



2025 - 2026

HUMAN COMPUTER INTERACTION

Chapter 4
Ergonomic Rules
in User Interfaces

INTRODUCTION

- Evaluation is a key phase in the design cycle of user interfaces (UI).
It allows us to **verify** and **validate** an interactive system.
- **Verification:** Does the system meet the specifications from requirements analysis?
- **Validation:** Does the system meet real user needs while respecting the application constraints?

ERGONOMICS – DEFINITIONS

- **Ergonomics** aims to adapt work environments to human physical and psychological characteristics.
- **Software ergonomics** focuses on adapting software to users — reducing errors, learning time, and making software easy and pleasant to use.

WHAT DO WE EVALUATE?

- **Reliability and technical quality**
- **Usability** – Can users easily reach their goals?
- **Interface quality (ergonomics)**
- **Utility** – Does the software serve the client's main goals?
- **Usage** – Is it actually used as expected?

WHEN TO EVALUATE

- **During design:** iterative prototypes, user-centered design (formative evaluation)
- **During development:** classical quality control
- **During deployment:** user satisfaction, incident analysis
- **Before purchase:** software comparison (summative evaluation)

EVALUATING SYSTEM UTILITY

- Does the system meet the high-level goals of the client?
- Can users complete their tasks with the available functions?
- It's important to **clearly define user tasks.**

EVALUATING SYSTEM USABILITY

- **Usability (ISO 9241):**

The extent to which a product can be used by specific users to achieve specific goals effectively, efficiently, and satisfactorily in a given context.

Main criteria:

- **Effectiveness** – achieving intended results
- **Efficiency** – using minimal resources
- **Satisfaction** – user comfort and positive perception
- **Ease of learning and ease of use**
- **Reliability** – few or no errors

USABILITY EVALUATION TECHNIQUES

- **Experimental evaluation:** user observation, data collection, interviews, questionnaires
- **Analytical evaluation:** expert inspection, usage scenarios, heuristic evaluation, ergonomic criteria

EXPERT EVALUATION

- Requires multiple experts (especially early in design)
- Helps detect major usability problems
- Needs usage scenarios and user profiles

 Experts can't fully replace real users and may have biased perspectives.

HEURISTIC ANALYSIS

- Heuristic evaluation involves inspecting an interface against a list of usability principles (“heuristics”) to identify strengths and weaknesses.
- It helps classify ergonomic issues, judge their severity, and find solutions.

 Focuses on usability standards, not functional adequacy.

ERGONOMIC EVALUATION FRAMEWORKS

Several well-known grids exist:

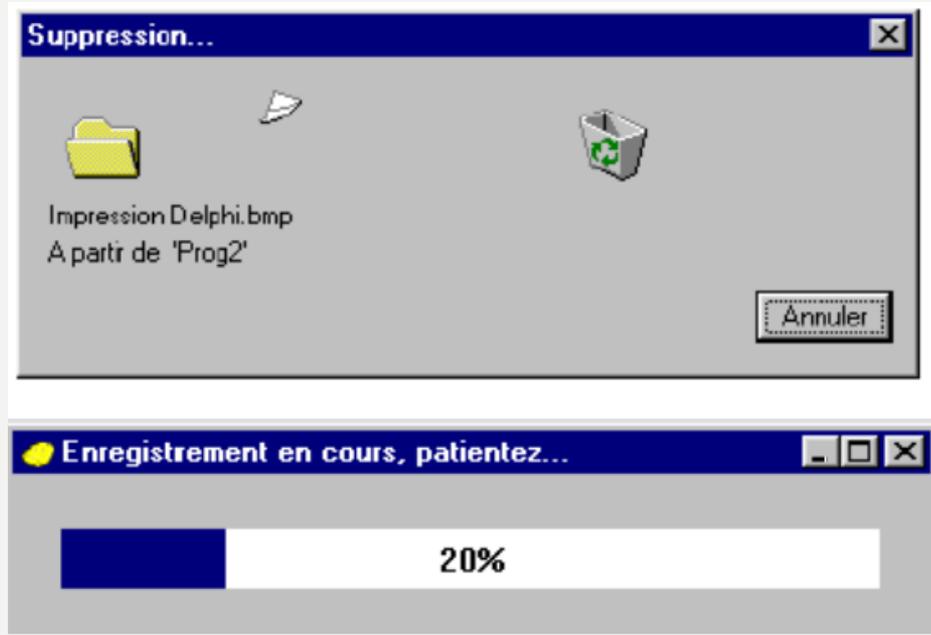
- **Bastien & Scapin's criteria**
- **Nielsen's heuristics**
- **Coutaz's golden rules**
- **Shneiderman's golden rules**
- **Tognazzini's principles, etc.**



***NIELSEN'S 10
HEURISTICS FOR USER
INTERFACE DESIGN***

1. VISIBILITY OF SYSTEM STATUS

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



2. MATCH BETWEEN THE SYSTEM AND THE REAL WORLD

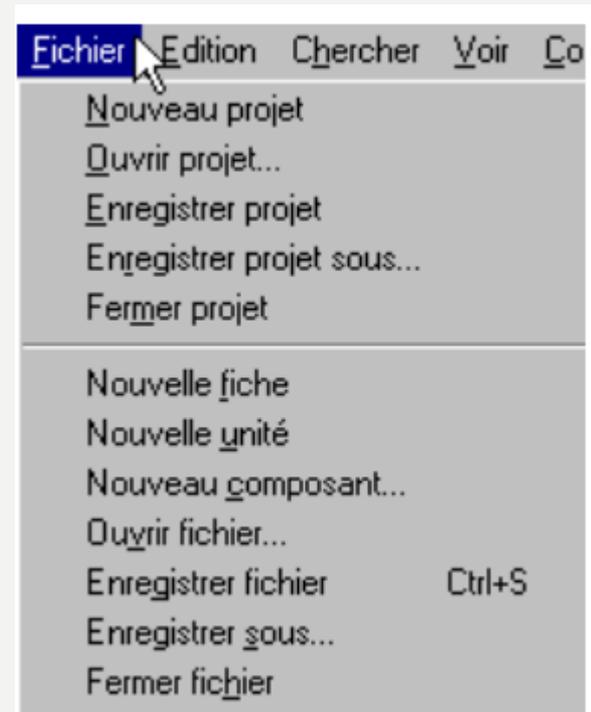
The interface should speak the users' language, using familiar words, phrases, and concepts — not system-oriented terms. Follow real-world conventions so that information appears in a natural and logical order.

3. USER CONTROL AND FREEDOM

- Users often perform actions by mistake. The system should provide clearly marked “emergency exits” to leave an unwanted state without having to go through long dialogs. Support *undo* and *redo* functions.

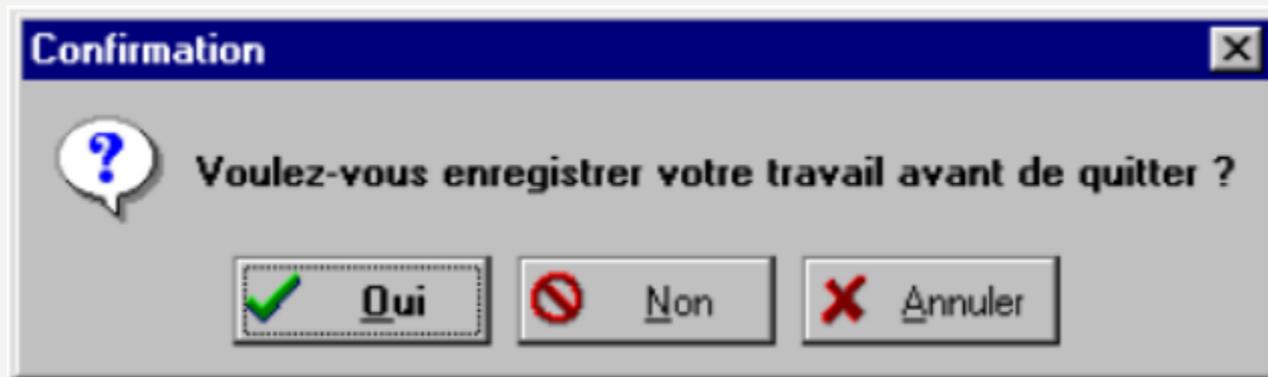
4. CONSISTENCY AND STANDARDS

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



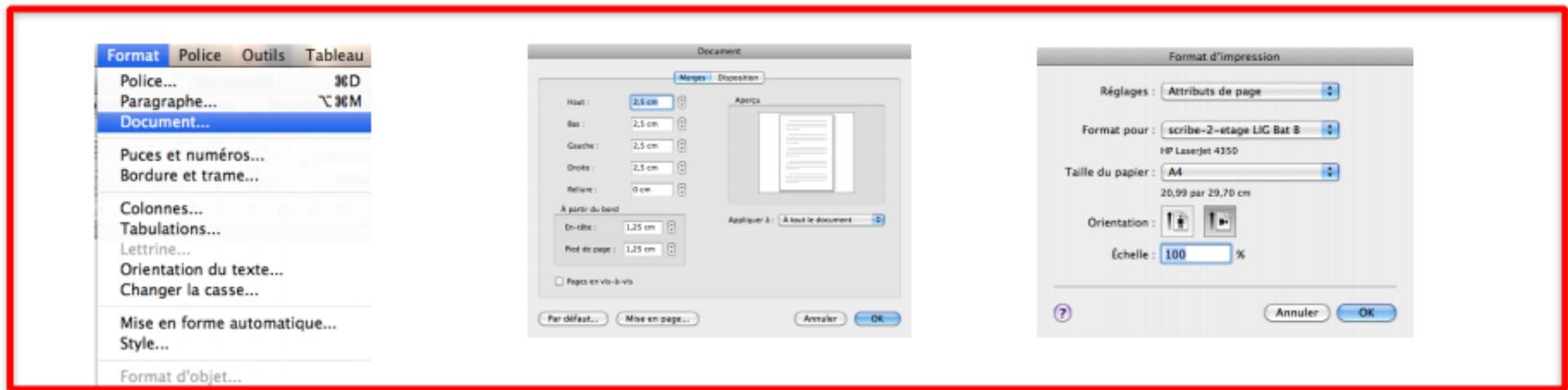
5. ERROR PREVENTION

Beyond good error messages, the design should prevent problems from occurring in the first place.



6. RECOGNITION RATHER THAN RECALL

Make objects, actions, and options visible. Users should not have to remember information from one screen to another. Instructions should be visible or easily retrievable whenever needed.



7. FLEXIBILITY AND EFFICIENCY OF USE

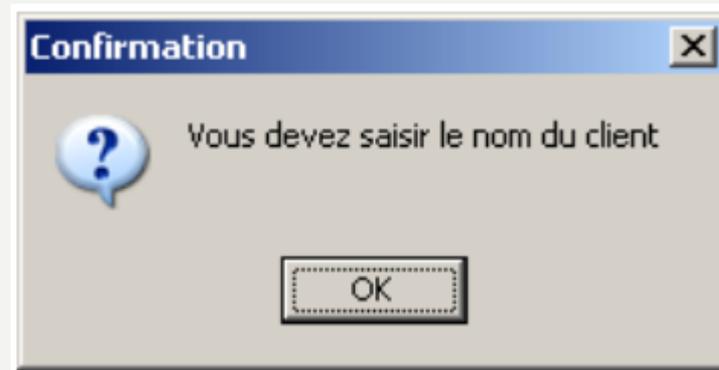
- Shortcuts—often invisible to novice users—can speed up interaction for experts. The interface should cater to both inexperienced and experienced users and allow customization of frequent actions

8. AESTHETIC AND MINIMALIST DESIGN

- Dialogs should not contain irrelevant or rarely needed information. Every extra piece of information competes with relevant information and reduces its visibility.

9. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS

- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and suggest a constructive solution.



10. HELP AND DOCUMENTATION

- Although it is best if the system can be used without documentation, it may still be necessary to provide help and documentation. Such information should be easy to search, task-focused, and concise.

THANK YOU