

# Le génie logiciel

---

Qu'est-ce que c'est le génie logiciel et  
sa place dans les sujets d'informatique

Le contenu est basé aux transparents du 7<sup>ème</sup> édition  
de [«Software Engineering» de Ian Sommerville](#)

# Questions de l'ingénierie de logiciel

---

- Qu'est ce que c'est logiciel?
- Qu'est ce que c'est génie logiciel?
- Quelle est la différence entre génie logiciel et informatique?
- Quelle est la différence entre génie logiciel et génie des systèmes?
- Qu'est ce que c'est processus unifié de logiciel?
- Qu'est ce que c'est modèle du processus de logiciel?

# Questions de l'ingénierie de logiciel

---

- Combien ça coûte?
- Quelles sont les méthodes du génie logiciel?
- Qu'est ce que c'est CASE (Computer-Aided Software Engineering)?
- Quelles sont les propriétés du bon logiciel?

# Logiciel

---

- Programmes et la documentation associée – cahier de charges, modèles, manuels
- Types
  - Générique
  - Individuel
  - Hérité

# Génie logiciel

---

- Par rapport d'informatique
- Par rapport du génie des systèmes

# Processus du logiciel

---

- Un ensemble d'activités dont l'objectif est le développement et l'évolution du logiciel.
- Activités :
  - Spécification
  - Développement
  - Validation
  - Evolution

# Modèle du processus

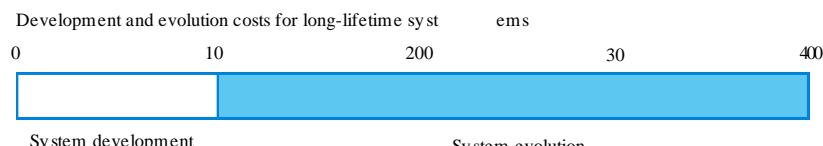
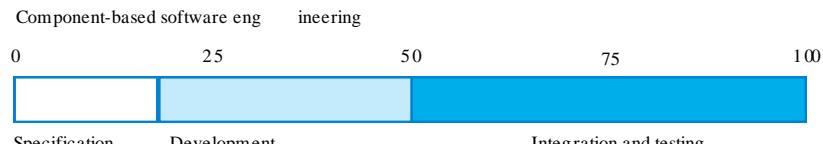
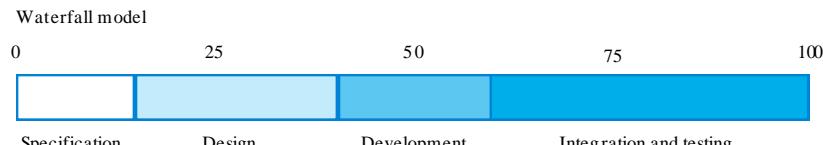
---

- C'est quoi un modèle?
- Points de vue:
  - Flux d'activités
  - Flux des données
  - Rôles/activités
- Modèles génériques
  - Cascade (Waterfall)
  - Itérative
  - Composants

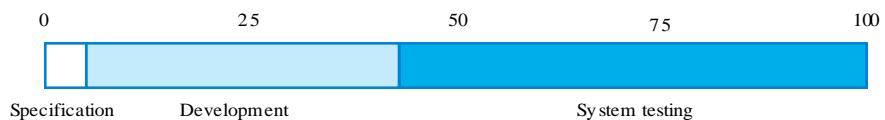
# Les coûts

- Activités

- Spécification du logiciel
- Développement
- Validation
- Evolution



- Production



# Méthodes de génie logiciel

---

- Composants des méthodes
  - Modèles
  - Règles
  - Recommandations
  - Direction et gestion

# CASE (Computer-Aided Software Engineering)

---

- Upper-CASE
  - Support les activités de conception et de definition des besoins
- Lower-CASE
  - Support les activités tards

# Les propriétés du bon logiciel

---

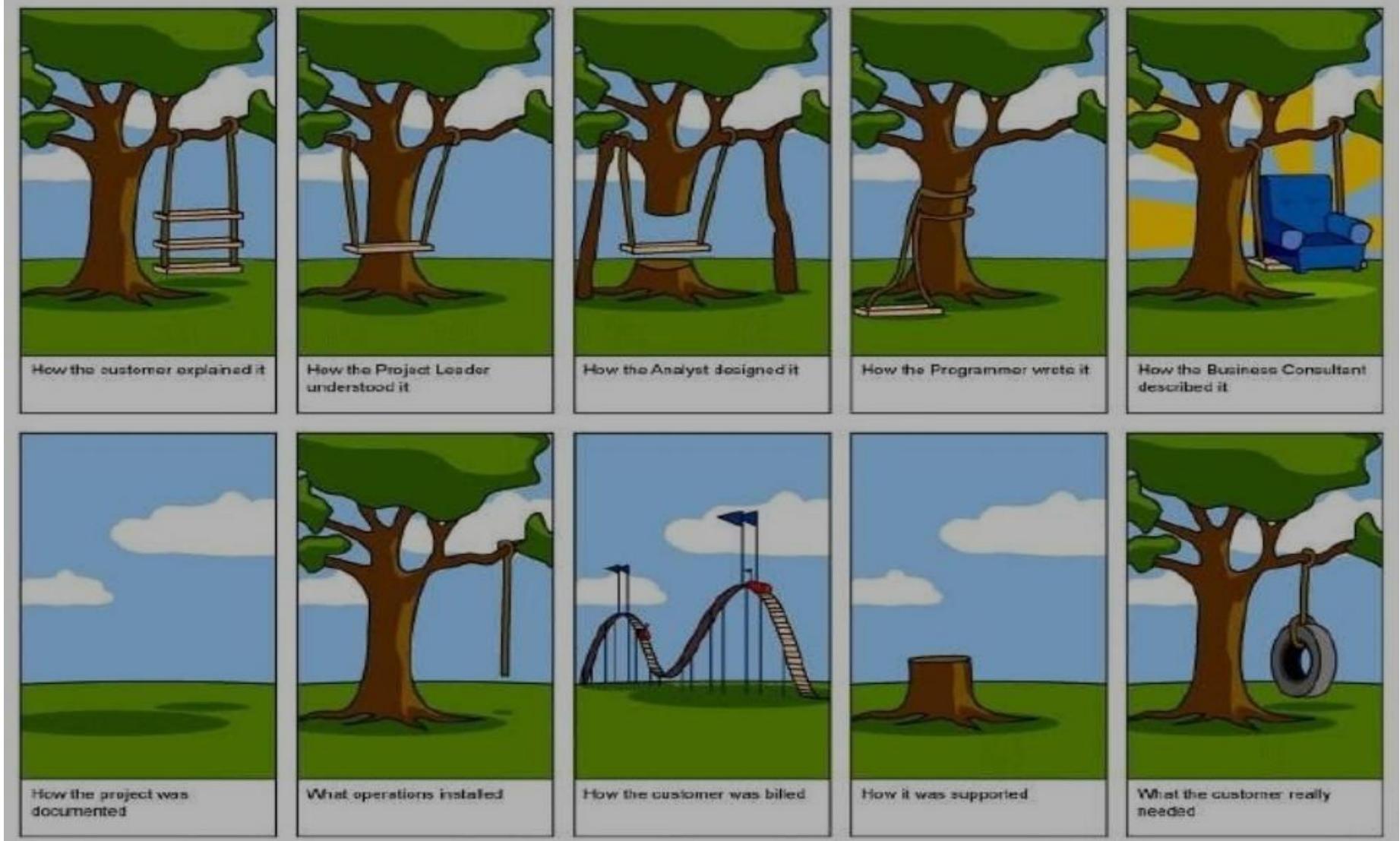
- Avoir la fonctionnalité désirée.
- Facilement maintenu
- Sûr
- Efficace
- Accepté, compris par les usagers

# Les défis devant le GL

---

- Hétérogénéité des plateformes
- Délivrance (respecter les termes et la qualité à la fois)
- Confiance des usagers
- Responsabilité professionnelles et éthiques

# Les défis devant le GL



# Systèmes critiques

---

- Système critique par rapport de sécurité
  - Perdre la vie ou la santé. Ex. Usine chimique
- Système critique par rapport de mission
  - Une activité essentielle est échouée
- Système critique par rapport de commerce
  - Grand pertes d'argent – système de comptabilité d'une banque

# Fiabilité

---

- Panne du matériel
- Echec du logiciel
- Erreur opérationnel – le plus souvent

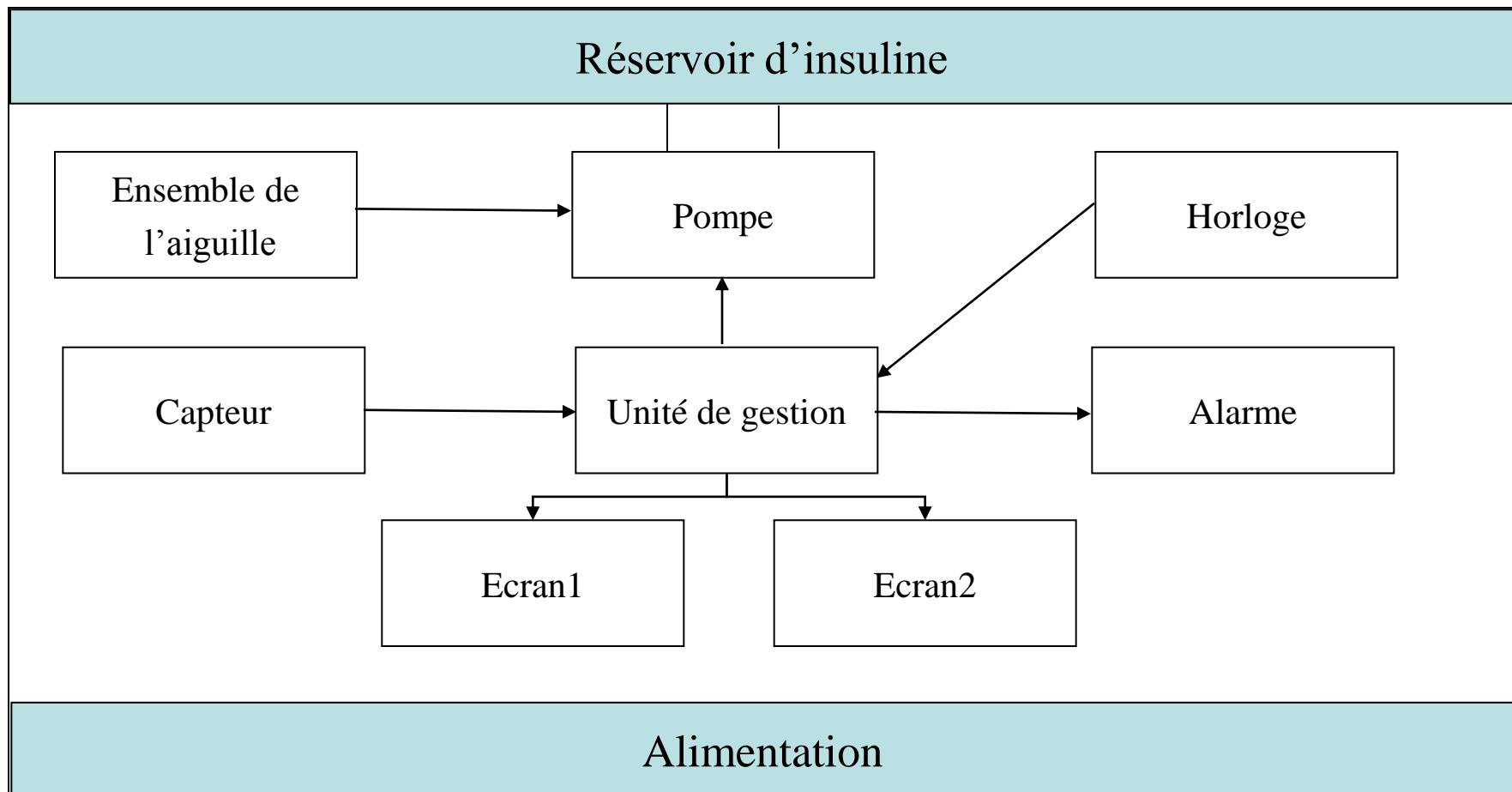
# Fiabilité

---

- Composants de la fiabilité (Dependability)
  - Disponibilité
  - *Fiabilité (Reliability)*
  - Sécurité
    - (Safety) De fonctionner sans échec catastrophique
    - (Security) De se protéger des attaques externes
  - Habilité de restauration après un échec
  - Habilité d'être maintenu
  - Habilité de survivre
  - Tolérance d'erreurs

# Exemple - Insuline pompe

- Organisation



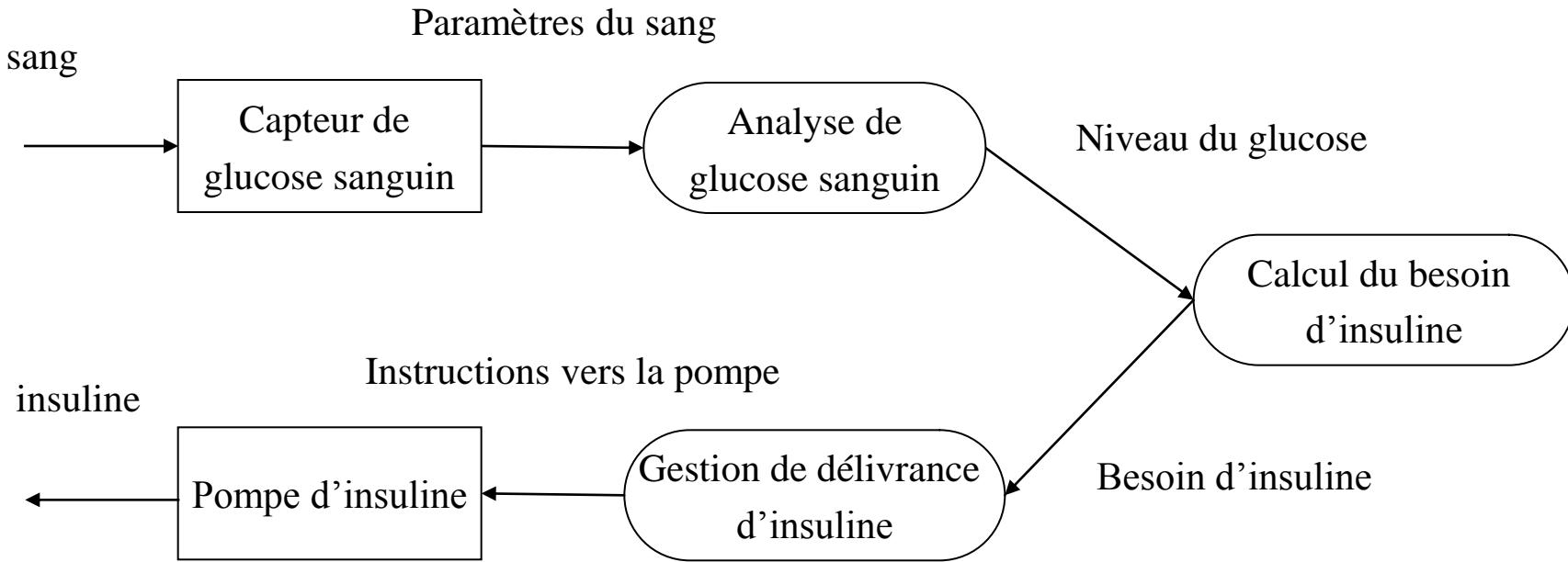
# Exercice

---

- Donner un flux des données pour l'exemple précédent.

# Exemple - Insuline pompe

- Flux de données



# Exercice

---

- Quelles sont les exigences de fiabilités et les coûts de fiabilités pour l'exemple précédent?

# Exigences de fiabilité

---

- Le système doit être capable de livrer l'insuline quand l'organisme a besoin et en quantité qui assez de neutraliser le glucose.
- L'exigence principale de sûreté et de ne pas livrer une surdose, qui peut être mortelle.

# Les coûts de fiabilité

---

