



University of M'sila Faculty of Mathematics and Computer
Science Department of Computer Science

PRINCIPLES OF SOFTWARE ERGONOMICS

Academic Year 2024-2025

Prepared and presented by SAOUDI Lalia

CONCEPT OF ERGONOMICS

- **Definition:**
- The term ergonomics comes from Greek and means the law of work:
 - ἔργον(ergon): work
 - νόμος(nomos): law
 - the science of work
- **Ergonomics of objects / of the workstation**
 - Ergonomics can roughly be defined as the study of people in their working environment. More specifically, an ergonomist (pronounced like economist) designs or modifies the work to fit the worker, not the other way around. The goal is to eliminate discomfort and risk of injury due to work.
 - aims to understand human/system interactions to adapt workstations to the physiological and psychological characteristics of the person, in order to
 - optimise system performance,
 - Enhancing safety, health, and comfort
- **Software ergonomics**
 - software ergonomics aims to understand the activity of a human interacting with software to adapt its interface according to human physiological and psychological characteristics
 - to improve the software, reduce errors, learning time
 - to design easily usable software
 - to envision new ergonomic interfaces

ERGONOMIC CRITERIA

- Ergonomic criteria are a set of principles, guidelines, and standards used to design and evaluate systems, to optimize overall system performance
- The establishment of ergonomic criteria primarily aims at two complementary objectives:
 - Assist in design: they serve as a guide during the design of a user interface with ergonomic aspects of the system.
 - Help in evaluation: They allow for the assessment of a system's usability (ergonomic audit) by serving as a basis for establishing evaluation grids.
- The ergonomic criteria allow for measuring the quality of the interface, according to several approaches

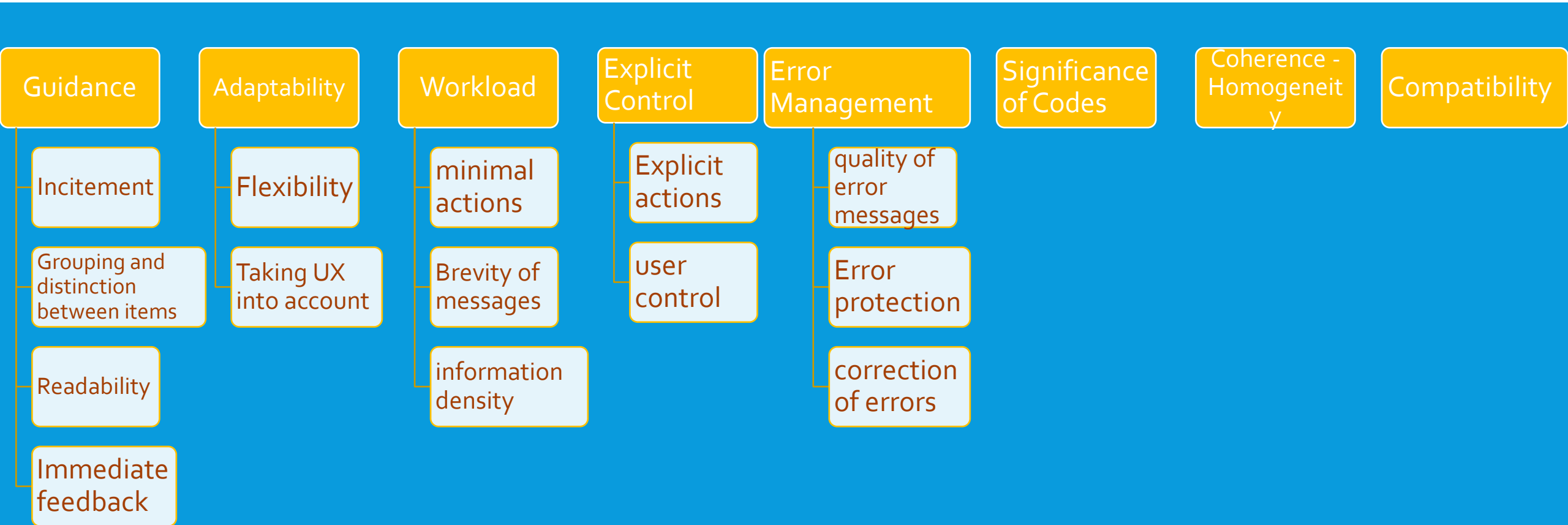
ERGONOMIC CRITERIA

- The ergonomic criteria allow for measuring the quality of the interface, according to several approaches:
 - The design guide by Smith and Mosier, Ergonomics in 128 points (BIT, IRSST)
 - The compilations of rules: Vanderdonckt (1992, + 3000 rules)
 - Nielsen's Usability Heuristics
 - Bastien and Scapin Criteria, 1993
- This course presents the framework of [Bastien and Scapin, 1993] which comprises 8 criteria refined into sub-criteria

ERGONOMIC CRITERIA: THE FRAMEWORK OF BASTIEN AND SCAPIN

- Christian Bastien and Dominique Scapin are two French researchers from INRIA.
- In 1993, they published universal and timeless rules to follow when designing interfaces or to identify the main usability issues
- The ergonomists Christian Bastien and Dominique Scapin synthesised about 900 recommendations gathered in the field of computer ergonomics.
- These recommendations are categorized into 8 main families :
 - Guidance, Workload, Explicit Control, Adaptability, Error Management ; Consistency, Significance of Codes, Compatibility
- Why these principles:
 - To have a common basis for discussion
 - To enable comparaison
 - To ensure consistency in evaluations by the same ergonomist
 - To be used in both design and evaluation

THE FRAMEWORK OF BASTIEN AND SCAPIN



THE BASTIEN AND SCAPIN FRAMEWORK:

1. GUIDANCE

Guidance:

Guidance involves assisting the user during the execution of a GUI by providing a set of means to advise, inform, and orient them during their interactions with the computer.

We distinguish:

- Explicit guidance (examples: messages, help pages, dialogue boxes,...)
- Implicit guidance examples: gray out inactive functions, limit the number of characters entered,...)
- The guidance criterion is broken down into four sub-criteria:
 1. Prompting
 2. Grouping/Distinction
 3. Immediate feedback
 4. Readability

THE BASTIEN AND SCAPIN FRAMEWORK:

1. GUIDANCE



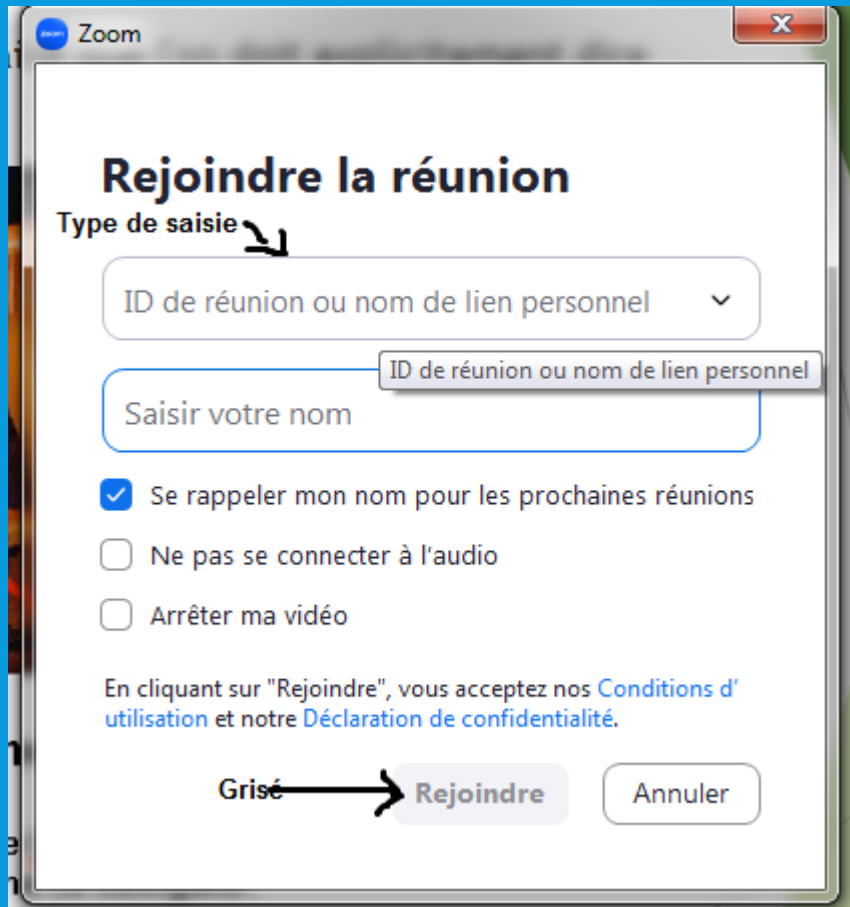
THE BASTIEN AND SCAPIN FRAMEWORK:

1.1. GUIDANCE-PROMPTING

- Prompting encourages the user to perform specific actions to relieve their cognitive load,
- It avoids tedious learning and reduces the risk of errors.
- **Recommendations:**
 - Grey out unavailable functions (menu options, buttons, ...)
 - Provide the list of expected inputs (drop-down lists, codes to use, ...)
 - Give the input format for the data (dates, dimensions, ...)
 - Change the shape of the cursor (mouse pointer) to provide indications on the operation to be performed
 - Clearly indicate mandatory fields (* or another indicator)
 - Clearly show how to go forward and backward
 - Display help bubbles (Tooltips) on non-trivial elements (toolbar icons, for example)

THE BASTIEN AND SCAPIN FRAMEWORK:

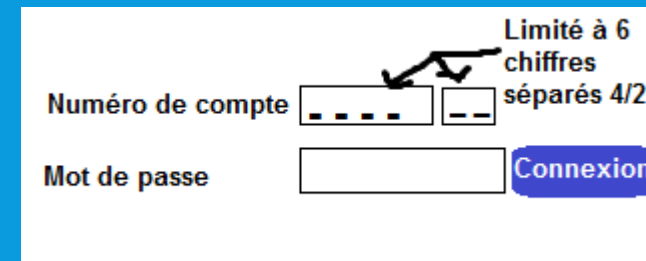
1.1. GUIDANCE-PROMPTING



A screenshot of a Zoom application window titled "Zoom". The main heading is "Rejoindre la réunion". Below it, the text "Type de saisie" is followed by a hand cursor icon pointing to a dropdown menu. The dropdown menu is open, showing the text "ID de réunion ou nom de lien personnel". Below the dropdown is a text input field with the placeholder "Saisir votre nom". There are three checkboxes: the first is checked and labeled "Se rappeler mon nom pour les prochaines réunions"; the second is unchecked and labeled "Ne pas se connecter à l'audio"; the third is unchecked and labeled "Arrêter ma vidéo". Below the checkboxes, there is a line of text: "En cliquant sur 'Rejoindre', vous acceptez nos [Conditions d'utilisation](#) et notre [Déclaration de confidentialité](#)." At the bottom, there are two buttons: "Rejoindre" and "Annuler". The "Rejoindre" button is highlighted with a grey background and has an arrow pointing to it from the word "Grisé" (Grayscale).



A screenshot of a light green notification box with a close button (X) in the top right corner. The text inside reads: "La page à l'adresse <https://www.blogger.com> indique :". Below this, it says: "Vous n'avez pas enregistré certains de vos changements, ils vont être perdus." At the bottom right, there are two buttons: "OK" and "Annuler".



A screenshot of a form for account creation. It has two rows. The first row is labeled "Numéro de compte" and has a text input field with four underscores. To the right of the input field, there is a label "Limité à 6 chiffres séparés 4/2" with an arrow pointing to the input field. The second row is labeled "Mot de passe" and has a text input field. To the right of the input field, there is a blue button labeled "Connexion".

THE FRAMEWORK OF BASTIEN AND SCAPIN:

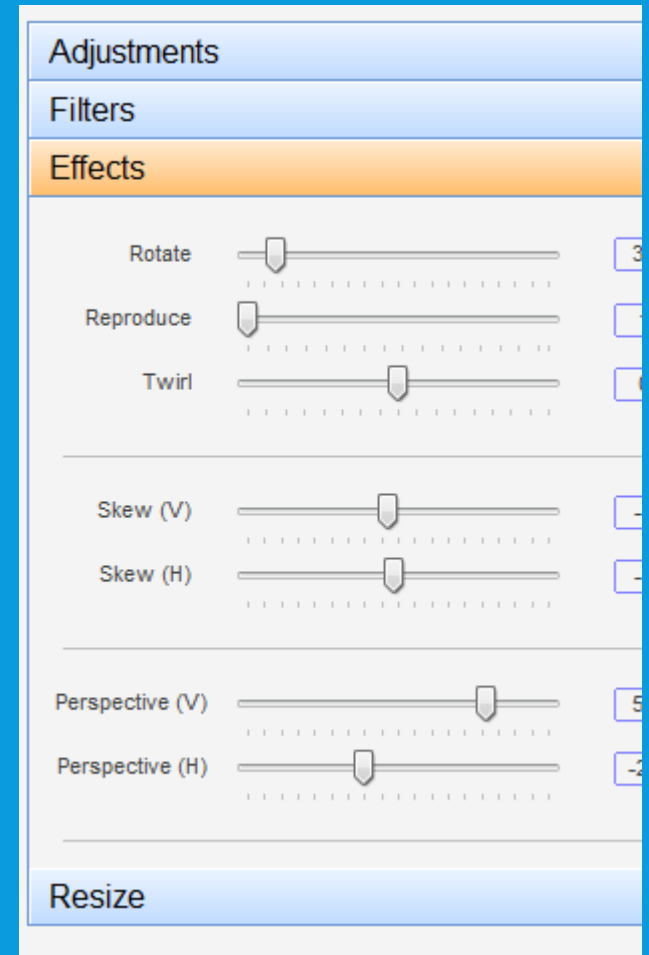
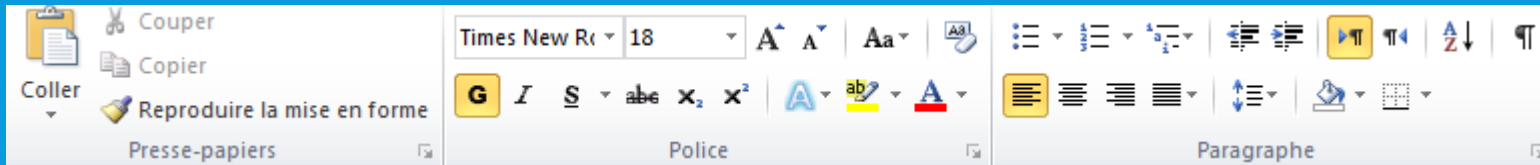
1.2. GUIDANCE-GROUPING/DIFFERENCE

- The grouping criterion consists of guiding the user by grouping information and functions of the same type.
 - A user generally considers that similar items will function in the same way (principle of similarity).
 - Two graphic attributes are used for this purpose:
 - The presentation format (color, shape, syntax,...)
 - The position in the interface (location, border,...)
- Conversely, the distinction criterion consists of clearly separating or assigning distinct properties to objects that are different or behave differently.
- **Recommendations:**
 - Group information of the same type by format or by position (grouping by menus, enclosing with a border,...)

THE FRAMEWORK OF BASTIEN AND SCAPIN:

1.2. GUIDANCE-GROUPING/DIFFERENCE

▪ Example



THE FRAMEWORK OF BASTIEN AND SCAPIN:

1.3. GUIDANCE- IMMEDIATE FEEDBACK

- The immediate feedback criterion brings together all the elements that serve to show the user what the system is doing and to provide feedback in response to each of their actions.
- Three main points to consider to respect this criterion when creating a user interface :
 - always display on the screen the inputs made by users
 - If a process is interrupted by the user or due to an event, the system should provide a message indicating a return to the previous state.
 - i In cases where processing takes a long time, information should be displayed informing users that processing is in progress.

THE FRAMEWORK OF BASTIEN AND SCAPIN:

1.3. GUIDANCE- IMMEDIATE FEEDBACK

- **Recommendations:**

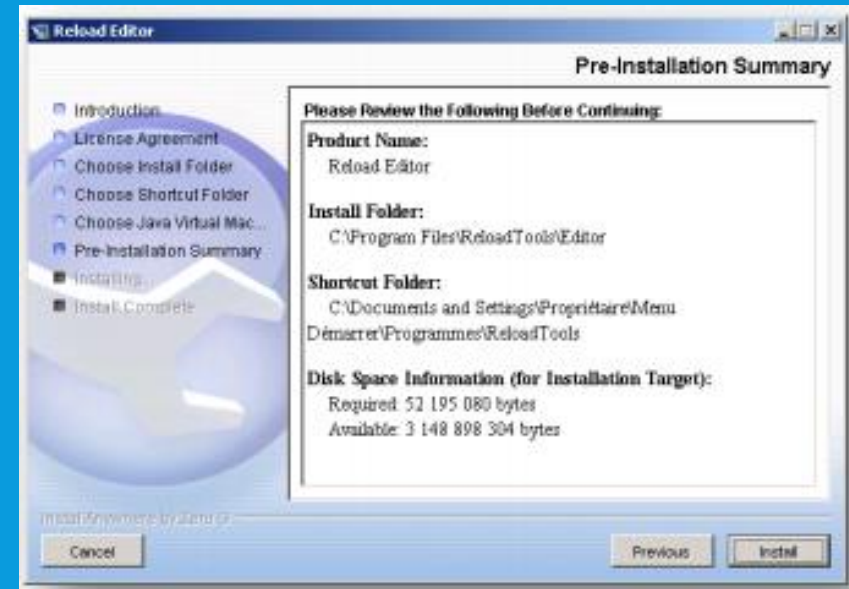
- Indicate long processes with a waiting indicator (disk, progress bar, animation, message, ...). Three orders of magnitude:
 - **0.1 s** : Not perceivable by the user
 - **1 s** : Perceptible (an indicator is needed)
 - **10 s** : The user can do something else (announce the wait time and progress, if possible)

THE REFERENCE FRAMEWORK OF BASTIEN AND SCAPIN: 1.4. GUIDANCE: READABILITY (GUIDANCE)

- The readability criterion consists of facilitating the perception of textual and iconographic information through a judicious choice of their properties and layout.
- Particular attention must be given to:
 - The fonts (variety, size, attributes, lowercase/uppercase,...)
 - The colors in general (choice, contrast, purpose,...)
 - The layout of elements (alignment, spacing, formatting,...)
 - The icons (dimensions, choice of symbols,...)

THE REFERENCE FRAMEWORK OF BASTIEN AND SCAPIN: 1.4. GUIDANCE: READABILITY (GUIDANCE)

- To avoid:



Test

THE REFERENCE FRAMEWORK OF BASTIEN AND SCAPIN: 2. ADAPTABILITY

- The adaptability of a system is characterized by the ability of its interface to react and adjust according to the context and according to the needs and preferences of its users.
- An interface cannot suit all its potential users simultaneously.
- The adaptability criterion is broken down into two elementary sub-criteria:
 - **Flexibility**
 - **Taking user experience into account**

THE FRAMEWORK OF BASTIEN AND SCAPIN:

2.1 ADAPTABILITY. FLEXIBILITY

- Flexibility is the ability of the interface to adapt to the varied actions of users, providing multiple ways to execute the same action.
- **Recommendations:**
 - Allow tasks (functions) to be performed using both the keyboard and the mouse
 - Allow triggering a frequent command from various locations within the application.
 - Allow the user to configure the software according to their preferences
 - Provide a quick way to access menu commands (shortcuts)
- Example:
 - Search for a contact: by number, by name, sequential search, in the contact list, in the call list

THE FRAMEWORK OF BASTIEN AND SCAPIN:

2.2 ADAPTABILITY. USER EXPERIENCE

- adapt the IHC to the different levels of user experience.
 - To meet this criterion, differentiated means must be provided to take into account the differences in experience that exist between users according to their profile.
 - It should be noted that user experience can vary over time.
- **Recommendations:**
 - Allow the user to define their level of experience. Inquire or determine it automatically using metrics (for example, the number or rate of errors, reaction times, etc.).
 - Guide the novice user step by step in completing the task.
 - Implement strong guidance by creating wizards.
 - Provide experienced users with ways to perform their tasks quickly and efficiently (even if this sometimes comes at the expense of guidance).
 - Example: the ZOOM of the window

THE FRAMEWORK OF BASTIEN AND SCAPIN:

3. COMPATIBILITY

- Compatibility is the software's ability to integrate into the users' actual activity.
- This criterion measures the software's suitability with the physical and social context in which it is used (work environment):
 - The agreement between user characteristics (memory, perceptions, habits, skills, age, expectations, etc.) and tasks, on one hand, and the organisation of outputs, inputs and dialogue on the other hand.
 - Compatibility also concerns the degree of similarity between various environments or applications.

THE FRAMEWORK OF BASTIEN AND SCAPIN:

3. COMPATIBILITY

Recommendations:

Product level:

- users know other products:
 - leverage this knowledge
 - in the same company, have a user interface style
 - Use familiar metaphors

Task level:

- think in terms of user tasks and facilitate the transition from one task to another (multi-windowing)
- matching the user's knowledge with the software's capabilities:
 - familiar environment and usual "facilitated learning"
- Speak the user's language (avoid IT jargon)

Example: same menu bar between Word and Excel.

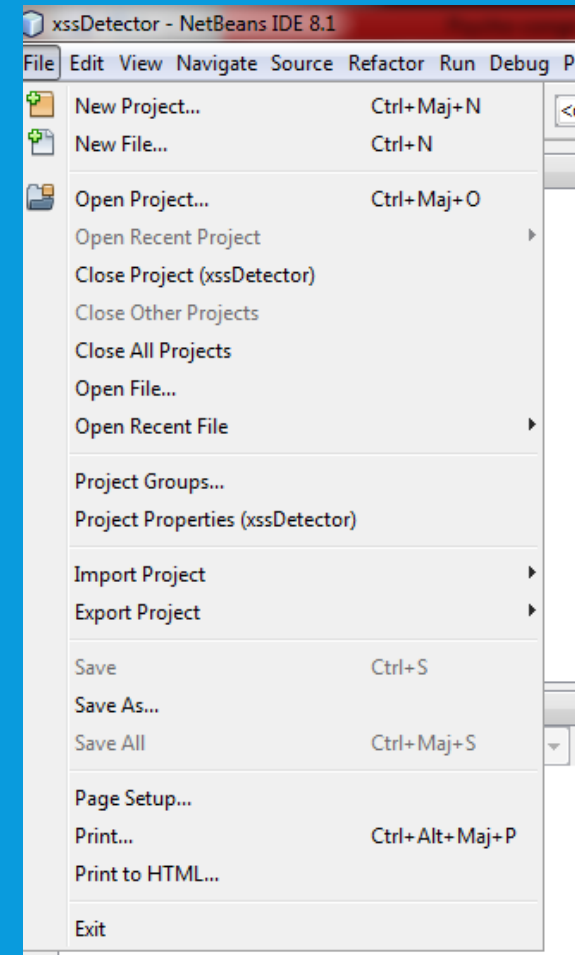
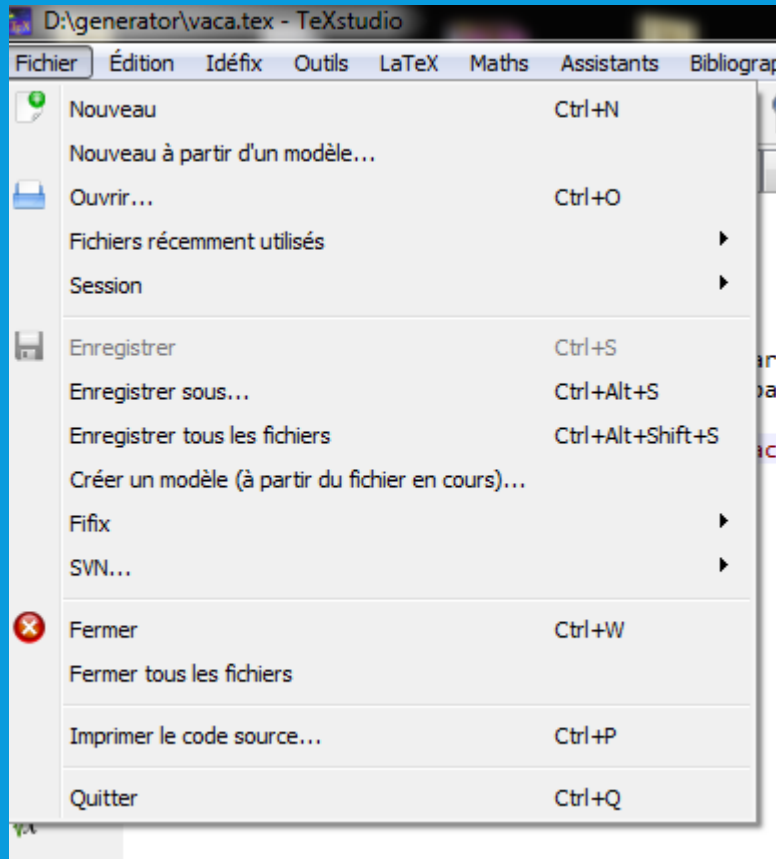
THE REFERENCE FRAMEWORK OF BASTIEN AND SCAPIN:

4. COHERENCE - HOMOGENEITY

- It is the ability of a computer system to maintain a constant logic of use within an application or across applications (procedural level and level of information presentation).
- A lack of uniformity can significantly increase execution time.
- **Recommendations:**
 - Use the same layout scheme for all windows (screen template also known as a regulatory layout)
 - The semantics of mouse buttons must be consistent
 - The same vocabulary should be used to denote the same functions
 - Use a consistent organisation and syntax for menus
 - Use graphic symbols consistently (icons, colours, ...)

THE REFERENCE FRAMEWORK OF BASTIEN AND SCAPIN:

4. COHERENCE - HOMOGENEITY



THE FRAMEWORK OF BASTIEN AND SCAPIN:

5. EXPLICIT CONTROL

- It represents the system's consideration of the explicit actions of the user.
- The more the users' inputs are explicitly defined by themselves, the more ambiguities and errors are limited.
- Two criteria for this category:
 - Explicit actions: the system must take into account the desired actions of the user at the moment they request them.
 - User control: the user must always have the ability to control their activities, always allowing the initiative to the user: letting them move between their activities, interrupt them, and stop them.

THE FRAMEWORK OF BASTIEN AND SCAPIN: 5. EXPLICIT CONTROL

1- Explicit actions:

the system must take into account the desired actions of the user at the moment they request them.

Recommendations

- Do not trigger operations without the explicit consent of the user.
- Trigger the operation immediately after the user's action or, if not, clearly indicate that the operation will be deferred (or cannot be performed)

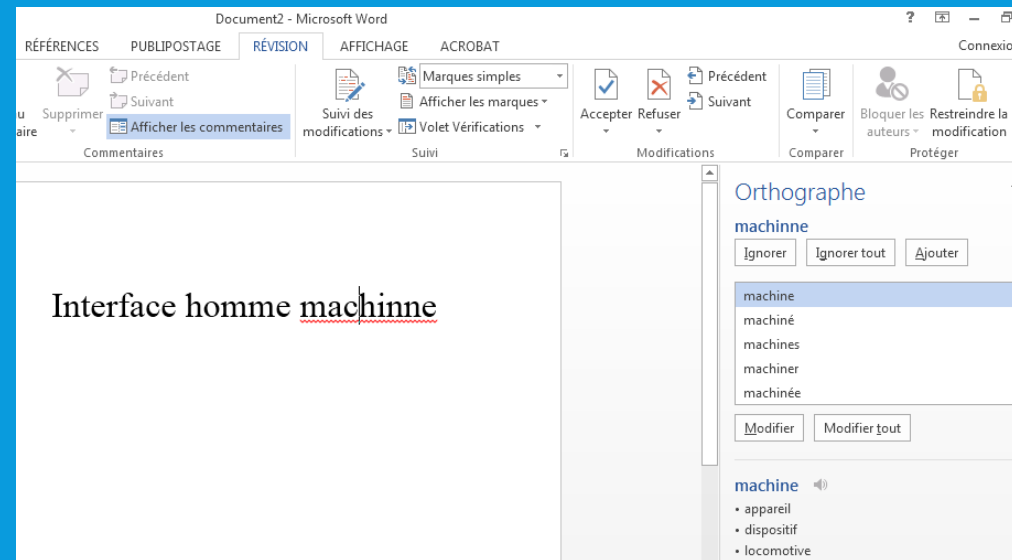
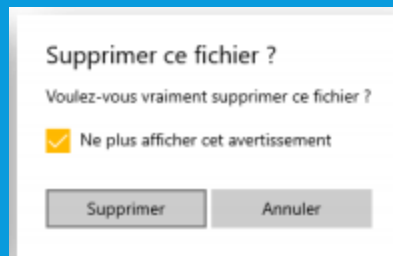
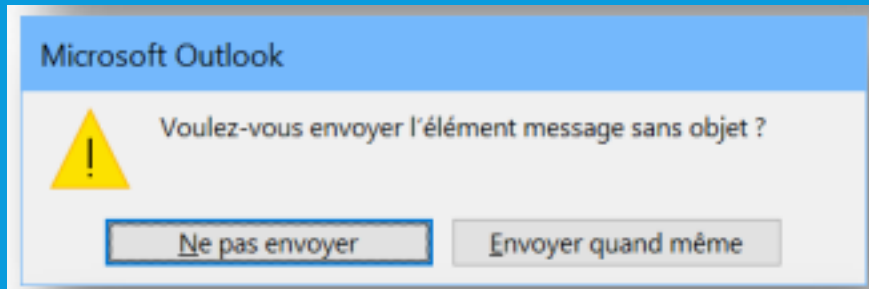
Things to avoid

- Automatic recording without explicit permission

THE FRAMEWORK OF BASTIEN AND SCAPIN: 5. EXPLICIT CONTROL

1- Explicit actions:

Example:



THE FRAMEWORK OF BASTIEN AND SCAPIN: 5. USER CONTROL

- 2. User control: the user must always have the ability to control their activities, always allowing the initiative to the user: letting them move between their activities, interrupt them, and stop them.
- Aims to make the user autonomous in their interaction with the system by giving them mastery of the process



THE FRAMEWORK OF BASTIEN AND SCAPIN:

6. ERROR MANAGEMENT

She is interested in ways to avoid or reduce errors on the one hand, and on the other hand to correct them when they occur.

- Three sub-criteria must be taken into account:
 - **Protection against errors (preventing the user from making them)**
 - **Quality of error messages (clearly informing the user)**
 - **Error correction (allowing the user to correct them)**

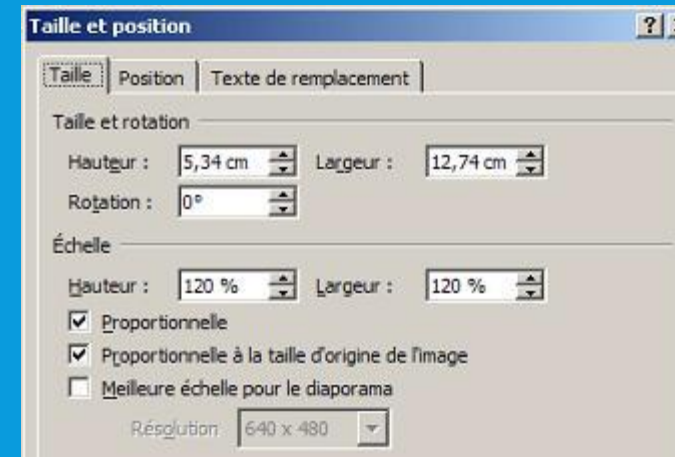
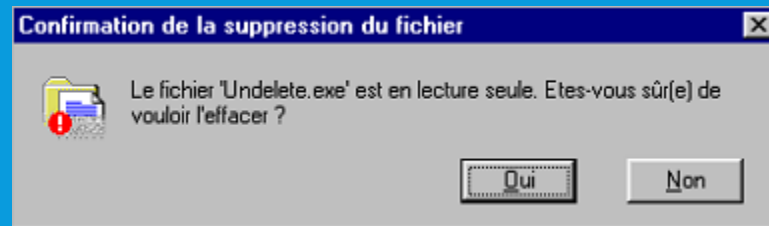
THE BASTIEN AND SCAPIN FRAMEWORK: 6.1 ERROR MANAGEMENT: PROTECTION AGAINST ERRORS

- **Protection against errors**
 - The aim is to prevent interruptions due to errors from hindering the smooth progress of the task.
 - Errors are prevented by guiding the user in such a way that they are not inclined to make them.
- **Recommendations:**
 - Highlight unavailable commands (grey them out) Prevention
 - Provide the list of possible values, units, etc. Prevention
 - Detect errors as early as possible and notify the user immediately
 - Minimize keyboard input (if possible, use dropdown lists)
 - Prevent the risk of data loss (ask for confirmation)

THE BASTIEN AND SCAPIN FRAMEWORK:

6.1 ERROR MANAGEMENT: PROTECTION AGAINST ERRORS

- Protection against errors
- Example:



THE BASTIEN AND SCAPIN FRAMEWORK:

6.2 ERROR MANAGEMENT: QUALITY OF ERROR MESSAGES

- Concerns the relevance, readability, and accuracy of the information on the nature of the errors made.
- For error correction to be easy, the error message must indicate the nature of the error, its cause, and how to correct it (the quality of messages promotes system learning).

THE BASTIEN AND SCAPIN FRAMEWORK:

6.3 ERROR MANAGEMENT: ERROR CORRECTION

- Providing the user with the means to correct their errors as easily and quickly as possible by allowing them, for example, to only correct the erroneous portion of the data and not forcing them to re-enter all the information.
- Different tactics depending on the type and severity of the error:
 - Block the user as long as the error persists (serious errors)
 - Allow them to continue after a warning (message, sound signal)
 - Do not respond to the erroneous command (a message is necessary)
 - Automatically correct the error (in a few rare cases)
- **Recommendations:**
 - Highlight the erroneous field or item
 - Allow the cancellation of an action or a series of actions (e.g. removing items from the virtual cart of an e-commerce application).
 - Offer alternatives (e.g. 'Related topics' for an unsuccessful search).

THE FRAMEWORK BY BASTIEN AND SCAPIN:

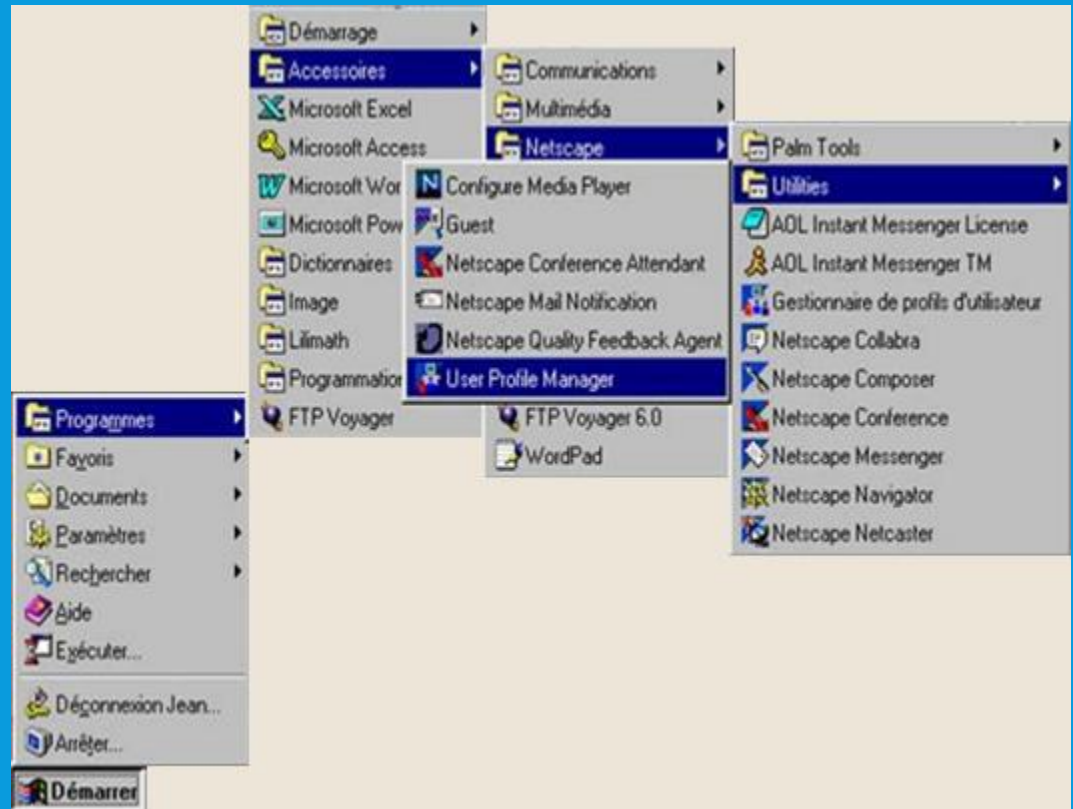
7. WORKLOAD

- Which mainly consists of reducing the perceptual activities, memorization, and execution of actions by the user (short data and action).
- We should therefore aim to minimize both the amount of information that the user has to take into account and the number of elementary actions they have to perform to complete a given task.
- The workload breaks down into two elementary sub-criteria:
 - **Brevity**
 - **Information density**

THE FRAMEWORK BY BASTIEN AND SCAPIN: 7 WORKLOAD: BREVITY

- **Brevity**
- It involves limiting as much as possible the reading, input, and the steps that users must go through (i.e., increasing the efficiency of the dialogue).
- **Recommendations:**
 - Limit the number of options in a menu or in a dropdown list
 - Avoid overly long labels
 - Reduce the number of elementary actions to achieve a given goal.
 - Prevent the user from having to remember information from one window to another, having to perform calculations, or input information that can be inferred by the system.

THE BASTIEN AND SCAPIN FRAMEWORK: 7 WORKLOAD: BREVITY



THE BASTIEN AND SCAPIN FRAMEWORK:

7. WORKLOAD: INFORMATIONAL DENSITY

- The criterion of informational density represents the workload for sets of elements. Only frequently used elements should be presented.
- User performance is negatively affected when the informational load is too high or (less commonly) too low.
- **Recommendations:**
 - Only display relevant information to carry out the task (simple dialogues, graphical representations, ...)
 - Avoid overly busy screens (break down if necessary)
 - Avoid too many links in a text displayed on a web page
 - Avoid overly verbose texts (simple dialogue, short sentences)
 - Prioritize recognition (symbols, icons)

THE BASTIEN AND SCAPIN FRAMEWORK: 7 WORKLOAD: INFORMATIONAL DENSITY

- Example:



THE BASTIEN AND SCAPIN FRAMEWORK: 8 SIGNIFICANCE OF CODES AND DENOMINATIONS

- Adequacy between the object or information displayed or entered and its referent.
- When coding is meaningful, recall and recognition are better:
- **Recommendations:**
 - Avoid technical terms (jargon), speak the user's language
 - Replicate the usual behavior of objects (e.g. the eraser)
 - Explicitly define and respect to abbreviation rules
 - Take into account applicable standards (formal or de facto standards) for all denominations
 - Example: icons:

