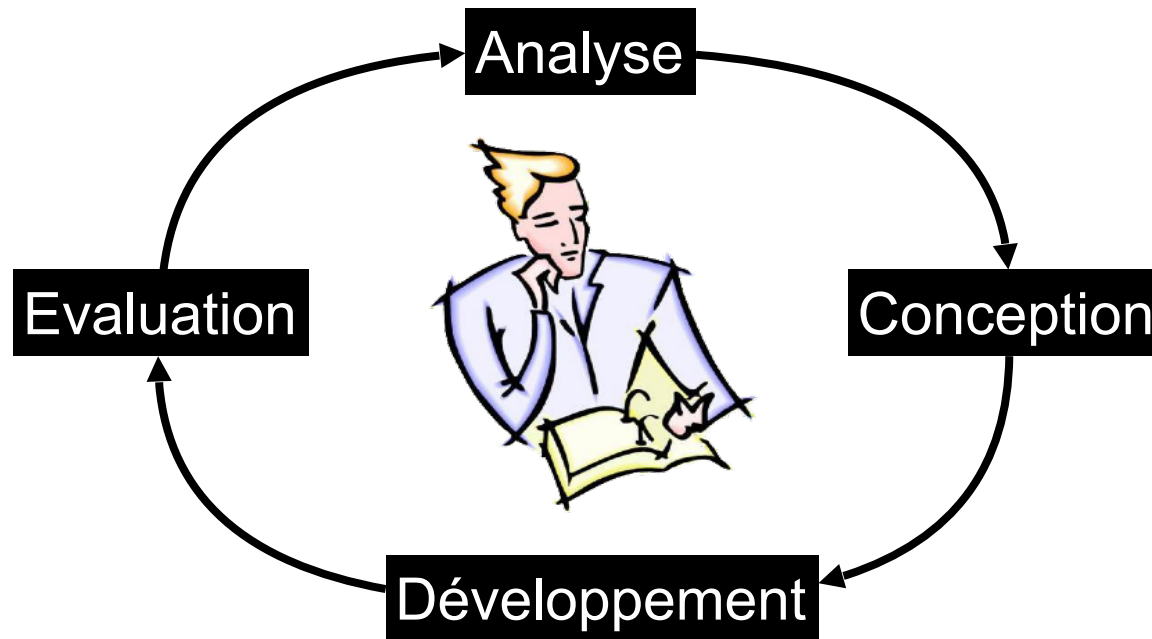


Analyse des besoins Idéation

Michel Beaudouin-Lafon
Université Paris-Saclay
mbl@lisn.fr
<http://ex-situ.lri.fr>

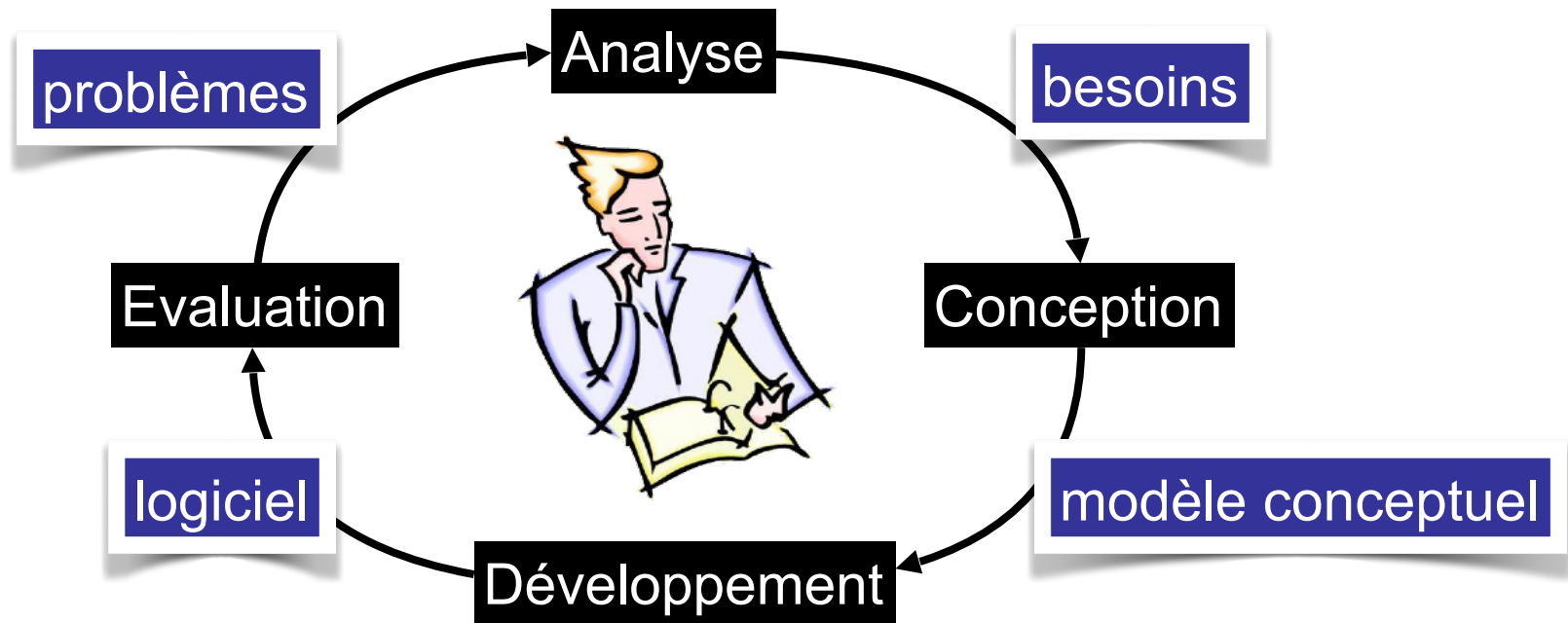
Merci à Wendy Mackay pour ses supports de cours et son expérience

Cycle de conception d'un système interactif



Cycle de conception d'un système interactif

Chaque étape produit des résultats utilisés par l'étape suivante



Conception Centrée sur l'Utilisateur

(UCD ou User-Centered Design)

(UXD ou User eXperience Design)

Se placer du point de vue de l'utilisateur (pas de l'informaticien)

Qui sont les utilisateurs ?

Quelles sont leurs pratiques ?

Quels sont leurs objectifs, leurs besoins ?

Dans quel(s) contexte(s) se situe leur activité ?

Si possible, impliquer les utilisateurs non seulement lors de l'analyse des besoins, mais aussi lors de la conception

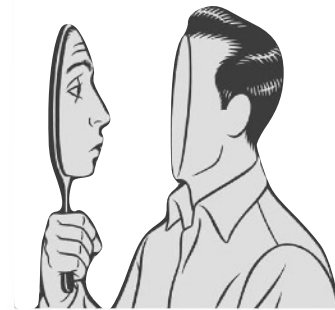
=> *Conception Participative (Participatory Design)*

Analyse des utilisateurs et de leurs besoins

Observation



Introspection



Questionnaires



Interviews



Règle fondamentale

Collecter des **détails concrets**, des « **histoires** » **réelles**

Aller du **spécifique** vers le général

Aller du **concret** vers l'abstrait

Aller des **faits** vers les opinions

Pour cela : commencer par des **questions précises**,
puis aller progressivement vers des questions ouvertes

POURQUOI ?

On a tendance à décrire l'activité telle qu'elle devrait être
et non pas telle qu'elle est

Ce sont souvent des détails (du point de vue de l'utilisateur)
qui révèlent de vrais problèmes



Observation : comment

Le concepteur observe des utilisateurs interagir avec le système

Observation non intrusive (“fly on the wall”)

Observer l’activité normale

Observation intrusive (“think aloud”)

Demander à l’utilisateur d’effectuer une tâche spécifique

Demander d’énoncer à haute voix ce qu’il/elle fait et pourquoi

Enregistrer les sessions

Audio et si possible vidéo

Prise de notes



Observation : quoi

Ce que l'on cherche à observer

Des **actions répétitives**

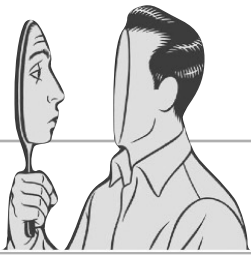
=> pistes pour des opérations à automatiser
ou à rendre plus efficaces

Des **erreurs** ou des **difficultés** de manipulation

=> signes d'un modèle conceptuel mal compris
ou de commandes mal conçues

Des **utilisations inattendues** des fonctionnalités

=> signes d'innovation par les utilisateurs
que l'on peut transformer en nouvelles opérations



Introspection : comment

Le concepteur s'observe en train d'utiliser le système

Choisir des **tâches précises**

Représentatives de l'usage du système
ou des problèmes à étudier

Passer en revue les **fonctions du système**

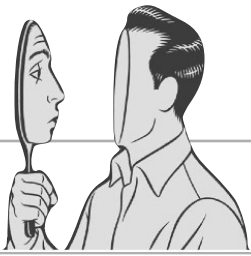
Analyse plus systématique

Enregistrer la session

Audio + vidéo de l'écran

Parler à haute voix

Eventuellement : autre personne pour prendre des notes



Introspection : quoi

Ce que l'on cherche à observer

Analyse détaillée des **actions** à effectuer

=> pour avoir une connaissance précise du système
car souvent, « le démon est dans les détails »

Points **positifs** & points **négatifs**

=> pour garder ce qui marche et corriger ce qui ne marche pas

Surprises éventuelles

Idées spontanées pour améliorer le système

=> pour alimenter la phase de conception

ATTENTION au **biais d'introspection** : vous n'êtes pas l'utilisateur



Questionnaires : comment

Récolter des informations d'un grand nombre d'utilisateurs

Questions **démographiques** (age, profession, ancienneté dans le poste, ...)

Questions **fermées factuelles**

« Combien de messages avez-vous envoyé hier »

Questions **fermées subjectives**

Choix multiples : « J'utilise cette fonction » jamais rarement souvent

Echelle de Likert à 5 degrés : « La fonction de recherche est utile »

pas du tout d'accord plutôt pas d'accord neutre plutôt d'accord tout à fait d'accord

Classement : « Classer ces fonctions par ordre d'importance pour vous »

Questions **ouvertes**

« Quelle nouvelle fonctionnalité vous serait la plus utile ? »

Penser au traitement des résultats, surtout si beaucoup de réponses



Questionnaires : quoi

Ce que l'on cherche à savoir

Tendances **générales** (via les question fermées)

ou bien l'identification de plusieurs types d'utilisateurs

=> avoir une vue d'ensemble de la population d'utilisateurs

Informations plus **spécifiques** (via les questions ouvertes)

=> détails concrets, idées d'améliorations

ATTENTION

Les utilisateurs ne répondent pas toujours honnêtement

Les questions peuvent tomber à côté des vrais problèmes

Les questionnaires sont rarement utiles seuls



Interviews : comment

Comprendre les usages et les besoins des utilisateurs

Préparer une **trame de questions**

Commencer par des **questions précises**, spécifiques

Demander de raconter un **événement récent mémorable**

Montrer comment le système a été utilisé

Poser des **questions plus générales** à la fin

Enregistrer la session

Audio et si possible vidéo

Seconde personne pour prendre des notes



Interviews : quoi

Ce que l'on cherche à savoir

Des **faits précis** qui illustrent les problèmes rencontrés par les utilisateurs et leurs besoins

=> mieux comprendre les problèmes rencontrés, en contexte

Des **usages non standards**

=> innovation par les utilisateurs

ATTENTION

Les utilisateurs ont tendance à généraliser
et à montrer l'usage « normal » et non pas l'usage réel



Analyser et synthétiser les résultats

Analyser les informations collectées en distinguant
Difficultés / Contournements / Innovations
Regrouper ces informations par catégories

Difficultés rencontrées par les utilisateurs
interactions trop complexes,
fonctions manquantes ou inadaptées, ...

Contournements qu'ils mettent en œuvre
solutions qui sont considérées comme « incorrectes »

Innovations
usages non anticipés par les concepteurs

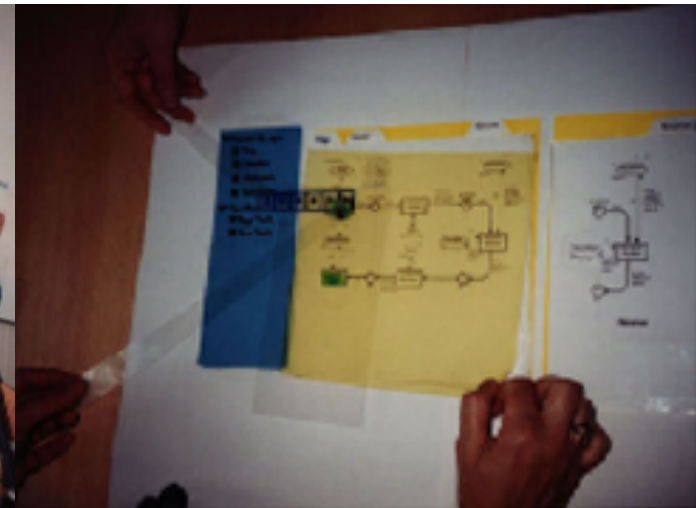
Exemple : StickyLines

Observations, introspection et interviews d'utilisateurs d'un logiciel de dessin de schémas (réseaux de Petri)



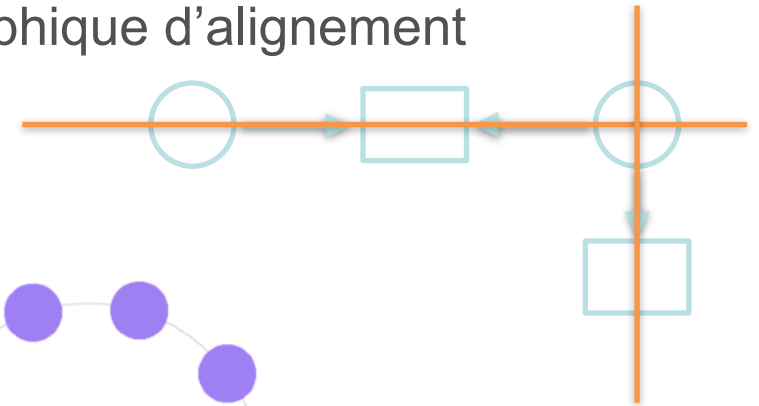
Difficultés rencontrées : aligner des objets et les maintenir alignés

Contournements : grouper / dégrouper / regrouper les objets

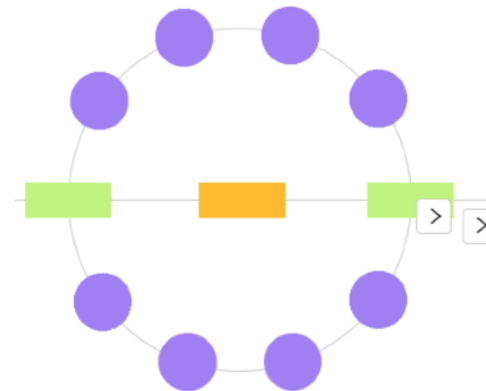


Exemple : StickyLines

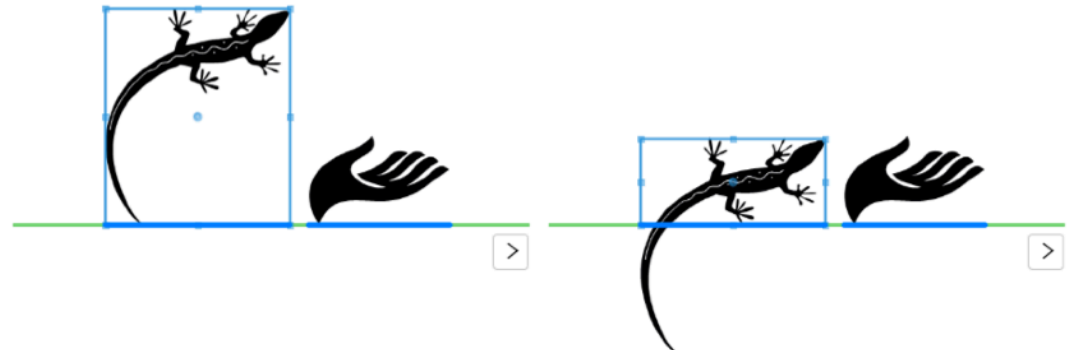
Solution : un guide magnétique = objet graphique d'alignement



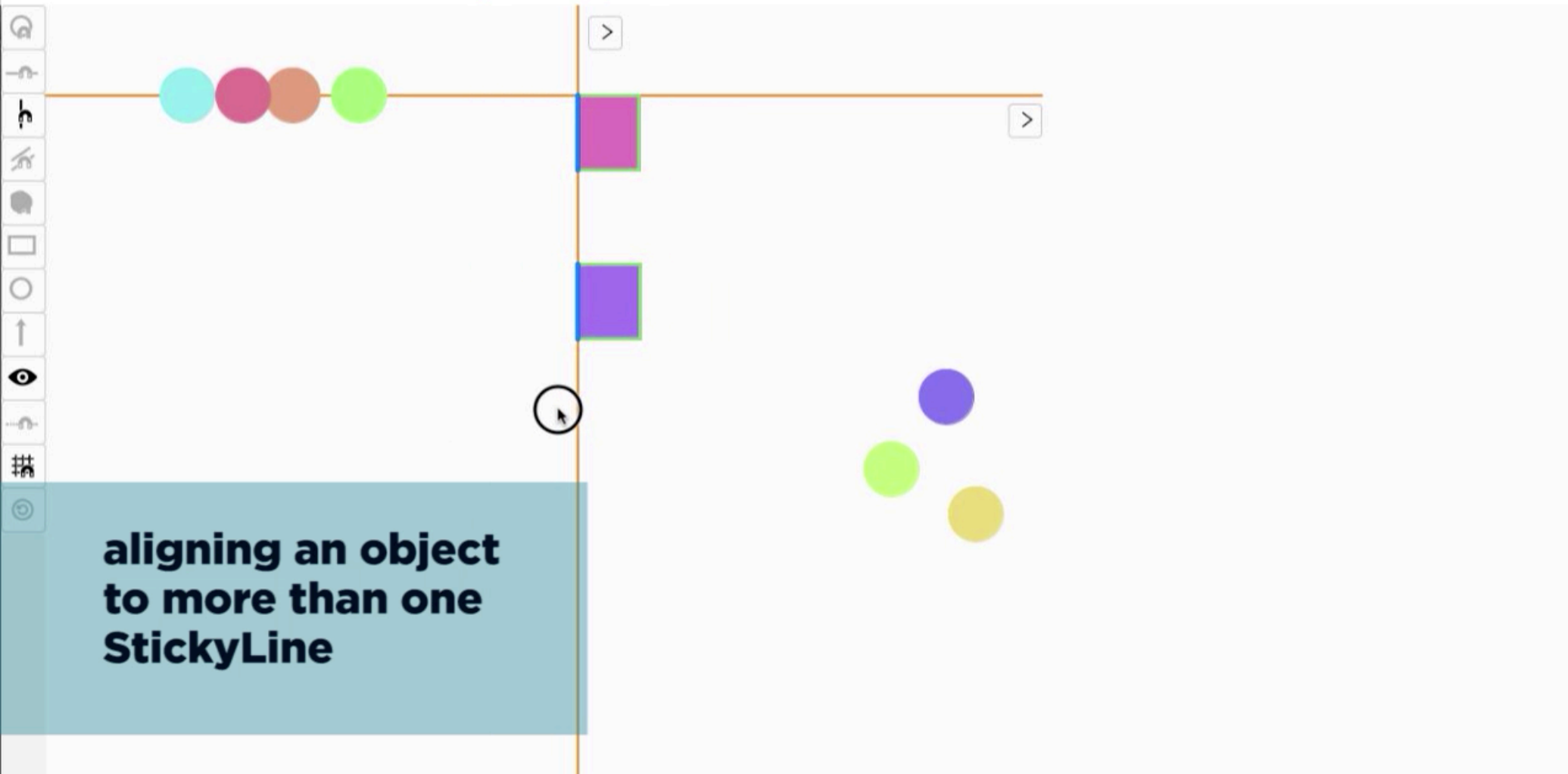
Généralisation : Alignement circulaire



Innovation : Ajustement

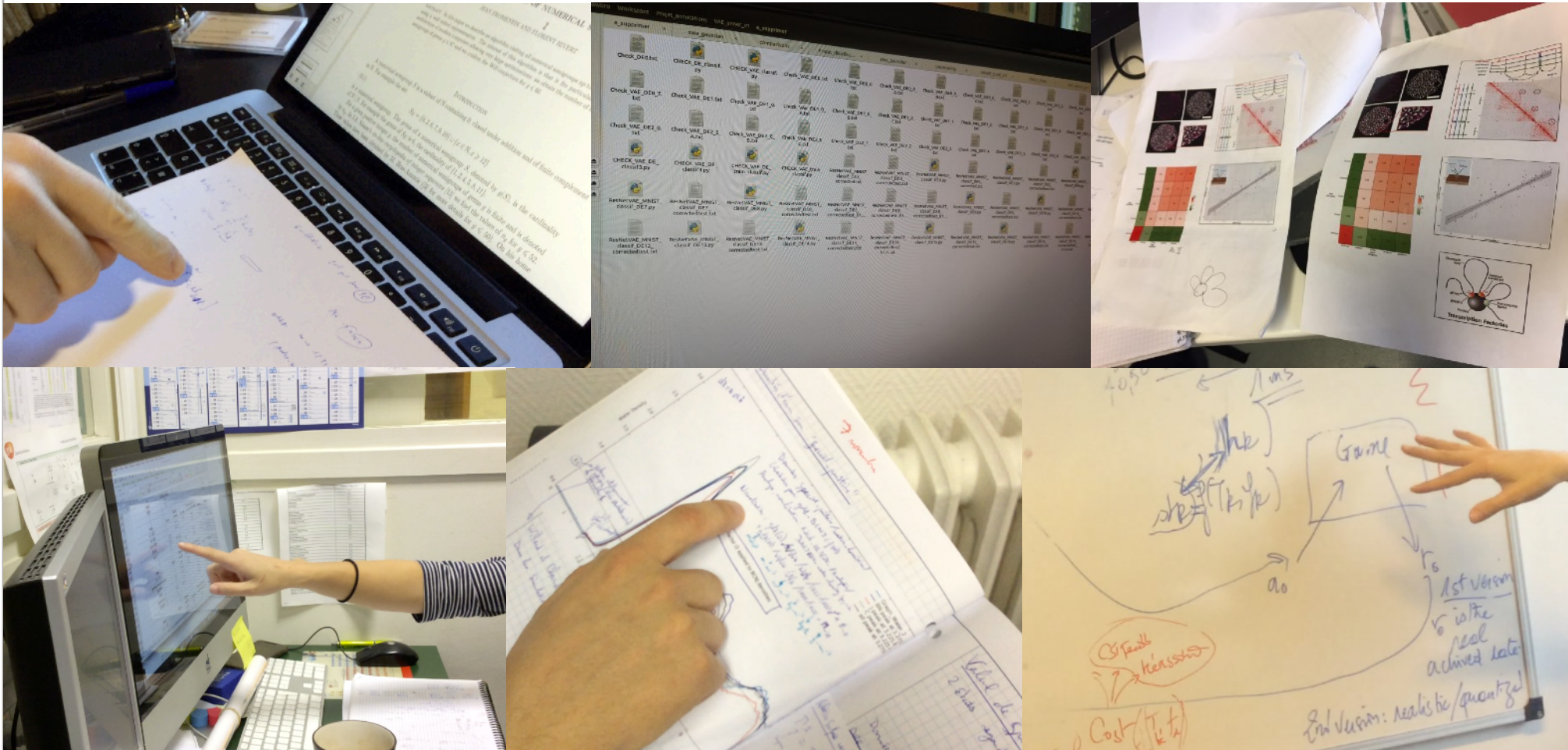


Exemple : StickyLines



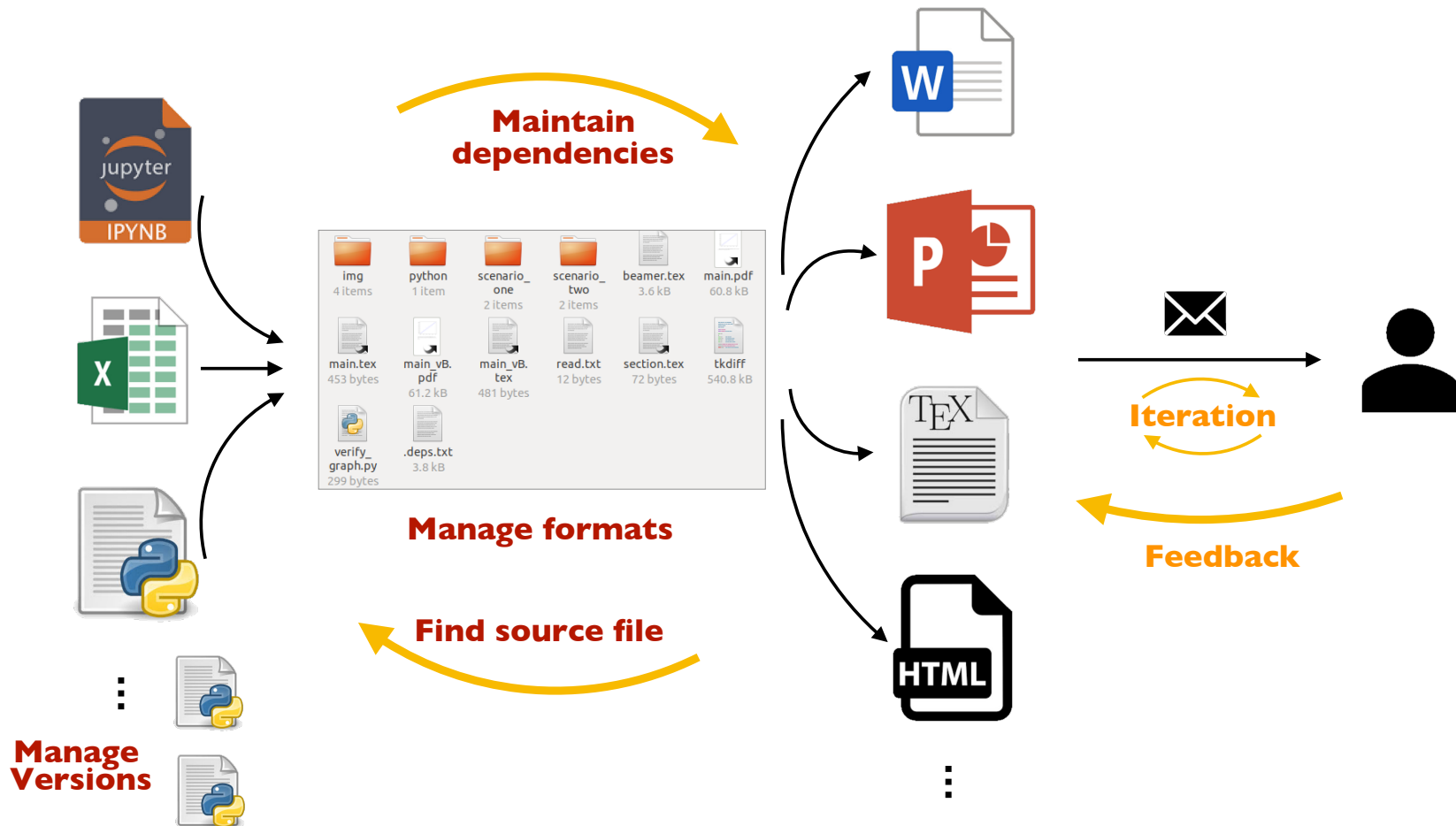
Exemple : FileWeaver

Interviews de chercheurs pour comprendre comment ils gèrent leurs données et leurs documents



Exemple : FileWeaver

Difficultés rencontrées : garder trace des dépendances

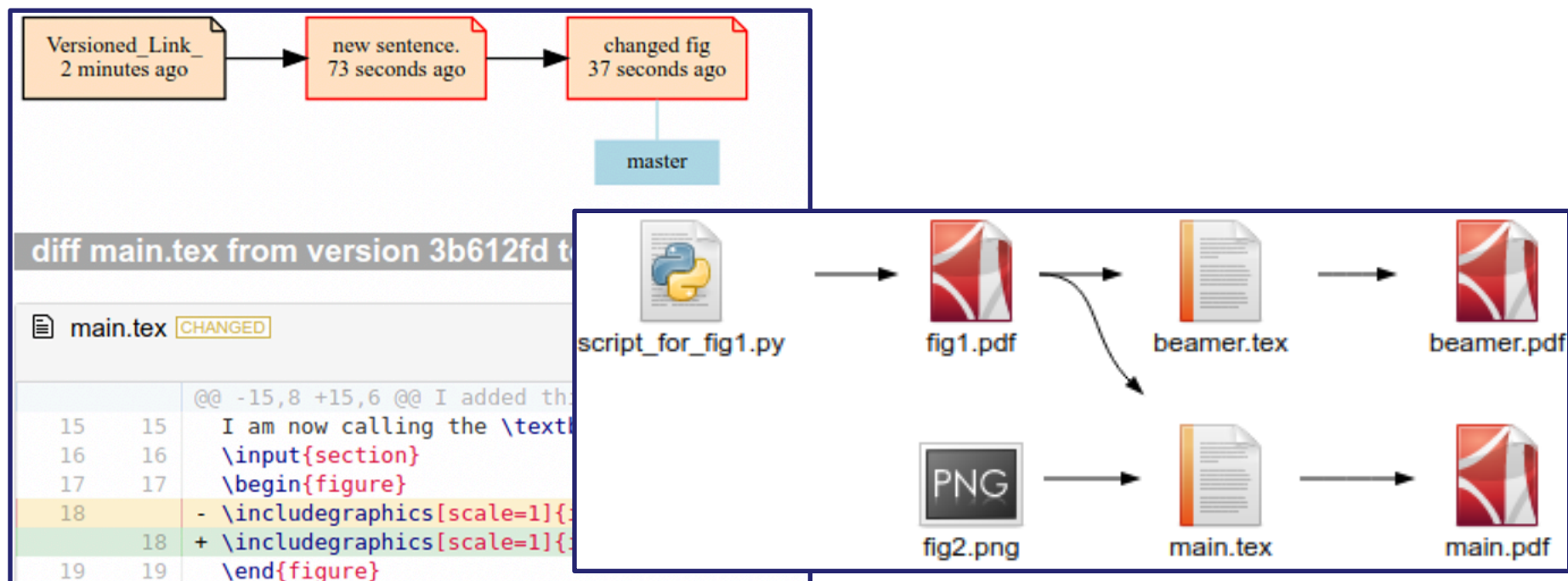


Exemple : FileWeaver

Solution :

Automatiser la découverte des dépendances
et la mise à jour des fichiers

Garder trace de l'historique des modifications



Exemple : FileWeaver

The image shows a file explorer window on the left and a FileWeaver Graph window on the right. The file explorer displays a directory named 'missing_link_testing_area' containing various files and folders. The FileWeaver Graph window shows a dependency graph for Beamer presentations.

File Explorer Contents:

- img (2 items)
- python (1 item)
- scenario_one (2 items)
- scenario_two (2 items)
- beamer.pdf (30.6 kB)
- beamer.tex (3.5 kB)
- main.pdf (60.8 kB)
- main.tex (453 bytes)
- read.txt (12 bytes)
- section.tex (72 bytes)
- tkdiff (540.8 kB)
- verify_graph.py (299 bytes)
- .deps.txt (18.7 kB)

FileWeaver Graph:

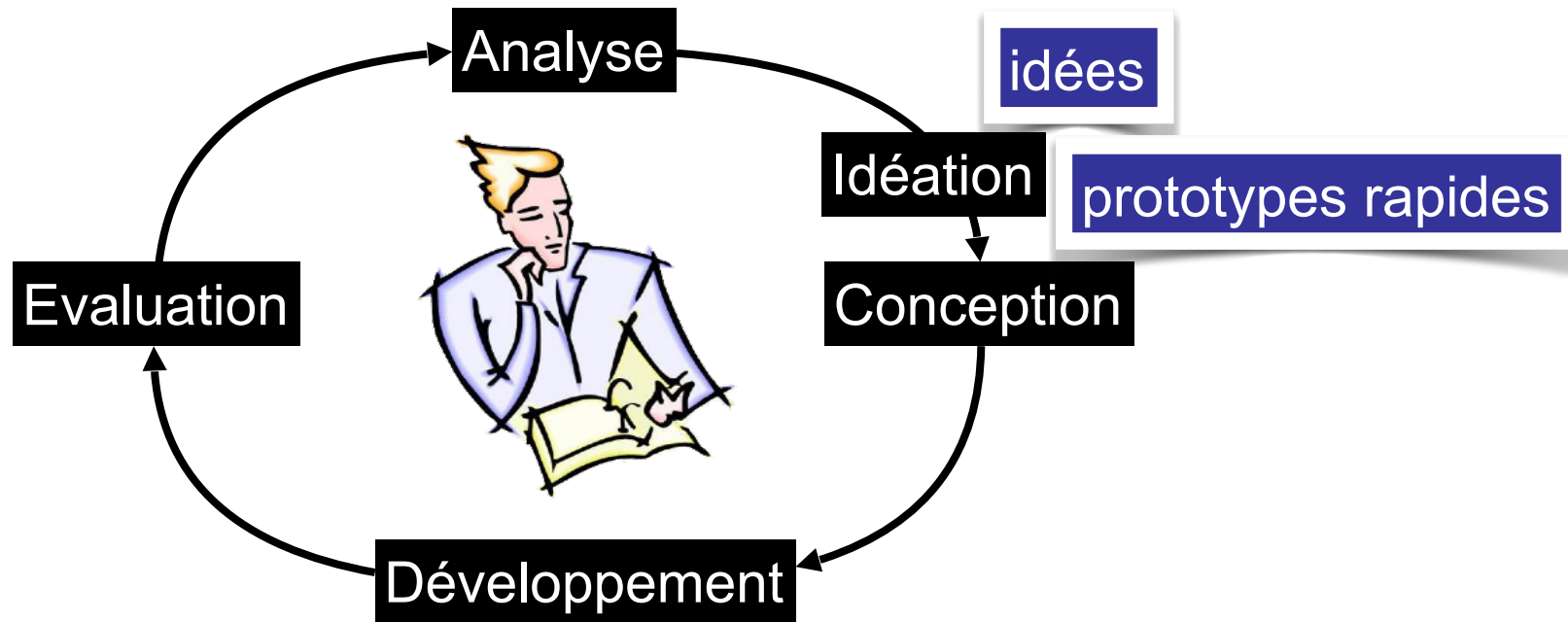
The graph illustrates the dependencies between files:

- Input Files:** fig2.jpg (JPG), fig2.png (PNG), fig2.svg (SVG), and script_for_fig1.py (Python).
- Intermediate Files:** fig1.pdf (generated from script_for_fig1.py), fig2.pdf (generated from fig2.jpg, fig2.png, and fig2.svg), and main.tex (generated from section.tex and fig2.pdf).
- Output Files:** beamer.pdf (generated from beamer.tex and fig1.pdf), and main.pdf (generated from main.tex).

```
graph LR;
  fig2_jpg[fig2.jpg] --> fig2_pdf[fig2.pdf];
  fig2_png[fig2.png] --> fig2_pdf;
  fig2_svg[fig2.svg] --> fig2_pdf;
  section_tex[section.tex] --> main_tex[main.tex];
  fig2_pdf --> main_tex;
  script_for_fig1_py[script_for_fig1.py] --> fig1_pdf[fig1.pdf];
  fig1_pdf --> beamer_tex[beamer.tex];
  beamer_tex --> beamer_pdf[beamer.pdf];
  main_tex --> main_pdf[main.pdf];
```

Cycle de conception d'un système interactif

Chaque étape produit des résultats utilisés par l'étape suivante





De l'analyse à la conception : l'idéation

Idéation : le processus créatif de production, développement, et communication de nouvelles idées [wikipedia]

Identifier le(s) problème(s) à résoudre

Générer des idées de solution

=> Brainstorming (remue-méninges)

Choisir les idées à explorer

=> Prototypage rapide

Se concentrer sur l'interaction en contexte

Se mettre à la place de l'utilisateur



Brainstorming (remue-méninges)

Phase 1

Générer le maximum d'idées

Tout le monde doit participer

Toutes les idées doivent être notées

Tout le monde doit proposer au moins une idée qu'il juge stupide

Phase 2

Relire les idées

Chacun peut voter pour ses 3 idées préférées

Classer les idées par nombre de votes

Discuter ces idées par rapport au problème traité



Brainstorming (remue-méninges)

NE PAS FAIRE

Discuter les idées

Critiquer les idées

Expliquer pourquoi une idée est
bonne / mauvaise

Ignorer les idées des autres

Changer de sujet

Proposer des généralités

Se bloquer

FAIRE

Exprimer chaque idée

Demander une clarification

Passer à l'idée suivante

S'en servir pour avoir de
nouvelles idées

Rester sur le sujet choisi

Rester spécifique

Penser différemment



Prototypage

Prototype = Représentation concrète d'une partie d'un système interactif

But : Tester des idées de façon concrète
notamment *l'interaction*

Considérer plusieurs alternatives et choisir la meilleure

S'assurer de l'utilisabilité de la solution dans divers cas d'usage

Aider les utilisateurs / le client à se faire une idée de la future interface

Se concentrer sur les parties qui posent un problème de conception



Prototypage

Tester des idées de façon concrète
notamment l'*interaction*

Plusieurs **types** de prototypes

selon la **représentation** : dessin, papier, wireframe, code

selon la **précision** : basse / moyenne / haute fidélité

selon l'**interactivité** : passif, simulé, interactif

Prototypage **rapide** :

représentation : papier ou wireframe

précision : basse ou moyenne précision

interaction : passive ou simulée

Prototypage rapide

Dessin à main levée (*sketching*)

Très rapide

Permet d'avoir une vue d'ensemble

Importance de l'aspect informel

Prototype papier

Rapide

Permet de mimer l'interaction

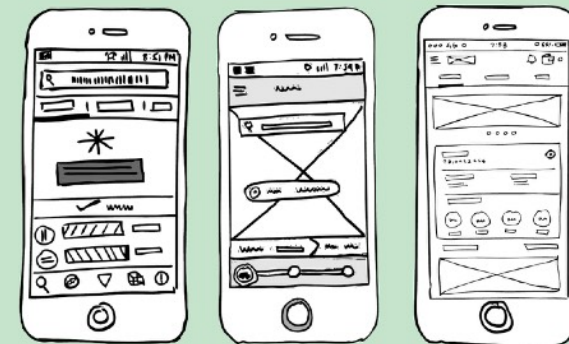
Capturer l'interaction en vidéo

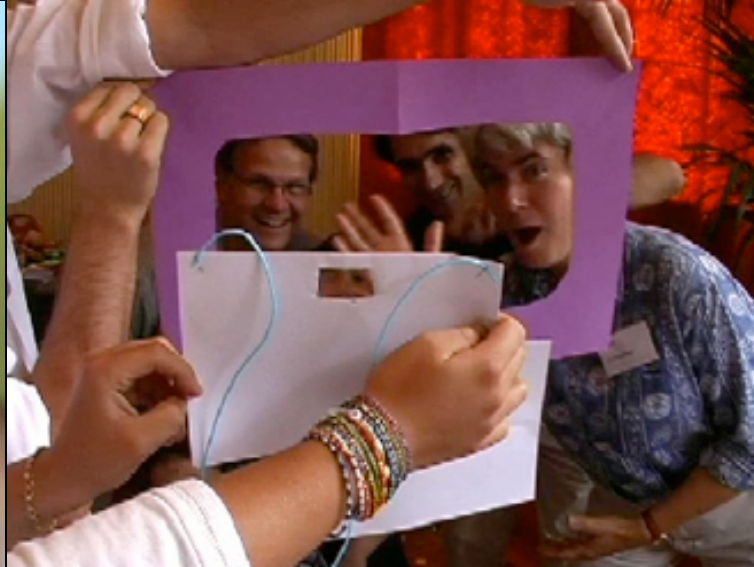
Wireframe

Moins rapide

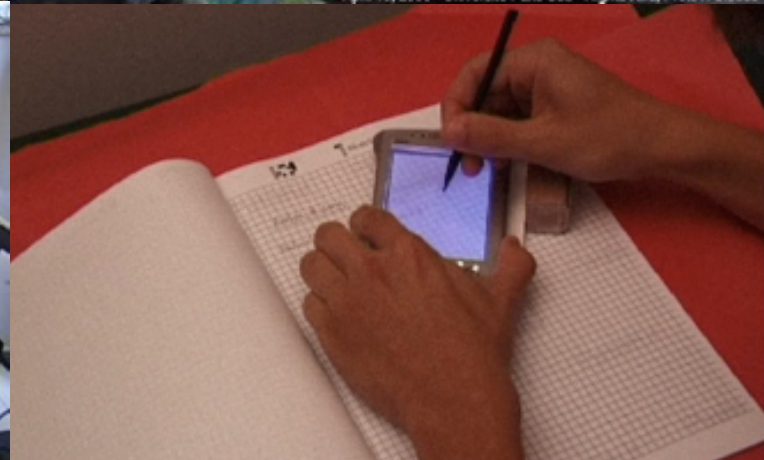
Permet d'animer l'interaction

Transition vers prototypes plus avancés





April 10, 2006 - Université Paris Sud - NightBoard, Proto1, Clouds



Conclusion

Analyse des besoins

Observation

Introspection

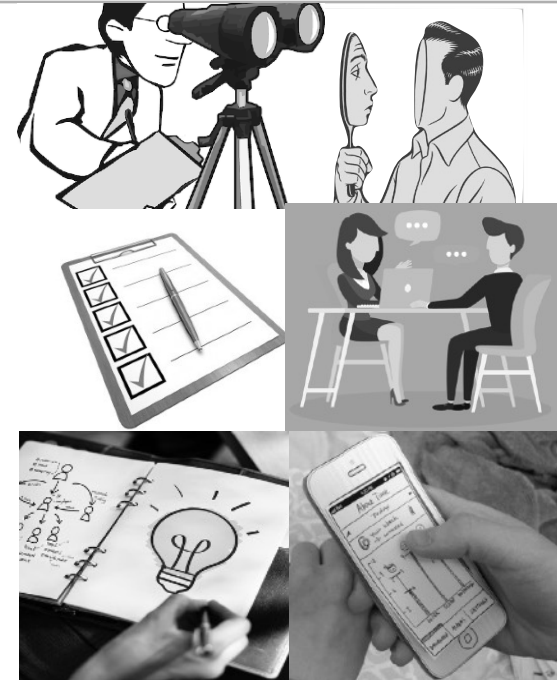
Questionnaires

Interviews

Idéation

Brainstorming

Prototypage



Conception centrée sur l'utilisateur

garder la perspective de l'utilisateur

partir des détails et généraliser - *pas l'inverse*