

# Génie Logiciel

---

# Introduction

---

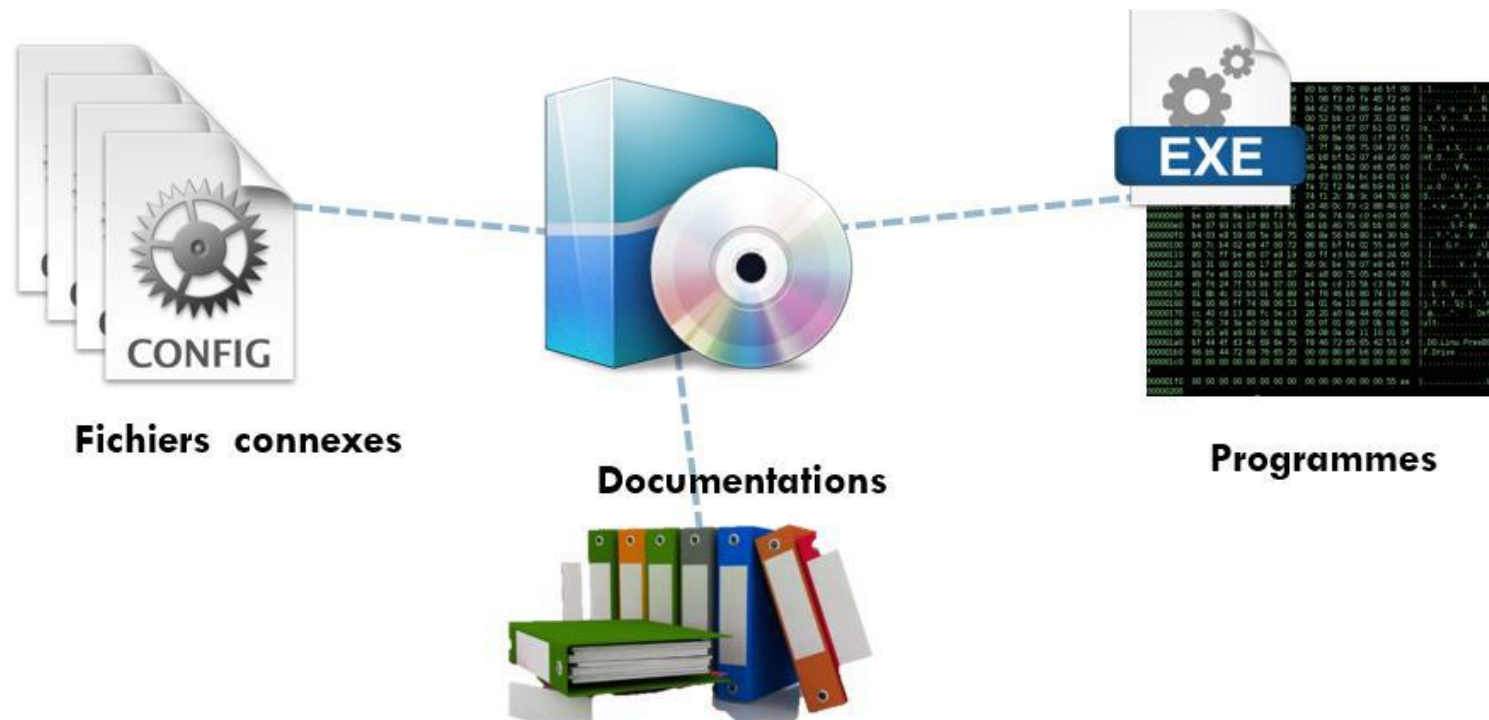
# Définitions (Système Informatique)

---

- Un **Système informatique** représente l'ensemble composé de matériels et de logiciels nécessaires pour réaliser des traitements de l'information.
- Depuis quelques années, la fabrication du matériel est assurée par quelques fabricants seulement.
  - Le matériel est relativement fiable
  - Le marché est standardisé
- Les problèmes liés à l'informatique sont essentiellement des problèmes de Logiciel !!!!

# Définitions (Logiciel)

- Un **logiciel** est un ensemble d'entités nécessaires au fonctionnement d'un processus de traitement automatique de l'information.



# Définitions (Logiciel – suite)

---

Un logiciel est en général un sous-système d'un système englobant.

➤ Il interagit avec son **environnement**:

- Des opérateurs humains (utilisateurs, administrateurs,...);
- D'autres logiciels;
- Des contrôleurs matériels.

➤ Il réalise une **spécification**:

- Son comportement vérifie un ensemble de critères qui régissent ses interactions avec son environnement.

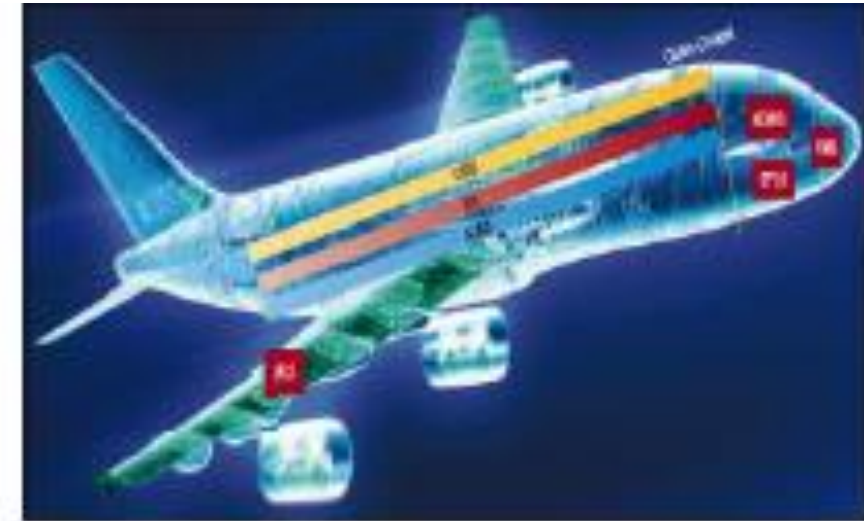
# Définitions (Logiciel – suite)

---

Exemple : Un Airbus A380

- Prix de 394 M Euros
- On estime que la partie informatique représente le tiers du coût
- Environ 1 Milliard de lignes de code

➔ **Un vrai challenge pour les informaticiens!!!!!!!**



# Définitions (Logiciel – suite)

---

Difficultés spécifiques du logiciel :

- Support immatériel
- Un objet technique fortement contraint:
  - Structure complexe
  - Relatif au métier du domaine traité (gestion, automobile, aéronautique,...)
- Difficile de mesurer la qualité
- Conséquences critiques causées par des modifications infimes
- Défaillances logicielles et principalement humaines

# Bug Informatique

---

- Décrit le dysfonctionnement d'un programme informatique.
- Sa gravité peut être plus ou moins importante et engendrer des conséquences plus ou moins désastreuses.
- Un **bug informatique** bénin se résume la plupart du temps au plantage d'un logiciel, à l'impossibilité d'effectuer une action, voire à une perte d'informations.
- Un **bug informatique** majeur peut par contre entraîner de graves accidents, comme l'a prouvé l'explosion du lanceur Ariane 5 en 1996.
- Le **bug** résulte, dans une grande majorité des cas, d'une erreur humaine lors de la conception du logiciel informatique.
- Pour le prévenir, les éditeurs et programmeurs de logiciels effectuent une batterie de tests lors de la conception du logiciel.
- Lorsqu'un bug se manifeste après la publication du logiciel, les développeurs ont la possibilité de le corriger en proposant un patch.



# Bugs Célèbres

---

- **Sonde Mariner 1, 1962(coût : \$18,5 millions )**

- Défaillance des commandes de guidage due à une erreur de spécification
- Erreur de programme fortran. Le point avait remplacé par une virgule

- **Accidents d'irradiation de Therac25 (1985-87):**

- Fautes de conception dans le matériel et le logiciel d'un appareil de traitement médical par irradiation
- La mort de plusieurs personnes

- **Le projet TAURUS,1993.** Ce fut un projet d'informatisation de la bourse de Londres. Il a été abandonné définitivement après quatre années de labeur et a engendré environ 300 millions de livres de perte.

# Bugs Célèbres (suite)

---

- **Ariane V vol 501, 1996** (coût : \$370 millions)
  - Explosion après 40 secondes de vol: Panne du système de navigation due à un dépassement de capacité (arithmetic overflow)
  - Réutilisation d'un composant d'Ariane IV non re-testé
- **Bug de l'an 2000** (coût : \$600 millions)
  - L'âge se faisait apparaître uniquement sur les deux chiffre.

# Raisons principales

---

- Erreurs humaines
- Taille et complexité des logiciels
- Taille des équipes de conception/développement
- Faible communication entre les parties prenantes
- Manque des méthodes de conception
- Spécifications incomplètes: phase d'analyse des besoins du client incomplète.
- Mauvaise gestion des changements
- Négligence et manque de méthodes et d'outils des phases de validation/vérification



**Comment le client  
a exprimé son besoin**



**Comment le chef de  
projet l'a compris**



**Comment l'ingénieur  
l'a conçu**



**Comment le  
programmeur l'a écrit**



**Comment le responsable  
des ventes l'a décrit**



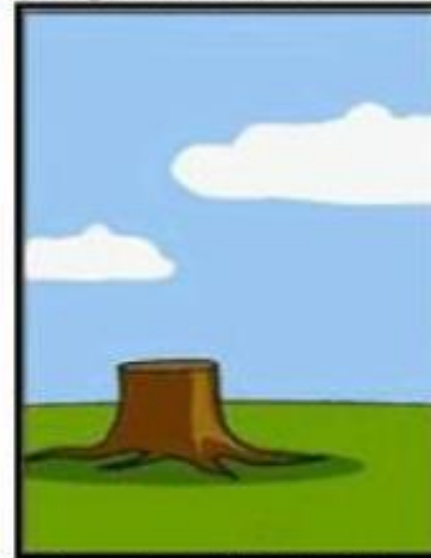
**Comment le projet  
a été documenté**



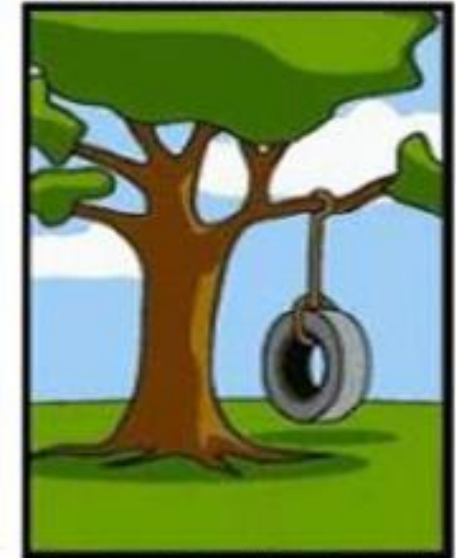
**Ce qui a finalement  
été installé**



**Comment le client  
a été facturé**



**Comment la hotline  
répond aux demandes**



**Ce dont le client avait  
réellement besoin**

# Quelques statistiques

---

Une étude du gouvernement américain fait apparaître les symptômes suivants:

- Beaucoup de logiciels ne sont pas livrés, pas utilisés ou abandonnés.
- Plus précisément, le coût se répartit de la façon suivante:

Logiciels	Coût
payés, jamais livrés	3.2 M \$
livrés, jamais utilisés	2.0 M \$
abandonnés ou recommencés	1.3 M \$
utilisés après modification	0.2 M \$
utilisés en l'état	0.1 M \$

# Génie Logiciel

---

## Naissance du génie logiciel

- les pères du génie logiciel se prénomment Friedrich Bauer et Louis Bolliet
- Cette spécialité a en effet vu le jour en 1968 sous le parrainage de l'OTAN
- Elle avait pour objectif de répondre à 2 constations:
  - D'une part, le logiciel n'était pas fiable
  - D'autre part, il était difficile à réaliser dans les délais et budgets prévus et en satisfaisant le cahier des charges

# Génie Logiciel

---

Constats:

- Le coût de construction du logiciel est devenu plus important que celui de la construction du matériel.
- délais de livraison non respectés
- budgets non respectés.
- ne répond pas aux besoins de l'utilisateur ou du client
- difficile à utiliser, maintenir, et faire évoluer

Introduction de l'expression : « **Génie Logiciel** » (**Software Engineering**)

- Comment faire des logiciels de qualité? Qu'attend-on d'un logiciel ? Quels sont les critères de qualité pour un logiciel ?

# Génie Logiciel

---

- Le génie logiciel est un domaine des sciences de l'ingénieur dont l'objet d'étude est la **conception**, la **fabrication**, et la **maintenance** des systèmes informatiques complexes.
- Le génie logiciel vise à garantir :
  - la **spécification** répond aux **besoins** réels de ses clients ;
  - le logiciel respecte sa **spécification**;
  - Les **coûts** alloués pour sa réalisation sont respectés ;
- les **délais** de réalisation sont respectés.
- ➔ Appliquer des méthodes fiables qui vont guider le développement du logiciel, de sa conception à sa livraison.



# Génie Logiciel

---

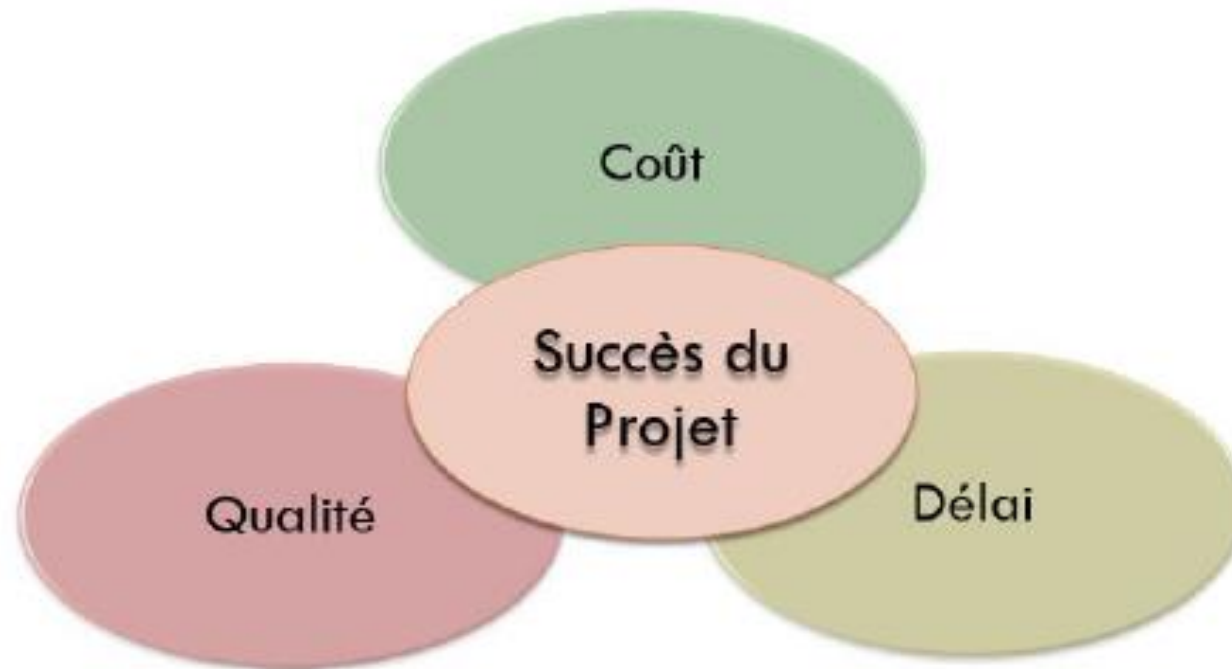
Pour résumer, nous dirons que:

- La motivation première du génie logiciel est la réduction des coûts de développement du logiciel
- Le Génie logiciel est un ensemble de moyens (techniques et méthodologiques) permettant la construction de systèmes informatiques répondant à des critères de qualité préalablement définis
- Sa mise en œuvre implique la prise en compte:
  - Des environnements de développement avec toute la variété d'outils et d'approches dont on peut disposer
  - Des méthodes et des techniques de gestion de processus
  - Des relations que celle-ci entretient avec les commanditaires et les utilisateurs du produit

# Réussite d'un projet informatique

---

## ➤ Piliers de réussite



# Qualités attendues d'un logiciel

---

- La norme ISO 9126 définit six groupes d'indicateurs de qualité des logiciels
  - ✓ Validité
  - ✓ Facilité d'utilisation
  - ✓ Performance
  - ✓ Fiabilité
  - ✓ Portabilité
  - ✓ Maintenabilité

# Qualités attendues d'un logiciel (validité)

---

➤ Adéquation entre le besoin effectif du client et les fonctionnalités offertes par le logiciel

➤ **Solutions :**

- Analyse exhaustive des besoins
- Améliorer la communication entre les intervenants
- Travailler avec rigueur

# Qualités attendues d'un logiciel (Facilité d'utilisation)

---

- Efficacité et satisfaction avec laquelle des utilisateurs accomplissent leurs objectifs selon leur environnement
  - ✓ **Facilité de compréhension:** comprendre ce que l'on peut faire avec le logiciel.
  - ✓ **Facilité d'apprentissage :** savoir comment travailler avec le logiciel
  - ✓ **Facilité d'exploitation :** effort nécessaire pour utiliser le logiciel
- **Solutions :**
  - Analyse du mode opératoire des utilisateurs.
  - Adapter l'ergonomie des logiciels aux utilisateurs.

# Qualités attendues d'un logiciel (Performance)

---

➤ Rapport entre les ressources dédiées (temps, matériel, utilisateurs) et la quantité des résultats produits.

➤ **Solutions :**

→ Logiciels plus simples

→ Veiller à la complexité des algorithmes

→ Machines plus performantes

# Qualités attendues d'un logiciel (Fiabilité)

---

- **Justesse et conformité** : le logiciel retourne des résultats corrects quellesque soient les conditions d'exploitation.
- **Robustesse et sureté** : le logiciel fonctionne raisonnablement en toutes circonstances (utilisation intensive), même en dehors des conditions d'utilisation prévues (tolérance aux pannes)
- **Mesures:**
  - ✓ MTBF : Mean Time Between Failures
  - ✓ Disponibilité (pourcentage du temps pendant lequel le système est utilisable) et Taux d'erreur (nombre d'erreurs par KLOC)
- **Solutions:**
  - Utiliser des méthodes formelles, des langages et des méthodes de programmation de haut niveau
  - Vérifications, tests

# Qualités attendues d'un logiciel (Portabilité)

---

- **Facilité d'adaptation** à des changements d'environnements opérationnels (matériel ou logiciel)
- **Facilité d'installation**
- **Coexistence avec d'autres logiciels**
  
- **Solutions:**
  - Rendre le logiciel indépendant des environnement d'exécution.
  - Machines virtuelles.



# Qualités attendues d'un logiciel (Maintenabilité)

---

- L'effort nécessaire à corriger ou transformer le logiciel.
- Solutions:
  - La conception par sous-ensembles faciles à démonter et à inter-changer.
  - Une parfaite communication entre opérateur et technicien de maintenance.