

# Introduction aux IHM

Module POO/IHM - SI3+SI4

<http://atelierihm.unice.fr/enseignements/pooihm/>

Extraits de cours de Philippe Renevier, Anne-Marie Dery-Pinna

## Au menu du module ...

<http://atelierihm.unice.fr/enseignements/nouihtm/>

## Plan et orientation pédagogique du module

- IHM pour applications installées sur le bureau (*desktop applications*) en SWING
- IHM pour applications web (*web applications*)



## Intervenants

Nom	Affiliation	Adresse
Christian Brel	I3S	Bâtiment Templiers ( <a href="#">www</a> )
Audrey Ocello	UNS / I3S	Bâtiment Templiers / Bureau 453 ( <a href="#">www</a> )
Romaric Pighetti	UNS / I3S	Bâtiment Templiers / Bureau 434 ( <a href="#">www</a> )

## Mode d'évaluation

Les coefficients suivants seront appliqués : 1/3 suivi de TD partie Desktop, 1/3 suivi de TD partie Web, 1/3 devoir surveillé



Vous êtes libre de développer avec IDE ou sans. Par contre, vous devez réaliser les IHM à la main, vous ne devez pas utiliser les éditeurs graphiques fournis par les IDEs.

# L'importance de l'IHM

- Pour les utilisateurs
  - "Le produit c'est l'IHM" (J. Raskin)
- Pour ceux qui développent les produits
  - "80% du code des systèmes interactifs est consacré à l'interface utilisateur" (L. Nigay)
- Pour ceux qui financent les produits
  - "63% des gros projets informatiques connaissent des dépassements de coûts" (S. Greenberg)

# Mauvaises pratiques de l'informaticien

- Focus uniquement sur les fonctions du système
- Traitement de l'interface en dernier
- IHM = côté "cosmétique" du produit
- Hypothèse que tous les utilisateurs leur ressemblent
- Pas de communication avec des utilisateurs

# Risques liés à une mauvaise IHM



Rejet pur et simple par les utilisateurs

Coût d'apprentissage/formation

Perte de productivité (hésitations, coup d'oeil à la doc)

Utilisation incomplète

Coût de maintenance (évolutivité de l'IHM)

Perte de crédibilité (réputation % satisfaction des clients)

Perte de parts de marché

# Problématiques d'IHM: utile et utilisable

- Réaliser des logiciels utiles et utilisables
  - Adéquation entre les fonctionnalités proposées et les besoins des utilisateurs : capacité fonctionnelle pour l'utilisateur d'atteindre ses buts de haut niveau
  - Adéquation entre l'interface et les utilisateurs : facilité d'apprentissage et d'utilisation
- Succès d'un produit informatique ou technologique  
=      Utilité    +    Utilisabilité



L'utilisabilité sert à poser la frontière entre l'utilité potentielle et l'utilité réelle

# Problématiques d'IHM : évolution

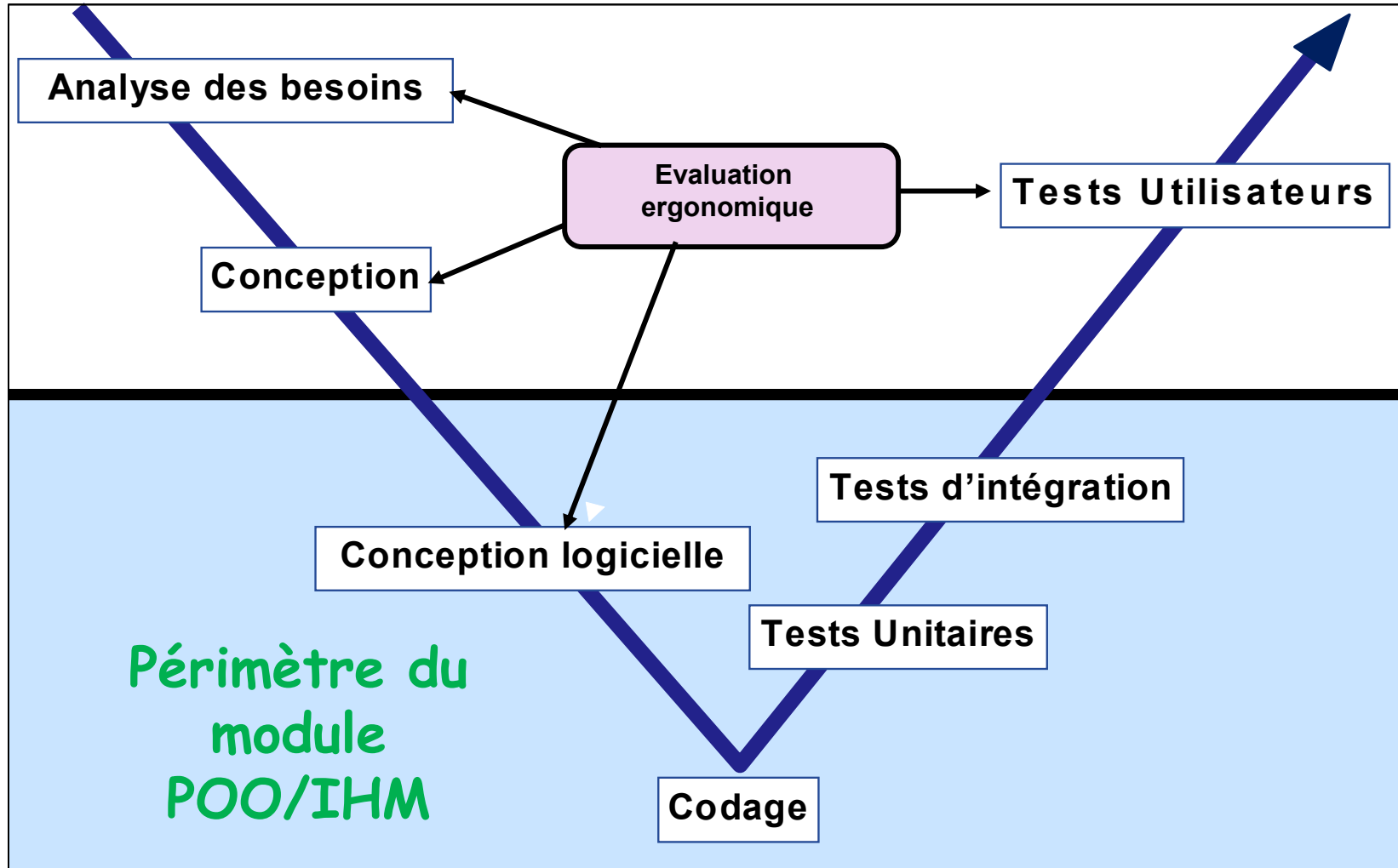
- "Tout logiciel doit évoluer au risque de devenir progressivement inutile" (M. Lehman)
  - Le matériel progresse et se diversifie sans cesse
  - Les fonctionnalités promises aussi
- Coûts de maintenance IHM / coût global d'un logiciel :
  - 50% à 60% dans les années 80 et 90
  - 80% à 90% maintenant !



Importance de bien concevoir et bien coder les IHM

Appliquer les principes de modularité pour éviter de tout modifier si on change la partie fonctionnelle ou la partie IHM

# Cycle de vie d'un projet et périmètre du module





# Compétences liées aux IHM et périmètre du module

## ■ Des interfaces flexibles

Module POO/IHM  
en SI3/SI4

- Sensibilisation à la conception d'IHM, considérations architecturales, développement modulaire d'IHM

## ■ Des interfaces utilisables

Module Conception  
& évaluation des  
IHM en SI5

- Conception centrée utilisateurs avec prise en compte des usages, mise en place de protocoles d'évaluations

## ■ Des interfaces adaptées au support visé

Autres  
modules du  
parcours  
IHM en SI5

- Acquisition de connaissances à la pointe de la technologie

Design non enseigné à  
l'école : Designers &  
informaticiens collaborent

## ■ Des interfaces "stylées".

# Que trouve-t'on dans une toolkit d'IHM ?

- 2 aspects à prendre en compte
  - La structure
  - Le comportement
- La structure sert à disposer l'information
  - C'est la partie "visible" de l'IHM
- Le comportement sert à déclencher des réactions face aux actions de l'utilisateur
  - Les actions de l'utilisateur sont traduites en des stimuli que l'on appelle **événements**

# Que trouve-t'on dans une toolkit d'IHM ?

Structure

- **Des éléments graphiques simples** prêts à l'emploi (widgets)
  - Définition d'un élément graphique avec une dimension, une position
  - et un type de contrôle associé permettant de saisir ou d'afficher de l'information et d'activer des fonctionnalités
- **Des éléments graphiques composés** (conteneurs) :
  - Définition d'un regroupement - éléments graphiques qui contiennent d'autres éléments graphiques
- **Du formattage** :
  - Définition de l'organisation (placement des éléments les uns par rapport aux autres) - En ligne, en tableau, avec des contraintes, etc
- **Des bouts d'écran** :
  - Définition de l'affichage à l'écran (dessiner les éléments graphiques) et gestion du redimensionnement

Comportement

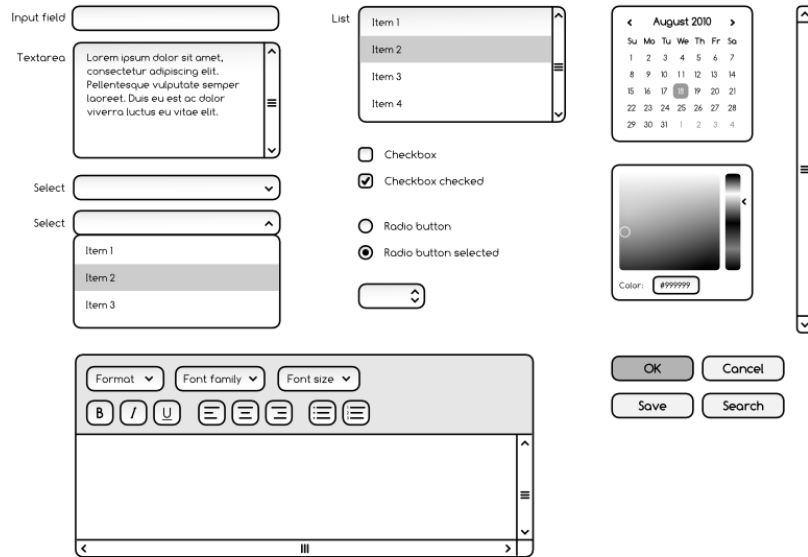
- **Des événements** pour écouter les actions de l'utilisateur (appuie sur des touches clavier /souris, mouvement de la souris/Wiimote, ...)

# Widgets génériques pour le maquettage

## Sketching kit

JankoAtWarpSpeed

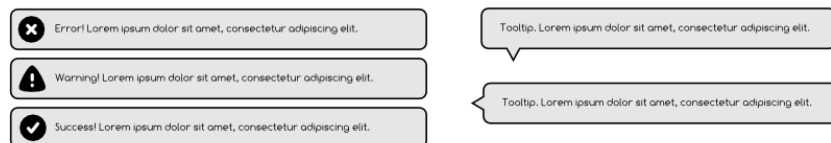
### Form elements



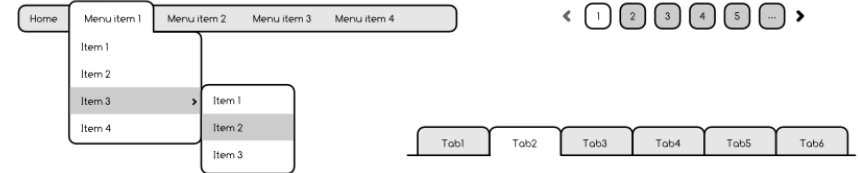
### Icons and indicators



### Feedback and tooltips



### Navigation



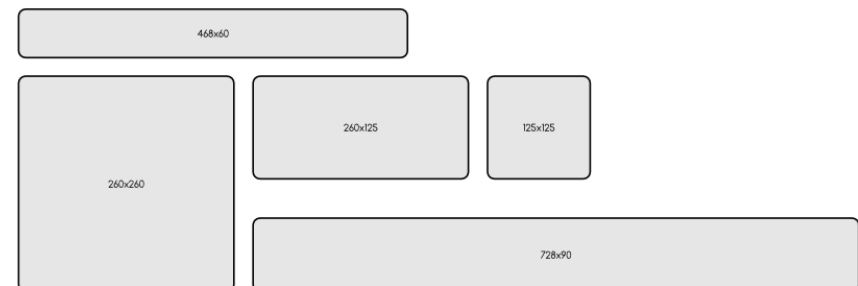
### Images



### Other controls



### Advertisement banners



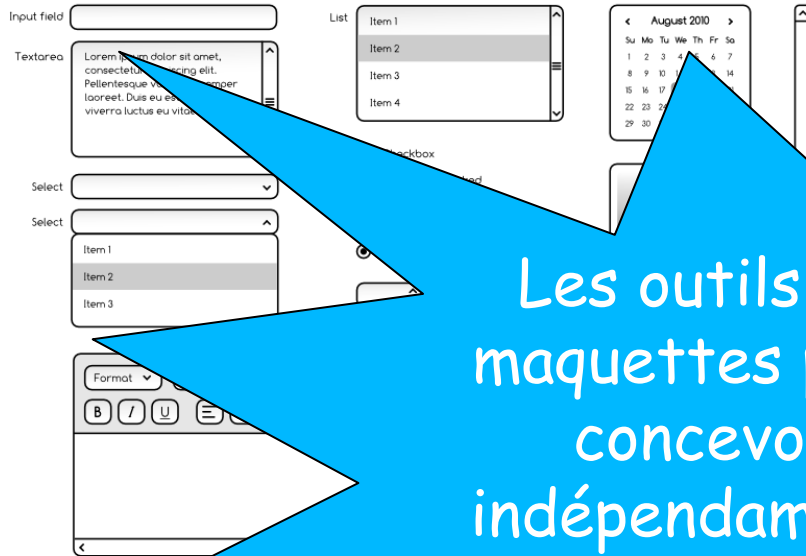
<http://www.jankoatwarpspeed.com/post/2009/12/24/sketching-wireframing-kit.aspx>

# Widgets génériques pour le maquettage

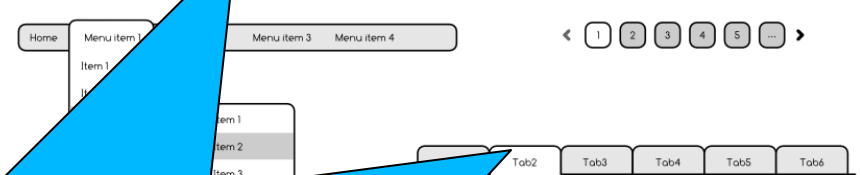
## Sketching kit

JankoAtWarpSpeed

### Form elements



### Navigation

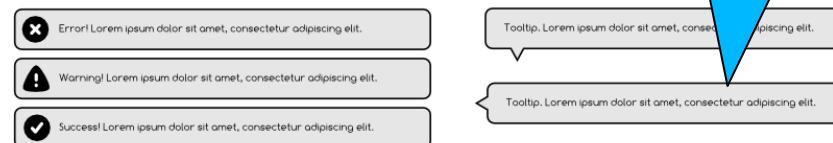


Les outils "dédiés" aux  
maquettes permettent de  
concevoir une IHM  
indépendamment de toute  
toolkit du marché

### Icons and indicators



### Feedback and tooltips



<http://www.jankoatwarpspeed.com/post/2009/12/24/sketching-wireframing-kit.aspx>

# Note importante

Appliquer les règles de conception et de programmation au développement d'IHM

Implémenter les IHM de façon *modulaire* (ex: pas directement dans un "main" en java !)

On parle de classes de vue : une IHM est la *somme* de toutes les classes de vue

Une classe de vue est donc :

Soit une *agrégation* de composants graphiques (dans ce cas la classe contiendra un objet de type JFrame par exemple en java SWING)

Soit une *extension* d'un composant graphiques (dans ce cas la classe héritera de JFrame par exemple en java SWING)