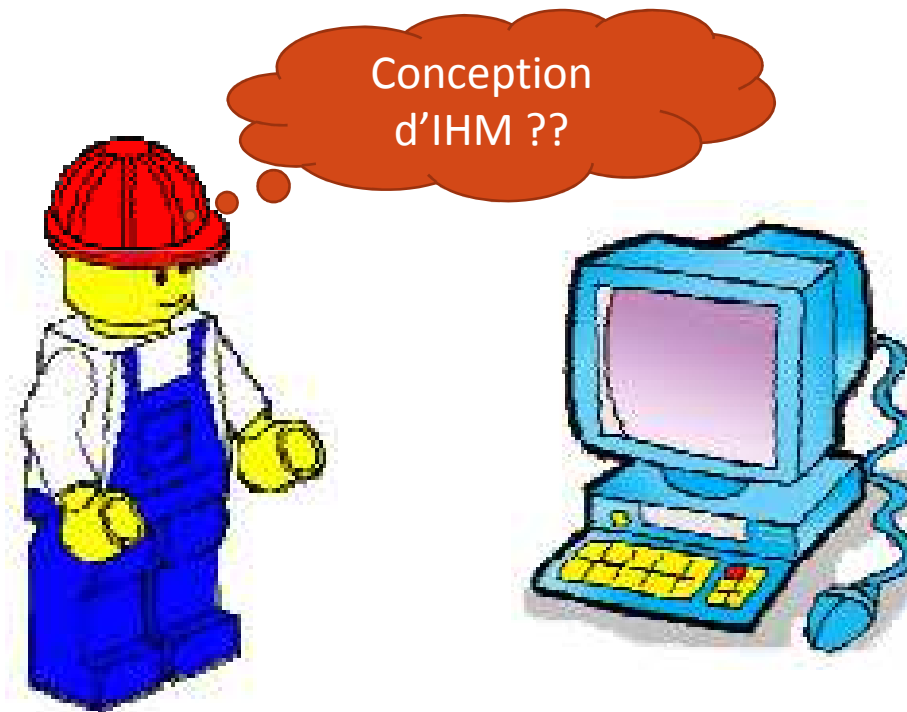


Chapitre VI

Méthodologies pour la Conception de Systèmes Interactifs



Plan du cours

- I. Introduction
- II. Modèles de Conception en Génie Logiciel
- III. Modèle de Seeheim
- IV. Etapes de Conception d'une IHM
- V. Méthodes de Conception d'une IHM
- VI. Norme ISO 13407 99
- VII. Techniques de Recueil d'Informations
- VIII. Conclusion

Une réflexion pour commencer ...

*J'ai toujours rêvé d'un ordinateur
qui soit aussi facile à utiliser qu'un
téléphone. Mon rêve s'est réalisé :
je ne sais plus comment utiliser
mon téléphone.*



Bjarne Stroustrup
(Concepteur du C++)

I. Introduction

De beaux boutons, des menus et des animations ne suffisent pas à rendre un système utilisable.

L'objectif de ce cours est de vous présenter les différentes méthodes et techniques permettant la conception de systèmes interactifs et d'IHM.

Une IHM facile à utiliser et à apprendre amplifie les sensations positives de succès et de contrôle. Elle contribue à la réduction des coûts et délais de formation tout en réduisant les coûts de maintenance.

Pourquoi une méthode de conception ?

- 97% des applications possèdent une interface utilisateurs (Brad A. Myers, 1995),
- Environ 70% des coûts d'un logiciel interactif sont consacrés à la conception de l'interface utilisateur (Bill Buxton, 1991).
- Les risques d'une mauvaise interface sont:
 1. Rejet pur et simple par les utilisateurs,
 2. Coût d'apprentissage (formation),
 3. Coût de maintenance,
 4. Perte de productivité,
 5. Perte de crédibilité (fiabilité),
 6. Utilisation incomplète.

Pourquoi une méthode de conception ?

➤ 1979: Enquête de l'US Government Accounting Office sur les dépenses en logiciel:

- 2% pour des softs livrés et utilisés,
- 25% pour des softs jamais livrés,
- 50% pour des softs livrés mais jamais utilisés.

➤ 1995 : Enquête de Software Engineering Institute:

- Plus de 1/3 des projets d'envergure de développement de logiciels sont annulés,
- En moyenne, un projet dure deux fois plus longtemps que prévu,
- Plus de 3/4 des applications informatiques d'envergure présentent des défaillances opérationnelles et ne fonctionnent pas tel que prévu initialement.

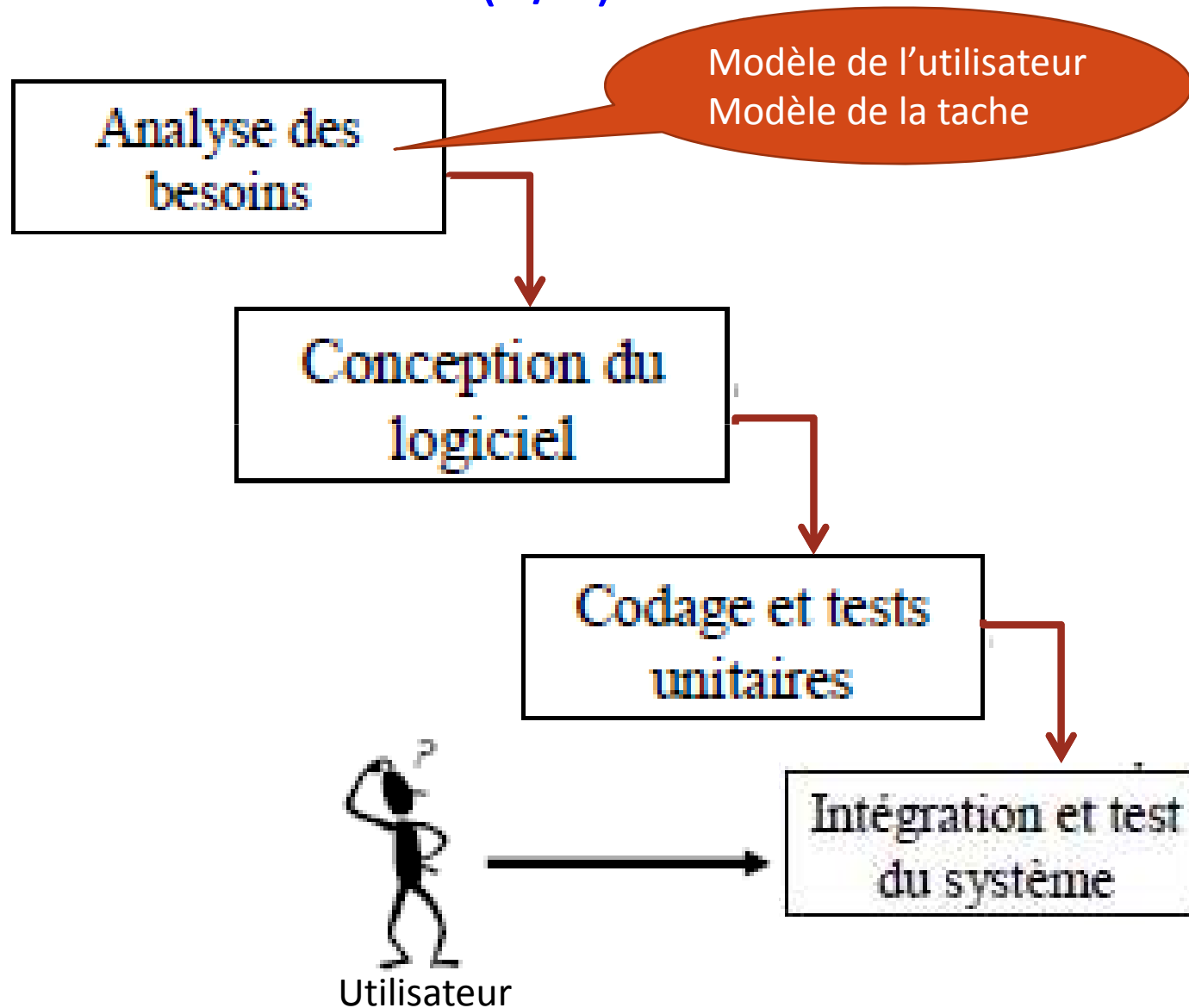
➤ Une solution proposée à l'époque : le Génie Logiciel

II. Modèles du Génie Logiciel

- Modèle en cascade
- Modèle en cascade avec iterations
- Modèle en V
- Modèle en spirale
- Modèle par incrément
- Modèle itératif
- Etc.

1. Modèle en Cascade (Royce 70)

(1/2)



1. Modèle en Cascade (2/2)

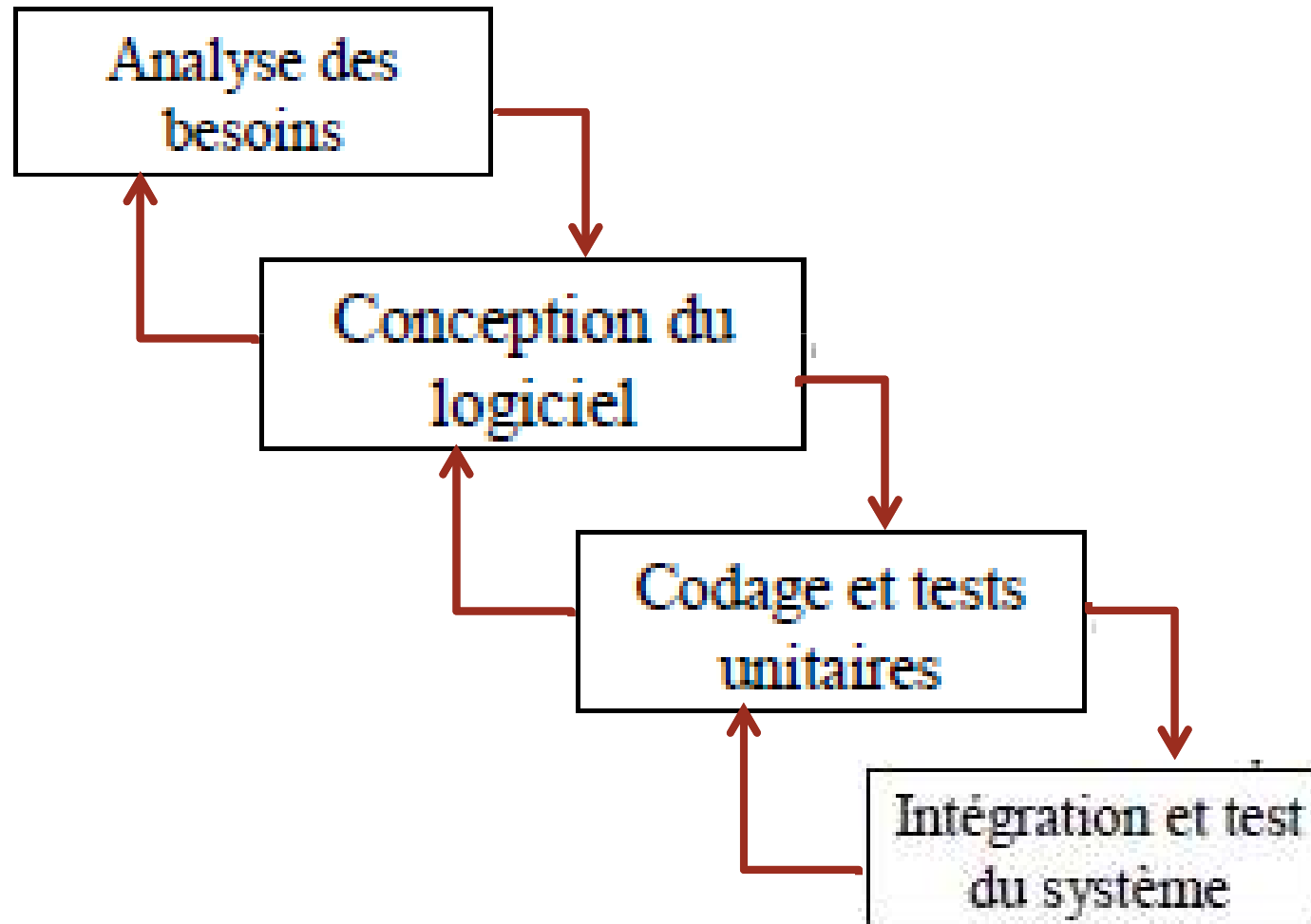
Principe

- Découpage en différentes activités prédéfinies.
- Documents livrés à chaque fin d'étape.
- Chacune de ces phases doit être terminée avant d'entamer la suivante.
- Un avancement séquentiel basé sur le principe de non-retour.

Avantages: Simple, "facile" à mettre en œuvre; idéal pour les projets stables.

Inconvénients: évaluation tardive, pas de retour en arrière, risque de rejet ou d'ajustements bricolés.

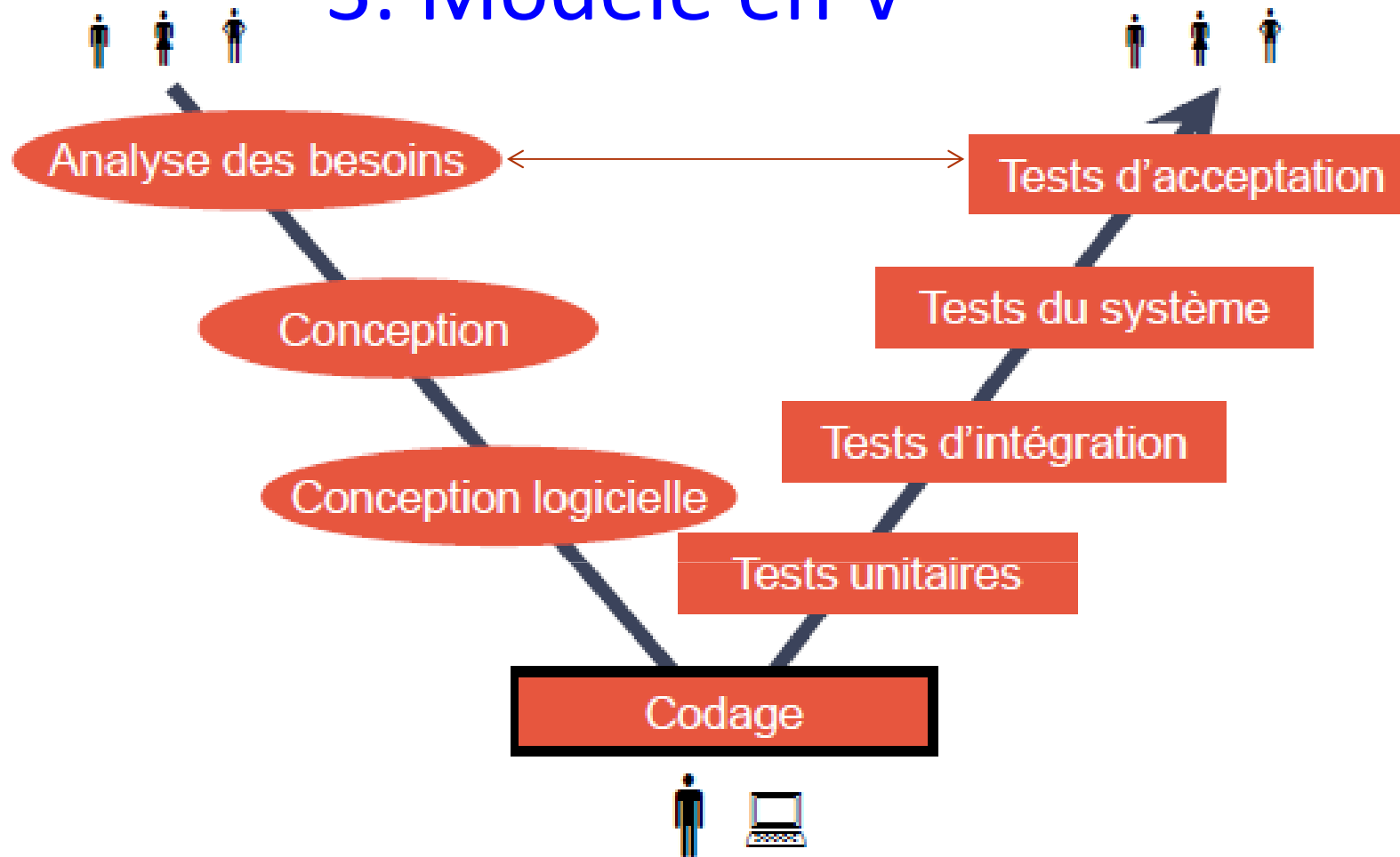
2. Modèle en Cascade avec itérations (1/2)



2. Modèle en Cascade avec itérations (2/2)

- Cycle de vie en cascade itératif
 - Etape suivante uniquement quand une étape est satisfaisante
 - Conception orientée vers l'implantation
 - Evaluation en dernier
 - Prise en compte de l'utilisateur trop implicite
- Modèle créé pour les grands projets
 - Importance des documents (cahier des charges, spécifications)
 - Signés par les clients

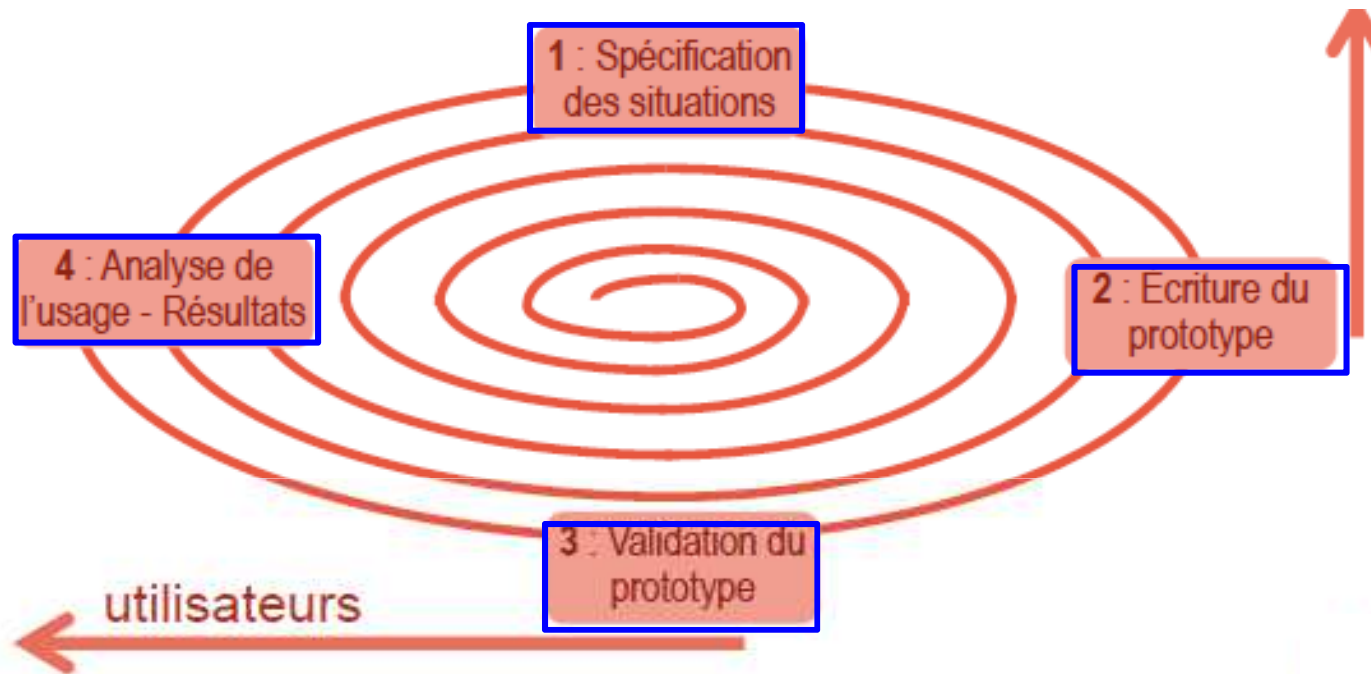
3. Modèle en V



- Devenu un **standard** de l'industrie logicielle depuis les années **1980**.
- Adapté aux projets de taille et de complexité moyenne
- L'évaluation se fait seulement après le codage
- Le modèle ne précise pas la portée des retours arrière
- Implication utilisateur limitée au début et à la fin

4. Modèle en spirale

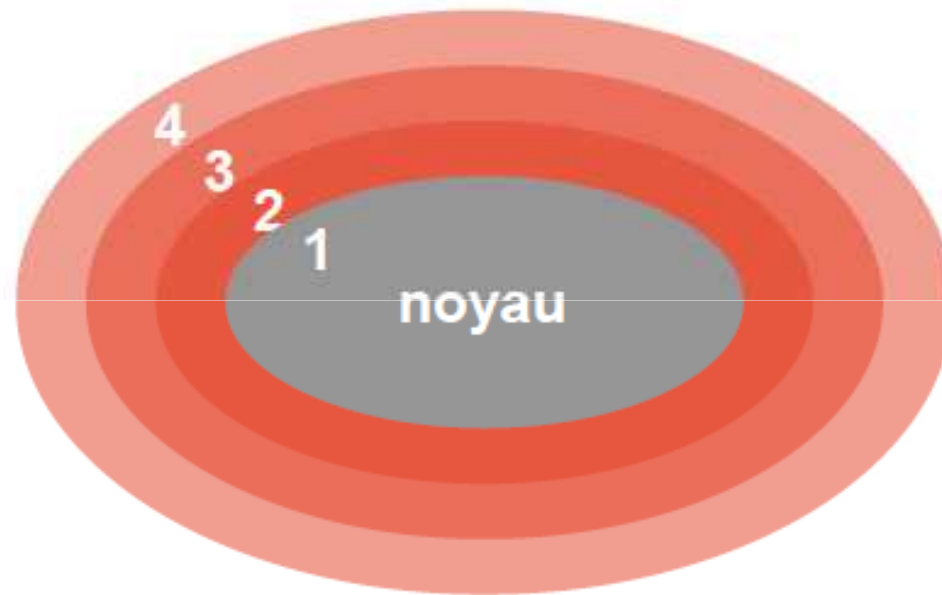
(Barry Boehm en 1988)



- Il propose des prototypes successifs
- Pour chaque cycle le modèle explicite
 - Les objectifs, alternative retenue et contraintes
 - L'analyse et résolution des problèmes
 - Le développement, validation et vérification de la phase
 - La planification de la phase suivante

5. Modèle par incréments

- On développe tout d'abord le noyau
- On ajoute petit à petit des fonctions



➤ Risques:

- Rencontrer un problème pour l'ajout d'un élément
- Remettre en question les éléments précédents
- Voire même le noyau

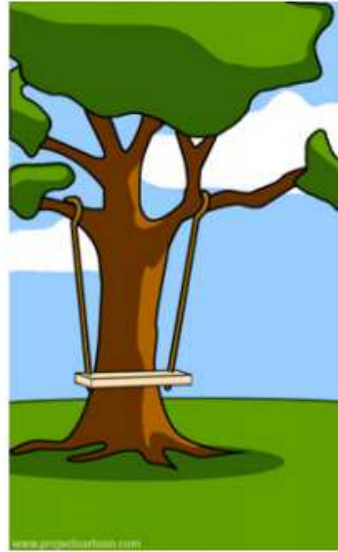
Modèles de Conception en GL: Bilan

- Implication **limitée** des utilisateurs (principalement lors de l'analyse et de l'évaluation),
- Principe **d'indépendance** entre le **noyau** fonctionnel et **l'interface** utilisateur:
 - ❖ interface et interaction ne sont définies qu'après. (dans les logiciels interactifs cette séparation n'est pas si nette),
 - ❖ il est indispensable de prévoir l'usage en même temps que les fonctionnalités,
- Évaluation tardive.

⇒ Méthode de conception spécifique pour les IHM



Ce que le client a expliqué



Ce que le chef de projet a compris



Ce que le designer a dessiné



Ce que le programmeur a écrit



Ce que le client avait vraiment besoin

III. Modèle de Seeheim (1/2)

Seeheim est une ville d'Allemagne où s'est tenu un atelier de travail sur le thème «Systèmes de Gestion d'Interface Utilisateur ».

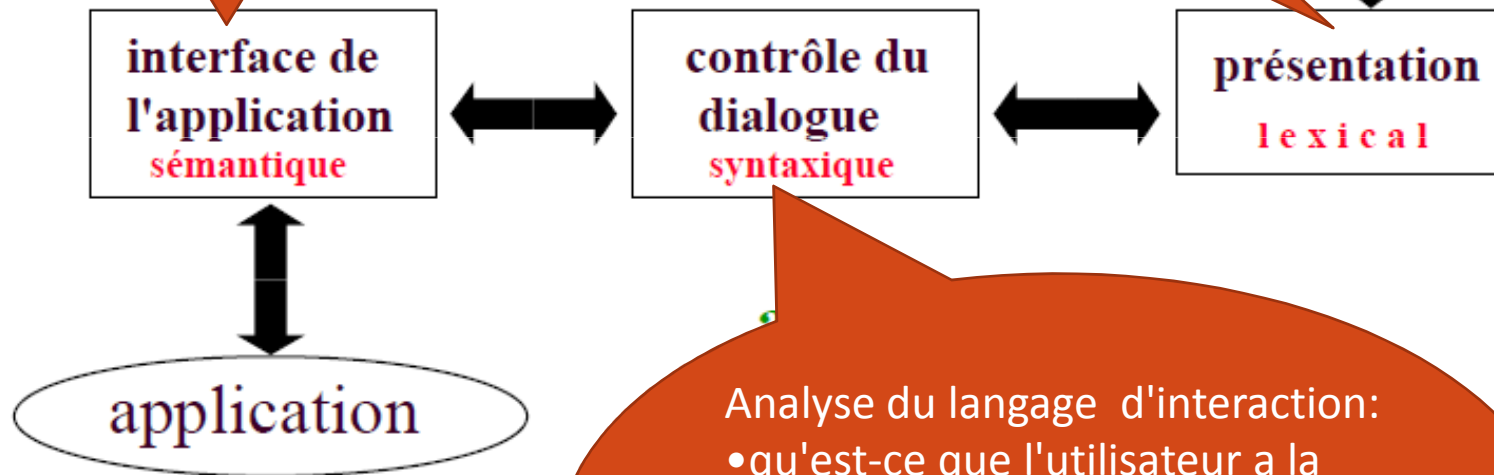
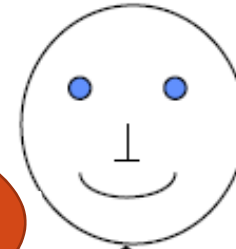
Le **modèle de Seeheim** est un patron d'architecture logicielle introduit en **1983** pour structurer l'interface homme machine dans un logiciel interactif.

Étant donné un **utilisateur** et un **ensemble de fonctions** (qu'on appelle le noyau fonctionnel), le modèle de Seeheim stipule qu'une **IHM** est composée de **trois parties** :

III. Modèle de Seeheim (2/2)

-Les fonctions réalisées par l'application ,
-Les données manipulées par l'application.

Ce que l'utilisateur voit de l'application.



Analyse du langage d'interaction:

- qu'est-ce que l'utilisateur a la possibilité de faire ?
- comment l'utilisateur agit sur la présentation ?
- l'influence de son action sur ce qu'il pourra faire ensuite.

IV. Etapes de Conception d'une IHM

Une méthode de conception IHM se découpe (généralement) en trois phases :

- 1. Analyse:** préciser les attentes et les besoins des utilisateurs, connaître leurs tâches et le contexte;
- 2. Développement:** réaliser toute ou partie d'une interface (sous une forme plus ou moins aboutie);
- 3. Évaluation:**
 - mesurer l'utilisabilité de l'interface réalisée (performance, satisfaction des utilisateurs, facilité d'utilisation),
 - identifier les points à améliorer pour la version suivante, etc.

Utilisateur : plusieurs profils, caractéristiques variées

Tâche: objectif de l'utilisateur (e.g., rechercher un livre)
Répétitive, régulière, occasionnelle, sensible aux modifications de l'environnement, contrainte par le temps, risquée, etc.

Contexte: environnement et contraintes d'utilisation

- Grand public (proposer une prise en main immédiate),
Loisirs (rendre le produit attrayant),
Industrie (augmenter la productivité),
Systèmes critiques (assurer un risque zéro), etc.
- Techniques (e.g., plate-forme, taille mémoire, écran, capteurs, réutilisation d'ancien code)



V. Méthodes de Conception des IHM

Différentes méthodes existantes (inspirées des méthodes de conception GL) :

1. Conception itérative,
2. Conception incrémentale,
3. Conception par prototypage,
4. Conception centrée utilisateur: CCU
5. Conception participative,
6. Conception par personas et scénarios.

1. Conception Itérative

Méthodologie basée sur une **succession de cycles** composés des trois phases (analyse, développement, évaluation).

Dans chaque cycle, la conception doit être élaborée, affinée et testée:

- Travail sur **l'intégralité** de l'interface,
- **Cycles répétés** jusqu'à obtention d'une interface satisfaisante,
- Prise en compte de nouveaux objectifs,
- Prise en compte de l'avis des utilisateurs qui peut changer.

2. Conception Incrementale

Méthodologie basée sur la réalisation d'une **première** partie, puis d'une **seconde**, etc.

- Travail sur une **seule zone** de l'interface jusqu'à satisfaction,
- Développement de solutions partielles, intermédiaires,
- Prise en compte de nouveaux objectifs,
- Prise en compte de l'avis des utilisateurs qui peut changer.

3. Conception par prototypage

de la **maquette**



au prototype

à jeter ou réutilisable



vers le **produit final**

Le prototypage permet :

- Aux concepteurs de travailler sur plusieurs ensembles de détails à la fois,
- Aux utilisateurs de voir ce que sera le système final,
- De se concentrer sur les parties problématiques de l'interface,
- D'étudier des alternatives de conception,
- De s'assurer de l'utilisabilité du système.

1. Maquette

Définition

- ❖ Ensemble d'objets graphiques donnant une image de l'écran-utilisateur,
- ❖ Support de communication entre les concepteurs (phase initiale),
- ❖ Ne contient aucun accès aux données, aucun calcul.

Intérêts

- ❖ Représentation accessible aux non informaticiens,
- ❖ Eviter de détecter des anomalies trop tard.

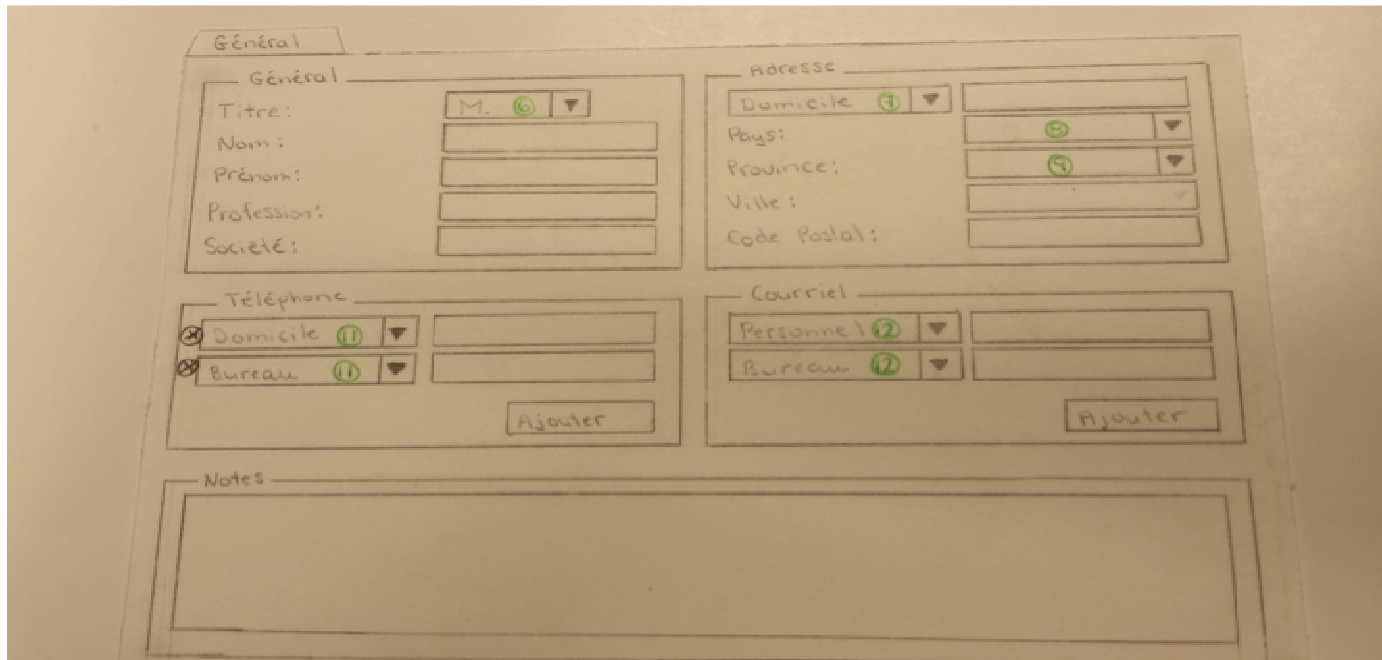
2. Prototype

Définition

- ❖ Représentation réduite d'une partie du système permettant une démonstration,
- ❖ Développement complet de l'interface pour certaines tâches,
- ❖ Fonctionnalités complètes (calculs, accès aux données).

Intérêts

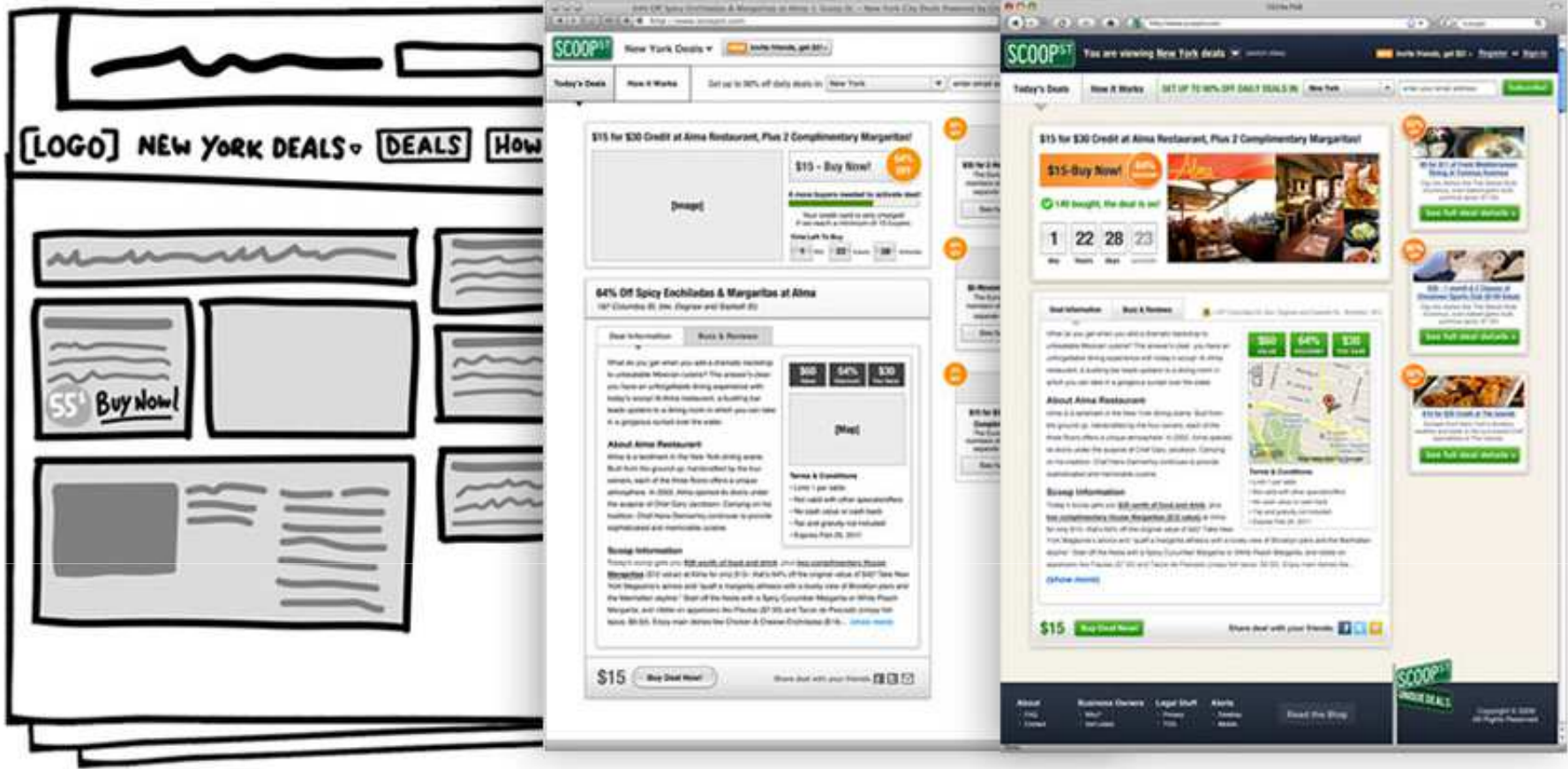
- ❖ Communiquer avec les clients,
- ❖ Vérifier la faisabilité technique,
- ❌ Valider une solution technique,
- ❖ Mesurer un temps de réponse.



Maquette papier.

Outils d'aide au prototypage :

- ❖ Papier, post-its
- ❖ Transparents, vidéo (e.g., Libre Office Impress)
- ❖ Logiciels de maquettage
 - haute fidélité, i.e., avec interactions (e.g., Invision, Maqetta)
 - basse fidélité, i.e., seulement des liens entre écrans (e.g., Mocking Bird, Pencil, Balsamiq)
- ❖ Logiciels de développement (e.g., frameworks web, Netbeans, Visual Studio)



Croquis, maquette et prototype d'une page web.

Lien qui regroupe plusieurs outils de prototypage:

<https://dzone.com/articles/12-kick-ass-software-prototyping-and-mockup-tools>

4. Conception Centrée Utilisateur (CCU)

- Terme inventé par D. Norman en 1986
- **Prise en compte des utilisateurs**
 - ❖ dès la phase d'analyse,
 - ❖ étude de l'utilisateur et de sa tâche.
- **Spécification des caractéristiques**
 - ❖ de l'utilisateur (Modèle de l'utilisateur),
 - ❖ de la tâche à réaliser (Modèle de la tâche),
 - ❖ de l'interaction (Modèle de l'interaction).
- **Relations concepteur–utilisateur**
 - ❖ utilisateur observé dans la résolution de sa tâche,
 - ❖ utilisateur interrogé sur ses attentes,
 - ❖ utilisateur questionné sur le logiciel conçu.

4. Conception Centrée Utilisateur (suite)

➤ Avantages

- ❖ prise en compte de l'utilisateur avant la phase d'évaluation,
- ❖ l'utilisateur est l'évaluateur.

➤ Difficultés

- ❖ Choix des utilisateurs représentatifs et disponibles,
- ❖ Contexte réel d'utilisation,
- ❖ Expliciter les comportements, les connaissances mises en jeu, etc.

➤ Techniques de recueil d'information auprès des utilisateurs

- ❖ Observation directe, entretiens, questionnaires, brainstorming, analyse des traces, etc.

Exemples d'apport de la CCU

- **Site web IBM:** 1 semaine après son « redesign »
aide: -84%, et vente: +400%
- **Vente en-ligne :** 39% d'échecs à cause du design
- **Production de logiciel :** 84% d'échecs (temps, fonctions, budgets)
- **63% d'excès du budget**
 - Demande de changements par utilisateurs, tâches trop grandes, manque de compréhension des propres requis des utilisateurs, mauvaise communication entre utilisateurs et concepteurs.
- **Maintenance:**
 - 33% de bugging, 67% changements par utilisateurs,
 - 40 à 100 fois plus cher de changer l'interface plutôt que de le prévoir pendant la conception.

4.1. Modèle de l'utilisateur

Identifier les caractéristiques pertinentes de l'utilisateur:

➤ Données générales

- ❖ taille, âge, sexe, handicaps, etc.
- ❖ niveau de formation, habitudes culturelles (format des dates, sens d'écriture), etc.

➤ Données liées à l'application

- ❖ compétences sur le domaine,
- ❖ compétences en informatique et sur le système (débutant, expert, usage occasionnel, quotidien)

4.2. Modèle de la tâche

□ Définitions

➤ Tâche

- ❖ but: ce qui doit être fait
- ❖ procédure:
 - ❖ un ensemble de sous-tâches
 - ❖ liées par des relation de composition
 - ❖ liées par des relations temporelles

➤ Tâche élémentaire

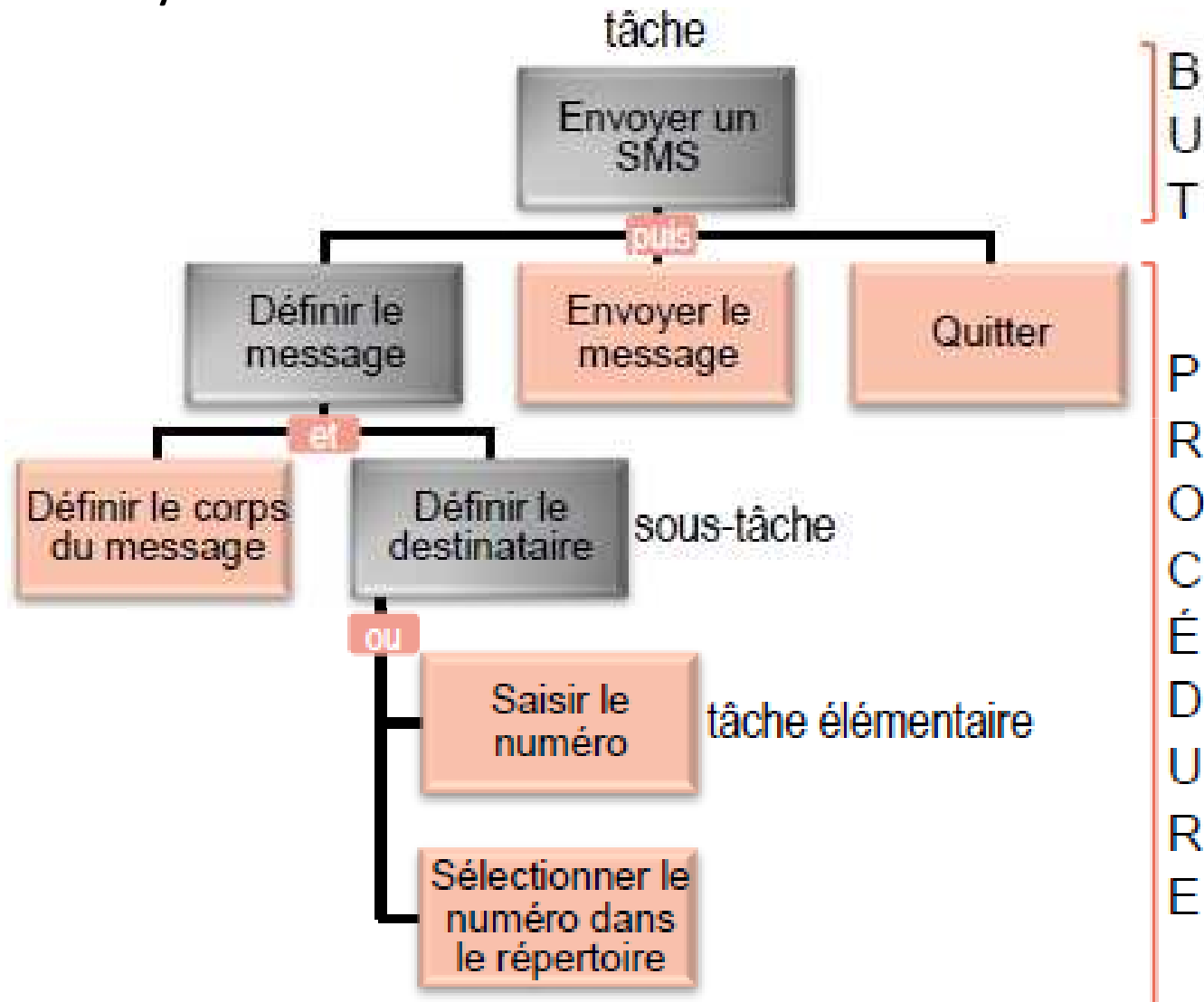
- ❖ tâche décomposable uniquement en actions physiques op. d'E/S

□ Méthode

- ❖ construire la hiérarchie des tâches du système,
- ❖ spécifier chaque tâche, penser aux exceptions,
- ❖ évaluer la décomposition avec l'utilisateur.

4.2. Modèle de la tâche (suite)

Exemple: envoyer un sms



4.3. Modèle de l'interaction

- Établir une **correspondance** directe entre:
 1. les objets conceptuels informatiques
ex : un fichier
 2. et les objets d'interaction et de présentation
ex : les représentations du fichier à l'écran:
 - fermé : représentation iconique
 - ouvert : représentation du contenules opérations sur le fichier:
 - modification
 - suppression, etc.

- Cette correspondance doit
 - ❖ apparaître comme « naturelle »
 - ❖ s'inscrire dans une cohérence d'ensemble: métaphore

5. Conception Participative (1/3)

Définition

est une méthode centrée utilisateur où l'accent est mis sur le rôle actif des utilisateurs (partenaire de conception à **part entière** et participe aux **choix** de conception **finaux**).

L'utilisateur intervient dans toutes les phases:

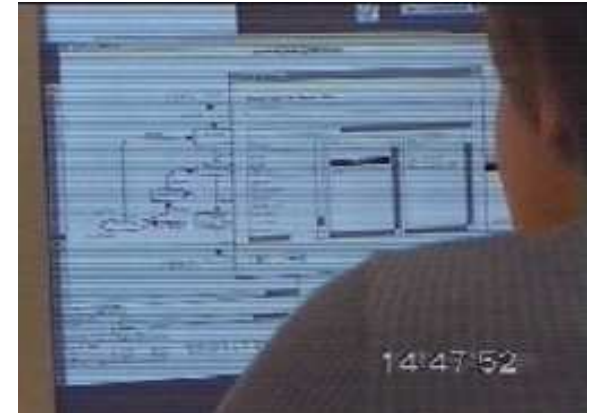
- ❖ Analyse, Conception, Évaluation,
- ❖ Processus itératif.

➤ La conception participative permet de

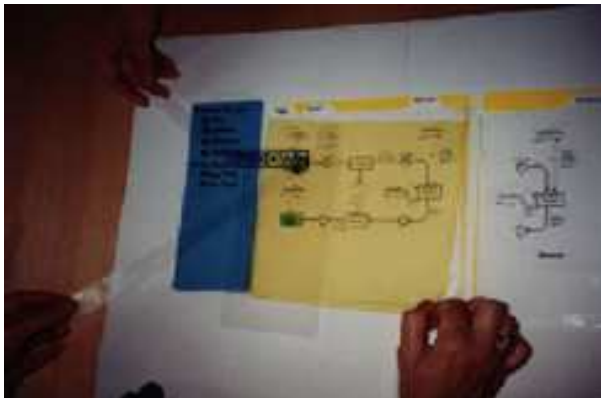
- ❖ Redéfinir le problème et la résolution du problème,
- ❖ Comprendre l'interaction dans un contexte réel,
- ❖ Intégrer le contexte dans le système,
- ❖ Faire communiquer toutes les personnes qui ont des bagages et langages différents.



Evaluation



Brainstorming



Prototypage



Observation

Conception
Participative

Étapes de la conception participative (3/3)

Étape 1 : observation

- Générer des exemples spécifiques de problèmes rencontrés par les utilisateurs,
- Produire une description réaliste du travail actuel d'un utilisateur (Processus, problèmes, apports).

Étape 2 : Brainstorming (oral, écrit, vidéo)

- Générer autant d'idées créatives que possible,
- Créer une description réaliste de l'utilisation du nouveau système.

Étape 3 : Prototypage vidéo *(de préférence)*

- Produire un prototype testable d'une nouvelle interface.

Étape 4 : Evaluation (voir chapitre VI)

- Etude d'utilisabilité.

6. Conception par Personas et Scenarios

Définitions

est une technique de conception centrée utilisateurs, initiée par Alan Cooper en 1999. Cette méthode permet d'offrir une vision commune et partagée des utilisateurs d'un service ou d'un produit, en insistant sur leurs **buts**, leurs **attentes** et leurs **freins** potentiels, et en proposant **un format des plus engageants**.

Persona

est **un utilisateur-type**, une représentation fictive des utilisateurs cibles, qu'on peut utiliser pour fixer des priorités et guider nos décisions de conception d'interface.

- Données générales de l'utilisateur (prénom, photo, éducation, environnement de travail, devise, etc.),
- Ce qui va déclencher ses actions,
- Ce qui peut l'influencer,
- Ce qui peut le freiner ou le faire fuir.

Scenario

Un scénario est une sorte d'histoire avec:

- Un persona,
- Un environnement,
- Un but (que la persona doit accomplir),
- Des obstacles.

Avantages de la conception par personas

- Empathie cognitive (comprendre les états ou croyances d'une autre personne),
- Applicable particulièrement au Web / large échelle.

Inconvénients de la conception par personas

- Mauvaise définition des personas conduit à un échec,
- Distance par rapport aux utilisateurs réels,
- Besoin de modifier les personas en cas de nouveaux résultats ou d'environnement différent.

Vous pouvez créer des personas via ce lien: <https://personagenerator.com/>

Elodie



Persona primaire

Ses attentes

- Travailler au calme.
- Pouvoit par moment travailler avec ses camarades.
- Utiliser son PC portable.
- Accéder au Wifi.
- Brancher son ordinateur.
- Accéder à des revues ou livres sur place.

Ses frustrations

- Pas assez de prises de courant disponibles.
- Mauvaise connexion Wifi à certains endroits de la BU.
- Pas de pause café possible en dehors d'un espace enfumé.
- Peu d'espaces de travail en groupe.

Bio

Elodie est en L3 de Lettres modernes. Elle a d'abord réalisé ses deux premières années universitaires à Nancy, puis s'est inscrite à l'Université de Metz. Certains jours, en fonction de son emploi du temps, elle dispose de 2 à 5 heures de "creux" pendant lesquelles elle aime aller à la BU pour travailler. Il lui arrive même de venir le samedi matin, afin d'éviter de déranger (et d'être dérangée par) les colocataires de son appartement qu'elle loue en centre ville.

Technologies utilisées

- Smartphone
- Ordinateur portable (MacBook)
- Tablette
- Internet

Personnalité

Introvertie	Extrovertie
Conservatrice	Créative
Passive	Active
Non fumeuse	Fumeuse
Isolée	Sociale

Texte de la BU : "La BU est un endroit calme, avec lequel j'aimerais en récupérer le maximum. J'aimerais également pouvoir brancher mon PC, avoir accès à des revues et livres sur place."

Age : 22 ans
Travail : Job étudiant au Mc Do
Situation : Célibataire
Quartier : Metz
Filière : L3 Lettres modernes

Lectrice



Allia : la prof

65 ans, Divorcée 3 enfants
Retraîtée, ancienne professeur de Français, Boulogne

Biographie

Allia a été professeur de Français au lycée de Thiers pendant 40 ans. Plutôt découragée par le niveau d'orthographe et le faible vocabulaire de ses élèves, elle a tout de même continué à transmettre sa passion. Avec un certain succès puisque plusieurs de ses élèves sont devenus des écrivains à succès. Aujourd'hui à la retraite, elle devore des livres à longueur de journée. Elle est souvent déçue mais parfois un auteur ravive sa flamme.

Pratique informatique

- Dialogue avec Skype depuis que son fils lui a installé
- Evite les réseaux sociaux, trop souvent bourrés de fautes et de stupidité

Attente

- Aider de jeunes auteurs à s'améliorer
- Satisfaire sa boulimie livresque
- Dialoguer avec des gens civilisés et cultivés

En conclusion

Allia sera exigeante sur le contenu et la forme du site mais pourra beaucoup s'investir et faire avancer les auteurs.

Sites clefs

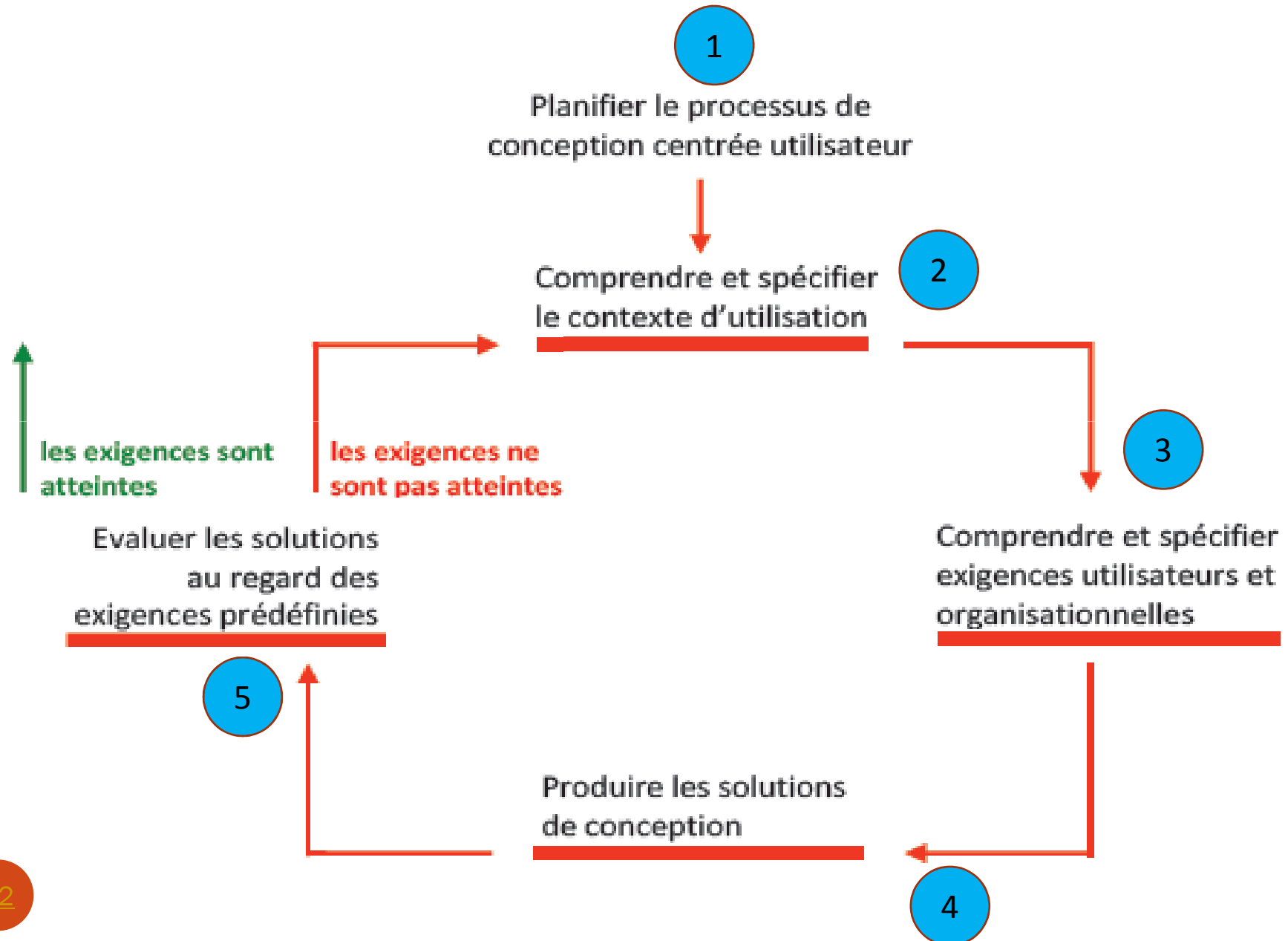
- LeMonde.fr
- Picasa
- Projet Gutenberg (soutien pas utilisation)

N'aime pas

- Les fautes d'orthographe
- Les gens malpolis
- San Antonio

« La littérature française aurait bien besoin d'un petit remontant »

VII. Norme ISO 13407 pour la CCU



VII. Norme ISO 13407 (suite)

1. Planification du processus de CCU

Adapter des outils et méthodes en se basant sur la consultation de **documentation** et des **discussions** autour des pratiques et des contraintes de la conception.

2. Spécification du contexte d'utilisation

- Saisir les caractéristiques, buts et tâches des **utilisateurs** ainsi que leur **environnement** d'utilisation.
- Décrire **l'environnement** des points de vue technique, matériel, social, organisationnel et législatif.



VII. Norme ISO 13407 (suite)

3. Spécification des besoins utilisateurs

Prendre en compte les besoins, les compétences et l'environnement de travail de **tous les intervenants** sur le système.

4. Production des solutions

Utiliser les connaissances acquises lors des étapes précédentes pour **matérialiser les solutions** sous forme de **prototypes**.

5. Evaluation des solutions

Utiliser les prototypes créés pour évaluer les solutions conçues en fonction des exigences.

L'évaluation permet de détecter les défauts.



VIII. Techniques de Recueil d'Informations

Une méthode de conception IHM nécessite la collecte des informations sur les utilisateurs, leurs tâches ou sur les évaluations des interfaces.

Il existe plusieurs techniques qui se distinguent par leurs buts et la procédure suivie:

1. **Scénario de conception**
2. **Inspection cognitive**
3. **Magicien d'Oz**
4. Enquête / Entretien
5. Observations
6. **Focus group**
7. **Tri par cartes**
8. Questionnaire
9. **Remue-méninges** (brainstorming)
10. Conception en parallèle
11. **Audit ergonomique**

VIII. Conclusion

- ✓ Conception centrée utilisateurs,
- ✓ Prendre en compte les **utilisateurs potentiels** dès le début,
- ✓ Créer une **équipe** pluri-disciplinaire,
- ✓ Prévoir des **évaluations** dans chaque phase de la conception.

Questions ...