COURS D'INGÉNIERIE LOGICIEL

ÉLABORÉ PAR

MR GAOUAR . A

gaouaradil@gmail

Promotion L3 IBM

20/01/2018

CHAPITRE I
INTRODUCTION AU GÉNIE LOGICIEL

Objectifs du cours

- Etre capable de définir les principes et les techniques du génie logiciel
- Comprendre la valeur ajoutée du génie logiciel

Références

- Beugnard, A. (1998). Introduction au génie logiciel. ENST Bretagne.
- Di gallo, F. (2001). Cours de génie logiciel. CNAM Bordeaux.
- Lonchamp, L. (2003). Cours génie logiciel. CNAM – Nancy.
- Notes de cours de Mme Latifa Rabii

I. Introduction

- **Section 1 : Rappels**
- **Section 2 : Définition d'un logiciel**
- **Section 3 : Domaines d'application du logiciel**
- **Section 4 : Crise du logiciel**
- **Section 5 : Solution**

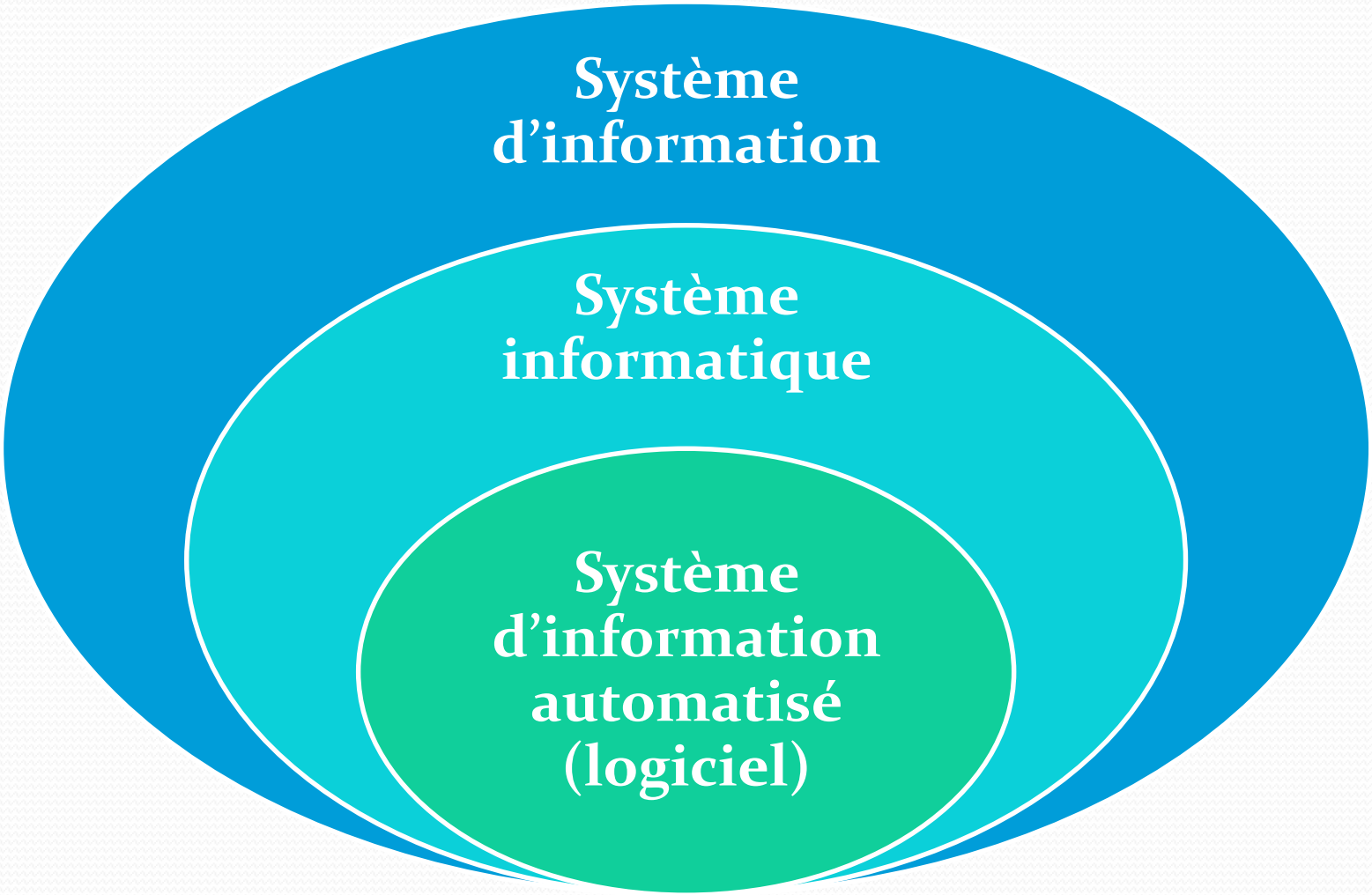
Rappels

- **Un système d'information : est l'ensemble des éléments** participants à la gestion, au **stockage**, au **traitement**, au **transport** et à la **diffusion** de l'information au sein de l'organisation
- **Un système informatique :** est **l'ensemble des équipements** destinés au traitement automatique de l'information permettant d'acquérir, de stocker, de traiter et de communiquer des données

Définition d'un logiciel (1/3)

- Un logiciel est **un système d'information automatisé**
- **Un système d'information automatisé** est l'ensemble des moyens et des méthodes qui se rapporte au traitement automatisé des données. Il constitue la partie logicielle du système informatique

Définition d'un logiciel (2/3)



Définition d'un logiciel (3/3)

- Un logiciel est un produit qui couvre un ensemble organisé de :
  - **Programmes**
  - **Procédés**
  - **Documentation**
  - **Services**

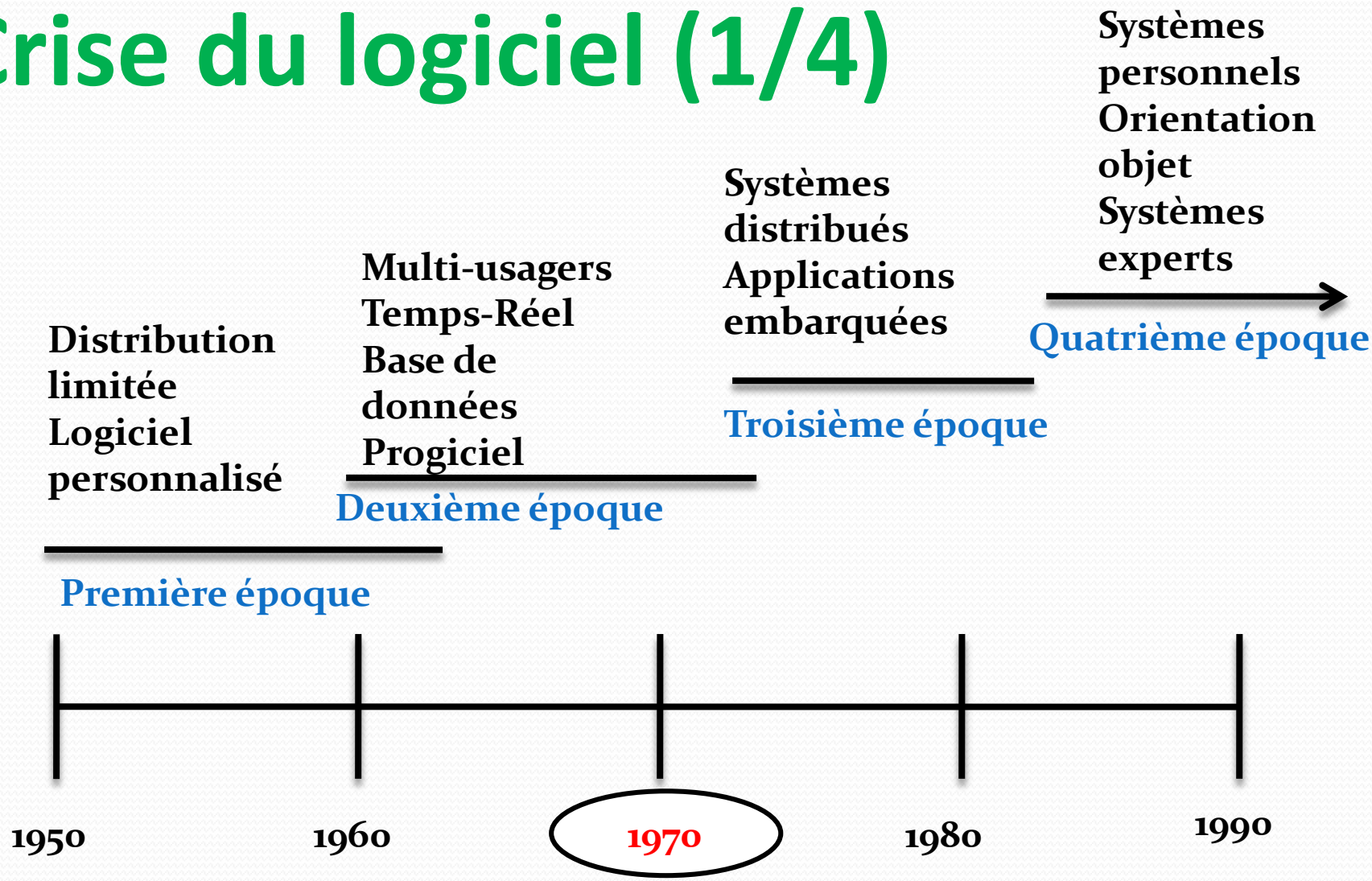
Caractéristiques du logiciel

- Un objet immatériel
- Ne s'use pas
- Le logiciel est facile à reproduire
- Ses fonctionnalités sont difficiles à figer au départ et souvent remises en cause
- La plupart des logiciels sont personnalisés

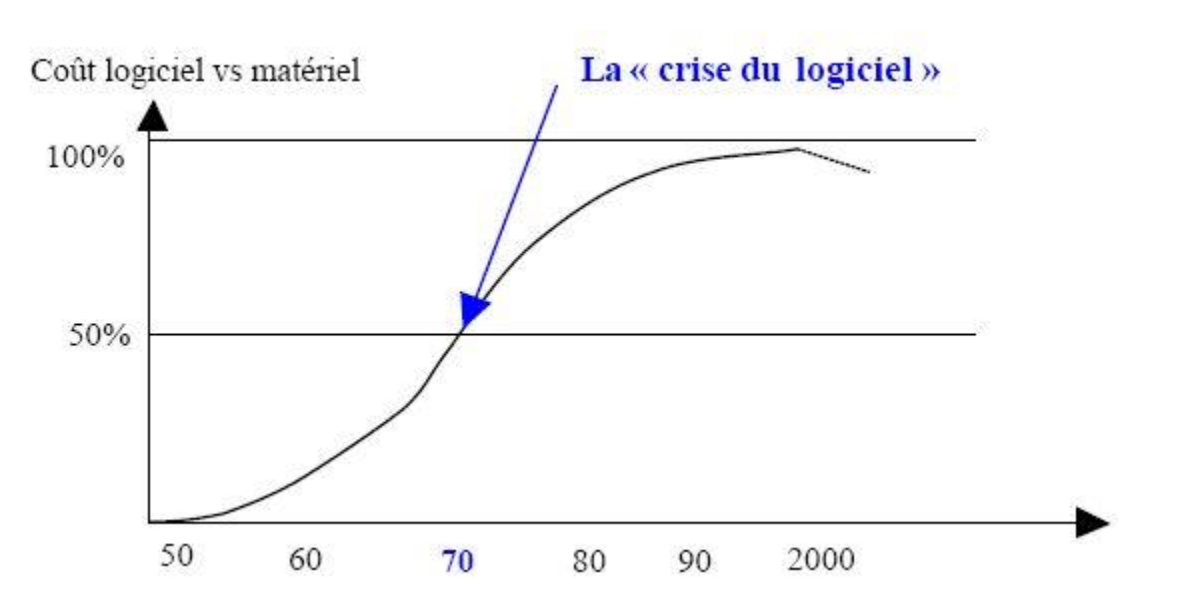
Domaines d'application du logiciel

- **Système**
  - Compilateurs (A-o System)
  - Editeurs
  - Gestion de fichiers
- **Temps réels** (Real-time)
  - Contrôle de machine
- **Affaires** (Data processing)
  - SGBD (Oracle)
  - ERP (SAP)
- **Embarqué** (Embedded)
  - Programme FPGA
  - Auto-contrôle
- **Scientifique**
  - Simulation
  - Conception assisté par ordinateur
  - Calcul numérique intensif
- **Bureautique**
- **Intelligence artificielle**
  - Système expert

Crise du logiciel (1/4)



Crise du logiciel (2/4)



Exemples d'échec de logiciel

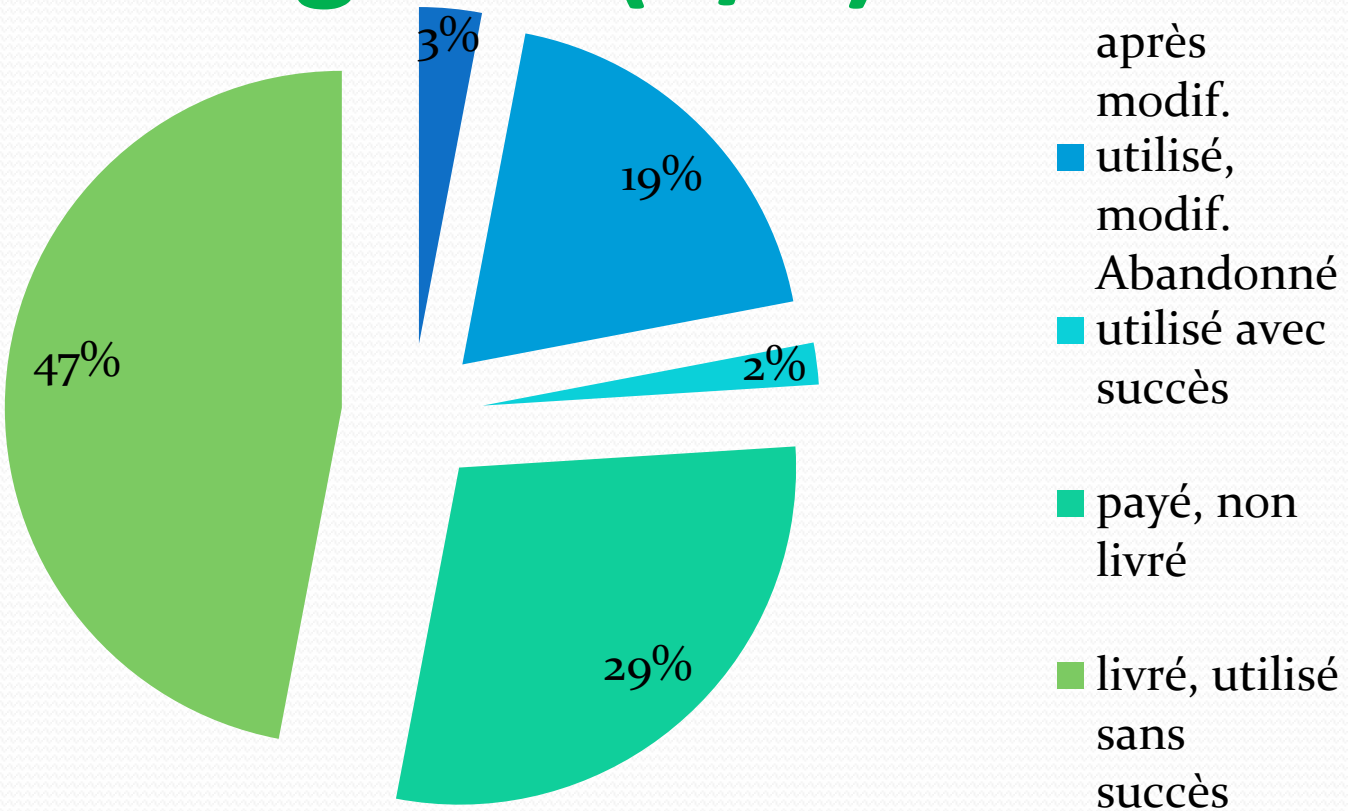
- Perte de la première sonde Mariner vers Venus suite à une erreur de programmation dans un programme Fortran
- Abandon du projet d'informatisation de la bourse londonienne après 4 ans de travail et 100 M£ de pertes
- Retard (2 ans) du premier vol de l'Eurofighter

→ Causes

Crise du logiciel (3/4)

- Difficulté de maîtrise **des coûts** (200 millions de dollars pour fabriquer OS-360)
- Difficulté de maîtrise **des délais** de réalisation (2 ans de retard pour les premiers compilateurs PL/1, Algol 68, ADA)

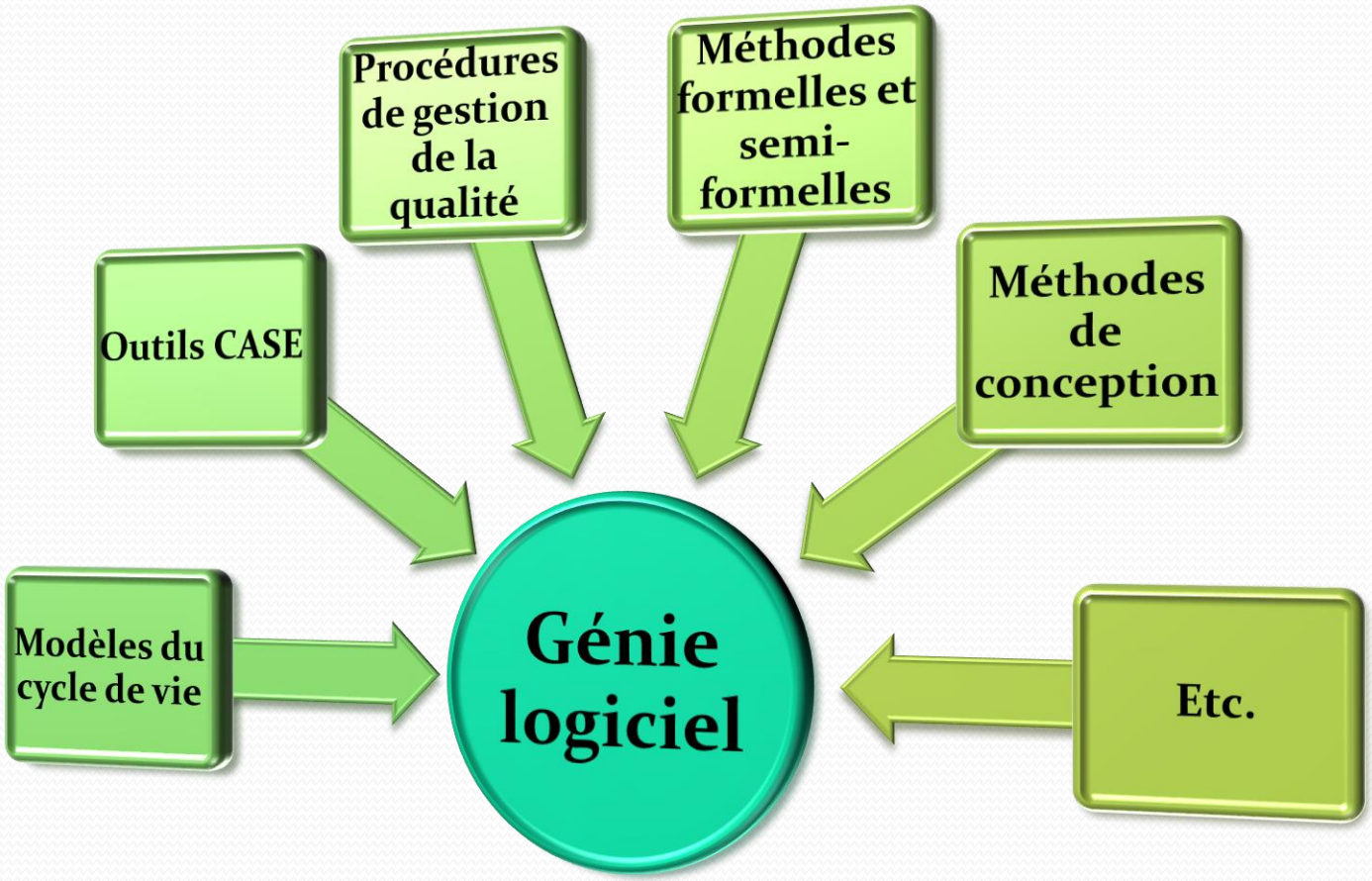
Crise du logiciel (4/4)



Rapports au congrès américains sur les logiciels

Solution

En octobre 1968 lors d'une conférence de l'OTAN à Garmisch-Partenkirchen en Allemagne



II. Définition et principes

- **Section 1 : Définition du génie logiciel**
- **Section 2 : Objectifs**
- **Section 3 : Caractéristiques**
- **Section 4 : Principes du génie logiciel**
- **Section 5 : Vue globale**

Définition du génie logiciel

- Le Génie logiciel (‘Software Engineering’ en anglais), GL, est un domaine des ‘sciences de l’ingénieur’ dont la finalité est la conception, la fabrication et la maintenance de systèmes logiciels complexes, sûrs et de qualité.
- Aujourd’hui, les économies de tous les pays développés sont dépendantes des systèmes logiciels. Par conséquent, l’ingénierie du logiciel a une place importante et une lourde responsabilité dans le bon fonctionnement des équipements et des institutions.

Définition du génie logiciel

- Le GL se définit souvent par opposition à la 'programmation', c'est-à-dire la production d'un programme par un individu unique, considérée comme 'facile'. Dans le cas du GL, il s'agit de la fabrication collective d'un système complexe, concrétisée par un ensemble de documents de conception, de programmes et de jeux de tests avec souvent de multiples versions, et considérée comme 'difficile'.

Définition du génie logiciel

- Ensemble de moyens humains et matériels mis en œuvre pour la construction de logiciels
- Le processus visant la résolution de problèmes posés par un client par le développement et l'évolution de systèmes logiciels de **grande taille** et de haute **qualité** en respectant les contraintes de **coûts** et de **temps**

Objectifs du génie logiciel

Le GL se préoccupe des procédés de fabrication des logiciels de façon à s'assurer que les quatre critères suivants soient satisfaits.

- Le système qui est fabriqué répond aux besoins (exigences) des utilisateurs (correction fonctionnelle).
- La qualité correspond au contrat de service initial. La qualité du logiciel est une notion multiforme qui recouvre notamment :
  - ***la validité :*** aptitude d'un logiciel à réaliser exactement les tâches définies par sa spécification,
  - ***la fiabilité :*** aptitude d'un logiciel à assurer de manière continue le service attendu,
  - ***la robustesse :*** aptitude d'un logiciel à fonctionner même dans des conditions anormales,

- ***l'extensibilité :*** facilité d'adaptation d'un logiciel aux changements de spécification,
- ***la réutilisabilité :*** aptitude d'un logiciel à être réutilisé en tout ou partie,
- ***la compatibilité :*** aptitude des logiciels à pouvoir être combinés les uns aux autres,
- ***l'efficacité :*** aptitude d'un logiciel à bien utiliser les ressources matérielles (mémoire, CPU…),
- ***la portabilité :*** facilité à être porté sur de nouveaux environnements matériels et/ou logiciels,
- ***la traçabilité :*** capacité à identifier et/ou suivre un élément du cahier des charges lié à un composant d'un logiciel,

- ***la vérifiabilité :*** facilité de préparation des procédures de recette et de certification,
- ***l'intégrité :*** aptitude d'un logiciel à protéger ses différents composants contre des accès ou des modifications non autorisés,
- ***la facilité d'utilisation, d'entretien, etc.***

- Les coûts restent dans les limites prévues au départ.
- Les délais restent dans les limites prévues au départ. Ces qualités sont parfois contradictoires. Il faut les pondérer selon les types d'utilisation. Il faut aussi distinguer les systèmes sur mesure et les produits logiciels de grande diffusion.

Caractéristiques du génie logiciel

- Le GL est en forte relation avec presque tous les autres domaines de l'informatique : langages de programmation (modularité, orientation objet, parallélisme…), bases de données (modélisation des données, accès aux données…), informatique théorique (automates, réseaux de Petri, types abstraits…), etc.
- Le GL est aussi en relation avec d'autres disciplines de l'ingénieur comme: ingénierie des systèmes et gestion de projets, sûreté et fiabilité des systèmes, etc.
- Les principales branches du GL couvrent :
  - la conception,
  - la validation/vérification,
  - la gestion de projet et l'assurance qualité,
  - les aspects socio-économiques.

Caractéristiques……*suite*

- Dans sa partie technique, le GL présente un spectre très large depuis des approches très *formelles* (spécifications formelles, approches transformationnelles, preuves de programmes) jusqu'à des démarches absolument *empiriques*. Cette variété reflète la variété des types de systèmes à produire :
  - Gros systèmes de gestion (ou systèmes d'information) ; le plus souvent des systèmes transactionnels construits autour d'une base de données;
  - Systèmes temps réel, qui doivent répondre à des événements dans des limites de temps prédéfinies et strictes ;

Caractéristiques……*suite*

- Systèmes distribués sur un réseau de machines (distribution des données et/ou des traitements), 'nouvelles architectures' liées à Internet ;
- Systèmes embarqués et systèmes critiques, interfacés avec un système à contrôler (ex: aéronautique, centrales nucléaires…).

- Le GL est difficile à étudier car très vaste, pas toujours très précis (beaucoup de discours généraux), foisonnant dans les concepts et le vocabulaire, sensible aux effets de modes. Les aspects techniques nécessitent une bonne maîtrise des outils fondamentaux de l'informatique (programmation, BD, système/réseau…). Dans les sections suivantes, nous allons passer en revue les *principes* généraux, les *techniques* spécialisées, les *méthodes* et *outils* pour le GL.

Principes du génie logiciel

- Rigueur
- Décomposition des problèmes en sous problèmes
- Modularité
- Généricité
- Construction incrémentale
- Anticipation et Application du changement
- abstraction.

Principes du génie logiciel….

- ***Rigueur.*** La production de logiciel est une activité créative, mais qui doit se conduire avec une certaine rigueur. Le niveau maximum de rigueur est la formalité, c'est-à-dire le cas où les descriptions et les validations s'appuient sur des notations et lois mathématiques. Il n'est pas possible d'être formel tout le temps : il faut bien construire la première description formelle à partir de connaissances non formalisées ! Mais dans certaines circonstances les techniques formelles sont utiles.
- ***Décomposition des problèmes en sous-problèmes.*** C'est une règle de bon sens qui consiste à considérer séparément différents aspects d'un problème afin d'en maîtriser la complexité. C'est un aspect de la stratégie générale du « diviser pour régner ».

- Elle prend une multitude de formes :
  - ✓ séparation dans le temps (les différents aspects sont abordés successivement),
  - ✓ séparation des qualités que l'on cherche à optimiser à un stade donné (ex : assurer la correction avant de se préoccuper de l'efficacité),
  - ✓ séparations des vues que l'on peut avoir d'un système (ex : se concentrer sur l'aspect données avant de considérer l'aspect traitements),
  - ✓ séparation du système en parties (sous-systèmes), - etc.

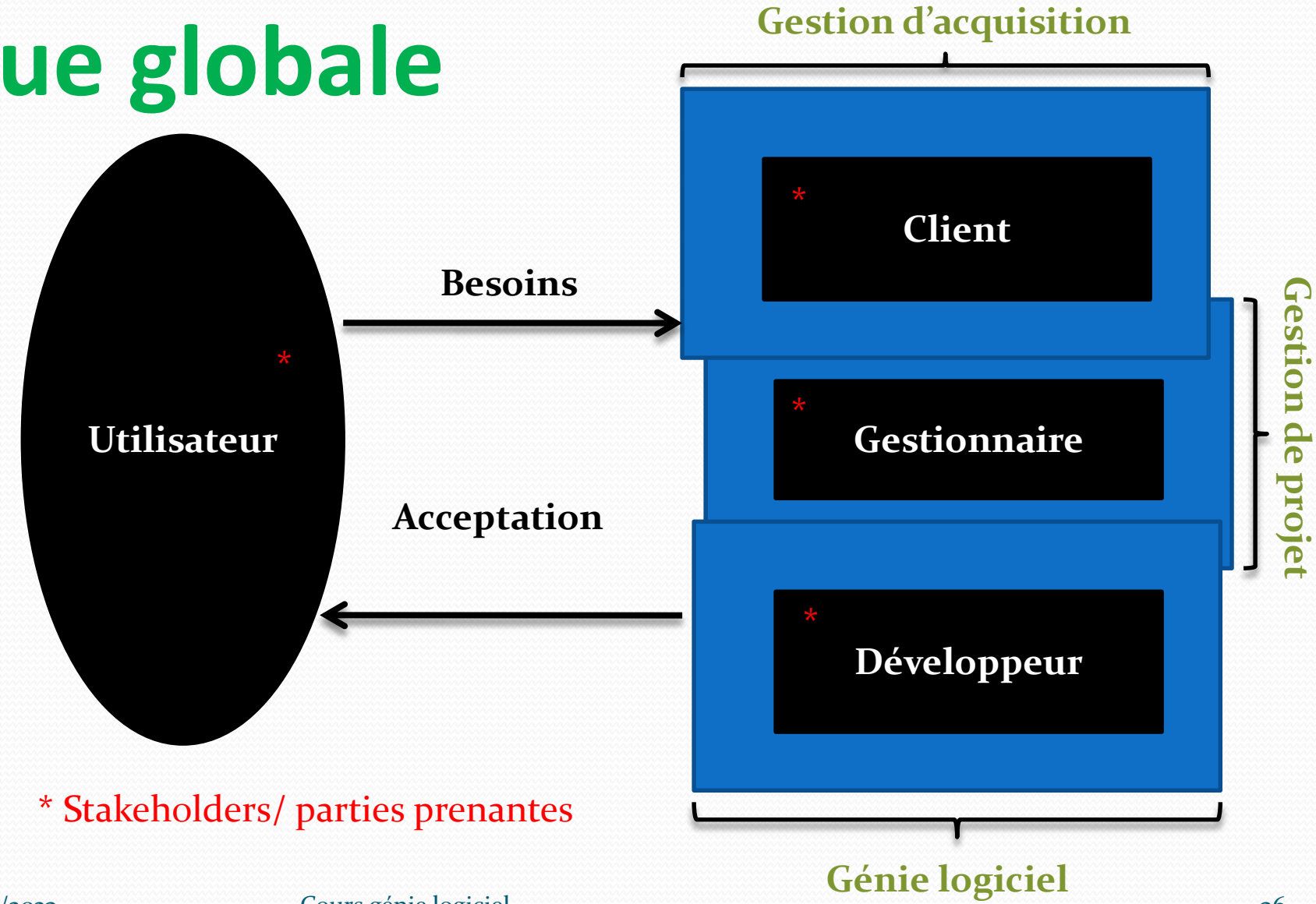
- ***Modularité.*** Un système est modulaire s'il est composé de sous-systèmes plus simples, ou modules. La modularité est une propriété importante de tous les procédés et produits industriels (cf. l'industrie automobile où le produit et le procédé sont très structurés et modulaires). La modularité permet de considérer séparément le contenu du module et les relations entre modules (ce qui rejoint l'idée de séparation des questions). Elle facilite également la réutilisation de composants bien délimités. Un bon découpage modulaire se caractérise par une forte cohésion interne des modules (ex : fonctionnelle, temporelle, logique…) et un faible couplage entre les modules (relations inter modulaires en nombre limité et clairement décrites). Toute l'évolution des langages de programmation (orientés objets notamment) vise à rendre plus facile une programmation modulaire, appelée aujourd'hui 'programmation par composants'.

- ***Abstraction.*** L'abstraction consiste à ne considérer que les aspects jugés importants d'un système à un moment donné, en faisant abstraction des autres aspects (c'est encore un exemple de séparation des problèmes). Une même réalité peut souvent être décrite à différents niveaux d'abstraction. L'abstraction permet une meilleure maîtrise de la complexité.
- ***Généricité.*** Il est parfois avantageux de remplacer la résolution d'un problème spécifique par la résolution d'un problème plus général. Cette solution générique (paramétrable ou adaptable) pourra être réutilisée plus facilement. Exemple : plutôt que d'écrire une identification spécifique à un écran particulier, écrire (ou réutiliser) un module générique d'authentification (saisie d'une identification - éventuellement dans une liste - et éventuellement d'un mot de passe).

- ***Construction incrémentale.*** Un procédé incrémental atteint son but par étapes en s'en approchant de plus en plus ; chaque résultat est construit en étendant le précédent. On peut par exemple réaliser d'abord un noyau des fonctions essentielles et ajouter progressivement les aspects plus secondaires. Ou encore, construire une série de prototypes 'simulant' plus ou moins complètement le système envisagé.
- ***Anticipation du changement.*** La caractéristique essentielle du logiciel, par rapport à d'autres produits, est qu'il est presque toujours soumis à des changements continuels (corrections d'imperfections et évolutions en fonction des besoins qui changent). Ceci requiert des efforts particuliers pour prévoir, faciliter et gérer ces évolutions inévitables. Il faut, par exemple, faire en sorte que les changements soient les plus localisés possibles (bonne modularité), ou encore être capable de gérer les multiples versions des modules et configurations des versions des modules, constituant des versions du produit complet.

- Il faut noter que les principes ci-dessus sont très *abstraits* et ne sont *pas utilisables directement*. Mais ils font partie du vocabulaire de base du génie logiciel. Ces principes ont un impact réel sur beaucoup d'aspects et constituent le type de connaissances le plus stable, dans un domaine où les outils, les méthodes et les techniques évoluent très vite.

Vue globale



* Stakeholders/ parties prenantes

Outils pour les activités du génie logiciel

Les activités de production du logiciel sont parfois complexes et nécessitent de la rigueur dans l'élaboration et l'exploitation des résultats fournis par chaque étape. C'est la raison pour laquelle on fait appel à des outils pour supporter les activités pendant tout le cycle de vie du logiciel. Ces outils sont appelés ***les outils CASE,*** (Computer-Aided Software Engineering). Les CASE sont classés selon leurs fonctions et phases où ils sont utilisés. On recense essentiellement les classes de CASE suivantes :

- outils de planification (comme PERT),
- outils d'édition de textes, d'images…,
- outils de traçabilité,

- outils de gestion de configurations,
- outils de prototypage,
- outils d'aide à la conception,
- compilateurs et environnements de programmation,
- outils d'analyse de programmes et code,
- outils de tests,
- outils de débogage,
- outils de documentation,
- outils de réingénierie,
- …