**Check list selon les 10 principes de Jacob Nielsen**

La technique d’évaluation heuristique est une variation des inspections d’utilisabilité où les spécialistes jugent en profondeur si chaque partie des interfaces suivent les principes d’utilisabilité. Pour appliquer ce type d’évaluation il faut suivre trois stratégies :

1. reconnaître le système en concentrant l’attention sur les tâches les plus communes et importantes ;
2. se maintenir au courant des critères et recommandations ergonomiques appartenant aux composantes du système;
3. réaliser une balayage critique du système et des connaissances ergonomiques disponibles.

Pour Nielsen, l’utilisabilité possède de multiples composantes et est associée traditionnellement à quatre attributs : apprentissage, efficience, gestion d’erreurs et satisfaction. Ces quatres attributs donnent naissance à des heuristiques (dialogues simples et naturels, parler le langage de l’utilisateur, minimiser l’effort de mémorisation de l’utilisateur, consistance, feed-back, sorties claires, action minimales, bons messages d’erreurs, aide et documentation). A partir de plusieurs erreurs qui ont été rencontrées par les spécialistes et pour assurer un meilleur rapport coût - bénéfice, Nielsen a analysé et affiné d’innombrable directives et de longues listes d’inspections (249 problèmes d’usage) en les regroupant en 10 principes.

**Les 10 principes :**

1. **Visibilité de statut du système**

Le système devrait maintenir l’utilisateur toujours informé sur sa localisation à travers des marquages appropriées dans un temps raisonnable. Dans ce principe, deux chose sont plus importantes : Les utilisateurs doivent savoir *où est-ce qu’ils sont* et *où est-ce qu’ils doivent aller.*

1. **Correspondance entre le système et le monde réel**

Le système devrait " parler " la langue des utilisateurs de telle façon que les informations apparaissent dans un ordre naturel et logique, en se conformant aux mots, expressions et conventions usuels.

1. **Contrôle de l’utilisateur et liberté**

Le système devrait être très " obéissant " aux actions de l’utilisateur et offrir des " sorties de secours " clairement marquées pour les opérations exécutées mais non désirées, ainsi que des options de faire/défaire les opérations dans le dialogue. Il faut aussi permettre à l’utilisateur de personnaliser sa zone de travail comme la couleur, la police de caractères, la largueur de l’écran, la version du navigateur, etc.

1. **Consistance et étalon (normalisation)**

Suivre les conventions de plate-forme est la façon la plus sûr de respecter ce principe. Les utilisateurs ne doivent pas se demander si les différentes mots, situations ou actions veulent dire la même chose. Les titres et entêtes doivent être représentatifs du contenu.

1. **Prévention contre les erreurs**

Le système devrait prévoir les erreurs que l’utilisateur peut commettre. Il doit offrir de bons messages d’erreurs et instructions compréhensibles pour les corriger afin d’éviter que des problèmes surviennent.

1. **Reconnaissance plutôt que rappel**

Le système devrait rendre les objets, les actions et les options visibles. L’utilisateur ne doit pas se rappeler des informations d’une partie du dialogue à l’autre. Les directives pour l’usage du système doivent être visible ou facilement récupérables à chaque fois que c’est nécessaire pendant le dialogue (boutons, cartes de navigation, bonnes étiquettes et liens descriptifs, par exemple).

1. **Flexibilité et efficience d’utilisation**

Le système devrait offrir des raccourcis pour les utilisateur expérimentés. Mettre des accélérateurs permettant aux utilisateurs de faire et/ou récupérer les actions plus fréquemment utilisées (par exemple, des signet et etc.).

1. **" Design " esthétique et minimaliste**

Les dialogues devraient contenir seulement les informations pertinentes et principales qui devraient être bien visible. Les informations non pertinentes ou rarement utilisées doivent être placées dans les niveaux les plus bas ou les plus détaillés dans la hiérarchie de distribution du contenu. Chaque unité d’information supplémentaire dans un dialogue rivalise avec les unités pertinentes et diminue leur visibilité relative.

1. **Fonctions de l’aide (reconnaissance, diagnostic et récupération d’erreurs)**

Les messages d’erreurs devraient être exprimés dans la langue ordinaire, sans code en indiquant précisément le problème de manière constructive, claire et compressible et de minière à offrir une solution aux problèmes ou en tout cas comment les résoudre.

1. **Aide et documentation**

Le système devrait fournir des aides et de la documentation. Toutes les informations devraient être facile à chercher et être concentrées sur la tâche (par exemple, tutoriels, spécifications, agents intelligents, etc.).

## Les critères de Bastien et Scapin

### 1. 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.).

Justification : un bon guidage facilite l’apprentissage et l’utilisation du système en permettant à l’utilisateur :

* de savoir, à tout moment où il se trouve dans une séquence d’interactions, ou dans l’accomplissement d’une tâche ;
* de connaître les actions permises ainsi que leurs conséquences ;
* d’obtenir de l’information supplémentaire.

La facilité d’apprentissage et d’utilisation qui s’ensuivent conduisent à de meilleures performances et occasionnent moins d’erreurs.

### 1.1. Incitation

Moyens pour l’utilisateur de connaître les actions disponibles, les alternatives, l’état du contexte où il se trouve.

### 1.2. Groupement/Distinction par le format ou le groupement

Organisation visuelle des items d’information les uns par rapport aux autres. Il prend en compte la localisation et le format pour indiquer les relations entre les divers items affichés, et leur appartenance ou non à une même classe d’items.

### 1.3. Feedback Immédiat

Une réponse doit être fournie à l’utilisateur le renseignant sur l’action accomplie et sur son résultat, ceci, avec un délai de réponse approprié et homogène selon les types de transactions (souvent, délai immédiat).

### 1.4. Lisibilité.

Caractéristiques matérielles de présentation des informations qui doivent en faciliter la lecture (typographie, espacement, ...)

### 2. 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.

Justification : Plus la charge de travail est élevé, 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.

### 2.1. Brièveté

Limiter le travail de lecture et d’entrée d’information :

* Concision : pour les éléments individuels d’entrée ou de sortie.
* Actions Minimales : limiter le nombre d’actions successives nécessaires pour atteindre un but.

### 2.2. Densité informationnelle

Réduire au maximum la charge informationnelle

### 3. CONTROLE 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.

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.

### 3.1. Actions Explicites

La relation entre les actions de l’utilisateur et les réponses du système doit être explicite, c’est-à-dire que le système doit exécuter :

* seulement les actions demandées par l’utilisateur
* au moment où il(elle) les demande.

### 3.2. Contrôle Utilisateur

L’utilisateur doit toujours pouvoir contrôler le déroulement des traitements informatiques en cours.

### 4. ADAPTABILITE

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

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 lui 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.

### 4.1. Flexibilité

Réfère aux moyens disponibles à l’utilisateur pour personnaliser son interface de façon à prendre en compte ses stratégies de travail et/ou ses habitudes et les exigences de ses tâches.

### 4.2. Prise en compte de l’expérience

Les différents moyens disponibles pour prendre en compte le niveau d’expérience de l’utilisateur (expérimenté, débutant, occasionnel).

### 5. 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…

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.

### 5.1. Protection contre les erreurs

Réfère aux moyens disponibles pour détecter et prévenir (avant validation) :

* les erreurs d’entrée de données,
* les erreurs de commandes,
* les actions à conséquences destructrices

### 5.2. Qualité des messages d’erreur

Réfère à l’expression et au contenu des messages d’erreur :

* leur pertinence,
* leur facilité de lecture,
* leur précision quant à la nature des erreurs,
* l’indication des actions de correction

### 5.3. Correction des erreurs

Réfère aux moyens disponibles pour l’utilisateur pour corriger immédiatement ses erreurs.

### 6. 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.

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 de refus d’utilisation.

### 7. SIGNIFIANCE DES CODES ET DENOMINATIONS

Définition : Le critère Signifiance 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.

Justification : Lorsque le codage est signifiant, le rappel et la reconnaissance sont meilleurs. De plus, les codes et dénominations non significatifs pour les utilisateurs peuvent leur suggérer des opérations inappropriées et ainsi conduire à des erreurs.

### 8. COMPATIBILITE

Définition : Le critère Compatibilité se réfère à l’accord pouvant exister entre les caractéristiques des utilisateurs (mémoire, perceptions, habitudes, compétences, âge, attentes, etc.) 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.

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. Les performances sont meilleures lorsque l’information est présentée sous une forme directement utilisable.

**7 règles d’or [Coutaz, 90]**

* + Lutter pour la cohérence
  + Lutter pour la concision
  + Réduire la charge cognitive
  + Mettre le contrôle entre les mains de l’utilisateur
  + Souplesse d’utilisation
  + Structurer le dialogue
  + Prédire les erreurs