

CENTRALE
L Y O N

Introduction au Génie Logiciel

Définition
Evolution
Cycles de vie
Différentes vues

BTD/GL/Intro 1

CENTRALE
L Y O N

Définition

Génie Logiciel :

Un ensemble de méthodes, de techniques et d'outils permettant d'améliorer le processus de production et de maintenance des logiciels dans ses aspects techniques et dans son organisation.

BTD/GL/Intro 2

CENTRALE
L Y O N

Deux facettes importantes pour la réalisation de logiciels

- **acquisition de techniques de base :**
 - **algorithmie**
 - **structures de données**
- **maîtrise d'un projet complexe :**
 - **organisation (structure du groupe)**
 - **planification (étapes et répartition du travail)**
 - **suivi de l'état d'avancement**
 - ...

BTD/GL/Intro 3

CENTRALE
L Y O N

Réalisations des TP

Problème
...
Solution algorithmique

Codage et test

```
graph TD; A[Problème  
...  
Solution algorithmique] --> B[Codage et test]; B --> C[Open Box];
```

Durée de vie : 15 jours
Taille : quelques lignes ou pages
Travail : individuel

BTD/GL/Intro 4

CENTRALE
L Y O N

Réalisations des projets informatiques industriels

Problème

```
graph LR; A[Problème] --> B[Recherche de solution]; B --> C[Codage et tests]; C --> D[Maintenance];
```

Durée de vie : plusieurs années
Taille : des milliers lignes
Travail : collectif

BTD/GL/Intro 5

CENTRALE
L Y O N

Quelques indications sur l'évolution de l'informatique

- Matériels

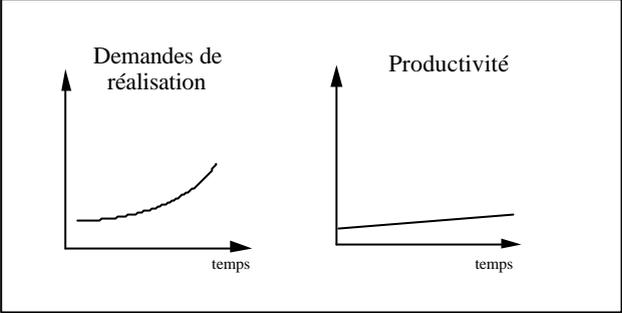
The figure contains three separate coordinate systems, each with a vertical y-axis and a horizontal x-axis labeled 'temps'.
1. The first graph, titled 'Performances', shows a curve that starts at a low point and increases exponentially over time.
2. The second graph, titled 'Prix', shows a curve that starts at a high point and decreases exponentially over time.
3. The third graph, titled 'Parc', shows a curve that starts at a low point and increases exponentially over time.

BTD/GL/Intro 6

CENTRALE
L Y O N

Quelques indications sur l'évolution de l'informatique

- Logiciels

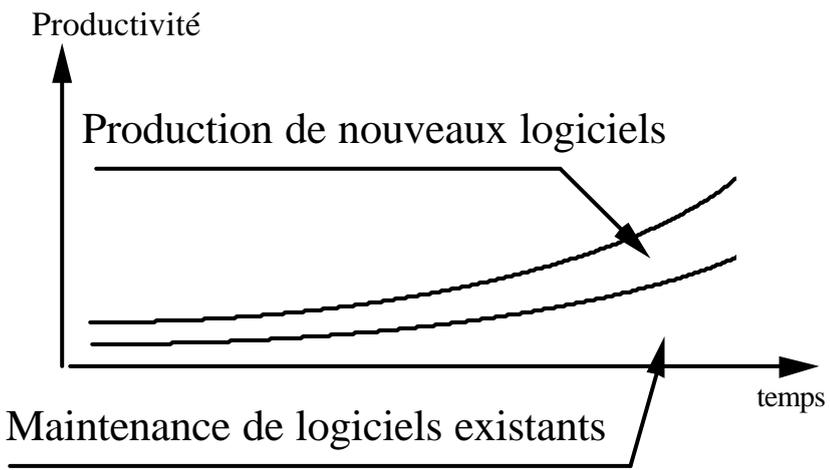


Ceci explique en partie la pénurie actuelle en informaticiens

BTD/GL/Intro 7

CENTRALE
L Y O N

Quelques indications sur l'évolution de l'informatique



Production de nouveaux logiciels

Maintenance de logiciels existants

BTD/GL/Intro 8

CENTRALE
L Y O N

Quelques données chiffrées

- **Coût relatif des phases de développement :**
 - Analyse et conception 30-35 %
 - Codage 15-20 %
 - Test 50 %
- **Nombre relatif d'erreurs :**
 - Conception 50 %
 - Programmation 30 %
 - Syntaxe 20 %
- **Coût relatif de correction de ces erreurs :**
 - Conception 80 %
 - Programmation et syntaxe 20 %

BTD/GL/Intro 9

CENTRALE
L Y O N

Maintenance

- **Coût de la maintenance :**
60 - 90 % de l'effort complet
- **Maintenance évolutive 60%**
 - nouvelles fonctionnalités
 - amélioration des performances
- **Maintenance adaptative 20%**
 - nouvel environnement matériel
 - nouvel environnement logiciel
- **Maintenance curative 20%**
 - débogage

BTD/GL/Intro 10

CENTRALE
L Y O N

Maintenance

- L 'activité de maintenance ne doit pas détruire la maintenabilité du logiciel
- Travail de maintenance doit se situer dans le cycle de vie du logiciel

BTD/GL/Intro 11

CENTRALE
L Y O N

Génie Logiciel

- Génie Logiciel => Engineering

Ingénierie de Logiciel

- Objectif :
Remplacer des scientifiques et des « artistes » par des ingénieurs

BTD/GL/Intro 12

CENTRALE
L Y O N

Travail d'ingénieur

- **Caractéristiques principales :**
 - Utiliser un processus de production systématique
 - Concevoir et de construire des programmes utilisables tenant compte des contraintes
 - Estimer, en avance, le coût et le délai
 - Utiliser des méthodes standards
 - S'appuyer sur des éléments réutilisables
 - Gérer le processus complet

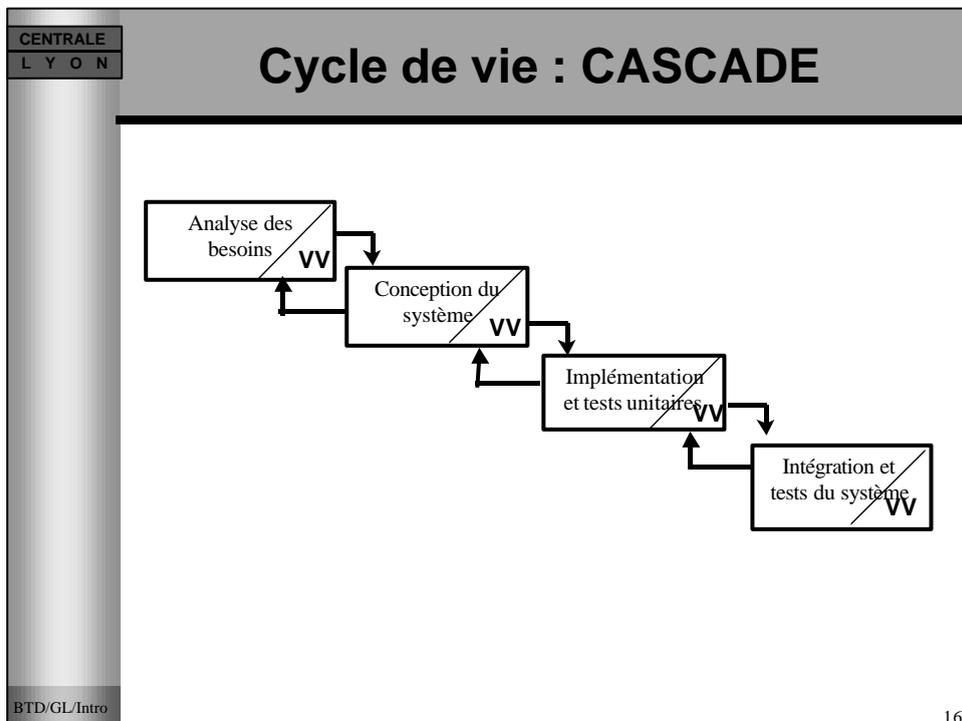
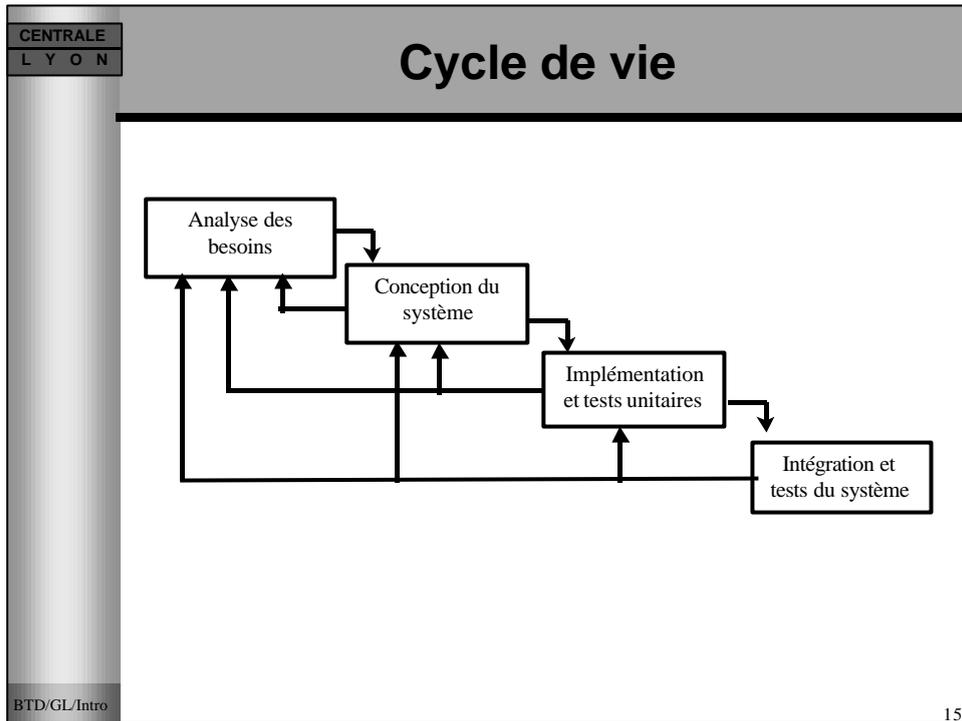
BTD/GL/Intro 13

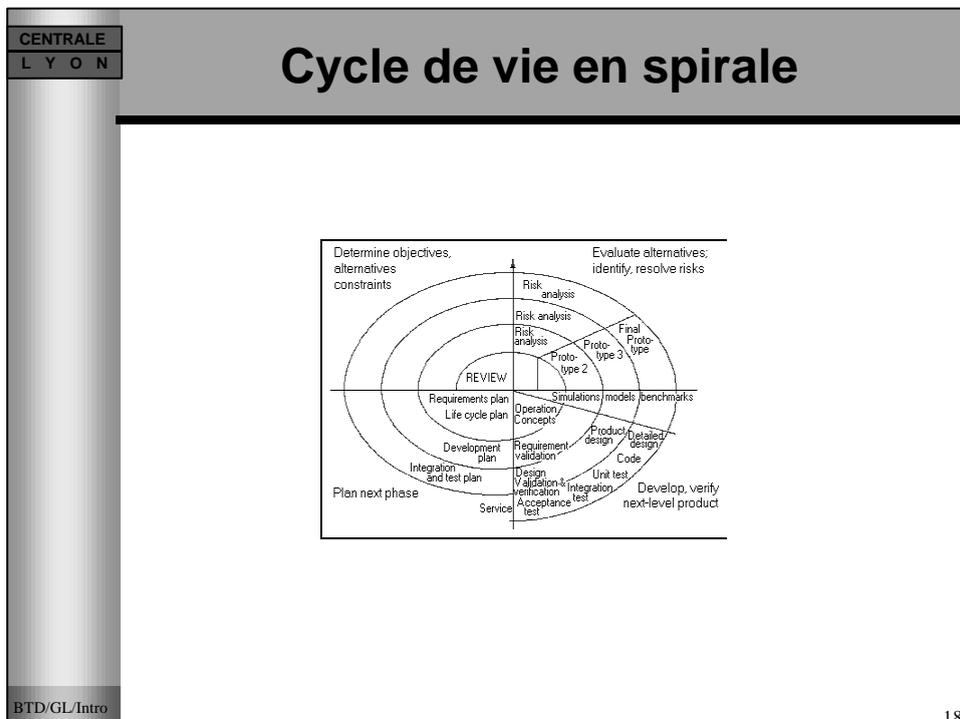
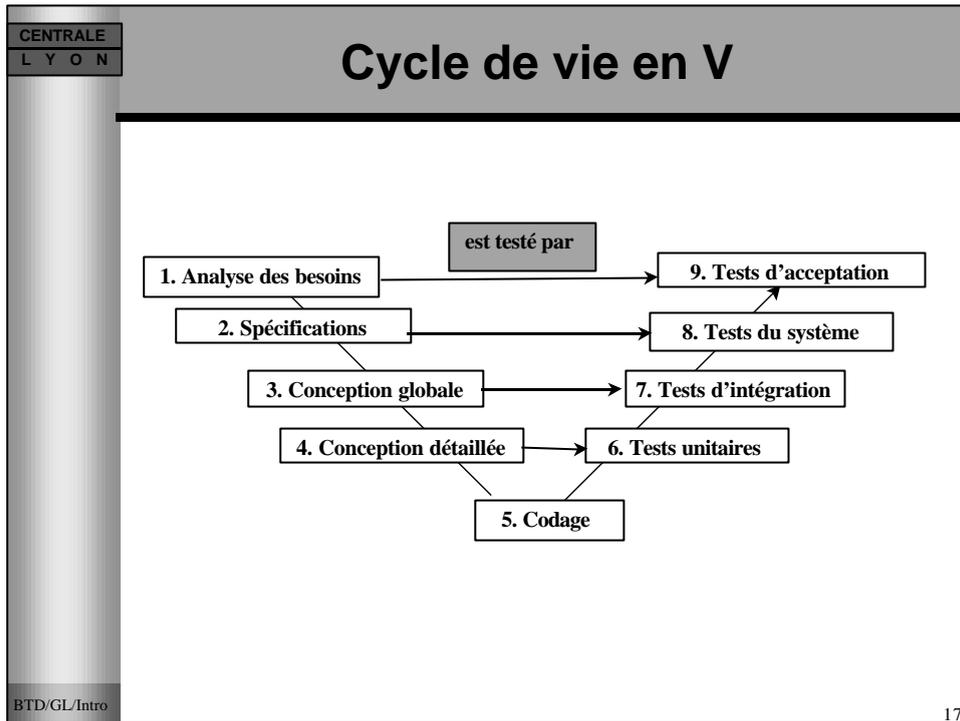
CENTRALE
L Y O N

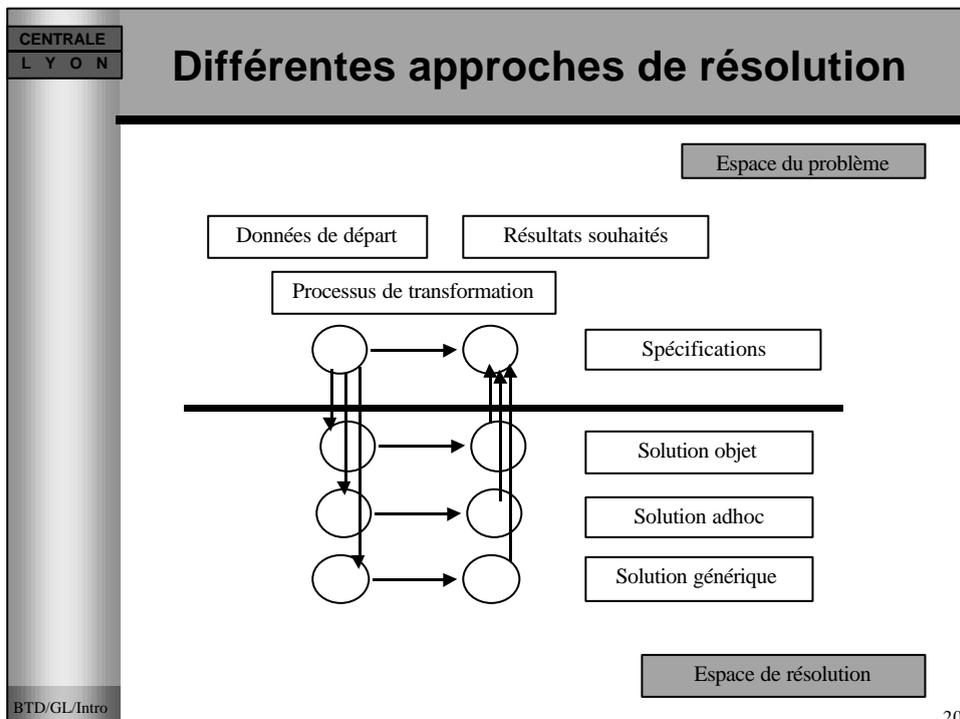
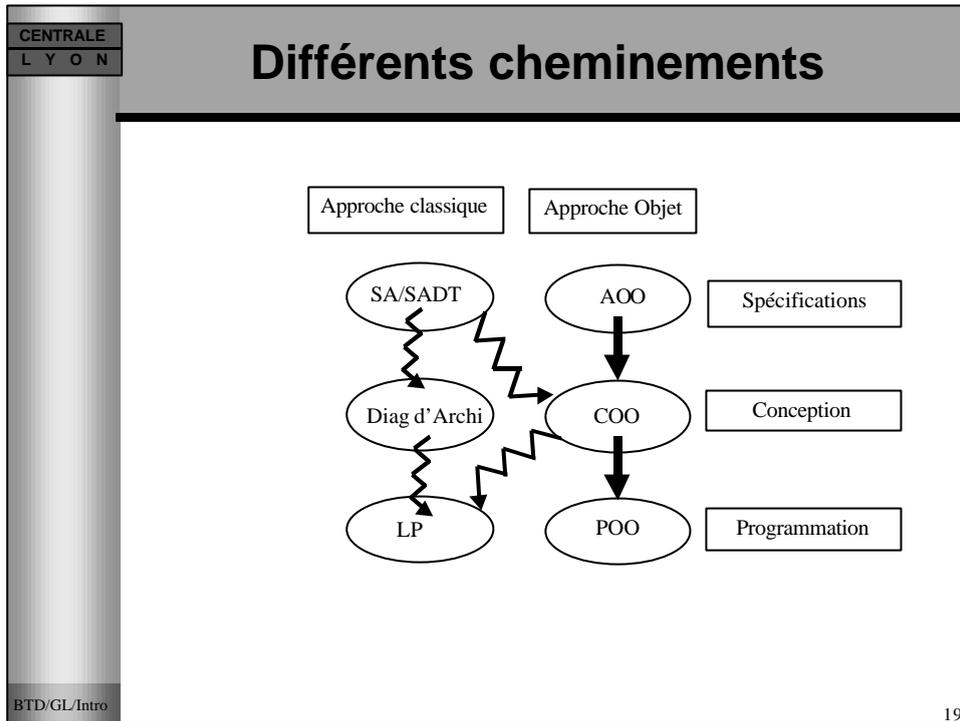
Gestion de projet logiciel

- **Six degrés de liberté d'un projet logiciel**
 - Axe du temps : déroulement chronologique appelé cycle de vie
 - Axe qualité : identifier la qualité souhaitée
 - Axe méthodes : façons de traiter le projet de façon technique et organisationnelle
 - Axe mesures : analyse quantitative et estimation à l'avance de certains paramètres
 - Axe langages : choix du moyen d'expression dans différentes étapes de réalisation du produit logiciel
 - Axe outils et environnement : choix des moyens de conception et de réalisation

BTD/GL/Intro 14







CENTRALE
L Y O N

Cycle de vie : Des unités de manipulation

BTD/GL/Intro

21

CENTRALE
L Y O N

L'état de l'art

- **Dans les années 1960**
 - Reconnaissance de la crise de logiciel et naissance de « SOFTWARE ENGINEERING »
- **Dans les années 1970**
 - Recherche de solutions
 - au début : programmation structurée
 - au milieu : concept de cycle de vie
 - vers la fin : spécification / conception
- **Dans les années 1980**
 - Méthodologies de développement de logiciel
 - Environnement de développement et matériel support
 - Mesures de productivité

BTD/GL/Intro

22

CENTRALE
L Y O N

L 'état de l'art (suite)

- **Dans les années 1990 :**
 - **Approche objet :**
 - Langages basés et orientés objets
 - Architectures orientées objets
 - Spécification objet
 - Réutilisation
 - Composants
- **Dans les années 2000 :**
 - **Unification des méthodes :**
 - Processus unifié (RUP)
 - UML
 - **Utilisation massive de Java**
 - composants JavaBeans, EntrepriseJavaBeans

BTD/GL/Intro 23

CENTRALE
L Y O N

Le défi de la qualité

- **Fiabilité (validité + robustesse) :** fabriquer des logiciels qui fonctionnent et qui garantissent le fonctionnement
- **Réutilisabilité :** fabriquer moins de logiciels
- **Extensibilité :** les modifications devraient être plus faciles
- **Efficacité :** exploiter « au mieux » les possibilités du matériel

BTD/GL/Intro 24

CENTRALE
L Y O N

Qualité par le langage ADA

- **Standardisation par l'adoption d'un langage unique pour tous et pour toutes les phases de développement (Conception, Réalisation, Maintenance)**
- **Augmentation de la qualité de la conception (mais aussi du temps nécessaire) et diminution du temps consacré aux tests**
- **Réduction du coût de formation (même outil pour tous les projets)**
- **Portabilité accrue et coût de portage diminué (très important avec l'évolution rapide des ordinateurs)**

BTD/GL/Intro 25

CENTRALE
L Y O N

Qualité par le langage ADA (suite)

- **Partage de logiciels encouragé car des outils, techniques et composants logiciels développés pendant un projet peuvent être utilisés dans les projets suivants**
- **Communication améliorée entre les ingénieurs utilisant le langage commun ; il en découle une compréhension durable des concepts de base**
- **Coûts de formation et de conversion réduits car une fois l'équipe formée, ses membres peuvent passer d'un projet à l'autre sans l'apprentissage d'un nouveau langage**

BTD/GL/Intro 26

CENTRALE
L Y O N

Qualité par la Conception et Programmation par Objets

- Une technique clé pour la réutilisabilité et extensibilité de logiciels
- Construction de logiciels par extension (héritage, surcharge) et combinaison
- Fiabilité par un concept de typage, introduction d'assertions et d'exceptions disciplinées

BTD/GL/Intro 27

CENTRALE
L Y O N

Architecture classique

- procédures
- données globales partagées

The diagram illustrates a classical architecture. At the top, a horizontal box labeled "Données partagées" (Shared Data) is connected by curved lines to four vertical boxes below, each labeled "Procédures" (Procedures). This indicates that all procedures have access to the same shared data.

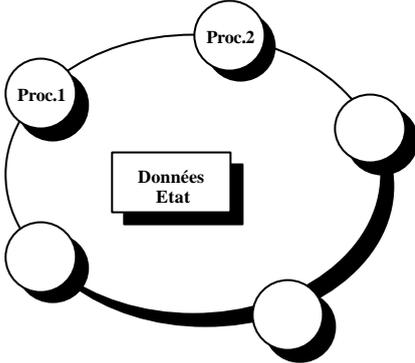
BTD/GL/Intro 28

CENTRALE
L Y O N

Approche objet

Objet: Données + Traitements
 Etat + Comportement

- Protection : encapsulation des données



BTD/GL/Intro

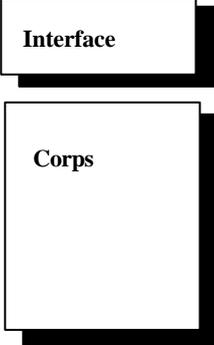
29

CENTRALE
L Y O N

Bases théoriques de l'approche objet

Les types abstraits :

- signature
- préconditions
- axiomes



BTD/GL/Intro

30

CENTRALE
L Y O N

Relations entre objets

BTD/GL/Intro

31

CENTRALE
L Y O N

Architectures objet

- Les objets proposent et utilisent des « services »
- Le fonctionnement découle de l'enchaînement d'évocation des services

BTD/GL/Intro

32

CENTRALE
L Y O N

Architectures prédéfinies

- Architecture pré-processeur - noyau - post-processeur
- Architecture Client-Serveur

```
graph LR; A[Pré-processeur] --> B[Noyau]; B --> C[Post-processeur]; D[Client] <--> E[Serveur]
```

BTD/GL/Intro 33

CENTRALE
L Y O N

Architecture de convergence

- A la fois prédéfinie et objet

Interface Homme-Machine
Objets métier (du domaine applicatif)
Infrastructure (base de données, gestion distribuée,...)

BTD/GL/Intro 34