

# Évaluation analytique

Interface Homme Machine  
Nathanaël Martel

# Évaluation analytique

**Soumettre l'interface à ses propres test  
... pour trouver et résoudre les problèmes**

**Attention à ne pas confondre avec**

- **Tests unitaire**
- **Tests fonctionnel**

# Évaluation analytique

- Évaluation pas à pas
- Keystroke-Level Model (KLM)
- Évaluation heuristique

# Évaluation pas à pas

- **Les évaluateurs regardent le système du point de vue de l'utilisateur**
- **Ils effectuent les tâches de l'utilisateur et prédisent où les utilisateurs auront des problèmes**
- **Ils se concentrent sur l'aspect intuitif de l'interface**

# Évaluation pas à pas

## Pour chaque étapes

- Commencez par une idée approximative de la tâche à accomplir
- Explorez l'interface et sélectionnez l'action la plus appropriée sur la base de la similitude avec l'objectif
- Surveillez les réactions de l'interface (est-ce que l'on approche de l'objectif?)
- Déterminer ce qu'il faut faire ensuite

# Évaluation pas à pas

## Préparation :

- Identifier les utilisateurs
- Identifier les tâches principales
- Pour chaque tâches, decrivez la séquence correcte d'action
- Faire une prototype
- Trouver des évaluateurs

# Évaluation pas à pas

**Pour chaque actions, est-ce que l'utilisateur :**

- **S'attend à faire cette action ?**
- **Voit le contrôle pour cette action ?**
- **Comprend que le contrôle produit l'action désirée ?**

# Évaluation pas à pas

**Si l'action correcte est effectuée,**

- **Les progrès sont-ils apparents ?**



# Évaluation pas à pas

**Exemple :**

# Évaluation pas à pas

## Évaluation pas à pas en groupe

- Avec un groupe hétérogène
- Relever des informations sur ce qui se dit
- Filmer le processus ?

# Évaluation pas à pas

## Avantages de l'évaluation pas à pas

- Identification des problèmes très tôt
- Pas cher
- Point de vue de la tâche

# Évaluation pas à pas

## Inconvénient de l'évaluation pas à pas

- **Demande un peu de pratique**
- **Besoin de comprendre la tâche de l'utilisateur**
- **Utile uniquement pour l'aspect intuitif de l'interface**
- **Pas réalisable hors de séquence de tâche**
- **Possibilité de rater des problèmes critique**
- **Chronophage**

# Keystroke-Level Model

- **Technique avancée et standardisé d'évaluation**
- **Entrée : description détaillé de l'interface et des tâches**
- **Sortie : mesure d'efficacité**

# Keystroke-Level Model

- **Basé sur le modèle de processeur humain**
- **Séparation du sujet humain en sous-système :**
  - Sensoriel
  - Moteur
  - Cognitif

# Keystroke-Level Model

## Liste des actions utilisateur spécifiques pour effectuer une tâche

- **K** : presser une touche
- **B** : appuyer sur le bouton de la souris
- **P** : pointer avec la souris
- **H** : déplacement de la main
- **D** : Dessiner un segment
- **M** : Se préparer à une action physique
- **R** : Réponse du système (si elle est significative)

# Keystroke-Level Model

- **Détailler les tâches, y compris les actions mentales.**
- **Prendre une correspondance de temps (exemple  $K=.1s$ )**
- **Faire la somme du temps d'exécution**



# Keystroke-Level Model

**Exemple : accéder à l'aide dans un logiciel**

- **H : déplacer la main vers la souris**
  - 0,4 s
- **P : déplacer le pointeur sur « aide »**
  - 1,1 s
- **B : presser le bouton de la souris**
  - 0,1 s
- **M : lire le menu et trouver l'intitulé correspondant à l'aide**
  - 1,35 s
- **P : déplacer la souris vers ce menu**
  - 1,1 s
- **B : relâcher le bouton de la souris**
  - 0,1 s

**Total : 4,15 s**

# Keystroke-Level Model

**Exercice : calculez le temps nécessaire pour**

- **Supprimer un fichier sur le bureau en le faisant glisser vers la corbeille.**
- **Supprimer un fichier sur le bureau en utilisant l'option « mettre à la corbeille » du menu**

# Keystroke-Level Model

## Supprimer en glissant vers la corbeille

- **H : déplacer la main vers la souris**
  - 0,4 s
- **M : rechercher l'icône du fichier**
  - 1,35 s
- **P : déplacer le pointeur sur l'icône**
  - 1,1 s
- **B : presser le bouton de la souris (et le maintenir appuyé)**
  - 0,1 s
- **P : déplacer le fichier dans la poubelle**
  - 1,1 s
- **B : relâcher le bouton de la souris**
  - 0,1 s

**Total : 4,15 s**

# Keystroke-Level Model

## Supprimer en utilisant l'option du menu

- **H : déplacer la main vers la souris**
  - 0,4 s
- **M : rechercher l'icône du fichier**
  - 1,35 s
- **P : déplacer le pointeur sur l'icône**
  - 1,1 s
- **B : presser le bouton droit de la souris**
  - 0,1 s
- **M : lire le menu et trouver l'intitulé correspondant**
  - 1,35 s
- **P : déplacer le pointeur sur l'icône**
  - 1,1 s
- **B : presser le bouton de la souris**
  - 0,1 s

**Total : 5,1 s**

# Keystroke-Level Model

## Limitations

- **Uniquement pour les utilisateurs expérimentés**
- **Ne prend pas en compte les erreurs ni la fatigue**
- **Ne prend pas en compte les différences individuel**
- **Chronophage**

# Évaluation heuristique

- **Les évaluateurs jugent si chaque éléments d'interface suit les règles et principes établis**
- **Ces principes sont disponible sous forme de check-list**

# Évaluation heuristique

- **L'évaluateur parcourt l'interface au moins deux fois**
  - la première, pour avoir une vision globale du système
  - la seconde, pour se concentrer sur les problèmes spécifiques d'interface
- **Noter les problèmes au fur et à mesure**
- **Comparer les différentes remarques entre évaluateurs**

# Évaluation heuristique

## Quelle liste utiliser ?

- **WCAG (Web Content Accessibility Guidelines)**
  - recommandation du W3C en matière d'accessibilité Web
  - obligation légale en France
- **Opquast**
  - 26 critères pour améliorer vos sites mobiles ... et d'autres check-lists
  - outil de test automatisé



# Évaluation heuristique

## Combien d'évaluateurs ?

- L'expérience montre que des personnes différentes trouvent des problèmes différents.
- Un évaluateur remonte à peine 35% des problèmes
- Entre 3 et 5
  - 5 experts remontent près de 80% des problèmes

# Évaluation heuristique

## Résultats attendus

- Liste des problèmes avec la référence au principe
- Description du problème la plus précise possible
- Trié par sévérité

# Évaluation heuristique

## Avantage

- **Peu cher**
- **Très intuitif et très motivant**
- **Pas besoins de planification**
- **Dès le début du développement**
- **Favorise l'apprentissage des bonnes pratiques à tous les membres de l'équipe**

# Évaluation heuristique

## Inconvénient

- Il peut arriver que l'on trouve des problèmes sans avoir de solution
- Les évaluateurs ne sont pas des utilisateurs réels

# Conclusion

Il est possible d'améliorer pas mal de problème en évaluant seul et en écoutant les autres membres de l'équipe.  
Mettez vous dans la peau de votre utilisateur.  
Apprenez et vérifiez les bonnes pratiques