Chapitre 6 – Personas

- **Personas are fictitious characters created to represent the different user types within a targeted demographic that might use a site or product.**
- A user persona is a representation of the goals and behavior of a real group of users.
- Personas Helps us to decide about a product

1. Building personas

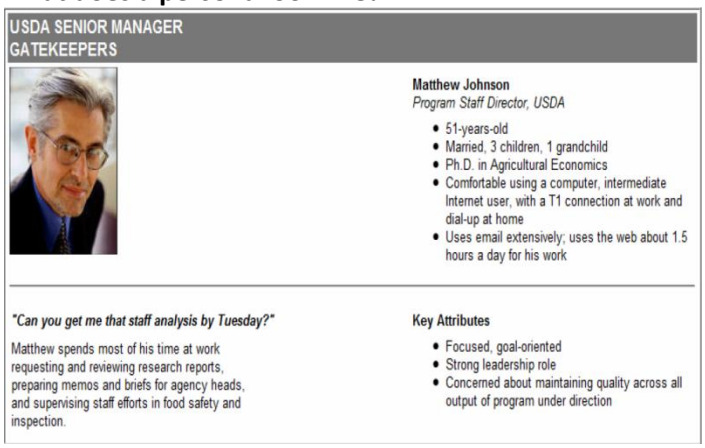
How do we get information for a persona?

By analyzing what you learned about your users from user research, including:
Contextual Interviews, individual Interviews, surveys (Online), usability testing
After getting information for a persona, you can select the characteristics which are most representative

2. "10 steps" approach to Personas

Step 1 - Finding the Users: Capture real user's data from ethnographic or other qualitative studies.
Step 2 - Building a Hypothesis: Identify the ways and context when the real user interacts with the system.
Step 3 - Verification: Break your information down into candidate Personas.
Step 4 - Finding Patterns: Try grouping candidates, breaking down a candidate into several
Step 5 - Constructing Personas: Define the psyche, the background and the traits for each candidate.
The following steps relate to the usage of Personas in the bigger picture of user centric design:
Step 6 - Defining Situations: Identify the needs and situations, and relate them to the Personas.
Step 7 - Validation and Buy-in: Socialize and ensure that all participants agree on the descriptions.
Step 8 - distribution of Knowledge: Share the Personas, situations and data with all the organization.
Step 9 - Creating Scenarios: Describe what happens in a given situation, when a given Persona uses the system.
Step 10 - Ongoing Development: Validate the Personas, needs and scenarios every time new data is captured.

What does a persona look like?



Characteristics

A persona usually includes:

- a name and picture
- demographics (age, education,ethnicity, family status)
- job title and major responsibilities
- goals and tasks in relation to your site
- environment (physical, social, technological)
- a quote that sums up what matters most to the persona with relevance for your site

3. Conclusion

- Personas allow you to identify and communicate user needs efficiently and effectively. By developing 'stand in' users, based on real user data, the design team can concentrate on designing for these typical users with the confidence that the needs of the user base will be met.

Chapitre 8 – Les recommandations AFNOR

I- 'Ergonomie et conception du dialogue homme-ordinateur'
II- 'Définition des critères ergonomiques de conception et d'évaluation des produits logiciels'

1. Comptabilité

a. Niveau produit – Les utilisateurs connaissent d'autres produits.
- **a.** exploiter cette connaissance.
- **b.** dans une même compagnie, avoir un style d'interface utilisateur.

b. Niveau tâche :
i. raisonner en termes de tâche utilisateur et faciliter le passage d'une tâche à une autre (multi-fenêtrage)

c. La compatibilité répond aux **objectifs** suivants :
i. correspondance entre les connaissances de l'utilisateur et la capacité du logiciel.
ii. univers familier et habituel ➔ apprentissage facilité.

d. Critère essentiel conditionnant la **pertinence** de tous les autres.

2. Guidage – C'est l'ensemble des moyens mis à disposition de l'utilisateur pour :
- **connaître** l'état du système.
- **établir** le rapport entre actions et état du système.
- **évaluer** le système et orienter son action sur celui-ci.

2 types de guidage :
1. **explicite** : message d'avertissement, évitement d'erreurs, aide en ligne, codes clairs, explicites et sans ambiguïté.
2. **implicite** : structuration de l'affichage, différentiation par typographie (couleur, attributs informatiques ...) des catégories d'information.

Objectifs :
- faciliter l'apprentissage.
- aider l'utilisateur à se repérer et à choisir ses actions.
- prévenir les erreurs.

3. Homogénéité (ou consistance) – C'est la similarité interne d'un produit :
- Capacité d'un système informatique à conserver une logique d'usage constante dans une application ou d'une application à une autre (niveau procédure et niveau présentation des informations)
- Stabilité des choix de conception.

Ex : Les menus dans MS Word, Paint, Excel sont très similaires.

Objectifs :
- rendre le comportement du système prévisible.
- diminuer le temps de recherche d'une information.
- faciliter la prise d'informations.

4. **Souplesse –** C'est la capacité de l'interface à s'adapter aux différentes exigences de la tâche, aux diverses habitudes et connaissances des utilisateurs :
   - Personnalisation de l'interface :
     - dans le **fonctionnement** (adaptation du logiciel à diverses populations d'utilisateurs)
     - dans **l'utilisation** (diverses procédures, options et commandes pour atteindre un même objectif)

   - Cette flexibilité permet d'atteindre les **objectifs** suivants :
     - adaptation à la diversité des utilisateurs.
     - l'outil doit s'adapter à l'homme et non l'inverse.

5. **Contrôle explicite –** C'est l'ensemble des éléments du dialogue qui permettent à l'utilisateur de maîtriser le lancement et déroulement des opérations :
   - sémantique des commandes rendant compte de leurs effets.
   - effets des commandes prédictibles.

   Objectifs :
   - favoriser la prévision des réactions de l'interface.
   - favoriser l'apprentissage.
   - diminuer les risques d'erreur.

6. **Gestion des erreurs –** C'est l'ensemble des moyens pour guider l'utilisateur dans la perception, l'identification de ses erreurs et conserver l'intégrité de l'application (Robustesse)

   Objectifs
   - favoriser l'exploration et l'apprentissage par un système tolérant les changements de décision des utilisateurs.
   - éviter les perturbations (crainte, blocage,...) associées à la difficulté de corriger les erreurs commises.
   - permettre à l'utilisateur de localiser, comprendre et corriger précisément.

7. **Concision –** C'est l'ensemble des moyens qui contribuent pour l'utilisateur à la réduction de ses activités de perception et mémorisation

   Objectifs
   - optimiser la prise d'informations et de décision en présentant des informations précises et brève.
   - minimiser le nombre d'actions ou d'opérations et le temps de manipulation.

Chapitre 9 – Les GUIDES de STYLE

Les styles de dialogues :

1. **Les menus**

Avantages	Inconvénients	Conseille pour
facilité d'apprentissage	flexibilité faible	utilisateur peu motivé
facilité de mémorisation	navigation parfois fastidieuse et difficile	faibles connaissances et expériences

Principes et règles de conception :

i. Structure des menus

- structure des menus = structure de la tâche.
- ordre des menus = ordre des actions utilisateurs.
- minimiser la profondeur, étendre en largeur.
- menus verticaux avec des labels courts.

ii. Ordre des choix :

Conventionnels, fréquence d'utilisation, ordre attendu, catégories sémantiques, alphabétiques, . . .

2. Les grilles de saisie

Avantages		Inconvénients
facilité d'apprentissage facilité de mémorisation	bonne utilisation de l'espace écran possibilité de saisies de données très variées	Connaissances supposées connues des formats d'entrées

a) **Principes et règles de conception :**

i. Organisation de la grille :

- partir du support papier s'il existe, sinon groupement sémantique par importance relative d'utilisation.
- éviter la mémorisation d'un écran à un autre
- définir la taille des groupes.

ii. Remplissage des champs :

- placer les labels à gauche pour alphabétique, à droite pour numérique.
- découper les listes longues par ligne blanche (par 5).
- distinguer les zones à remplir par attribut visuel (couleur, inverse vidéo, soulignement,..).

iii. Format des entrées :

- tolérer différentes entrées si non ambiguïté.
- découper les formats d'entrée trop longs.
- proposer des valeurs par défaut.

b) **Entrée des données :**

- rendre simples les entrées très utilisées.
- l'utilisateur spécifie l'unité de mesure mais n'effectue pas les conversions.
- codes et abréviations familiers.
- garder les champs les plus courts possibles.
- éviter les passages minuscules/majuscules, les combinaisons lettres/chiffres, les remplissages de Zéros.

c) **Navigation :**

- positionner curseur dans la zone la plus probable de remplissage.
- tolérer les mouvements avant/arrière entre les champs et dans les champs.
- numéroter avec titre commun si plusieurs écrans nécessaires.

d) **Traitement des erreurs :**

- utiliser la surbrillance des zones d'erreurs avec messages.

- tolérer l'édition dans les champs (insert, supprimer, overstrike).

3.Les langages de commandes

Avantages	Inconvénients	Conseille pour
puissance et flexibilité	apprentissage difficile	attitude positive, motivation forte
efficacité, rapidité	forte mémorisation, saisie importante par frappe clavier	expériences et connaissances : fortes (tâches, systèmes, informatiques)
faible occupation de l'écran et ressources	risques importants d'erreurs	habitude frappe clavier

a)Principes et règles de conception

i. Aspects sémantiques :
- choisir entre langage riche ou langage minimal.

ii. Aspects syntaxiques :
- utiliser une syntaxe forme impérative: verbe-objet.
- utiliser des paramètres par défaut.
- éviter l'usage fréquent de touches "shift" ou de touches de contrôle.

b) **Interaction :**	c)**Vocabulaire :**
fournir des retours d'informations (aide-mémoire et référence en ligne, prompts)	utiliser le vocabulaire courant, bien différentié (pas de jargon informaticien : utiliser le vocabulaire de l'utilisateur)
utiliser des clés de fonction pour les commandes très utilisées	construire les abréviations selon une règle simple
permettre la configuration du langage (abréviations, synonymes)	dans le manuel : mots complets (novices), ou non (experts)

4.L'organisation des écrans

a)**Nombres:**	b)**Techniques de codage, on trouve sur les stations:**
entiers : justifiés à droite ; décimaux : alignement sur la virgule éviter les zéros non significatifs	clignotement, gras, taille, fonte, soulignement, formes, caractères spéciaux et icônes, encadrement, son et couleur
découper les nombres par tranches de 3 ou 4 chiffres avec les séparateurs usuels	ces techniques permettent d'attirer l'attention de l'œil de l'utilisateur, mais en abuser fait disparaître l'effet attendu

c)**Texte : 3 types**

1. Messages	2. Prompts (indications courtes)	3. Instructions (indication plus complexes)
brefs et concis, adapté a l'utilisateur	bien localisés et adaptés	texte: simples et clairs
constructifs plutôt que critiques	grammaticalement simples	
plaçant l'utilisateur en situation de commande	ordre d'utilisation	

Conseils

lecture d'un texte
un texte écrit en minuscules
se lit beaucoup plus vite
qu'un texte en majuscules. La vitesse de lecture en majuscules a été estimée 13% plus lente qu'en minuscules, ceci provenant d'une différentiation plus forte des minjuscules que des majuscules. Estimation faite par Tullis en 1988. De même, la lecture d'un texte est améliorée si la
longueur d'une ligne est
supérieure à 26 caractères (longueur conseillée 50 à 55 caractères ou doubles colonnes de 30à 35 car)

JUSTIFICATION ET COLONNES

lecture d'un texte
un texte écrit en minuscules se lit beaucoup plus vite qu'un texte en majuscules. La vitesse de lecture en minuscules a été estimée 13% plus lente qu'en majuscules , ceci provenant d'une différentiation plus forte des majuscules que des minuscules.

lecture d'un texte
un texte écrit en minuscules se lit beaucoup plus vite qu'un texte en majuscules. La vitesse de lecture en majuscules a été estimée 13% plus lente qu'en minuscules , ceci provenant d'une différentiation plus forte des minuscules que des majuscules.

Lecture d'un texte

lecture d'un texte
un texte écrit en minuscules se lit beaucoup plus vite qu'un texte en majuscules. La vitesse de lecture en majuscules a été estimée 13% plus lente qu'en minuscules, ceci provenant d'une différentiation plus forte des minuscules que des majuscules. Estimation faite par Tullis en 1988. De même, la lecture d'un texte est améliorée si la longueur d'une ligne est supérieure à 26 caractères.(longueur conseillée 50 à 55 caractères ou doubles colonnes de 30à 35 car)

Nouveau
Ouvrir
Fermer
Enregistrer
Options
Mise en Page
Quitter

Menu : caractère gras et inverse-vidéo

Nouveau
Ouvrir
Fermer
Enregistrer
Options
Mise en Page
Quitter

un texte écrit en minuscules se lit beaucoup plus vite qu'un texte en majuscules.
La vitesse de lecture en majuscules a été estimée 13% plus lente qu'en minuscules,
ceci provenant d'une différentiation plus forte des minuscules que des majuscules.
Estimation faite par Tullis en 1988. De même, la lecture d'un texte est améliorée
si la longueur d'une ligne est supérieure à 26 caractères.(longueur conseillée 50 à
55 caractères ou doubles colonnes de 30à 35 car)

➔ Ne pas souligner de longs textes, ceci réduit la lisibilité

ce n'est pas en mettant du rouge à lèvres à un bull-dog qu'on a envie de l'embrasser

ce n'est pas en mettant du rouge à lèvres à un bull-dog qu'on a envie de l'embrasser

ce n'est pas en mettant du rouge à lèvres à un bull-dog qu'on a envie de l'embrasser

Conseils:

1. fond monochromatique
2. texte couleur brillante
3. éviter fonds marrons ou verts
4. contraste élevé en brillance et saturation

Chapitre 10 – Les évaluations des systèmes interactives

Evaluation de l'interaction :

1. Pourquoi évaluer?

- L'intuition du concepteur du système *ne peut suffire*.
- La modélisation formelle du système et de l'interaction ne couvre pas tous les *choix de conception*.
- Les recommandations (*guidelines*) restent générales pour couvrir tous les aspects d'une interaction spécifique.

2. Quand évaluer?

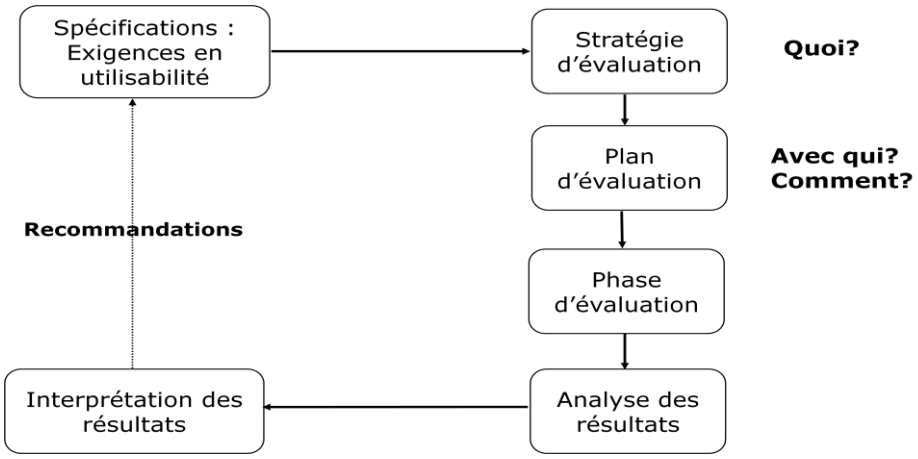
Le cycle de vie logiciel concerne aussi l'interaction

- Cycle de vie avec prototypage.
- Evaluation à **toutes** les étapes du développement.
- **Prototype papier** (se focalise sur l'essentiel) vs. **Prototype interactif** (plus artificiel avec une basse fidélité)

L'évaluation ne donne de résultat que si elle est bien préparée.

3. Comment procéder à l'évaluation?

Processus d'évaluation



a)Stratégie d'évaluation

i. **Quoi** : que chercher à évaluer ?
- Nombreux facteurs d'utilisabilité donc nombreuses façons d'envisager une évaluation.
- Définir des objectifs précis, liés à des facteurs bien identifiés.

○ **Critères d'utilisabilité de Nielsen**
✓***Efficacité*** : taux de réussite de la tâche à réaliser.
✓***Efficience*** : importance de l'effort nécessaire.
✓***Satisfaction*** : ressenti subjectif de l'utilisateur.

○ **Critères d'utilisabilité de Quesenbery**
✓***Efficacité.***
✓***Efficience.***
✓***Attirance (caractère engageant)***
✓***Tolérance aux erreurs.***
✓***Apprenabilité.***

	Métrique d'évaluation quantitative - Mesures d'utilisabilité		
	Efficacité	**Efficience**	**Satisfaction**
Adéquation à la tâche	% de buts atteints	temps pour réaliser la tâche	Echelle de jugement de satisfaction
Approprié pour l'utilisateur entraîné	Nombre de fonctions importantes utilisées	Efficacité relative comparée à un expert	Echelle de jugement de satisfaction
Apprenabilité	% de fonctions apprises	Temps d'apprentissage	Echelle de jugement de la facilité d'utilisation
Tolérance à l'erreur	% d'erreurs corrigées	Temps passé à corriger les erreurs	Echelle de jugement de facilité de correction

ii. **Méthodologie**

1. **Préciser les critères d'utilisabilité importants pour le projet** (public visé, objectifs principaux,…)
2. **Hiérarchiser les priorités** (Noter l'importance des différents facteurs d'utilisabilité pour le projet)
3. **Choisir le type de données adaptées à l'évaluation de chaque critère**

Evaluation **qualitative** ou **quantitative** ?
- **Quantitatives** - Basées sur des mesures objectives (durées, % d'erreurs…) obtenues lors des expérimentations.
- **Qualitatives** - Données subjectives non numériques (texte, entretien) obtenues auprès d'utilisateurs ou d'experts en ergonomie des **IHMs(interface homme-machine)**.

4. **Spécifications** – Peuvent donner des indications sur le type de données à privilégier. Elles imposent parfois des mesures quantitatives très précises (**Ex** : Pas plus de 10 sec pour afficher la page d'accueil)

Sinon, c'est à vous de proposer une stratégie d'évaluation que vous pourrez faire valider.

b) Plan d'évaluation

Il reste de nombreux choix à faire pour mettre en place de manière opérationnelle l'évaluation :

- **Qui** utilisera le système lors de l'évaluation ?
- **Où** (dans quelles conditions) se déroulera l'évaluation ?
- **Quand** (quelle état d'avancement du projet) se déroulera l'évaluation ?

Le plan d'évaluation répond à ces questions en définissant :
✓La **technique d'évaluation** la plus appropriée.
✓La **préparation de l'évaluation** en fonction de cette technique.

- **Avec qui évaluer ?**
  - **Avec des utilisateurs** : **expérimentation**
    - Observer les utilisateurs.
    - Demander leur avis.
    - Mesurer leurs performances.

Recrutement : Profile de l'utilisateur (âge, …), Nombre de sujet (5 min), Ne jamais faire appel au concepteur du system.

  - **Sans utilisateur** : analyse a priori (ou **heuristique**)
    - Demander leur avis à des experts.
    - Simuler le fonctionnement en vérifiant des heuristiques ergonomiques.
    - Estimer le comportement du système à l'aide d'un modèle.

- **Paradigmes d'évaluation** (Chaque titres sera définit en détails en ce qui suit)

A) **Evaluation analytique / heuristique**
   i. Évaluation a priori
      I- revue d'expert (analyse heuristique)
      II- promenade cognitive (*cognitive walkthrough*)
      III- évaluation participative (*brainstorming*)
   ii. Modèles prédictifs.

B) **Evaluation expérimentale : objective ou subjective**
   i. Évaluation subjective : questionnaires ou interviews (entretiens)
   ii. Tests d'utilisabilité avec des utilisateurs potentiels.
   iii. Tests d'acceptabilité auprès de populations échantillon.
   iv. Évaluation post-commercialisation.
   v. Expérimentations cognitives.

Exemple de fiche d'évaluation heuristique

Tâche n°: **Version du système**	**Expert :** **Date :**		
Localisation dans la tâche	**Heuristique**	**Description du problème**	**Commentaires ou suggestion de solution**
Un nouveau mail arrive dans la boîte aux lettres	***Observabilité***	*L'utilisateur n'est pas informé de cette arrivée.*	*L'utilisateur aimerait sans doute être alerté (par un son par exemple) de cette arrivée.*

A) Evaluation Heuristique

i. Evaluation a priori

I- **Analyse heuristique**

- <u>Definition</u>
  - Revue du système par des experts.
  - Validation d'un certain nombre d'heuristiques ergonomiques (**Heuristiques de Nielsen**)
  - Sur spécifications d'écran et d'interaction ou système ou prototype existant.
  - Doit être validée par des tests utilisateurs.

- <u>Mise en œuvre pratique</u>
  - **Définir des tâches** comme pour une évaluation expérimentale avec des sujets naïfs + **préciser les heuristiques** à étudier.

- **Mettre en place l'évaluation** : pas besoin de conditions d'utilisations réelles.
- **Recueil des données** : l'expert rempli une fiche d'évaluation.
- **Synthèse** : hiérarchiser les problèmes en termes de sévérité et de criticité.

Heuristique de Nielsen :
La prise en compte de ces principes lors de la conception de l'interface permet d'obtenir au mieux une interface utile et utilisable (IHM ou Interface Homme-Machine).

1. **Observabilité** (visibility of the system status) :
   Un feedback doit être fourni non seulement lorsqu'une erreur (de l'utilisateur ou du système) survient, mais aussi lors du déroulement normal des actions.

Le feedback fourni par le système a deux caractéristiques **importantes** :

- le **temps de réponse** : délai précédant la survenance du feedback suite à l'action effectuée par l'utilisateur.
- la **persistance** : période durant laquelle le feedback peut être perçu par l'utilisateur.

2. **Familiarité** (match between system and the real world)

Le système doit s'exprimer dans le langage de l'utilisateur, avec des mots, des phrases et des concepts familiers à celui-ci, plutôt qu'avec des termes "orientés système". Il faut respecter les conventions du monde réel, en faisant apparaître les informations dans un ordre logique. (**Ex** : Les formes d'icones d'un message qui presente l'action de check email)

3. **Contrôle** (user control and freedom)

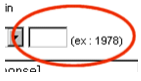
Les utilisateurs choisissent souvent par erreur des fonctions du système et ont besoin d'issues de secours clairement balisées pour quitter la situation non désirée sans devoir parcourir un long dialogue. (**Ex** : Undo , Redo)

4. **Consistance** (consistency and standards)

Les utilisateurs ne devraient pas se demander si différents mots, situations, ou actions veulent dire la même chose. Il faut se conformer aux conventions de la plate-forme.
De la même manière, des opérations sémantiquement identiques, même utilisées dans des contextes différents, sont mieux accomplies si elles sont présentées de manière uniforme, et actionnées par les mêmes mécanismes.
Ex : Les commandes "Copier/Coller" sont identiques dans à peu près toutes les applications de Microsoft Windows.

5. **Prévention des erreurs** (errors prevention)

Les dialogues ne devraient pas contenir des informations qui sont non pertinentes ou rarement utilisées. Toute information superflue dans un dialogue entre en compétition avec les informations pertinentes et diminue la visibilité relative de ces dernières.
En effet, ajouter trop d'informations dans une interface peut détourner l'attention de l'utilisateur de l'information principale. L'information principale, identifiée par l'analyse de la tâche, est celle qui doit se retrouver sur l'écran principal. Les informations moins importantes peuvent être accessibles mais, par exemple sur d'autres écrans…

6. **Faciliter la remémoration** (recognition rather than recall)
   Il faut rendre les objets, les actions et les options visibles. Il ne convient pas que l'utilisateur doive se rappeler des informations données à un endroit du dialogue lorsqu'il est à un autre endroit du dialogue. Les instructions d'utilisation du système devraient être visibles ou faciles à retrouver lorsqu'elles sont nécessaires 

7. **Flexibilité et efficacité** (flexibility and efficiency of use)

Des accélérateurs - non vus par l'utilisateur novice - peuvent souvent accélérer l'interaction pour l'utilisateur expert de telle façon que le système puisse s'adresser à la fois aux novices et aux experts. Il faut permettre aux utilisateurs d'adapter des actions fréquentes.

Dans un langage à ligne de commande, l'utilisateur doit pouvoir faire appel aisément et rapidement aux dernières commandes utilisées sans avoir à les réinsérer entièrement.
Les navigateurs Web actuels incluent tous la fonction de complétion automatique lorsque l'utilisateur insère un début d'URL déjà visitée. L'historique des adresses des sites dernièrement visités permet de retrouver rapidement un site.

8. **Esthétique** (aesthetic and minimalist design)

Les messages d'erreur doivent être exprimés dans un langage clair (pas de codes), indiquer précisément le problème et suggérer une solution d'une manière constructive.
Les messages d'erreur doivent suivre quatre recommandations :

1. Ils doivent être **formulés dans un langage clair**. (éviter les codes "orientés système")
2. Ils doivent indiquer **clairement** le problème.
3. Ils doivent aider l'utilisateur à **solutionner le problème**.
4. Les messages d'erreurs ne doivent **pas intimider ou rendre responsable** explicitement l'utilisateur.

9. **Gestion des erreurs** (help users recognize, diagnose and recover from errors)

Plutôt qu'un bon message d'erreur, il vaut mieux un design soigneux qui empêche un problème de se produire.
Si l'utilisateur désire réaliser une commande dont les conséquences peuvent être sérieuses, le système devrait lui demander au préalable une confirmation (Ex : Dans Linux, lors de l'execution de la commande « rm »(remove directory))

10. **Aide** (help and documentation)

Même s'il est préférable qu'un système puisse être utilisé sans documentation, il peut être nécessaire de fournir une aide et une documentation. Toute information de ce type doit être facile à chercher et ciblée sur la tâche de l'utilisateur. Elle doit reprendre la liste des étapes concrètes à suivre et ne doit pas être trop volumineuse.
De telles informations doivent :

1. être faciles à rechercher.
2. être focalisées sur la tâche de l'utilisateur.
3. détailler la liste des étapes concrètes à réaliser.

II- Promenade cognitive

1. Spécification des utilisateurs visés et du système à réaliser sous forme d'enchaînement d'écrans.
2. Évaluation a priori par des experts en présence du concepteur.
3. L'évaluateur se « promène » à travers les écrans en simulant la réalisation de la tâche suivant un scénario crédible. Il évalue :
   - si l'action apparait de manière évidente à l'utilisateur.
   - si l'utilisateur percevrait aisément que l'action à réaliser est disponible.
   - si l'utilisateur pourra voir le résultat de son action et l'interprétera correctement.
4. Revue critique de l'évaluation avec le concepteur.
5. Document de synthèse.

III- Evaluation participative

- **Objectifs**
  - ➢ Réservée aux experts, technique issue du marketing
  - ➢ Utilisée avant tout en conception initiale pour faciliter la créativité mais peut aussi concerner l'évaluation
- **Protocole**

  1) **Lancement** - Un évaluateur lance le thème qui doit être exploré (Ex : rendre notre site plus visibles)

  2) **Production d'idées** – chaque participant lance à tour de rôle des idées qui sont notées les unes après les autres, sans tentative de regroupement et sans qu'un participant ne critique les propositions des autres. Idées visibles de tous (tableau par exemple)

3) **Synthèse** : on revient ensuite sur les idées proposées.

Brainstorming vs. Focus Group
Focus Group plus centré sur la résolution d'un problème précis.

B) Evaluation expérimentale OBJECTIVE

a. **Principe : observation de sessions d'utilisation** (Avec ou sans participation visible de l'expérimentateur)
b. **Intérêt** : conditions d'usages aussi réelles que possible.
   1. Session d'utilisation du système par un sujet suivant une tâche ou un scénario clairement défini.
   2. Observation/enregistrement de la session et dépouillement des données.
   3. Analyse des données et rédaction d'un rapport d'évaluation.
c. **Différentes analyses**
   - **Évaluation qualitative** : protocole verbal (*thinking aloud*), questionnaires.
   - **Évaluation quantitative** : métriques par observation ou fichiers de logs.

d. **Limitations**
   - Évalue le plus souvent la première prise en main du logiciel : pas de suivi de l'apprentissage au cours du temps.
   - Ne permet pas une couverture large des fonctionnalités.

e. **Où évaluer (tests avec utilisateurs) ?**
   i. **Sur le terrain**

Analyse des données moins évidentes, mais permet de détecter des problèmes auquel on n'aurait pas pensé.

   ii. **En laboratoire d'utilisabilité**

Etude plus contrôlée, récupérer des données objectives plus facilement analysables.

f. **Déroulement d'une session d'évaluation : script d'évaluation**

Rédiger un **script d'évaluation** qui sera lu au participant permet de nous assurer que le déroulement des sessions est rigoureusement identique (du point de vue de l'utilisateur) d'un sujet à un autre.

1. **Préparation** de la session d'évaluation (Réception du sujet, Acceptation, Mise en œuvre)
2. **Session d'expérimentation** (Exécution d'une tache précise, Recueil de données et observations)
3. **Discussion post-session** (facultatif) (questionnaire âpres la session, analyse et visionnage sur le sujet déroulé)

g. **Quoi : quelles tâches tester ?**
   - **Type d'évaluation coûteuse** : impossibilité de tester toutes les fonctionnalités.
   - **Focalisation** sur quelques tâches, souvent mises en avant par les spécifications

h. **Tâches prédéfinies : intérêt et limites**
   - **Contrôle de l'évaluation** : facilite l'analyse des résultats.
   - Ne peut vérifier que des situations déjà anticipées, alors qu'on ne peut prévoir le comportement des utilisateurs

Solutions de compromis : définir des tâches générales, ou laisser l'utilisateur suggérer une tâche.

i. **Observables**
   - Postures, direction du regard, productions orales.
   - Actions informatiques, chemin suivi sur un site WWW (fichiers de **logs**)

Un observable particulier : les erreurs
   - Essentielles et révélatrices des intentions de l'utilisateur.
   - Difficiles à observer : l'utilisateur veut généralement les cacher (intérêt d'une capture vidéo / audio)

- Difficiles à analyser : une analyse avec le sujet permet comprendre ce qu'il a pensé au moment de l'erreur.

j. **Méthodologie de recueil des observables**
- Observation : papier-crayon, analyse de vidéos…
- Recueil manuel d'observation
- **Tâche lourde** : prévoir un minimum de soutien matériel, par exemple à l'aide de feuilles pré-remplies.

k. **Dépouillement**
- **Evaluation qualitative** – Problèmes les plus flagrants : cas exemplaires
- **Evaluation quantitative** – Calcul de mesures synthétiques à partir des données d'observation.

Ex: Temps d'exécution d'une tâche, % de la tâche exécutée complètement, temps perdu sur des erreurs.

l. **Principe : opinion post-utilisation**
1. Session d'utilisation, généralement suivant une tâche ou un scénario clairement définis.
2. Recueil post-session de l'avis des sujets : questionnaire, interview ou visionnage de la vidéo de la session d'utilisation.
3. Synthèse qualitative ou quantitative suivant les données recueillies : peut donc faire partie intégrante d'une évaluation expérimentale objective.

B) Evaluation expérimentale SUBJECTIVE

a. **Principe : opinion post-utilisation**
1. Session d'utilisation, généralement suivant une tâche ou un scénario clairement définis.
2. Recueil post-session de l'avis des sujets : questionnaire, interview, visionnage de la vidéo/session d'utilisation.
3. Synthèse qualitative ou quantitative suivant les données recueillies : peut donc faire partie intégrante d'une évaluation expérimentale **objective**.

b. **Intérêt**
- Point de vue de l'utilisateur + explications sur les difficultés rencontrées.
- Evaluation moins contrôlée et plus aisée à mettre en œuvre : plus couvrante.
- **Problème** : données subjectives parfois différentes à analyser.

c. **Différentes techniques**

i. **Interview (évaluation quantitative)**
1. Libre - Le sujet aborde des points qui l'ont marqué et qui n'avaient pas retenu l'attention du concepteur.
2. Dirigé - Questions ouvertes/fermées testant des points précis : la liste des questions doit être prédéfinie.

ii. **Questionnaires**: échelles de valeurs sur des points précis **(évaluation quantitatif)**
- Évaluation subjective : échelle multivaluée.
- Synthèse finale : analyse statistique de données.

Exemple sur les questionnaires :
Évaluez de 1 (pas du tout) à 4 (tout à fait) votre accord sur :
- Cette fonctionnalité est intéressante □1 □ 2 □ 3 □ 4
- Il est facile de réserver avec le système □1 □ 2 □ 3 □ 4
- Le temps de réservation est acceptable □1 □ 2 □ 3 □ 4

d. **Construire son propre questionnaire : méthodologie**
- **Importance de l'ordre des questions**

Questions (générales et simples) ou (précises et complexes) ou dépendant du profil utilisateur.
- **Importance de la rédaction des questions**, de l'ordre des réponses
- **Importance de l'échelle de notation** – plusieurs échelles envisageables (Oui, Non, …)

ii. **Modèles prédictifs**

Modèles psycho-cognitifs permettant de simuler les conséquences d'un choix de conception (temps d'exécution…)