

---

# Modélisation des Systèmes d'Information

---

**Jean-Yves Antoine**

<http://www.info.univ-tours.fr/~antoine>



# Objectifs de l'enseignement

---

Conception des **systemes d'information** ... avec une  
vision plus générale sur la conception de tout

**systeme informatique** (génie logiciel)

SIR

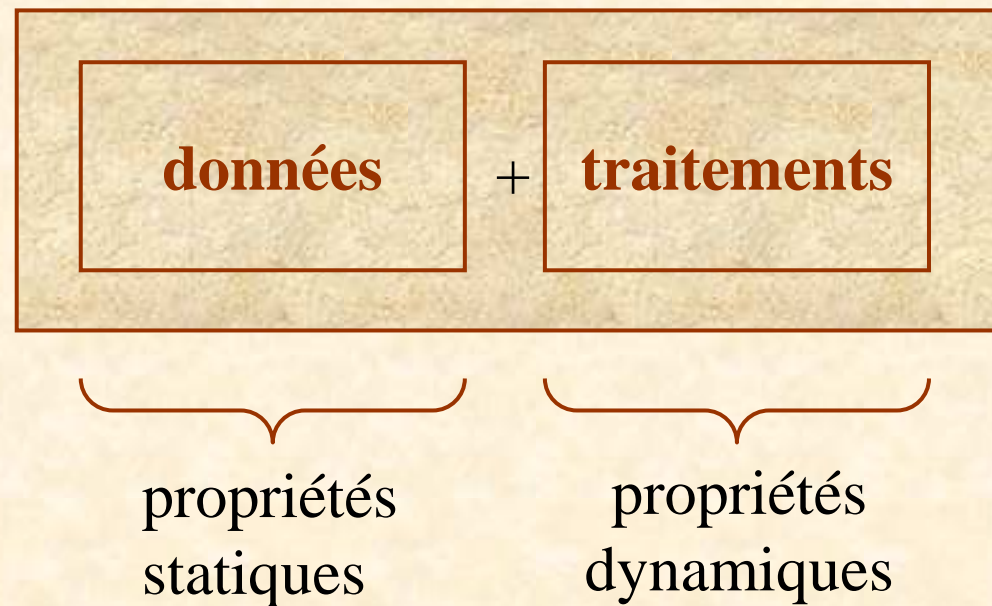


- Systeme d'information
- Conception / génie logiciel

# Objectifs de l'enseignement

---

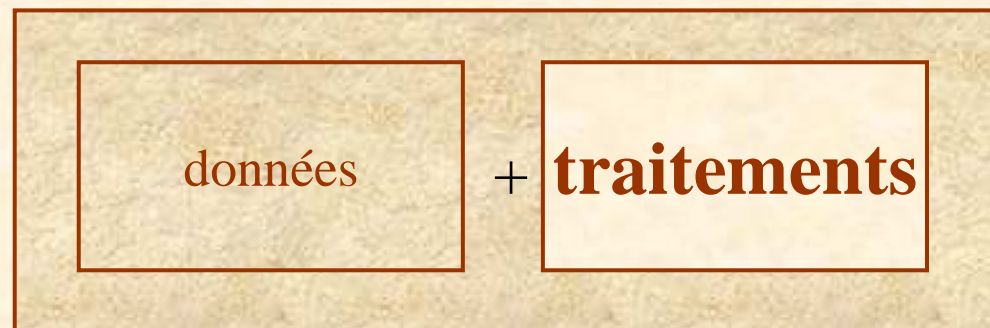
## ■ Système informatique



# Objectifs de l'enseignement

---

## ■ Logiciels « classique »

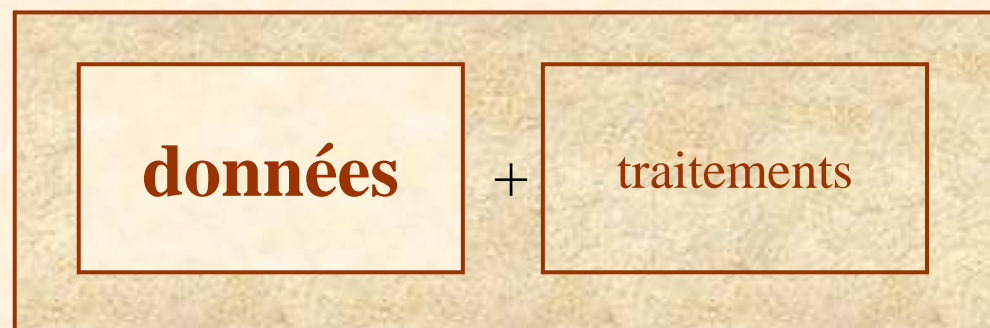


Les traitements revêtent le plus souvent une part prépondérante dans la conception du logiciel

# Objectifs de l'enseignement

---

## ■ Systèmes d'information (bases de données)



- Orientation données : «entrepôts de données»
- Persistance des données
- Les systèmes actuels accordent une part croissante aux traitements : «fouille de données» (*data mining*), aide à la décision (SIAD)

# Objectifs de l'enseignement

---

- **Connaître les méthodes** qui permettent de développer des systèmes logiciels complexes
  - **DEUG** : systèmes d'information (composante statique)
  - **Licence** : traitements (composante dynamique)
  - Forte orientation **conception objet**
- **Vous allez détester ...** car vous n'aimez pas être méthodique
- **Vous allez aimer ...** car proche de la «vie réelle» : développement en **équipe** et sur **projets importants**

# MSI : intervenants

---

## ■ Jean-Yves Antoine

- **Mél** : Jean-Yves.Antoine@univ-ubs.fr
- **Toile** : <http://www.univ-ubs.fr/valoria/antoine>

⇒ intro au GL ; concepts objets (semaines 1 à 4)

## ■ Frédéric Julliard

- **Mél** : Frederic.Julliard@univ-ubs.fr

⇒ Merise et UML (semaines 5 à 13)



# Bibliographie

---

## ■ Ouvrages francophones disponibles à la B.U.

- N. Lopez, J. Migueis et E. Pichon. (1997). Intégrer UML dans vos projets. Eyrolles.
- M. Bouzeghoub, G. Gardarin, P. Valduriez (1997) Les objets. Eyrolles.
- N. Kettani *et al.* (1999). De Merise à UML. Eyrolles
- I. Jacobson (1993) Le génie logiciel orienté objet : une approche fondée sur les cas d'utilisation. ACM Press, Addison-Wesley
- P-A Muller et N. Gaertner (2000) Modélisation objet avec UML. Eyrolles.
- M. Lai (2000). UML la notation unifiée de modélisation objet. Dunod Informatiques.



# Bibliographie

---

## ■ Cours francophones disponibles sur la Toile

- <http://www.uml.free.fr> (2002) (texte)
- D. Buchs. Génie Logiciel. EPFL, Lausanne. (slides)
- O. Sigaud. (1999) Introduction à la modélisation objets avec UML (poly)

*Et bien d'autres choses de qualité variable...*

# Validation

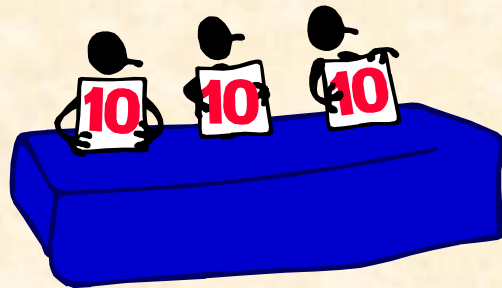
---

## ■ Contrôle continu : C

- a priori un examen en cours de semestre
- d'autres interrogations peuvent tomber...

## ■ Contrôle terminal : T

## ■ Note finale : $F = 1/2 (C + T)$



---

# Introduction au Génie Logiciel

---

**Jean-Yves Antoine**

**U. Bretagne Sud - UFR SSI - IUP Vannes**



année 2001-2002



# Pourquoi un cours en GL ?

---

- Vos compétences : « *programming-in-the-small* »
    - programmation individuelle sur de petits problèmes
    - algo, langages de programmation, structures de données
    - (parfois) un peu de méthodologie : analyse descendante
  - En entreprise : « *programming-in-the-large* »
    - travail en équipe sur des projets longs et complexes
    - spécifications de départ peu précises
    - dialogue avec le client/utilisateur : parler métier et non info
    - organisation, planification, gestion du risque
- ⇒ **démarche ingénierique : génie logiciel**

# Génie logiciel

---

## ■ Une définition

Méthodologie de construction en équipe d'un logiciel complexe et à multiples versions

## ■ Programmation vs. Génie Logiciel (approximation)

- programmation : activité personnelle
- génie logiciel : activité d'équipe

Suivant les projets, la partie programmation (codage) ne représentera qu'entre 10 % et 30 % du coût total.

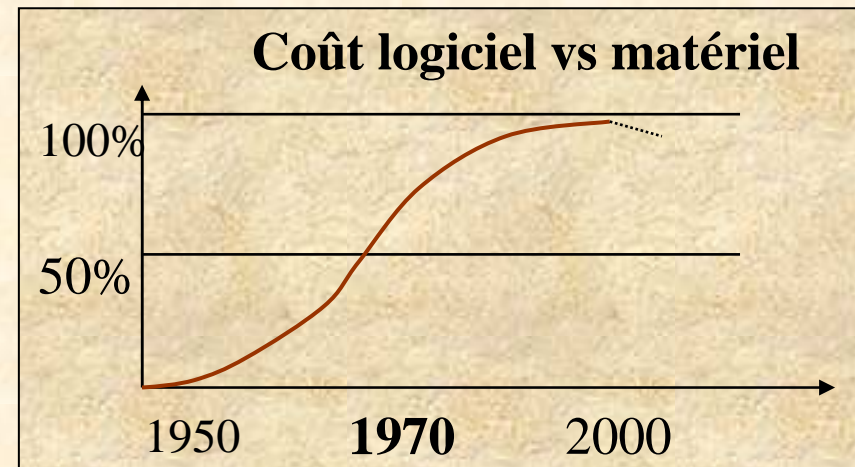
# Logiciel : aspects économiques

---

## ■ Importance économique du logiciel

- importance croissante de l'informatique dans l'économie (1985 : 150 Mrd\$ ; 1995 : 450 Mrd\$)

- coût du logiciel supérieur à celui du matériel



- coût maintenance supérieur au coût de conception  
⇒ **améliorer la qualité du logiciel !**

# La « crise » du logiciel

---

- Démarche ingénierique encore mal intégrée
  - Pannes logicielles
    - avion F16 retourné au passage de l'équateur : non prise en compte du référentiel hémisphère sud
    - échec sonde Venus : virgule remplacée par un point
    - bogue de l'an 2000 ...
    - instabilité de Windows 95 ...
- ⇒ **Coût global** : 80 Mrd\$ (1998) ; 175 Mrd\$ (2000)
- ⇒ **Systemes critiques** : nucléaire, transport ( $\approx 10^{-10}$  pannes/h), système bancaire...

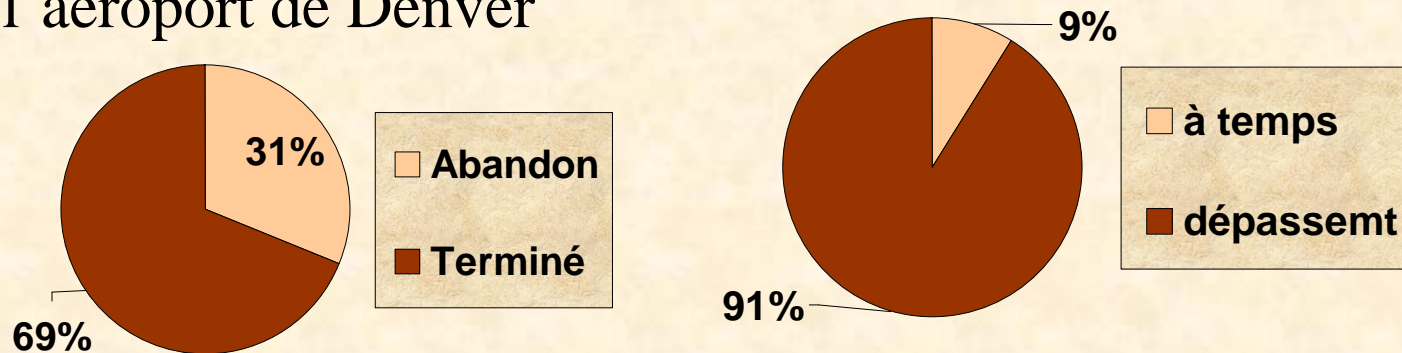


# La « crise » du logiciel

---

## ■ Echec de projet

- abandon du projet d'informatisation de la bourse de Londres : 4 années de travail et 100 M£ perdus
- Abandon du système de trafic aérien américain
- Retard (1 an) du système de livraison des bagages de l'aéroport de Denver

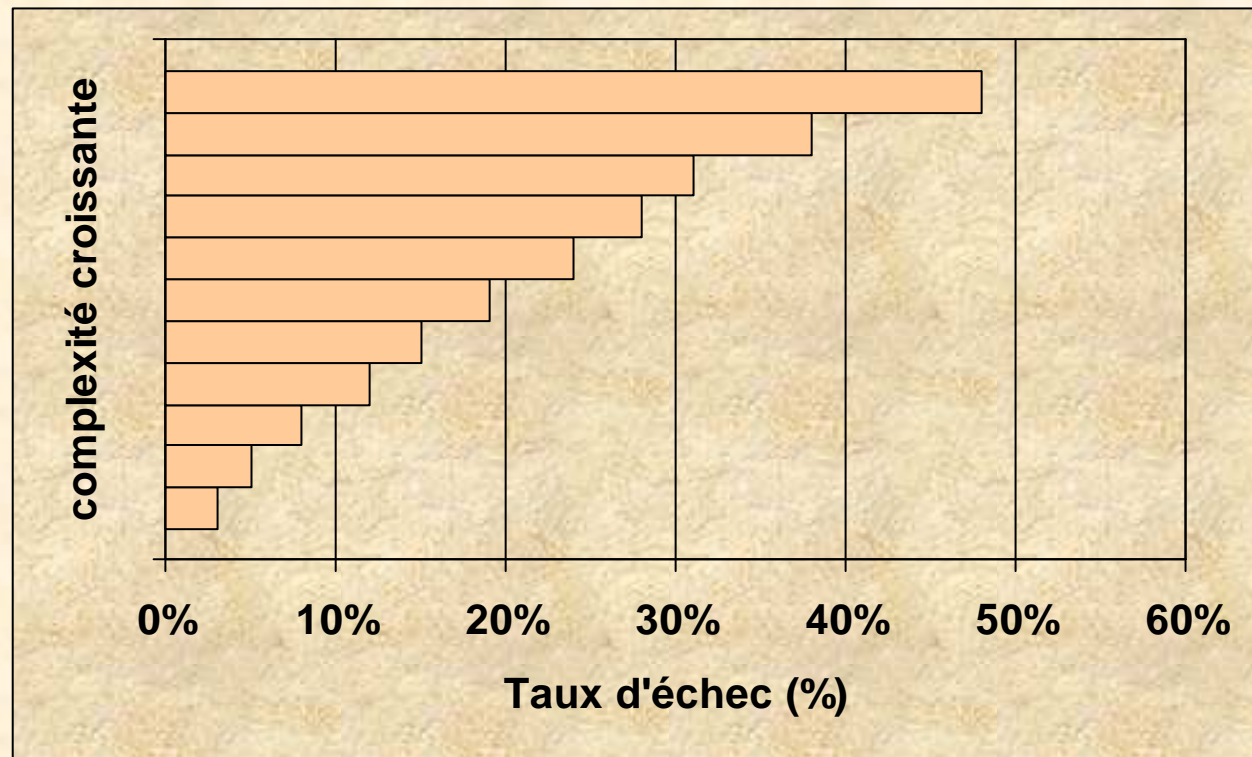


⇒ **Coût global (USA) : 81 Mdr\$ (1995)**

# La « crise » du logiciel

---

## ■ Influence de la taille du projet



# Qualité du logiciel

---

## ■ Privilégier la qualité à l'efficacité

- La prévention des erreurs coûte des dizaines de fois moins cher que leur correction
- Démarche qualité : **ISO 9126** ([www.osil.ch/eval/node15.html](http://www.osil.ch/eval/node15.html))

## ■ Qualité externe vs. Qualité interne

- Externe : vision client
- Interne : vision développeur

# Facteurs de qualité

---

## ■ Qualité externe

- **Complétude fonctionnelle** : réalise les tâches attendues (ISO)
- **Ergonomie** : facilité d'utilisation (ISO : *usability*)
- **Fiabilité** : fonctionne même dans des cas atypiques (ISO)
- **Adaptabilité** : adaptation aux modifs (ISO : *maintenability*)

## ■ Qualité interne

- **Réutilisabilité** : de plus en plus importante aujourd'hui !
- **Tracabilité**
- **Efficacité** : bonne utilisation des ressources matérielles (ISO)
- **Portabilité** : adaptation à de nouveaux environnements (ISO)

# Complexité croissante du logiciel

---

## ■ Complexité croissante du logiciel

- systèmes offrant de plus en plus de **fonctionnalités**.  
*Exemple* : système d'information = BD + **SIAD**
- **systèmes distribués** : machines hétérogènes en réseau
- **mutations technologiques** rapides : langages et environnements de développement, O.S.
- **évolutions des besoins du client** en cours de projet

## ■ Solutions proposées par le génie logiciel

- séparer aspects fonctionnels et technologiques
- décomposition en sous-systèmes  $\Rightarrow$  approche objet
- démarche itérative  $\Rightarrow$  approche objet

# Génie logiciel

---

- Le GL doit fournir des méthodes de conception de systèmes complexes permettant :
  - une prise en compte du client,
  - une démarche qualité
  - une organisation du travail en équipe
  
- Le GL, ce sont aussi des outils associés :
  - AGL (atelier de génie logiciel)
  - AGL méthodologiques.

*Exemple : Rational Rose (UML)*