.motif.MotifLookAndFeel`

Guidelines - Java GUI Example

- For Java GUI there are several guidelines

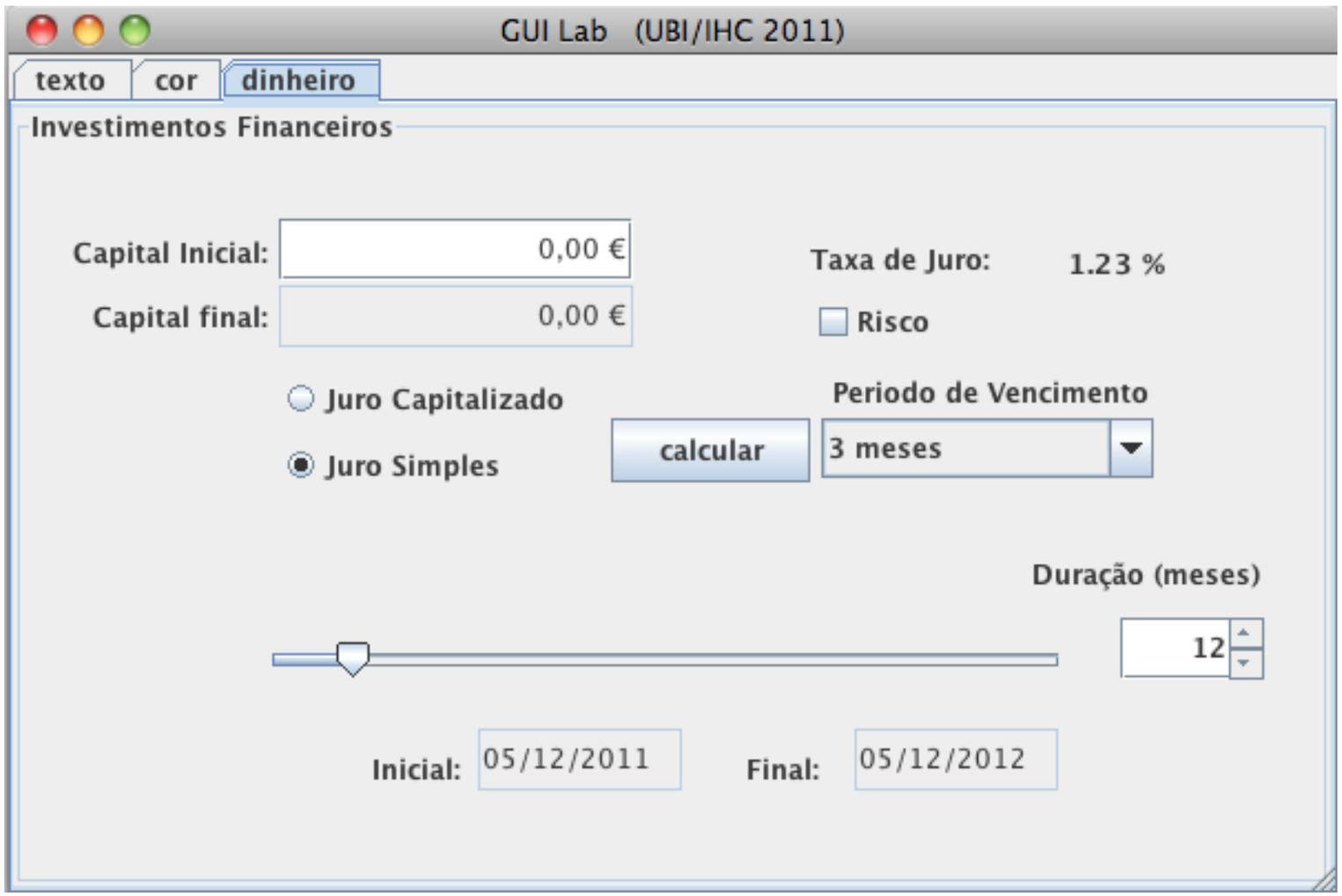
"GTK"

`com.sun.java.swing.plaf.gtk.GTKLookAndFeel`

Guidelines - Java GUI Example

- For Java GUI there are several guidelines

"Metal"



`javax.swing.plaf.metal.MetalLookAndFeel`

Guidelines - Java GUI Example

- For Java GUI there are several guidelines

Exemple o a specific definition o a “Look & Feel”.

```
public static void main(String args[]) {
    try {
        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new MainFrame().setVisible(true);
        }
    });
}
```

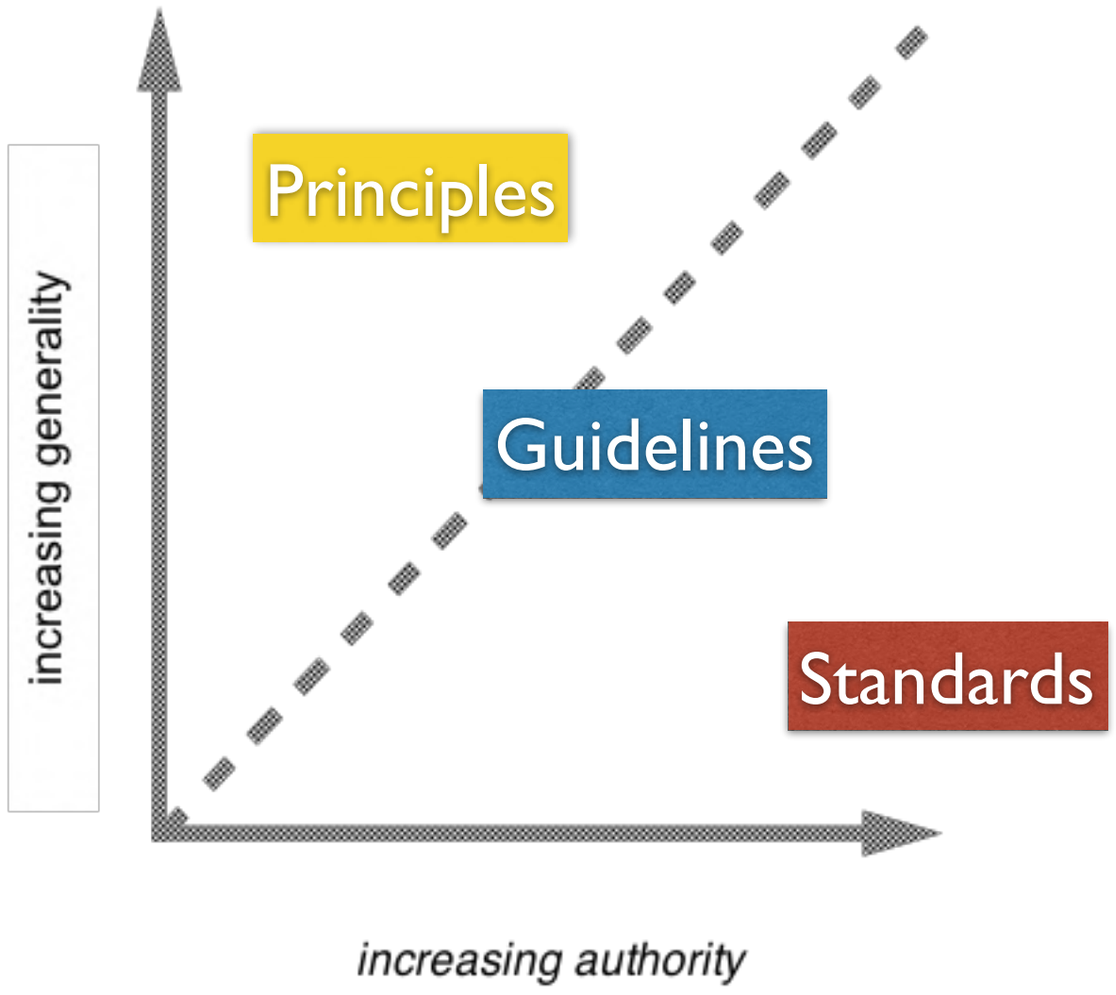
Guidelines - Java GUI Example

- In JavaFX use CSS skins

http://javafxtuts.com/javafx-css/

```
13 public class Javafxtuts extends Application {
14
15     @Override
16     public void start(Stage primaryStage) {
17         HBox root = new HBox();
18         //Set space or padding using setPadding() method
19         root.setPadding(new Insets(20));
20
21         //assiging a class to the button
22         Button button=new Button("my button");
23         //Adding a class to the button
24         button.getStyleClass().add("btn");
25
26         //assiging a class to the button1
27         Button button1 =new Button("Button1");
28         //set id to the button.
29         button1.setId("btn1");
30
31
32     root.getChildren().addAll(button,button1);
33     Scene scene = new Scene(root, 300, 150);
34     //To add a external css file we do as
35     String  style= getClass().getResource("New.css").toExternal
36     //now add the external css file to the scene
37     scene.getStylesheets().add(style);
38
39         primaryStage.setTitle("javafxtuts.com");
40         primaryStage.setScene(scene);
41         primaryStage.show();
42     }
43
44     /**
45      * @param args the command line arguments
46      */
47
48     public static void main(String[] args) {
49         launch(args);
50     }
```

Using design rules



Design rules

- Suggest how to increase usability
- Different kinds, containing different levels of **generality** and **authority**.

Golden Rules and Heuristics

- **We can find a kind of "Broad brush" rules**
- **They are useful "check list" for a good design**
- **You always get better with these instead of using none**
- **There are different collections:**
  - **Nielsen:** The 10 Heuristics.
  - **Norman:** The 7 Principles
  - **Shneiderman:** The 8 Golden Rules

The Nielsen's 10 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*

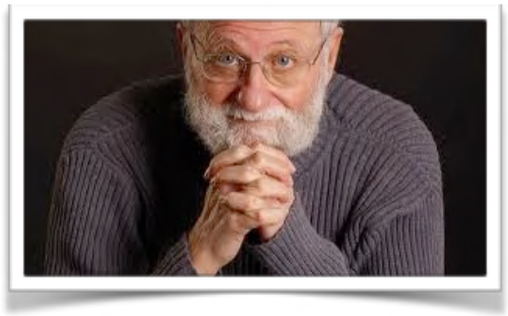
Jakob Nielsen

http://www.useit.com/papers/heuristic/heuristic_list.html

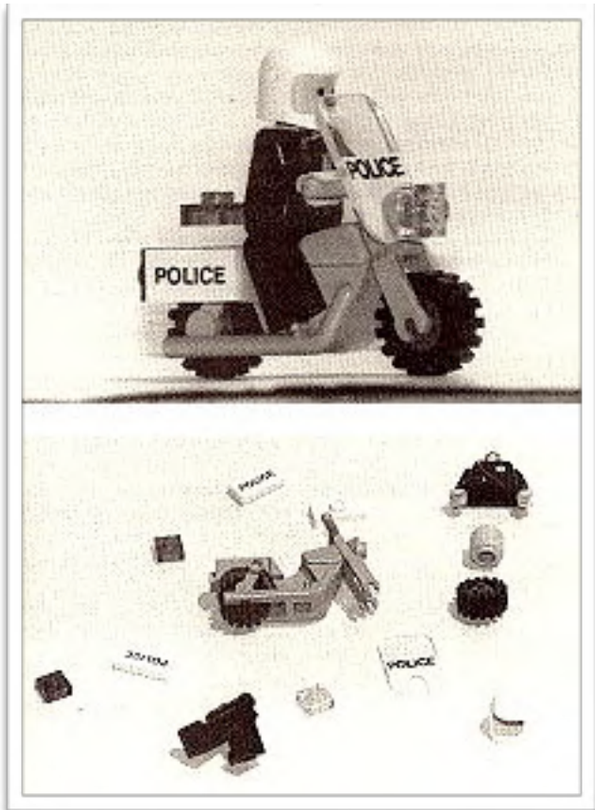
Norman's 7 Principles

1. *Use both knowledge in the world and knowledge in the head.*
2. *<u>Simplify</u> the structure of tasks*
3. *Make things <u>visible</u>: bridge the gulfs of Execution and Evaluation*
4. *Get the <u>mappings</u> right*
5. *Exploit the power of <u>constraints</u>, both natural and artificial.*
6. *Design for <u>error</u>*
7. *When all else fails, standardize*

- dicas mentais
- automação
- redesenhar.



Donald Norman

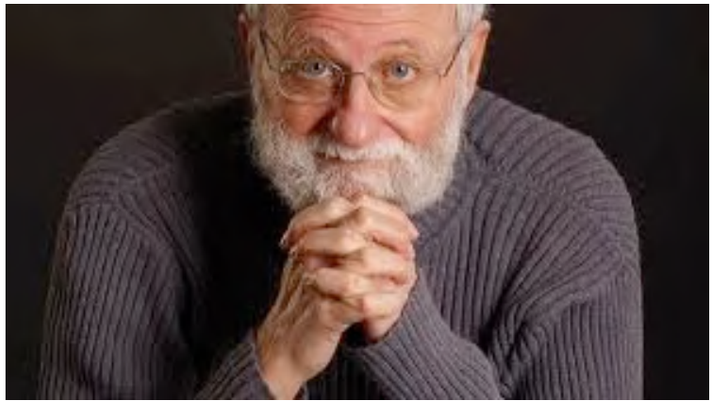


When there is no mapping we should define standards.
Note: The example of critical controls on a car.

Norman's 7 Principles

4. Get the <u>mappings</u> right

A funny story of bad mapping:



Donald Norman

The *Leitz* projector

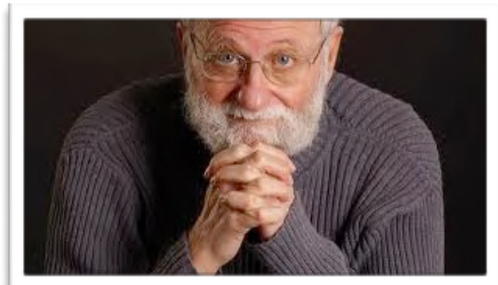
Taste (7) für Diawechsel am Gerät
Diawechsel vorwärts = kurz drücken,
Diawechsel rückwärtz = länger drücken.

Button (7) for changing the slides
Slide change forward = short press,
Slide change backward = longer press.

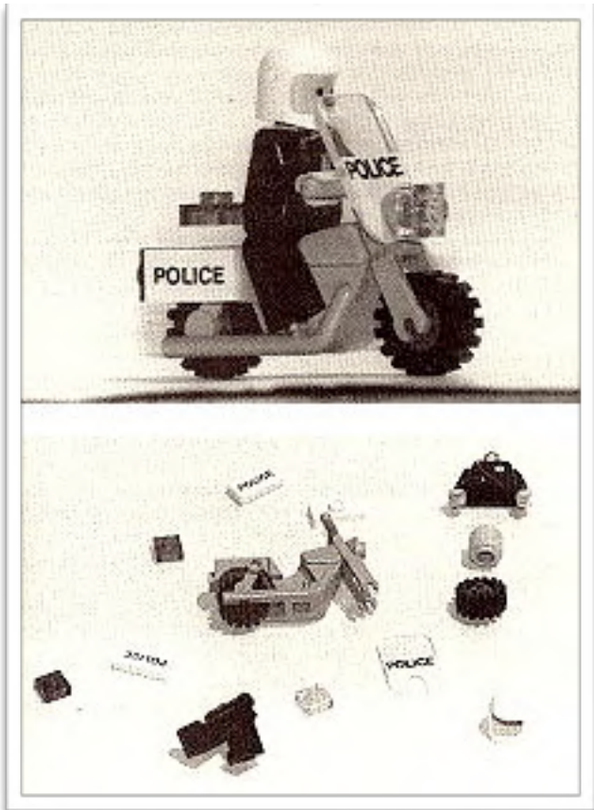
Norman's 7 Principles

1. *Use both knowledge in the world and knowledge in the head.*
2. *<u>Simplify</u> the structure of tasks*
3. *Make things <u>visible</u>: bridge the gulfs of Execution and Evaluation*
4. *Get the <u>mappings</u> right*
5. *Exploit the power of <u>constraints</u>, both natural and artificial.*
6. *Design for <u>error</u>*
7. *When all else fails, standardize*

- dicas mentais
- automação
- redesenhar.



Donald Norman



When there is no mapping we should define standards.
Note: The example of critical controls on a car.

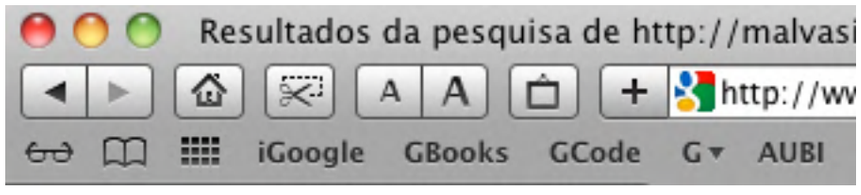
Shneiderman's 8 Golden Rules

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

Ben Shneiderman

HCI Design Patterns

- Way to **reuse** knowledge from previous good design experiences
- A pattern is an **invariant** solution for a recurrent problem, within a specific **context**.
- Examples:
  - "Natural light in two walls" (Architecture)
  - "Return to a safe place" (HCI)



- Patterns do not exist in isolation, but are **linked** to other patterns in languages allowing the generation of complete designs

HCI Design Patterns

- **Pattern Attributes:**
  - Capture the practical, non-theoretical design
  - Captures the common essential properties of good design examples
  - Represent the knowledge of design at **several levels**: social, organizational, conceptual, detailed
  - Incorporate values and **express** the human **dimension** in interaction design
  - Are **intuitive** and easy to read and can therefore be used for communication between all stakeholders
  - The language of the standards is **generative**, and can completely assist the development of design

HCI Design Patterns

- **Pattern Attributes**
  - Incorporate values and express the human dimension in interaction design

HCI Design Patterns

- **Pattern Attributes**
  - Incorporate values and express the human dimension in interaction design

HCI Design Patterns

- **Pattern Attributes**
  - Incorporate values and express the human dimension in interaction design

HCI Design Patterns

- **Pattern Attributes:**
  - Capture the practical, non-theoretical design
  - Captures the common essential properties of good design examples
  - Represent the knowledge of design at **several levels**: social, organizational, conceptual, detailed
  - Incorporate values and **express** the human **dimension** in interaction design
  - Are **intuitive** and easy to read and can therefore be used for communication between all stakeholders
  - The language of the patterns is **generative**, and can completely assist the development of design