

# **SI3 – Projet S6 2017-2018**

## **ergonomie logiciel**

Université Nice Sophia Antipolis (Polytech)

**23 Janvier 2018**

**Marco Winckler**, Sébastien Mosser, Jean Yves Tigli,  
Stéphane Lavirotte, Claudine Peyrat, Anne-Marie  
Pinna-Dery, Clément Duffau, Fabian Pinel, Jean-Yves  
Delmotte et Rémi Pourtier

Université Nice Sophia (Polytech) | I3S | SPARKS team | bureau 446

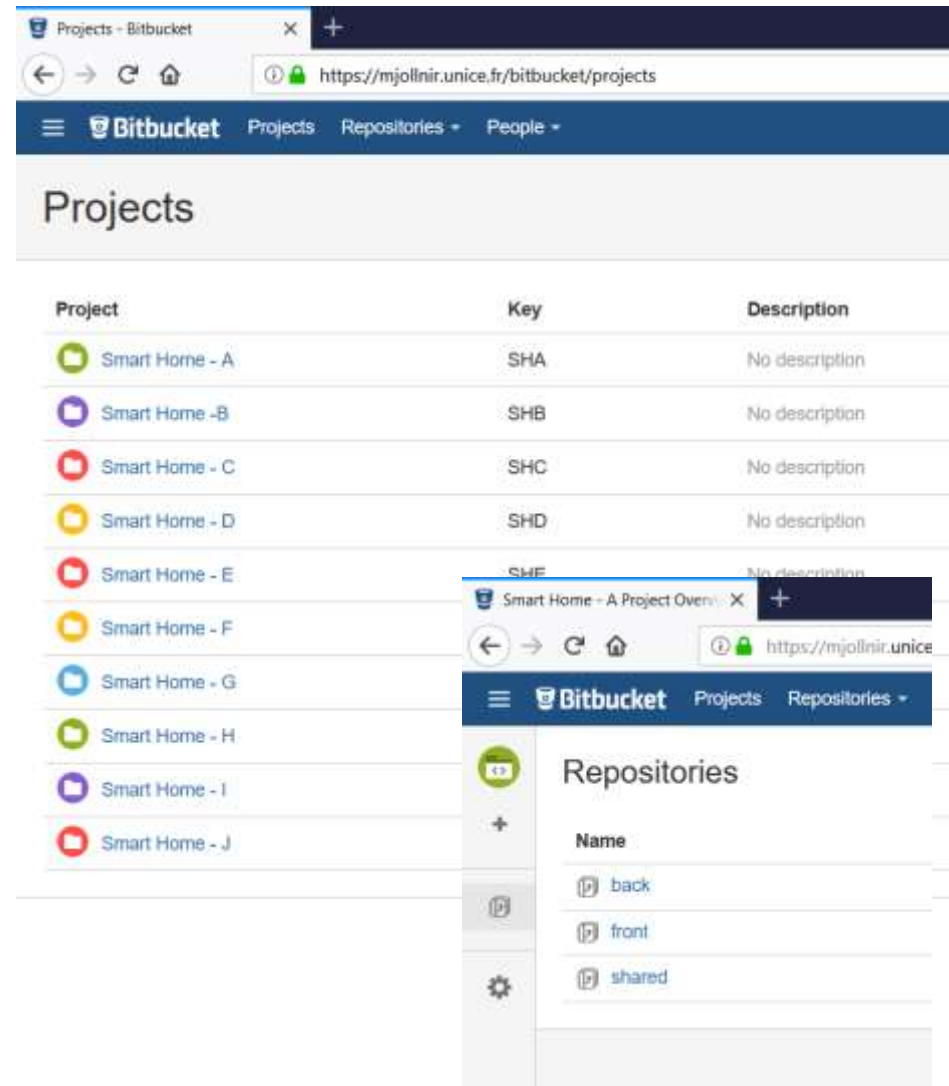
[winckler@i3s.unice.fr](mailto:winckler@i3s.unice.fr)

<http://www.i3s.unice.fr/~winckler/>



# Consignes pour le projet

- Groupes créés sur <https://mjollnir.unice.fr/bitbucket/projects>
- Attention au noms de groups:
  - Smart Home – A... J
- Repositories
  - Back → visualization
  - Front → declaration
  - Shared
- Il faut **créer un dossier livrables**
- A vous de trouver l'outil de gestion de la communication qui va bien pour le groupe
- Rappel: **déposer les livrables la veille des séances**

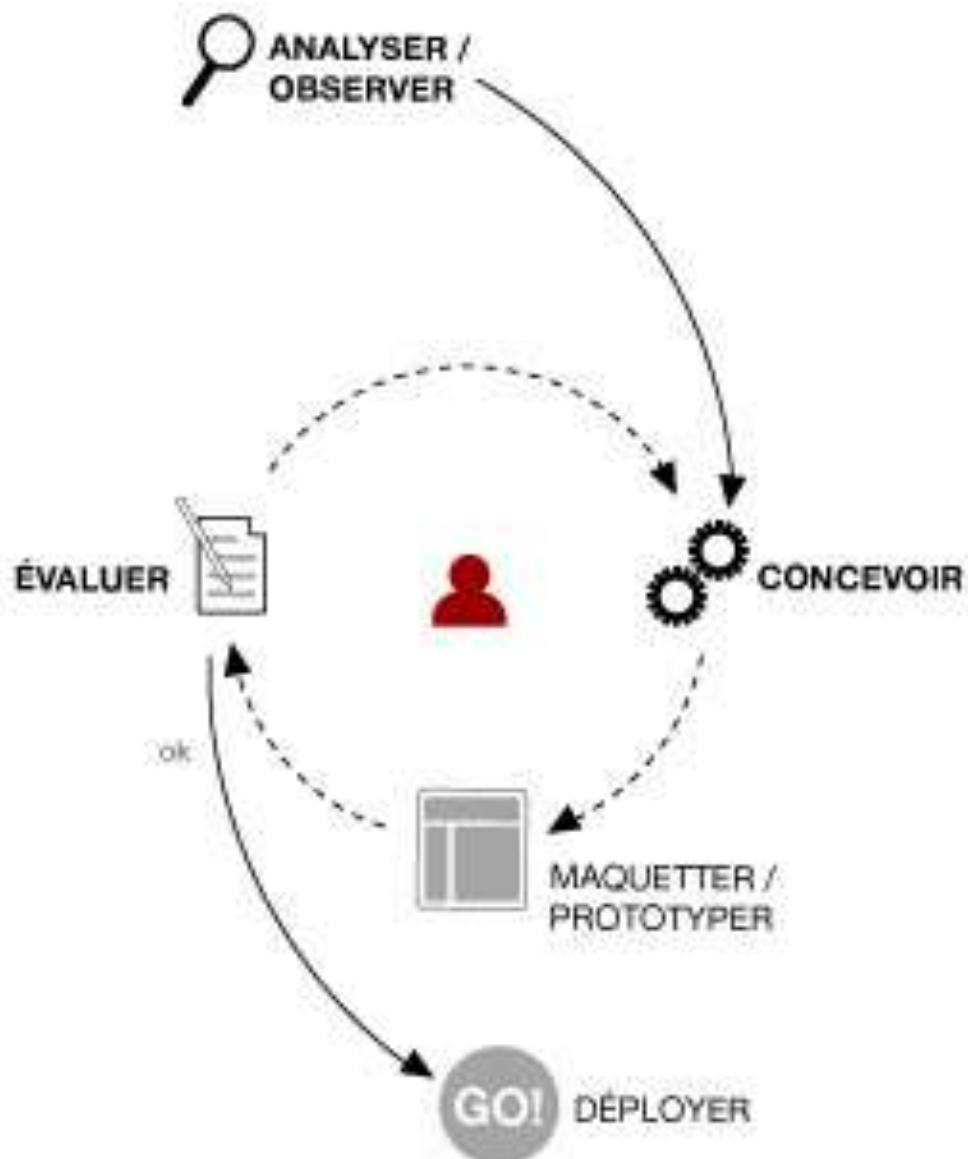


# Agenda

- Conception centrée utilisateur
- Méthode d'évaluation
  - Méthode par inspection
- Inspection des scénarios sur les prototypes

# **Processus de conception centrée utilisateurs**

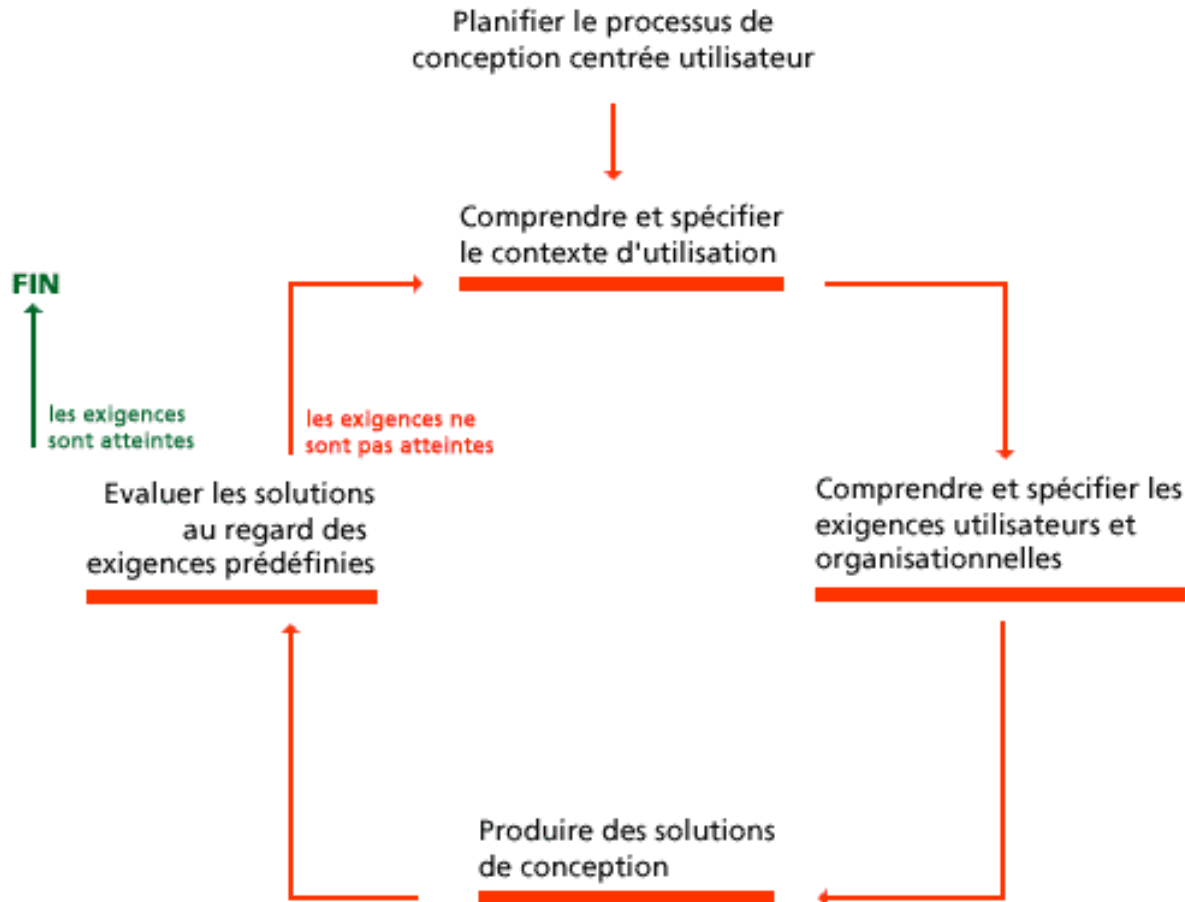
## DESIGN CENTRÉ UTILISATEUR



# La conception centrée utilisateur

- La norme ISO 13407 définit les conditions :
  - Une préoccupation amont des utilisateurs, de leurs tâches et de leur environnement
  - La participation active de ces utilisateurs, ainsi que la compréhension claire de leurs besoins et des exigences liées à leurs tâche
  - Une répartition appropriée des fonctions entre les utilisateurs et la technologie
  - L'itération des solutions de conception : on peut s'imaginer le cycle comme une spirale, une démarche qui boucle et reboucle jusqu'à ce que le système satisfasse aux exigences définies au départ
  - L'intervention d'une équipe de conception multi-disciplinaire

# Etapes du processus de conception centrée utilisateur



# Planifier le processus de conception centrée sur l'utilisateur

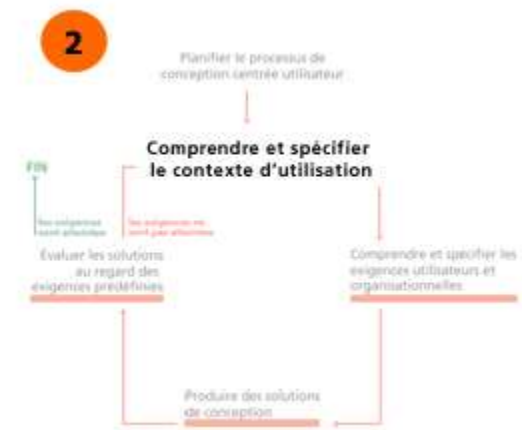
- Cette pré-étape consiste à planifier les activités de développement dans une optique de conception centrée utilisateur
- Les avantages doivent être connus de l'équipe, notamment :
  - le retour sur investissement
  - la satisfaction des utilisateurs
  - l'utilisabilité du système et
  - l'adaptation aux caractéristiques des opérateurs
- Les exigences doivent aussi être clairement explicitées
- Consulter les documents sur l'ergonome et charte graphique
- Dans le cas de la conception d'une application métier, on essaie aussi de constituer un groupe de travail (composé notamment des opérateurs, futurs utilisateurs du système).





# Comprendre et spécifier le contexte d'utilisation

- comprendre la population cible et ses caractéristiques, ses buts et tâches, ses environnements
- décrire les environnements technique, physique, ambiants, social, organisationnel et législatif
- identification des profils utilisateur
- Méthodes à utiliser:
  - Card sorting (tri de cartes)
  - Entretiens
  - Focus Groups
  - Scenarii
  - Personnas



# Spécifier les exigences liées à l'utilisateur et à l'organisation

- Objectifs d'utilisabilité :
  - "Qu'est-ce que je vais mesurer et qu'est-ce que j'attends en termes de performance homme-machine?"
  - Critères:
    - Taux de succès
    - Nombre d'erreurs
    - Temps d'exécution des tâches
    - Nombre d'étapes nécessaires à la complétion des tâches
    - Eventuels recours à une aide interne ou externe au produit
    - Rythme d'apprentissage
    - Satisfaction des utilisateurs...
- Objectifs opérationnels et financiers
- Exigences organisationnelles



# Produire des solutions de conception

- Production de solutions de conception vise à utiliser les connaissances acquises lors des étapes précédentes pour matérialiser les solutions afin de pouvoir les modifier en fonction des feedback utilisateurs
- Méthodes à utiliser:
  - Prototypage



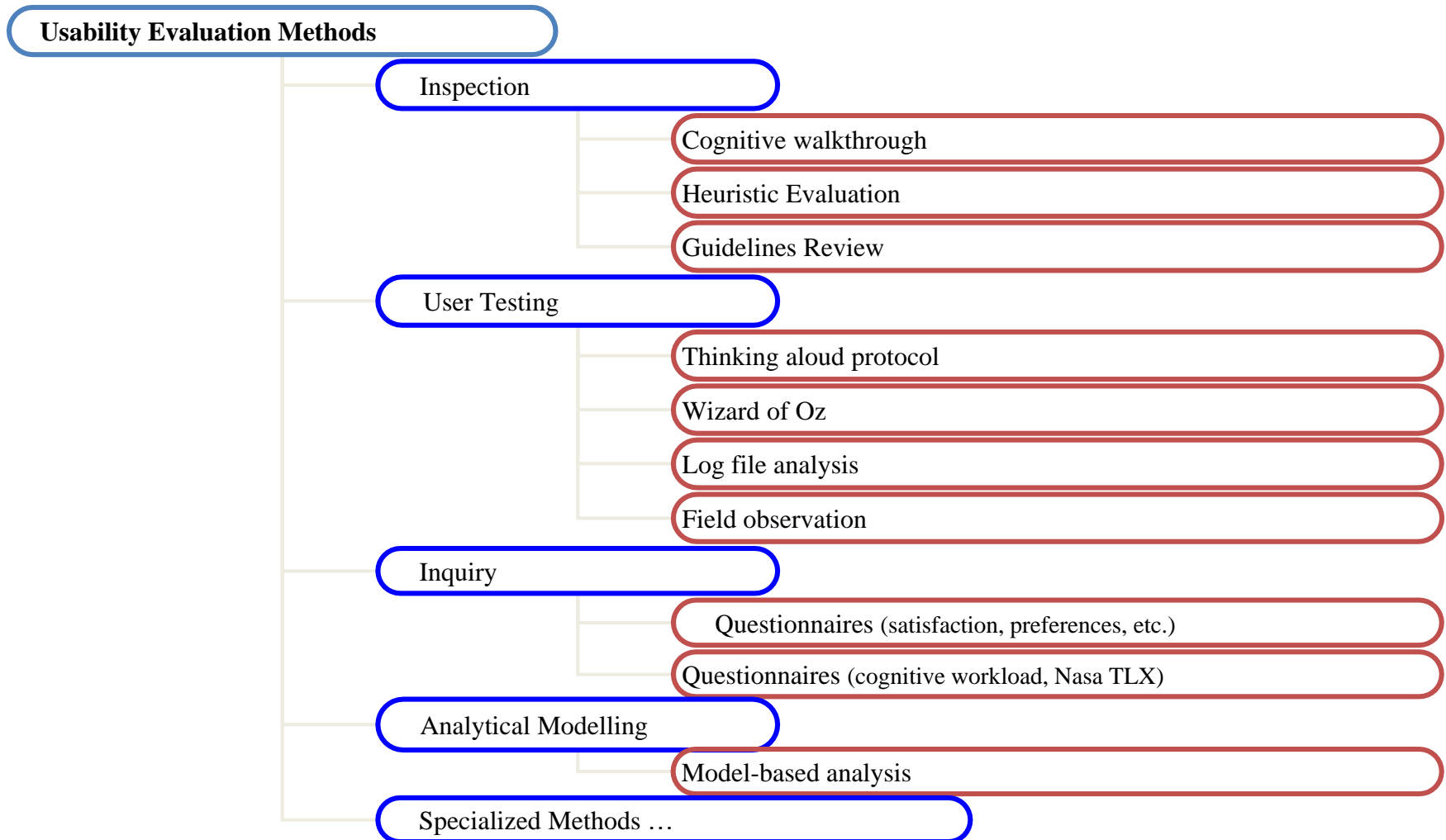
# Evaluer les solutions conçues au regard des exigences

- Réaliser de testes utilisateurs
- Décrire les problèmes d'utilisabilité rencontrés
- Ordonner les défauts de conception selon leur importance
- Recueillir un feedback sur la conception développée



# **Méthodes d'évaluation d'utilisabilités**

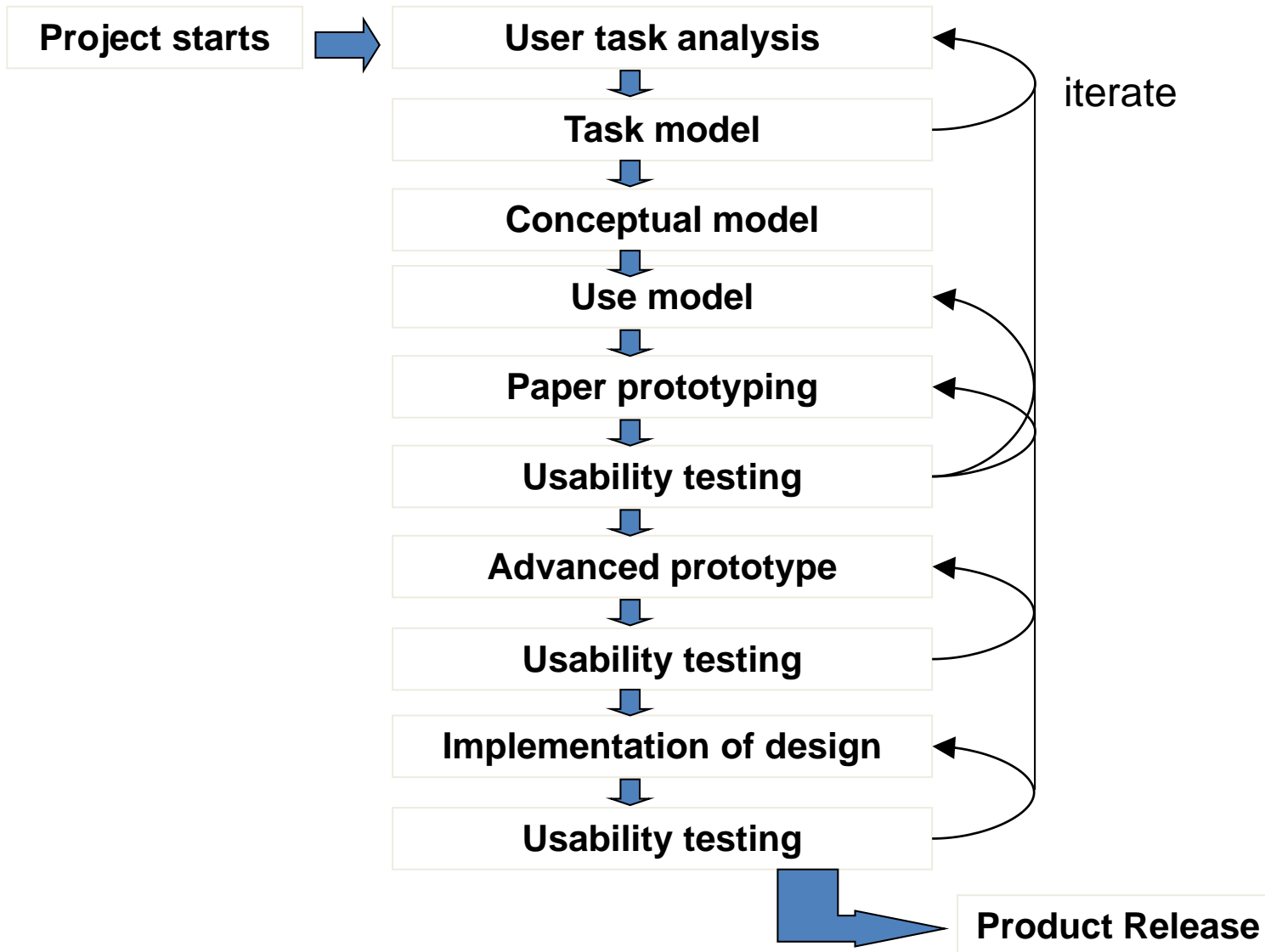
# Classification of Usability Evaluation Methods



# Où faire l'évaluation d'utilisabilité?

	<b>Lab</b>	<b>Field</b>
<b>Design</b>	Analytical, review or model based methods; usability inspection methods (performed by experts)	Evaluation using prototypes (Card sorting, usability studies using paper prototypes)
<b>Implementation</b>	Experimental methods, observation, interviews (experts or users)	Experimental Methods, observation, interviews (users)

# Quand faire l'évaluation d'utilisabilité?





# Phase de design (early phases)

- Méthodes par inspection
  - Basés sur l'option des experts IHM
  - Ex.:
    - Cognitive Walkthrough, Pluralistic Walkthrough
    - Heuristic Evaluation
    - Guideline Review
- Méthodes basés sur modèles

## Phase de development (late phases)

- Evaluations expérimentales:
  - Test d'utilisabilité en laboratoire des usages
- Observation des utilisateurs avec:
  - Thinking aloud
  - Protocol exp
- Enquêtes:
  - Entretiens
  - Questionnaires spécialisés

# **Exemple de problème d'utilisabilité**

# Critères pour déterminer les problèmes d'utilisabilité

- Pas de correspondance avec les besoins des utilisateurs
- Liens avec les tâches et les scénarios:
  - Degrée de réalisation de tâches (succès, succès partiel, échec)
  - Temps pour réaliser une tâche
  - Parution d'erreur pendant la réalisation d'une tâche
- Satisfaction, stress, etc.
- Respect de règles ergonomiques





# Lien avec tâche et scénario

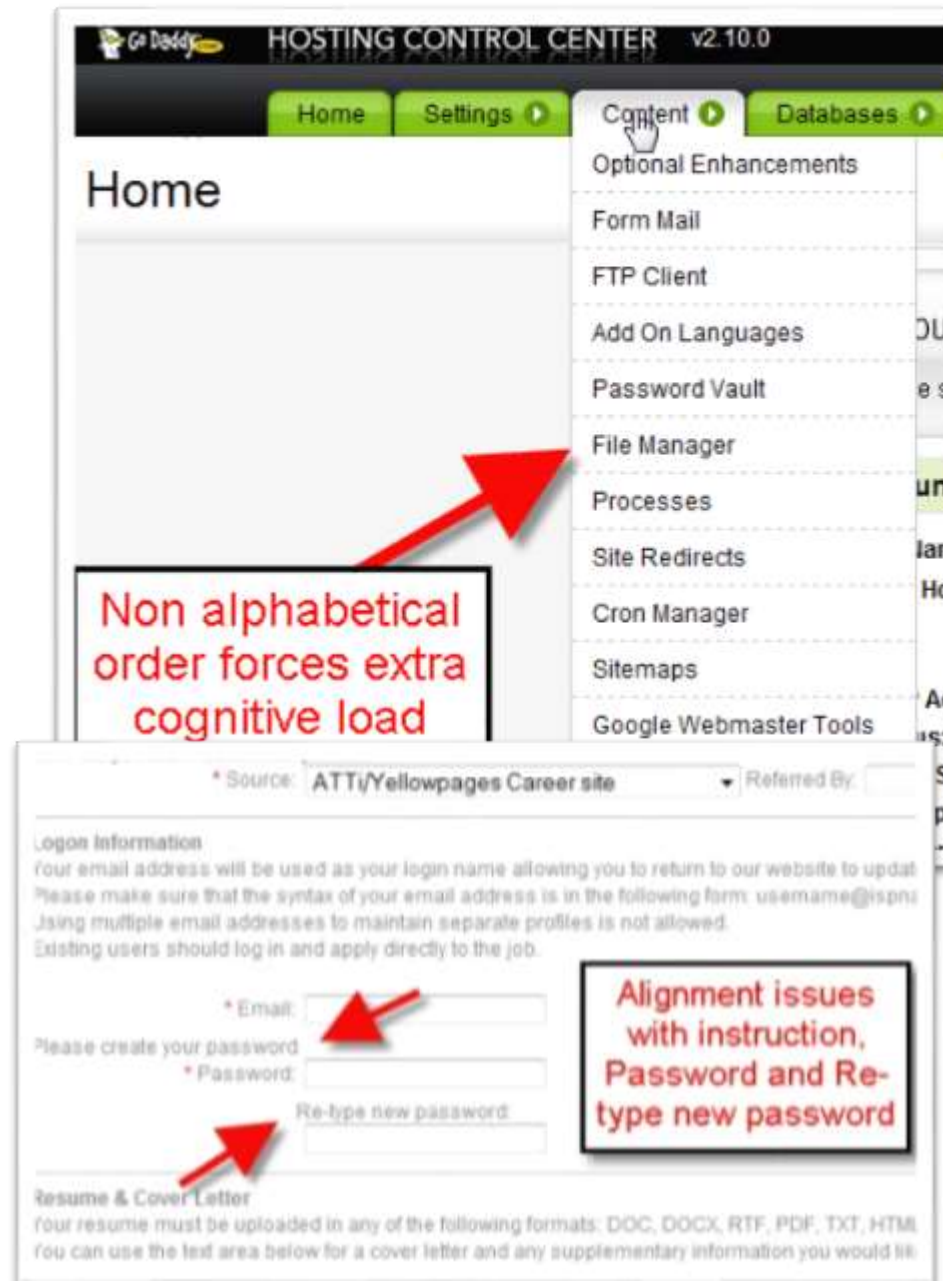


Problème en image...

La solution...



## Non-respect de règles ergonomiques

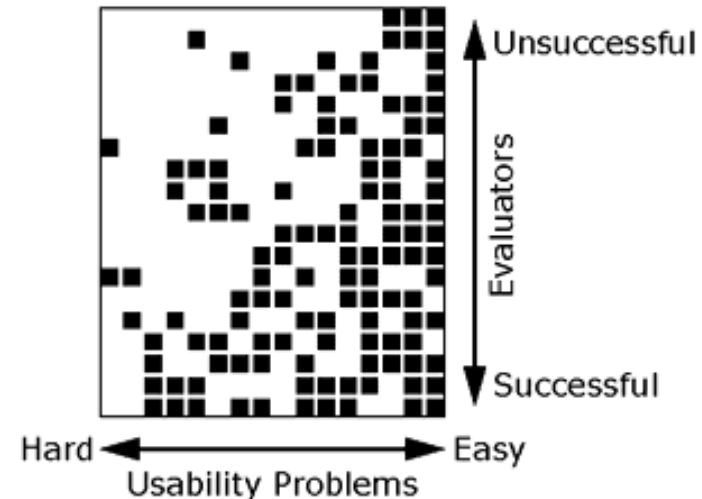


# **Heuristic Evaluation (méthode d'inspection)**



# « Evaluation Heuristique »

- Méthode proposé par Nielsen et Molich (1993)
- Le principe: plusieurs évaluateur vérifient l'interface avec l'aide de seulement 10 règles heuristiques
- Les heuristiques aident l'évaluateurs a se rappeler des aspects importants a évaluer

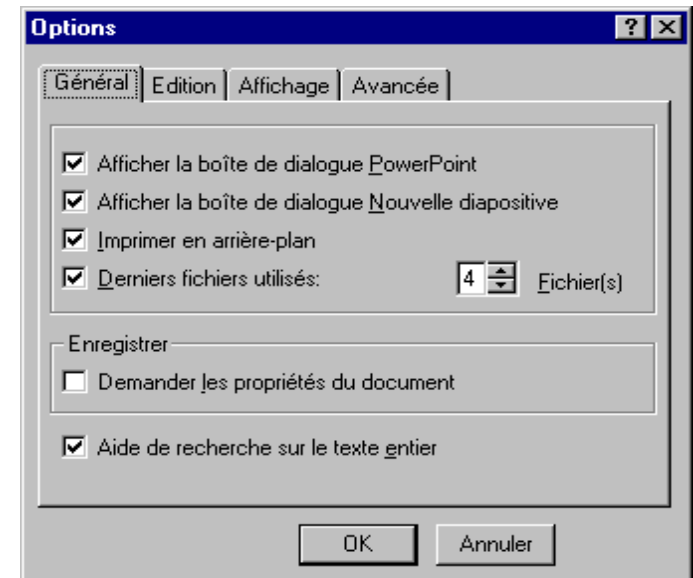


# Les 10 heuristiques...

1. Utiliser de dialogues simples et naturels
2. Parler le langage de l'utilisateur
3. Minimiser la charge cognitive
4. Cohérence
5. Donner du feedback
6. Fournir de sorties explicites
7. Fournir des raccourcis
8. Fournir de messages d'erreurs
9. Concevoir pour les erreurs
10. Aide et documentation

# 1. Utiliser de dialogues simples et naturels

- Utiliser une logique d'utilisation et pas de fonctionnement
  - Concevoir en fonction de la tâche
  - Réduction du gouffre de l'exécution
- Présenter seulement les informations pertinentes
  - «Less is more»
  - Groupage logique des informations
  - Peu de fenêtres et de navigation



## 2. Parler le langage utilisateur

- Terminologie inspirée du langage de la tâche, et pas du système
  - «Ce produit n'est pas référencé dans le stock» ***plutôt que***
  - «Foreign key not found in table T\_STOCK»
- Présenter l'interaction du point de vue de l'utilisateur
  - «Vous venez d'acheter 100 actions de la société XYZ» ***plutôt que***
  - «Nous venons de vous vendre 100 actions de la société XYZ»
- Mnémoniques, abréviations, icônes
  - S'assurer qu'elles sont significatives pour l'utilisateur

# 3. Minimiser la charge cognitive

- Pallier la faible capacité de la mémoire à court terme
- Privilégier la reconnaissance par rapport à la mémoire
  - Menus, icônes, choix explicites
  - Attention : Less is more !
- Décrire les formats d'entrée
  - Date de naissance : \_\_\_\_/\_\_\_\_/\_\_\_\_ (JJ/MM/AA, ex : 12/12/70)
- Utiliser un petit nombre de commandes génériques
  - cut, copy, paste, drag'n drop
  - Utilisables pour un mot, un paragraphe, un document...

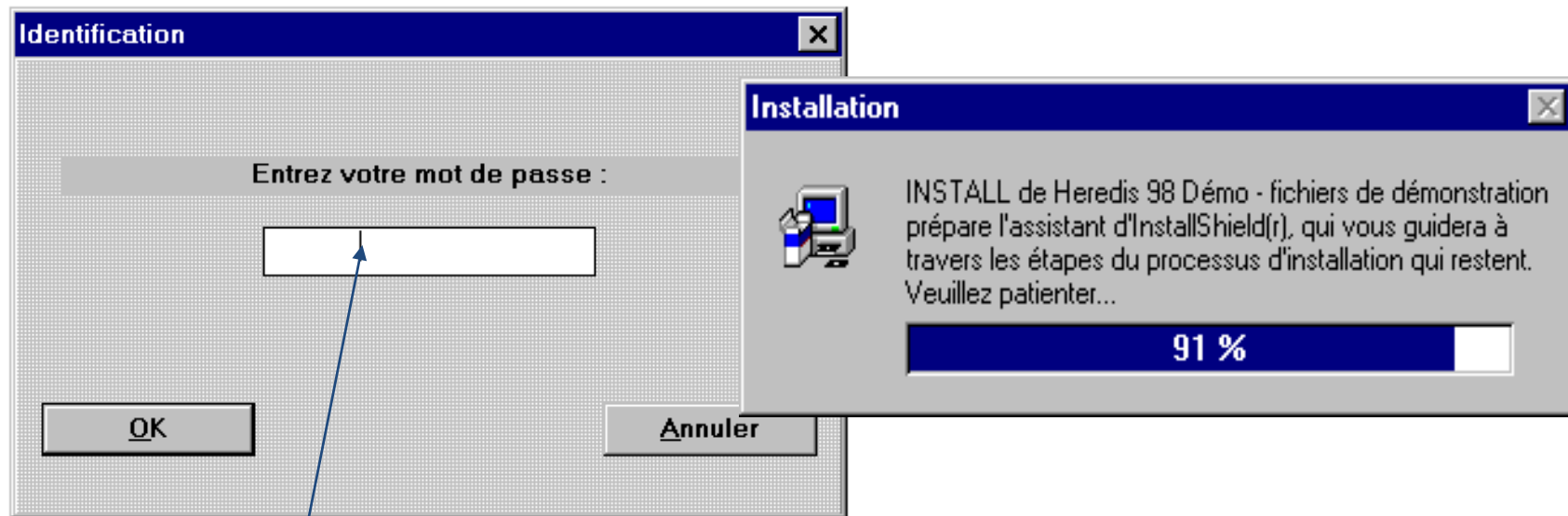
# 4. Cohérence

- Cohérence des effets
  - La même commande aura toujours les mêmes effets dans des situations comparables
  - Prédicibilité de l'interface
- Cohérence du langage et des graphiques
  - Dispositions spatiale des contrôles et texte des étiquettes
  - Apparence visuelle des contrôles (ex: scrollbars)
- Cohérence des entrées
  - Même syntaxe dans l'ensemble du système

# 5. Donner du feedback

- Minimiser le gouffre de l'évaluation
- Etre le plus spécifique possible
  - «Enregistrement du fichier 'Toto' dans le répertoire 'bidon' en cours...» plutôt que
  - «Enregistrement en cours...»
- Temps de réponse
  - 0,1 seconde max : perçu comme instantané
  - 1 seconde max : Le dialogue n'est pas interrompu, mais le délai est perçu
  - 10 secondes : limite pour conserver l'attention de l'utilisateur sur le dialogue
  - >10 secondes : L'utilisateur souhaite faire qq. chose d'autre

# Feedback immédiat



Non

Oui

Saisie non  
visible



# 6. Fournir de sorties explicites

- Stratégies
  - Bouton «Annuler» pour les dialogues modaux
  - «Undo» universel (retour à l'état précédent)
  - «Interrompre» pour les opérations longues
  - «Quitter» le programme n'importe quand
  - «Défauts» pour réinitialiser des propriétés

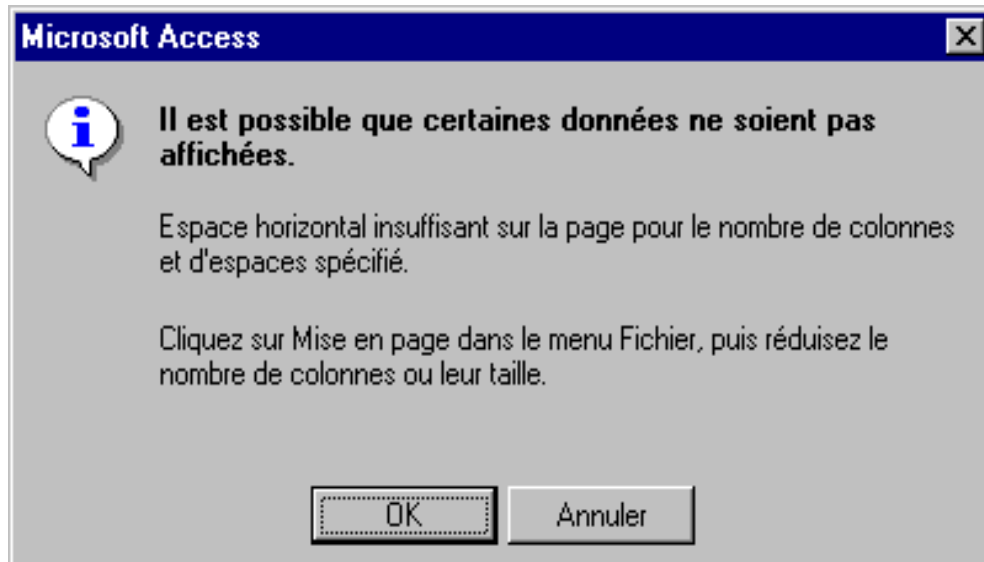
# 7. Fournir des raccourcis

- Adaptabilité aux utilisateurs experts
- Stratégies :
  - Accélérateurs clavier et souris
  - Complétion automatique des commandes
  - Touches de fonctions
  - Réutilisation
    - «History» du système Unix : 75% des commandes ont déjà été tapées auparavant
  - Barre d'outils personnalisable

## 8. Fournir de messages d'erreurs

- Les phrases doivent être concises, complètes et claires
  - « cannot open this document » X « cannot open this document because the application is not on the disk »
- La document de doit pas être une priorité pour comprendre l'erreur
- Proposer de solutions
  - « ... because the MS Word is not on the disk »
- Respecter l'utilisateur
  - « illegal user action, job abort »

# Qualité des messages d'erreurs



Oui



Non

Message  
peu détaillé

# 9. Concevoir pour les erreurs

- Deux types d'erreurs
  - Mistakes :
    - Une décisions consciente qui conduit à une action erronée
  - Slips :
    - Comportement inconscient qui conduit à une mauvaise séquence d'actions
    - Fréquent chez les utilisateurs expérimentés

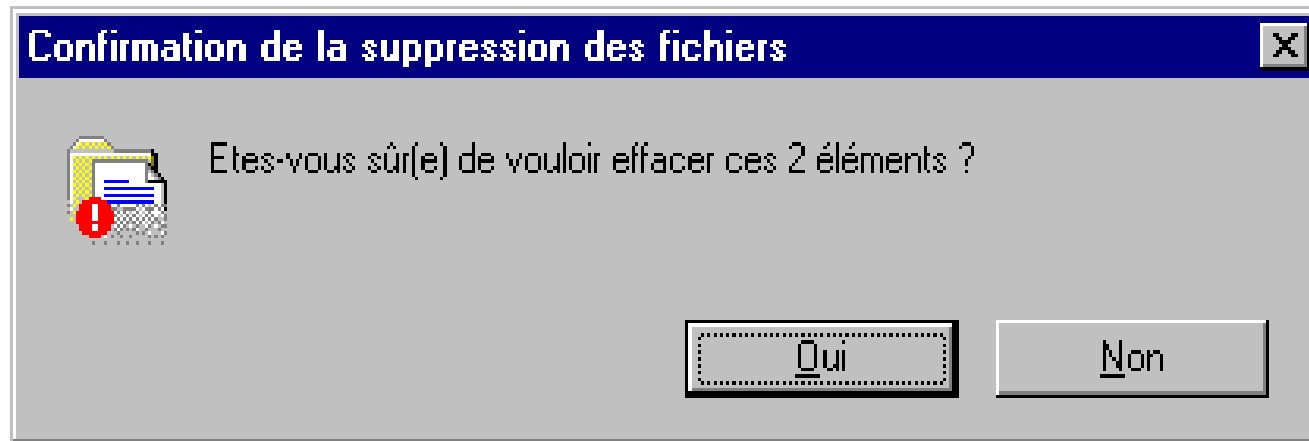
# Types de «Slips»

- Erreur de capture
  - L'activité la plus fréquente est déclenchée, au lieu de celle désirée
    - ex : confirmer la suppression d'un fichier, alors qu'on ne voulait pas le supprimer
    - Prévention : au lieu de confirmer, permettre le «Undo»
- Erreur de description
  - Quand deux actions possibles ont beaucoup en commun
    - ex : glisser un fichier dans la poubelle au lieu de la disquette
    - Prévention : différencier les icônes, disposition spatiale étudiée
- Erreur «Data driven»
  - ex: téléphoner à quelqu'un pour lui donner un numéro de chambre, et composer le numéro de chambre
- Activation associative
  - ex: le téléphone sonne, on crie «Entrez!»

# Types de «Slips» (suite)

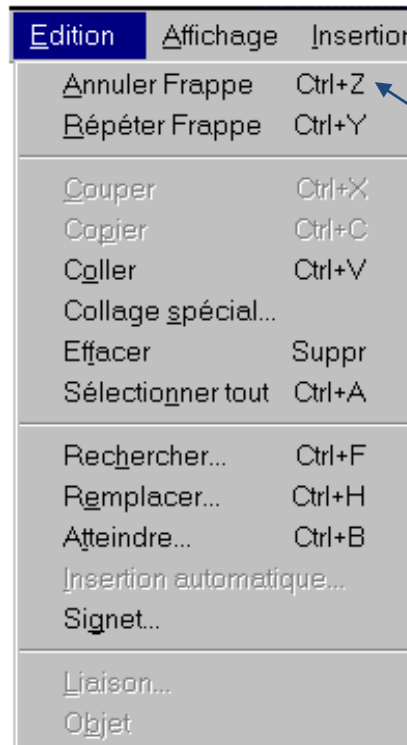
- Perte d'activation
  - Oublier le but final lorsqu'on exécute une séquence d'actions
    - ex: «Qu'est-ce que je fais ici ?»
    - Prévention : rendre le but explicite (si le système le connaît), rendre le cheminement explicite
- Erreurs de mode
  - Action effectuée dans un contexte inadéquat
    - ex: VI, référencer un fichier qui est dans un autre répertoire, ...
    - Prévention : minimiser les modes, rendre les modes visibles

# Protection contre les erreurs





# Correction des erreurs



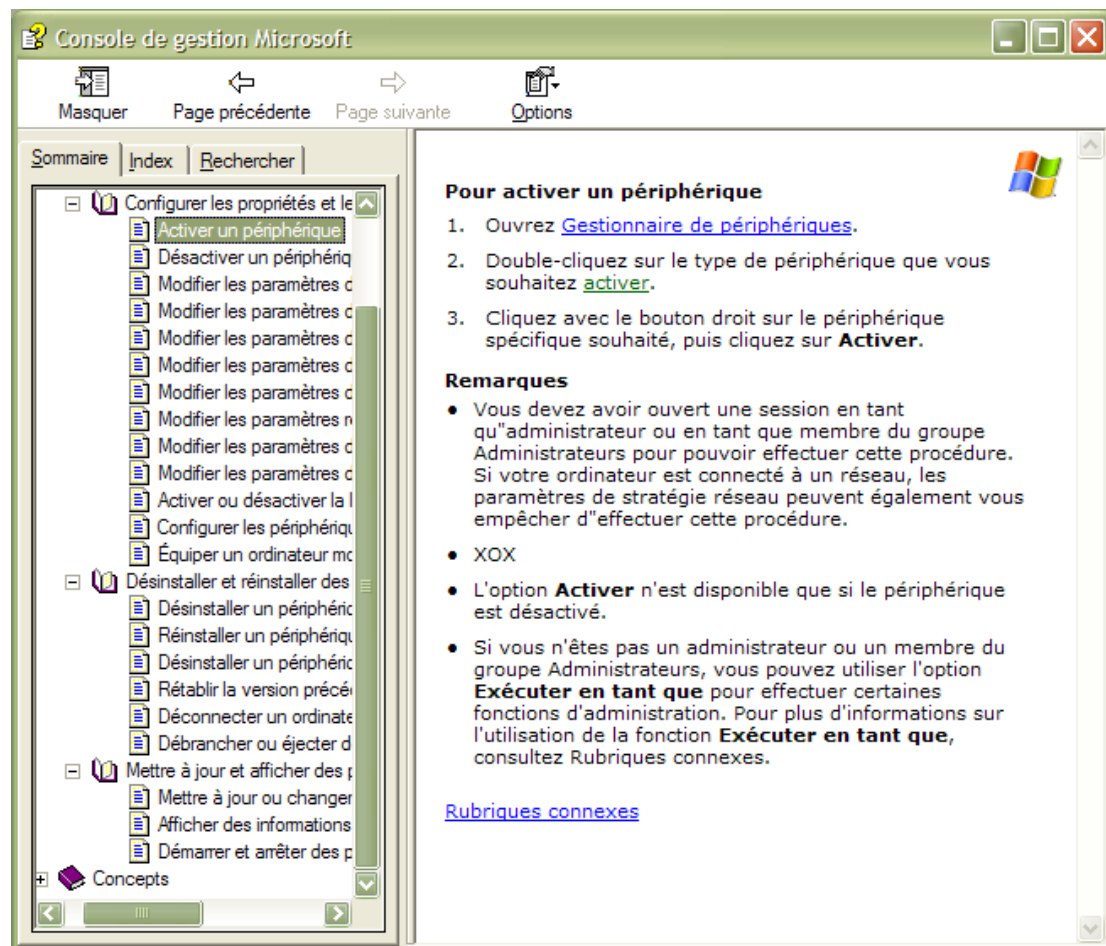
Commande  
Annuler

Oui

# 10. Aide et documentation

- Fournir de renseignement par écrit (où en ligne) sur le fonctionnement de l'application
- Les utilisateurs souvent ne lisent pas la doc...
- La documentation doit aider l'utilisateur à:
  - Chercher l'info
  - Comprendre l'info
  - Appliquer la procédure

# Exemple d'aide et documentation



# Example



**Description du problème:** dans la rubrique sur le livre "*O tempo e o Vento*", le bouton « critique » semble active mais il ne fournit aucun retour;

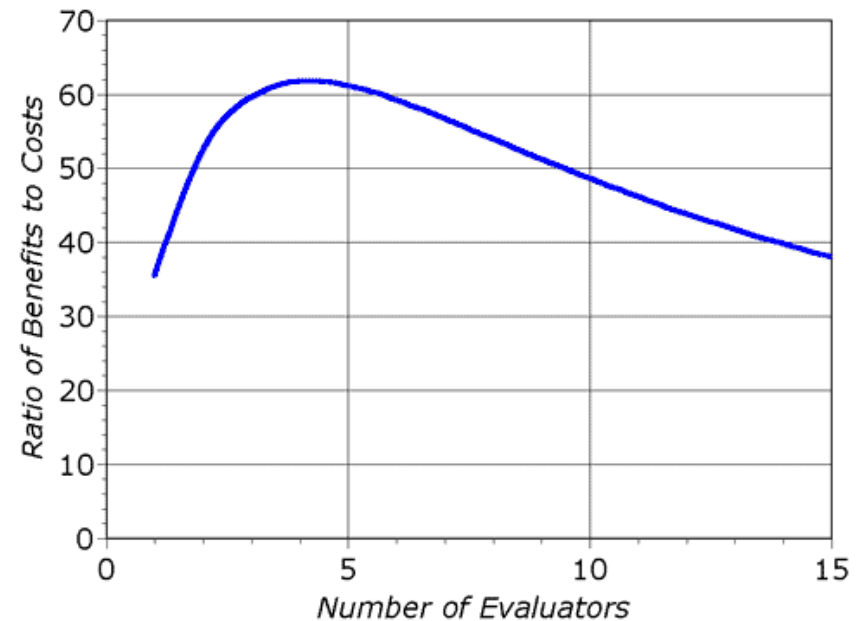
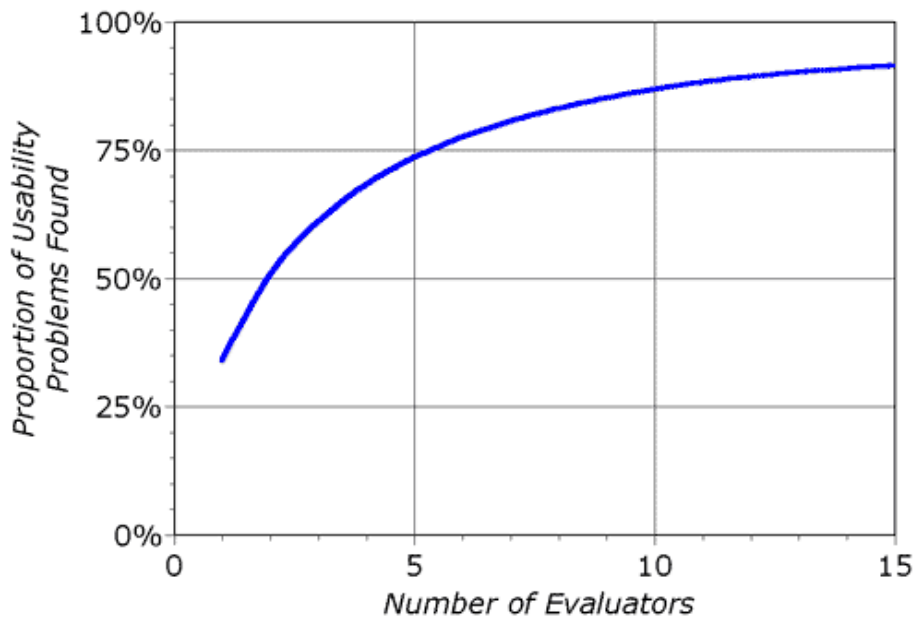
**Sévérité (1-3):** 3

**Heuristique violée:** *feedback* (H5).

# Le processus d'évaluation

- Nombre d'évaluateurs
- Expertise des évaluateurs
- L'approche Walkthroughout
- Description détaillé des problèmes
- Classification de problèmes par rapports a leur sévérité et les heuristiques

# Nombre d'évaluateur [ discount method ]



# Planifier l'évaluation

- Proposer aux évaluateur les tâches principales à inspecter
- Expliquer aux évaluateurs la raison d'être de l'outil pour qu'ils puisse réaliser leurs tâches
- Demander aux évaluateurs d'évaluer l'interaction avec le système
- Sélectionner les évaluateurs:
  - 5 évaluateur expérimentés trouverons 81%-90% des problèmes avec l'interface...
  - 5 évaluateurs novices (ex. des étudiants) trouverons 22%-29% de problèmes ...

# L'approche Walkthrough ...

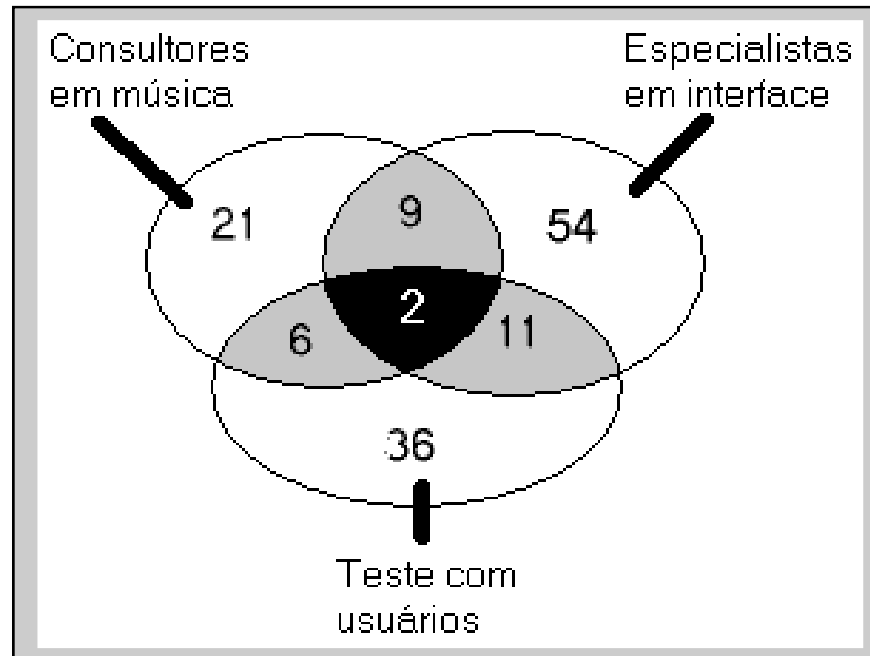
- Il faut un coordinateur et des évaluateurs...
- Le coordinateur sélectionne les évaluateur et fait la synthèse des évaluations individuelles
- Les évaluateurs ne doivent pas communiquer avant la fin de l'évaluation
- 1-2 heure pour chaque session de test
- L'évaluateur doit d'abord se familiariser avec l'outil avant de se lancer dans l'évaluation
- Chaque problème doit être classé selon une (ou plusieurs heuristique), une sévérité doit également être attribué
- Si aucune heuristique ne correspond au problème trouvé, l'évaluateur peut proposer une nouvelle heuristique et la justifier..



# Les Heuristiques pour le Web, les jeux, les téléphone portables, l'interaction avec les robots,...

- Nielsen, J. Designing Web Usability. Peachpit Press Publications. 432 pages.
- Pinelle, D., Wong, N., and Stach, T. 2008. Heuristic evaluation for games: usability principles for video game design. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 1453-1462. DOI= <http://doi.acm.org/10.1145/1357054.1357282>
- Clarkson, Edward C. Arkin, Ronald C. Applying Heuristic Evaluation to Human-Robot Interaction Systems. GVU Technical Report;GIT-GVU-06-08
- Bertini, E., Gabrielli, S., and Kimani, S. 2006. Appropriating and assessing heuristics for mobile computing. In *Proceedings of the Working Conference on Advanced Visual interfaces* (Venezia, Italy, May 23 - 26, 2006). AVI '06. ACM, New York, NY, 119-126. DOI= <http://doi.acm.org/10.1145/1133265.1133291>

# Evaluation Heuristique X Test utilisateurs



# Heuristic Evaluation

[ Comments ]

- + easy to apply
- + anyone could be training for using this method (!!!)
- + costs X benefits
- results are directly related to evaluators experience
- it covers only some kind of usability problems...

# **Rapport d'utilisabilité**

# Common Industry Format (CIF)

- Standard format for reporting usability problems
- US National Institute of Standards and Technology (NIST)

Formative test	Summative test
<ul style="list-style-type: none"><li>• During the development process</li><li>• To mould or improve the product</li><li>• Virtually anywhere (don't need a lab)</li><li>• With the test administrator and the participant co-present</li></ul>	<ul style="list-style-type: none"><li>• At the end of a development process</li><li>• To measure or validate the usability of a product</li><li>• To answer the question: "How usable is this product"</li><li>• To compare against competitor products or usability metrics</li><li>• To generate data to support marketing claims about usability</li><li>• In a usability lab</li><li>• With the participant working alone</li></ul>
<ul style="list-style-type: none"><li>• Participant comments in the form of a "thinking aloud" narrative (ex. attitudes, sources of confusion, reasons for actions)</li><li>• Photographs and highlights videos</li><li>• Usability problems and suggested fixes</li></ul>	<ul style="list-style-type: none"><li>• Statistical measures of usability (for example, success rate, average time to complete a task, number of assists)</li><li>• Reports or white papers</li></ul>

Session test form

Summary report

# Ce qu'il reste à faire aujourd'hui

- Inspecter les prototypes à l'aide de scénarios
- Utilisez la méthode d'évaluation heuristique
- Décrire les problèmes d'utilisabilités
- Proposer des solutions aux problèmes d'utilisabilité identifiés
- Corriger les prototypes
- Poser de questions 😊