

# Critères Ergonomiques pour l'Évaluation d'Interfaces Utilisateurs d'après [Bastien & Scapin 1993].

Renaud Blanch <blanch@imag.fr>

Université Joseph Fourier, Polytech'Grenoble & UFR IMAG

février 2011

# Plan (1/2)

- 1** Introduction
- 2** Guidage
- 3** Charge de travail
- 4** Contrôle explicite
- 5** Adaptabilité

...

# Plan (2/2)

...

**6** Gestion des erreurs

**7** Homogénéité/cohérence

**8** Signifiante des codes et dénominations

**9** Compatibilité

**10** Bibliographie

# Ressources en ligne

Le page consacrée au cours est ici :  
<http://iihm.imag.fr/blanch/M1/EIHM/>.

# Objectifs du cours

Après avoir suivi ce cours, vous saurez :

- **citer** et **justifier** les critères ergonomiques de Bastien & Scapin ; et
- **évaluer** au regard de ces critères des interfaces utilisateurs et **donner des recommandations** pour les améliorer le cas échéant.

# Ergonomie

D'après la **Société d'Ergonomie de Langue Française** [1988], l'ergonomie est :

## définition

“la mise en œuvre de **connaissances scientifiques relatives à l'Homme**, et nécessaires pour **concevoir des outils, des machines et des dispositifs** qui puissent être utilisés avec le **maximum de confort, de sécurité et d'efficacité** pour le plus grand nombre”.

# Compatibilité Homme-machine

L'ergonomie travaille à trois niveaux :

- au niveau **physique**, pour une adéquation entre l'homme d'un point de vue morphologique et son poste de travail ;

# Compatibilité Homme-machine

L'ergonomie travaille à trois niveaux :

- au niveau **physique**, pour une adéquation entre l'homme d'un point de vue morphologique et son poste de travail ;
- au niveau **social**, pour un environnement de travail compatible avec les attentes des utilisateurs ; et



# Compatibilité Homme-machine

L'ergonomie travaille à trois niveaux :

- au niveau **physique**, pour une adéquation entre l'homme d'un point de vue morphologique et son poste de travail ;
- au niveau **social**, pour un environnement de travail compatible avec les attentes des utilisateurs ; et
- au niveau **cognitif**, pour un traitement et une représentation des informations conformes aux attentes et capacités des utilisateurs.

# Ergonomie des logiciels

L'**ergonomie des logiciels** se situe au **niveau cognitif** et recherche trois niveaux de compatibilité :

- **perceptivo-moteur** : compatibilité entre dispositifs d'interaction et caractéristiques physiques de l'utilisateur ;
- **linguistique** : codage des informations ; et
- **activité** : structure logicielle adaptée et adaptable aux modes de raisonnement de l'utilisateur dans la réalisation de sa tâche.

# Critères ergonomiques

Il existe de **nombreuses recommandations** recensées dans des **guides ergonomiques**.

Ces recommandations sont organisées par **critères** au sein de **référentiels** : ISO, Schneiderman, IFIP, Nielsen, etc.

# Critères ergonomiques

Il existe de **nombreuses recommandations** recensées dans des **guides ergonomiques**.

Ces recommandations sont organisées par **critères** au sein de **référentiels** : ISO, Schneiderman, IFIP, Nielsen, etc.

Ce cours présente le **référentiel de [Bastien & Scapin 1993]** qui comporte 8 critères affinés en sous-critères.

Ces critères ne sont **pas orthogonaux**, et il n'est fourni **ni métrique, ni fonction d'évaluation**.

# Guidage

## définition

Le **guidage** est l'ensemble des moyens mis en œuvre pour **conseiller, orienter, informer, et conduire l'utilisateur** lors de ses interactions avec l'ordinateur (messages, alarmes, labels, etc.), y compris dans ses aspects lexicaux.

# Guidage (cont.)

## justification

**Facilite l'apprentissage et l'utilisation** du système en permettant :

- **de savoir où on se trouve** dans une séquence d'interactions, ou dans l'accomplissement d'une tâche ;
- **de connaître les actions permises** ainsi que leurs conséquences ; et
- **d'obtenir de l'information** supplémentaire

pour de **meilleures performances** et **moins d'erreurs**.

# Guidage (cont.)

## sous-critères

- Incitation
- Groupement / distinction entre items
- *Feedback* immédiat
- Lisibilité

# Incitation

## définition

L'**incitation** recouvre les moyens mis en œuvre pour **amener les utilisateurs à effectuer des actions spécifiques** (e.g., entrée de données ou autre).

Ce critère englobe aussi tous les mécanismes ou moyens pour faire **connaître les alternatives**, lorsque plusieurs actions sont possibles, selon les états ou contextes courants.

L'incitation concerne également les informations permettant aux utilisateurs de **savoir où ils en sont, d'identifier l'état ou contexte dans lequel ils se trouvent**, de même que les outils d'aide et leur accessibilité.



# Incitation (cont.)

## justification

Une bonne incitation guide les utilisateurs et leur **évite** par exemple **d'avoir à apprendre** une série de commandes.

Elle permet aussi aux utilisateurs de **savoir quel est le mode ou l'état en cours**, où ils se trouvent dans le dialogue et ce qu'ils ont fait pour s'y trouver.

Une bonne incitation **facilite** donc **la navigation** dans une application et **permet d'éviter les erreurs**.

# Incitation (cont.)

## exemples de recommandations

- Guider les entrées de données en indiquant le format adéquat et les valeurs acceptables ; par exemple, fournir, au niveau du label, des indices supplémentaires sur le format d'entrée des données.
- Afficher les unités de mesure des données à saisir.
- Indiquer toutes les informations d'état (e.g., modes, valeurs)

# Incitation (cont.)

- Pour chaque champ de données, fournir un label.
- Fournir des indices sur la longueur autorisée des entrées dans un champ.
- Donner un titre à chaque fenêtre.
- Fournir des aides accessibles en ligne.

# Groupement/distinction entre items

## définition

Le critère **groupement/distinction entre items** concerne l'**organisation visuelle des items** d'information **les uns par rapport aux autres**.

Ce critère prend en compte la **topologie** (localisation) et certaines **caractéristiques graphiques** (format) afin d'illustrer les relations entre les divers items affichés, leur appartenance ou non-appartenance à une même classe, ou encore dans le but de **montrer la distinction** entre différentes classes d'items. Ce critère concerne aussi l'organisation des items à l'intérieur d'une même classe.

# Groupement/distinction entre items (cont.)

## justification

La **compréhension** d'un écran **dépend**, entre autres choses, **de l'arrangement, du positionnement et de la distinction des objets** (images, textes, commandes, etc.) qui y sont présentés.

## Groupement/distinction entre items (cont.)

Les utilisateurs auront plus de **facilité à repérer les items** et/ou groupes d'items et à connaître leurs liens si ils sont **présentés de façon organisée** (e.g., alphabétique, fréquence d'utilisation) d'une part, et si ces items ou groupes d'items sont **présentés dans des formats ou codages** qui permettent d'illustrer **leurs similitudes ou leurs différences**, d'autre part. De même, les utilisateurs pourront mieux les apprendre et s'en rappeler. Le **groupement/distinction** entre items entraîne un meilleur **guidage**.

# Groupement/distinction entre items (cont.)

## (sous) sous-critères

- Groupement/distinction par la localisation
- Groupement/distinction par le format

# Groupement/distinction — par la localisation

## définition

Le critère **groupement/distinction par la localisation** concerne le **positionnement** des items les uns par rapport aux autres dans le but d'indiquer leur **appartenance ou non-appartenance** à une même classe, ou encore dans le but de montrer **la distinction** entre différentes classes. Ce critère concerne aussi l'organisation des items à l'intérieur d'une même classe.



# Groupement/distinction — par la localisation (cont.)

## exemples de recommandations

- Organiser les items en listes hiérarchiques.
- Grouper les options de menus en fonction des objets sur lesquels elles s'appliquent.
- Lorsque plusieurs options sont présentées, l'organisation de ces dernières doit être logique, i.e., représenter une organisation fonctionnelle pertinente ou significative (arrangement alphabétique, fonctionnel, fréquence d'utilisation, etc.)

# Groupement/distinction — par le format

## définition

Le critère **groupement/distinction par le format** concerne plus particulièrement les **caractéristiques graphiques** (format, couleur, etc.) permettant de faire apparaître l'**appartenance ou la non-appartenance** d'items à une même classe, ou encore permettant d'indiquer des distinctions entre classes ou bien encore des distinctions entre items d'une même classe.

# Groupement/distinction — par le format (cont.)

## exemples de recommandations

- Établir une distinction visuelle entre des aires ayant des fonctions différentes (commande, message, etc.)
- Établir une distinction visuelle entre les labels et les champs d'entrée.

# Feedback immédiat

## définition

Le **feedback immédiat** concerne les **réponses de l'ordinateur consécutives aux actions des utilisateurs**, lesquelles peuvent être le simple appui sur une touche ou l'entrée d'une séquence de commandes. Dans tous les cas, l'ordinateur doit répondre, **dans les plus brefs délais**, avec un délai de réponse approprié et homogène selon les types de transactions. Dans tous les cas, une réponse **aussi immédiate que possible** doit être fournie à l'utilisateur le renseignant sur l'action accomplie et sur son résultat.

## Feedback immédiat (cont.)

### justification

La **qualité** et la **rapidité du *feedback*** sont deux **facteurs importants** pour l'établissement **de la confiance et de la satisfaction** des utilisateurs ainsi que pour leur compréhension du dialogue.

Ces facteurs permettent aux utilisateurs de se faire une **bonne représentation du fonctionnement du système**.

L'absence de *feedback* ou des délais trop importants entre les actions utilisateur et le *feedback*, peuvent déconcerter les utilisateurs, ce qui augmente les chances que les utilisateurs entreprennent des actions qui **risquent d'entraver les transactions en cours**.

## Feedback immédiat (cont.)

### exemples de recommandations

- Toujours faire apparaître sur l'écran les entrées effectuées par les utilisateurs, sauf pour les entrées confidentielles. Cependant, même dans ce cas, un *feedback* perceptible doit être fourni (e.g., l'affichage d'astérisques correspondant aux entrées clavier).

## Feedback immédiat (cont.)

- Suite à l'interruption d'un traitement par l'utilisateur, le système devrait fournir un message indiquant le retour à l'état précédent.
- Dans les cas où les traitements sont longs, une information indiquant aux utilisateurs que les traitements sont en cours devrait leur être fournie.

# Lisibilité

## définition

Le critère **lisibilité** concerne les **caractéristiques lexicales** de présentation des informations sur l'écran pouvant entraver ou **faciliter la lecture** de ces informations

(luminance des caractères, contraste caractères fond, dimension des lettres, espacement entre les mots, espacement entre les lignes, espacement entre les paragraphes, longueur des lignes, etc.)

Par convention, le critère **lisibilité** ne concerne ni le *feedback* ni les messages d'erreurs.



# Lisibilité (cont.)

## justification

La **performance est accrue** lorsque la présentation des informations à l'écran tient compte des caractéristiques cognitives et perceptives des utilisateurs.

Une **bonne lisibilité facilite la lecture** des informations présentées. Ainsi par exemple, les lettres sombres sur fond clair sont plus faciles à lire que l'inverse ; le texte présenté en lettres majuscules et minuscules est lu plus rapidement que le texte présenté seulement en lettres majuscules.

# Lisibilité (cont.)

## exemples de recommandations

- Les titres doivent être centrés.
- Les labels doivent être en majuscule.
- Le curseur doit être facilement repérable.
- Il est préférable de présenter un texte avec quelques lignes longues plutôt que de nombreuses lignes courtes.
- Les lignes de texte continu doivent être d'au moins 50 caractères.

## Lisibilité (cont.)

- La justification à droite des textes ne devrait être utilisée qu'avec un espacement variable, de sorte qu'un espacement proportionnel constant entre les lettres et les mots soit respecté.
- Dans les affichages de textes, utiliser le moins possible la césure des mots.

# Charge de travail

## définition

Le critère **charge de travail** concerne l'ensemble des éléments de l'interface qui ont un rôle dans la **réduction de la charge perceptive ou mnésique** des utilisateurs et dans **l'augmentation de l'efficacité du dialogue**.

## Charge de travail (cont.)

### justification

Plus la **charge de travail** est **élevée**, plus **grands** sont les **risques d'erreurs**.

De même, moins l'utilisateur sera distrait par des informations non pertinentes, plus il pourra effectuer sa tâche efficacement. Par ailleurs, plus les actions requises seront courtes, plus rapides seront les interactions.

# Charge de travail (cont.)

## sous-critères

- Brièveté
- Densité informationnelle

# Brièveté

## définition

Le critère **brièveté** concerne la charge de travail au niveau **perceptif et mnésique** à la fois pour les éléments individuels d'entrée ou de sortie et les séquences d'entrées (i.e., les suites d'actions nécessaires à l'atteinte d'un but, à l'accomplissement d'une tâche).

Il s'agit ici de **limiter autant que possible le travail de lecture, d'entrée et les étapes** par lesquelles doivent passer les utilisateurs.

## Brièveté (cont.)

### justification

Les **capacités de la mémoire à court terme** sont **limitées**. Par conséquent, **plus courtes** sont les **entrées**, **plus limités** sont les **risques d'erreurs**. Par ailleurs, **plus succincts** sont les **items**, **plus court** est le **temps de lecture**.

Aussi, plus les actions nécessaires à l'atteinte d'un but sont nombreuses et compliquées, plus la **charge de travail** **augmente** et par conséquent plus les **risques d'erreurs** sont **élevés**.



# Brièveté (cont.)

## (sous) sous-critères

- Concision
- Actions minimales

# Brièveté — Concision

## définition

Le critère **concision** concerne la charge de travail au niveau perceptif et mnésique pour ce qui est des **éléments individuels d'entrée ou de sortie**.

Par convention, la **concision** ne concerne pas le *feedback* ni les messages d'erreurs.

## Brièveté — Concision (cont.)

### exemples de recommandations

- Pour les données numériques, la saisie des zéros précédant les nombres ne devrait pas être nécessaire.
- Si des codes sont supérieurs à 4 ou 5 caractères, utiliser des mnémotechniques ou abréviations.
- Permettre aux utilisateurs des entrées de données courtes.
- Lorsqu'une unité de mesure est associée à un champ de donnée, celle-ci doit faire partie du label du champ plutôt qu'être saisie par les utilisateurs.

# Brièveté — Actions minimales

## définition

Le critère **actions minimales** concerne la charge de travail quant aux actions nécessaires à l'atteinte d'un but, à l'accomplissement d'une tâche. Il s'agit ici de **limiter** autant que possible **les étapes** par lesquelles doivent passer les utilisateurs.

## Brièveté — Actions minimales (cont.)

### exemples de recommandations

- Minimiser le nombre d'étapes dans la sélection de menus.
- Ne pas demander aux utilisateurs d'entrer des données qui peuvent être déduites par l'ordinateur.
- Éviter les ponctuations pour les entrées de commandes.
- Pour la saisie de données, afficher dans les champs appropriés, les valeurs par défaut.
- Pour des documents contenant plusieurs pages, il devrait être possible d'atteindre une page donnée sans avoir à parcourir les pages intermédiaires une à une.

# Densité informationnelle

## définition

Le critère **densité informationnelle** concerne la **charge de travail** du point de vue perceptif et mnésique, **pour des ensembles d'éléments** et non pour des items.

## Densité informationnelle (cont.)

### justification

Dans la plupart des tâches, la **performance** des utilisateurs est **influencée négativement** quand la charge informationnelle est trop élevée ou trop faible. La probabilité d'erreur augmente.

Il faut donc supprimer les éléments sans lien avec le contenu de la tâche en cours. Il faut aussi **éviter** d'imposer à l'utilisateur la **mémorisation de longues et nombreuses informations** ou procédures (la mémoire à court terme est limitée), ou toute activité nécessitant de sa part la mise en œuvre d'**activités cognitives complexes** lorsque la tâche ne le requiert pas.

# Densité informationnelle (cont.)

## exemples de recommandations

- Limiter la densité informationnelle de l'écran, en affichant seulement les informations nécessaires.
- L'information ne doit pas nécessiter des traductions d'unités.
- Utiliser le minimum de quantificateurs, notamment dans les langages de requêtes.



## Densité informationnelle (cont.)

- Éviter à l'utilisateur d'avoir à se rappeler des données d'une page écran à une autre.
- Les données qui peuvent être calculées à partir de celles saisies par l'utilisateur doivent l'être automatiquement.
- On ne doit pas exiger de l'opérateur d'effectuer des calculs qui peuvent être faits automatiquement.

# Contrôle explicite

## définition

Le critère **contrôle explicite** concerne à la fois la **prise en compte** par le système **des actions** explicites des utilisateurs et le **contrôle** qu'ont les utilisateurs **sur le traitement** de leurs actions.

## Contrôle explicite (cont.)

### justification

Quand les entrées des utilisateurs sont explicitement définies par eux-mêmes et sous leur contrôle, les ambiguïtés et les erreurs sont limitées. De plus, le **contrôle** qu'ont les utilisateurs **sur le dialogue** est un **facteur d'acceptation du système**.

# Contrôle explicite (cont.)

## sous-critères

- Actions explicites
- Contrôle utilisateur

# Actions explicites

## définition

Le critère **actions explicites** concerne la **relation** pouvant exister **entre le fonctionnement de l'application et les actions des utilisateurs**.

Cette relation doit être **explicite**, c'est-à-dire que le système doit **exécuter seulement les opérations demandées par l'utilisateur** et pas d'autres et ce, **au moment où il les demande**.

## Actions explicites (cont.)

### justification

Quand les opérations du système résultent des actions des utilisateurs, on observe **moins d'erreurs** et la **compréhension** du fonctionnement de l'application est **facilitée**.

## Actions explicites (cont.)

### exemples de recommandations

- Le système doit requérir une action explicite de validation par l'utilisateur (e.g., Entrée, Validation, OK) suite à une entrée de données ; aucun traitement (e.g., sauvegarde d'un fichier) ne devrait être la conséquence d'une autre action (e.g., une demande d'impression).
- Lors d'une sélection d'options de menu par pointage, prévoir une action explicite de validation.
- L'entrée de commandes doit se terminer par une indication de fin (e.g., ENTER, OK) à laquelle des possibilités d'édition doivent être préalables.

# Contrôle utilisateur

## définition

Par **contrôle utilisateur** on entend ici le fait que **l'utilisateur doit toujours avoir la main**, pouvoir contrôler le déroulement (e.g., interrompre, reprendre) des traitements informatiques en cours.

Ses actions devraient être anticipées et des options appropriées fournies pour chaque cas.



# Contrôle utilisateur (cont.)

## justification

Quand l'utilisateur a le contrôle du dialogue, les **réactions** de ce dernier sont **prévisibles**. L'**apprentissage** s'en trouve **facilité** et le **risque d'erreurs diminué**.

# Contrôle utilisateur (cont.)

## exemples de recommandations

- Autoriser les utilisateurs à contrôler le rythme de leurs entrées plutôt que de faire dépendre ce rythme des traitements du système ou d'événements extérieurs.
- Le curseur ne doit pas être déplacé sans contrôle des utilisateurs (sauf s'il s'agit de procédures stables et bien connues, e.g., remplissage de formulaires).
- Ne pas passer d'une page écran à une autre sans contrôle de l'utilisateur.

## Contrôle utilisateur (cont.)

- Autoriser l'utilisateur à interrompre à tout moment une action ou un traitement en cours (e.g., annuler).
- Fournir la possibilité de retour arrière conduisant à l'annulation des modifications en cours et au retour à la version immédiatement précédente.

# Adaptabilité

## définition

L'**adaptabilité** d'un système concerne sa **capacité à réagir selon le contexte**, et **selon les besoins et préférences des utilisateurs**.

## Adaptabilité (cont.)

### justification

Plus les façons d'effectuer une même tâche sont diverses, plus les chances que l'utilisateur puisse choisir et maîtriser l'une d'entre elles, au cours de ses apprentissages, sont importantes.

Il faut donc **fournir** à l'utilisateur **des procédures, options, et commandes différentes** leur permettant d'atteindre **un même objectif**. Par ailleurs, une interface ne peut convenir à la fois à tous ses utilisateurs potentiels.

Pour qu'elle n'ait pas d'effets négatifs sur l'utilisateur, **cette interface doit**, selon les contextes, **s'adapter à l'utilisateur**.

# Adaptabilité (cont.)

## sous-critères

- Flexibilité
- Prise en compte de l'expérience de l'utilisateur

# Flexibilité

## définition

Le critère **flexibilité** concerne les moyens mis à la disposition des utilisateurs pour **personnaliser l'interface** afin de rendre compte de leurs stratégies ou habitudes de travail et des exigences de la tâche.

Le critère **flexibilité** correspond aussi au nombre de **façons différentes** mises à la disposition des utilisateurs pour atteindre un objectif donné.

Il s'agit en d'autres termes de la **capacité de l'interface à s'adapter** à des actions variées des utilisateurs.

# Flexibilité (cont.)

## justification

Plus les façons d'effectuer une même tâche sont diverses, plus les chances que l'utilisateur puisse choisir et maîtriser l'une d'entre elles, au cours de ses apprentissages, sont importantes.



## Flexibilité (cont.)

### exemples de recommandations

- Quand les exigences utilisateurs sont imprécises, fournir aux utilisateurs une certaine latitude dans les possibilités de contrôle des affichages.
- Pour l'entrée de données, lorsque des valeurs par défaut ne sont pas connues à l'avance, le système doit permettre aux utilisateurs de définir, changer ou supprimer ces valeurs.
- Quand certains affichages sont inutiles, les utilisateurs doivent pouvoir les désactiver temporairement.

## Flexibilité (cont.)

- La séquence des entrées de données doit pouvoir être modifiée pour s'adapter à l'ordre souhaité par les utilisateurs.
- Lorsqu'on ne peut spécifier à l'avance le format d'un document, on doit permettre aux utilisateurs d'en définir un et de le sauvegarder pour une utilisation ultérieure.
- Les utilisateurs devraient pouvoir assigner eux-mêmes le nom des champs de données qu'ils créent.

# Prise en compte de l'expérience

## définition

Le critère **prise en compte de l'expérience de l'utilisateur** concerne les moyens mis en oeuvre pour **respecter le niveau d'expérience** de l'utilisateur.

## Prise en compte de l'expérience (cont.)

### justification

Des utilisateurs expérimentés n'ont pas toujours les mêmes besoins informationnels que les novices. Il peut être souhaitable de fournir aux utilisateurs inexpérimentés des transactions très guidées, au pas-à-pas.

Pour des **utilisateurs expérimentés**, des dialogues à la seule initiative de l'ordinateur peuvent les ennuyer et ralentir leurs interactions ; par contre, des **raccourcis** peuvent leur permettre d'**accéder plus rapidement aux fonctions** du système. Des moyens différenciés doivent donc être prévus pour tenir compte de ces différences d'expérience.

## Prise en compte de l'expérience (cont.)

Cependant, **l'expérience des utilisateurs peut varier**. Les utilisateurs peuvent devenir plus experts à force d'utilisation, ou moins experts après de longues périodes de non-utilisation. L'interface doit aussi être conçue afin de tenir compte de ces variations du niveau d'expérience.

# Prise en compte de l'expérience (cont.)

## exemples de recommandations

- Autoriser les utilisateurs expérimentés à contourner une série de sélections par menu en formulant directement des commandes ou par des raccourcis clavier.
- Prévoir des choix d'entrées pas-à-pas ou multiples selon l'expérience des utilisateurs.
- Autoriser différents modes de dialogue correspondant aux différents groupes d'utilisateurs (e.g., permettre une incitation adaptée au niveau d'expérience des utilisateurs).

## Prise en compte de l'expérience (cont.)

- Permettre la saisie de plusieurs commandes avant confirmation pour les expérimentés.
- Lorsque les techniques de guidage ralentissent les utilisateurs expérimentés, fournir des moyens à ces utilisateurs de contourner ce guidage.
- Les utilisateurs devraient pouvoir demander un niveau de détail des messages d'erreurs qui soit fonction de leur niveau de connaissance.

# Gestion des erreurs

## définition

Le critère **gestion des erreurs** concerne tous les moyens permettant d'une part d'**éviter ou de réduire les erreurs**, et d'autre part de les **corriger lorsqu'elles surviennent**.

Les erreurs sont ici considérées comme des saisies de données incorrectes, des saisies dans des formats inadéquats, des saisies de commandes avec une syntaxe incorrecte, etc.



## Gestion des erreurs (cont.)

### justification

Les interruptions provoquées par **les erreurs** ont des conséquences négatives sur l'activité des utilisateurs. De manière générale, elles **rallongent** les transactions et **perturbent** la planification.

Plus les erreurs sont limitées, moins il y a d'interruptions au cours de la réalisation d'une tâche et meilleure est la performance.

# Gestion des erreurs (cont.)

## sous-critères

- Protection contre les erreurs
- Qualité des messages d'erreur
- Correction des erreurs

# Protection contre les erreurs

## définition

Le critère **protection contre les erreurs** concerne les moyens mis en place pour **détecter et prévenir les erreurs** d'entrées de données ou de commandes ou les actions aux conséquences néfastes.

# Protection contre les erreurs (cont.)

## justification

Il est préférable de détecter les erreurs lors de la saisie plutôt que lors de la validation : ceci évite de perturber la planification.

# Protection contre les erreurs (cont.)

## exemples de recommandations

- Quand les utilisateurs terminent une session et qu'il y a un risque de perte de données, il doit y avoir un message le signalant et demandant confirmation de fin de session.
- Les labels de champs doivent être protégés.

## Protection contre les erreurs (cont.)

- Les aires d'affichage doivent être protégés : les utilisateurs ne doivent pas pouvoir changer les informations contenues dans ces champs.
- Toutes les actions possibles sur une interface doivent être envisagées et plus particulièrement les appuis accidentels des touches du clavier afin que les entrées non attendues soient détectées.

# Qualité des messages d'erreur

## définition

Le critère **qualité des messages d'erreur** concerne la **pertinence, la facilité de lecture et l'exactitude de l'information** donnée aux utilisateurs **sur la nature des erreurs** commises (syntaxe, format, etc.) et sur les actions à entreprendre pour les corriger.

## Qualité des messages d'erreur (cont.)

### justification

La qualité des messages **favorise l'apprentissage** du système en indiquant aux utilisateurs les raisons ou la nature de leurs erreurs et en leur indiquant ce qu'il faut ou ce qu'ils auraient dû faire.



## Qualité des messages d'erreur (cont.)

### exemples de recommandations

- Si l'utilisateur sélectionne une touche fonction non valide, aucune action ne doit en résulter, si ce n'est un message indiquant les fonctions appropriées à cette étape de la transaction.
- Fournir des messages d'erreurs orientés tâches.
- Utiliser des termes aussi spécifiques que possibles pour les messages d'erreur.
- Utiliser des messages d'erreur aussi brefs que possible.
- Adopter un vocabulaire neutre, non personnalisé, non réprobateur dans les messages d'erreur ; éviter l'humour.

# Correction des erreurs

## définition

Le critère **correction des erreurs** concerne les moyens mis à la disposition des utilisateurs pour leur permettre de **corriger leurs erreurs**.

# Correction des erreurs (cont.)

## justification

Les erreurs sont d'autant **moins perturbatrices** qu'elles sont **faciles à corriger**.

# Correction des erreurs (cont.)

## exemples de recommandations

- Fournir la possibilité de modifier les commandes lors de leur saisie.
- Suite à une erreur de saisie d'une commande ou de données, donner aux utilisateurs la possibilité de corriger seulement la portion de données ou de commande qui est erronée.
- Si les utilisateurs se rendent compte qu'ils ont commis une erreur d'entrée de données, leur donner la possibilité d'effectuer, au moment de leur détection d'erreur, les corrections souhaitées.

# Homogénéité/cohérence

## définition

Le critère **homogénéité/cohérence** se réfère à la façon avec laquelle les **choix de conception** de l'interface (codes, dénominations, formats, procédures, etc.) sont **conservés pour des contextes identiques**, et sont **différents pour des contextes différents**.

## Homogénéité/cohérence (cont.)

### justification

Les procédures, labels, commandes, etc., sont d'autant **mieux reconnus, localisés et utilisés**, que leur **format, localisation, ou syntaxe sont stables** d'un écran à l'autre, d'une session à l'autre.

Dans ces conditions le **système est davantage prévisible** et les **apprentissages plus généralisables** ; les erreurs sont réduites.

Le manque d'homogénéité peut augmenter considérablement le temps de recherche. Le manque d'homogénéité est aussi une raison importante du refus d'utilisation.

# Homogénéité/cohérence (cont.)

## exemples de recommandations

- Localisation similaire des titres des fenêtres.
- Formats d'écrans similaires.
- Procédures similaires d'accès aux options de menus.
- Lors du guidage, toujours utiliser les mêmes ponctuations et les mêmes constructions de phrases.
- Afficher à la même position les "prompts" pour la saisie des données ou des commandes.
- Le format des champs d'entrée de données doit toujours être le même.

# Signifiante des codes et dénominations

## définition

Le critère **signifiante des codes et dénominations** concerne l'**adéquation entre l'objet ou l'information affichée ou entrée, et son référent**. Des codes et dénominations "signifiants" disposent d'une **relation sémantique forte** avec leur référent.



## Signifiante des codes et dénominations (cont.)

### justification

Lorsque le codage est significatif, **le rappel et la reconnaissance sont meilleurs.**

De plus, **des codes et dénominations non significatifs** pour les utilisateurs **peuvent** leur suggérer des opérations inappropriées et ainsi **conduire à des erreurs.**

# Signifiante des codes et dénominations (cont.)

## exemples de recommandations

- Les titres doivent véhiculer ce qu'ils représentent, et être distincts.
- Rendre les règles d'abréviation explicites.
- Utiliser des codes et dénominations significatifs et familiers plutôt que des codes et dénominations arbitraires (e.g., M pour masculin et F pour féminin plutôt que 1 et 2).

# Compatibilité

## définition

Le critère **compatibilité** se réfère à l'**accord** pouvant exister **entre les caractéristiques des utilisateurs** (e.g., mémoire, perceptions, habitudes, compétences, âge, attentes) **et des tâches**, d'une part, **et l'organisation des sorties, des entrées et du dialogue** d'une application donnée, d'autre part. De plus, la **compatibilité** concerne également le **degré de similitude entre divers environnements ou applications**.

## Compatibilité (cont.)

### justification

Le **transfert d'information** d'un contexte à un autre est d'autant plus **rapide et efficace** que le **volume d'information** à recoder par l'utilisateur est **réduit**.

L'efficacité est accrue lorsque :

- les **procédures** nécessaires à l'accomplissement de la tâche sont **compatibles avec les caractéristiques psychologiques des utilisateurs** ;
- les procédures et les tâches sont organisées de manière à respecter les attentes, ou habitudes des utilisateurs ;
- les traductions, les transpositions, les interprétations, ou références à la documentation sont minimisées.

# Compatibilité (cont.)

## exemples de recommandations

- L'organisation des informations affichées doit être conforme à l'organisation des données à entrer.
- Les procédures de dialogue doivent être compatibles avec l'ordre tel que se l'imagine l'utilisateur ou celui dont il a l'habitude.
- Le format de date en français est "jour/mois/année". En anglais, ce format devient "mois/jour/année".

## Compatibilité (cont.)

- Les termes employés doivent être familiers aux utilisateurs, et relatifs à la tâche à réaliser.
- Les unités de mesure doivent être celles qui sont normalement utilisées.
- L'affichage de texte à l'écran doit être conforme aux conventions utilisées pour la présentation des textes sur papier.

# À lire



C. Bastien et D. Scapin.

Critères Ergonomiques pour l'Évaluation d'Interfaces  
Utilisateurs.

INRIA RT n°156, 1993.