

CENTRALE
L Y O N

Génie Logiciel

Définition
Constats
Ingénierie
Cycles de vie
Principaux aspects
Organisation

BTD/IC/GL 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.

Bertrand DAVID : Génie Logiciel

BTD/IC/GL 2

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Deux facettes importantes pour la réalisation de logiciels

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

BTD/IC/GL

3

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Réalisations des TP

Problème
...
Solution
algorithmique

Codage
et test

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

BTD/IC/GL

4

CENTRALE
L Y O N

Réalisations des projets informatiques industriels

Problème

```

graph LR
    P[Problème] --> R[Recherche de solution]
    R --> C[Codage et tests]
    C --> M[Maintenance]
        
```

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

BTD/IC/GL

5

CENTRALE
L Y O N

Quelques indications sur l'évolution de l'informatique

- Matériels

Performances

Prix

Parc

BTD/IC/GL

6

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Quelques indications sur l'évolution de l'informatique

- Logiciels

Ceci explique en partie la pénurie actuelle en informaticiens

BTD/IC/GL 7

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Quelques indications sur l'évolution de l'informatique

Productivité

Production de nouveaux logiciels

Maintenance de logiciels existants


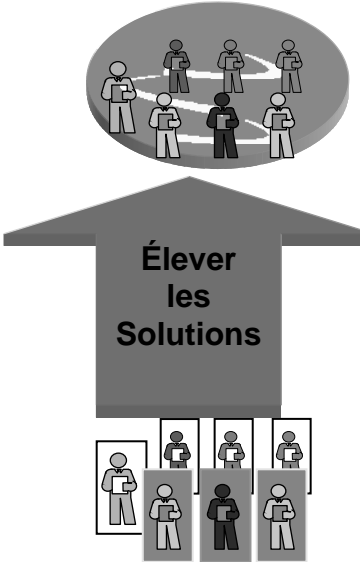
temps

BTD/IC/GL 8

CENTRALE
L Y O N

Le défi ...

Bertrand DAVID : Génie Logiciel

Le développement logiciel est un sport d'équipe

BTD/IC/GL

9

CENTRALE
L Y O N

Le problème

Bertrand DAVID : Génie Logiciel

“Seuls 26% des projets réussissent.”

Standish Group, CHAOS Rapport, 1998

- Ne répond aux besoins
- Est livré tard
- Coûte plus cher que prévu

BTD/IC/GL

10

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Qu'est ce qui rend les Systèmes Complexes ?

- Pressions de mise sur le marché
- Changement des besoins et des technologies
- Besoins de fiabilité
- Contraintes de performance
- Besoins de certification
- Besoins de distribution
- Prise en compte de l'existant
- Taille et distribution géographique de l'équipe
- Combinaison de tous ces facteurs

Coût

Coûts des logiciels complexes

Coût exponentiel

taille/échelle

$\text{Coût} = E \cdot (\text{Taille})^P$

BTD/IC/GL

11

CENTRALE
L Y O N

Les symptômes

- Inadéquation aux besoins
- Évolution des besoins
- Intégration logicielle
- Livraison chaotique
- Conflits d'équipe
- Tests inappropriés
- Découvertes tardives
- Mauvaises performances
- Difficile à maintenir

BTD/IC/GL

12

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Les principales causes

- Besoins ad-hoc
- Communications ambiguës
- Architectures fragiles
- Complexité écrasante
- Incohérences
- Tests insuffisants
- Évaluation subjective
- Développement en cascade
- Changements non contrôlés
- Automatisation insuffisante

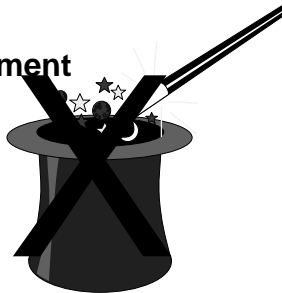
BTD/IC/GL 13

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Où est la solution magique ?

- Matériel plus performant
- Langages de programmation
- Méthodes formelles
- Environnements de développement
- Bases de données
- Middleware
- Processus
-



BTD/IC/GL 14

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Que sont les “*Best Practices*” ?

Un ensemble organisé et documenté de principes, méthodes et processus éprouvés qui augmentent la qualité et la productivité du développement logiciel.

BTD/IC/GL 15

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

Les meilleures pratiques

- Développement itératif
- Gestion des besoins
- Modélisation Visuelle
- Architecture à base de composants
- Vérification continue de la qualité
- Gestion des changements

BTD/IC/GL 16

CENTRALE L Y O N	<h2>Quelques données chiffrées</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> ● Coût relatif des phases de développement : <ul style="list-style-type: none"> → Analyse et conception 30-35 % → Codage 15-20 % → Test 50 % ● Nombre relatif d'erreurs : <ul style="list-style-type: none"> → Conception 50 % → Programmation 30 % → Syntaxe 20 % ● Coût relatif de correction de ces erreurs : <ul style="list-style-type: none"> → Conception 80 % → Programmation et syntaxe 20 %
BTD/IC/GL	17

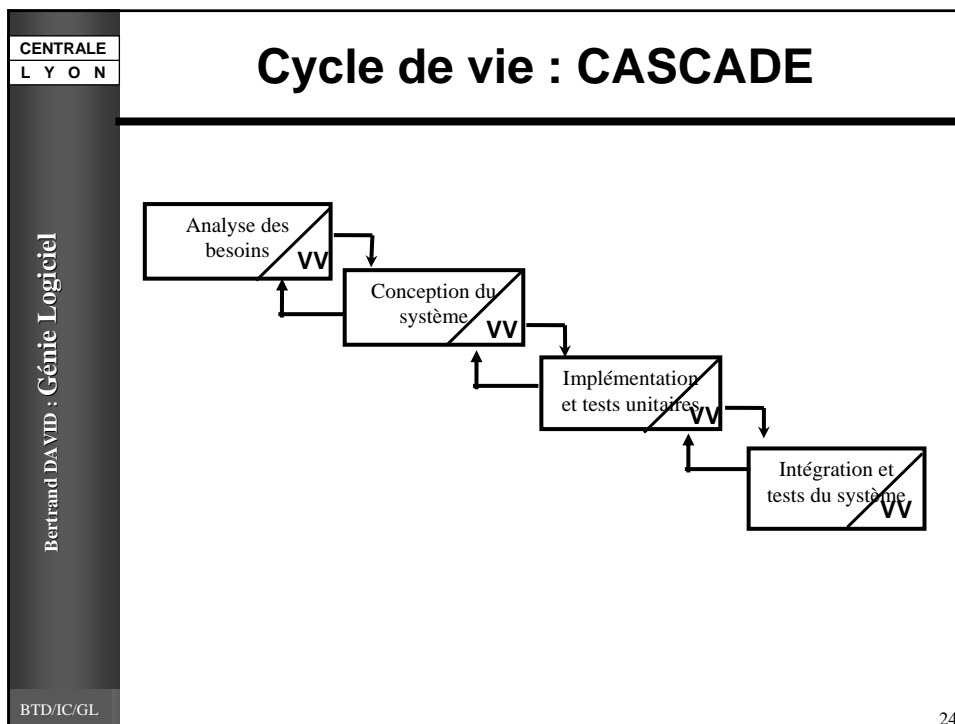
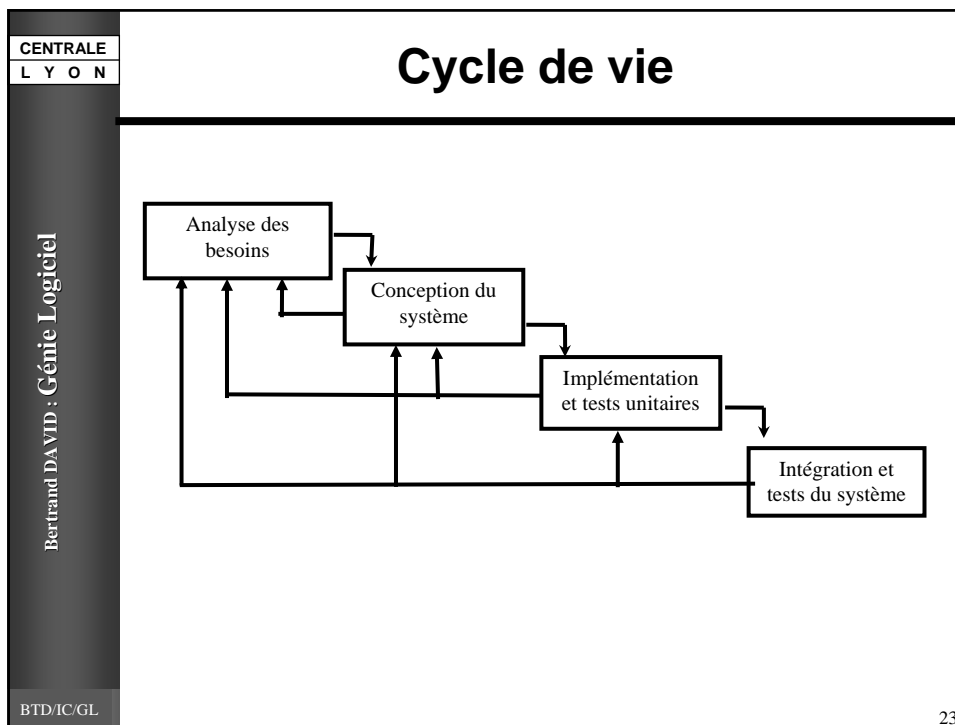
CENTRALE L Y O N	<h2>Maintenance</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> ● Coût de la maintenance : 60 - 90 % de l'effort complet ● Maintenance évolutive 60% <ul style="list-style-type: none"> → nouvelles fonctionnalités → amélioration des performances ● Maintenance adaptative 20% <ul style="list-style-type: none"> → nouvel environnement matériel → nouvel environnement logiciel ● Maintenance curative 20% <ul style="list-style-type: none"> → débogage
BTD/IC/GL	18

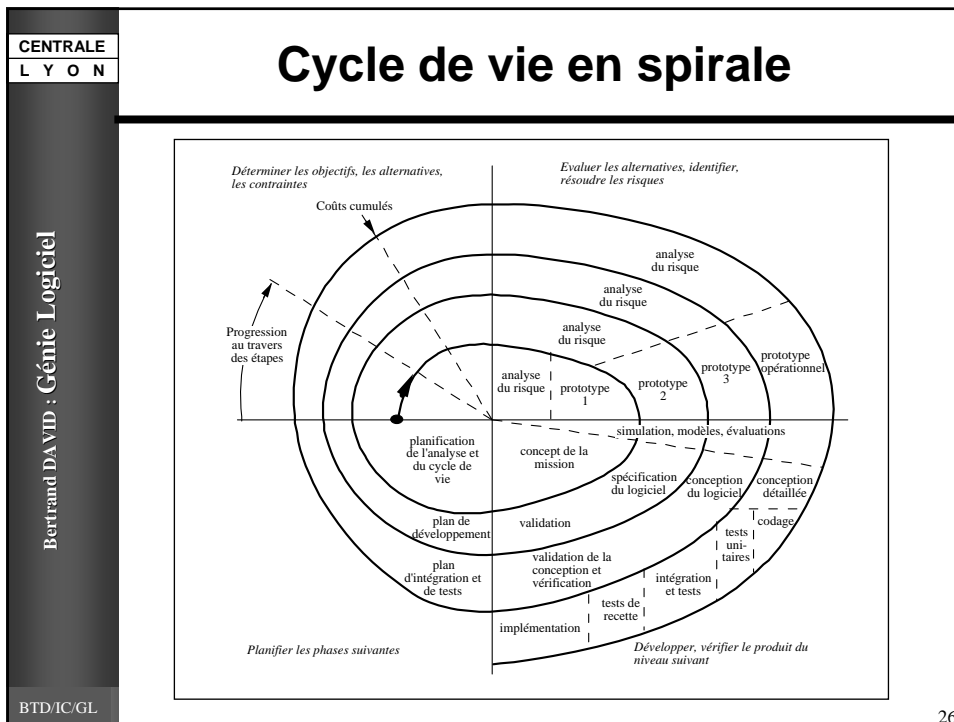
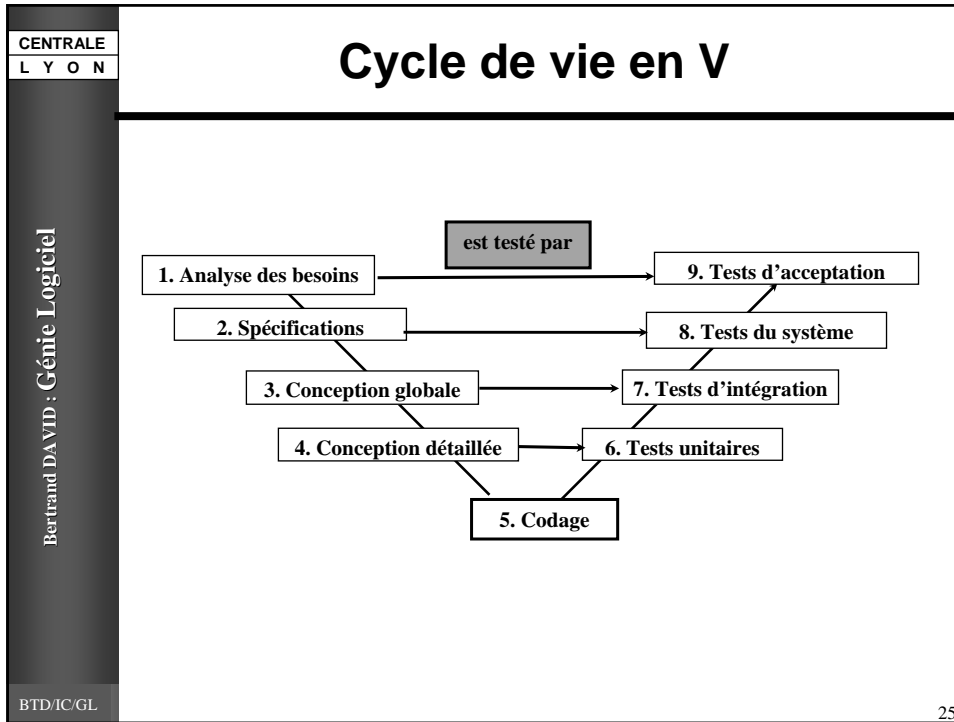
CENTRALE L Y O N	<h1>Maintenance</h1>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none">→ L'activité de maintenance ne doit pas détruire la maintenabilité du logiciel→ Travail de maintenance doit se situer dans le cycle de vie du logiciel
BTD/IC/GL	19

CENTRALE L Y O N	<h1>Génie Logiciel</h1>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none">● Génie Logiciel => Engineering <p style="text-align: center;"><u>Ingénierie de Logiciel</u></p> <ul style="list-style-type: none">● Objectif : Remplacer des scientifiques et des « artistes » par des ingénieurs
BTD/IC/GL	20

CENTRALE L Y O N	<h2>Travail d'ingénieur</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none">● <u>Caractéristiques principales :</u><ul style="list-style-type: none">→ 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/IC/GL	21

CENTRALE L Y O N	<h2>Gestion de projet logiciel</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none">● <u>Six degrés de liberté d'un projet logiciel</u><ul style="list-style-type: none">→ Axe du <u>temps</u> : déroulement chronologique appelé cycle de vie→ Axe <u>qualité</u> : identifier la qualité souhaitée→ Axe <u>méthodes</u> : façons de traiter le projet de façon technique et organisationnelle→ Axe <u>mesures</u> : analyse quantitative et estimation à l'avance de certains paramètres→ Axe <u>langages</u> : choix du moyen d'expression dans différentes étapes de réalisation du produit logiciel→ Axe <u>outils et environnement</u> : choix des moyens de conception et de réalisation
BTD/IC/GL	22





CENTRALE
L Y O N

Approches de développement (1/3)

- **Approche classique**

Approche classique

BTD/IC/GL

27

CENTRALE
L Y O N

Approches de développement (2/3)

- **Approches mixtes**

Approche classique

Approche Objet

BTD/IC/GL

28

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Approches de développement (3/3)

- Approche tout objet

```

    graph TD
      AO[Approche Objet] --> AOO((AOO))
      AOO --> COO((COO))
      COO --> POO((POO))
      AOO --- S[Spécifications]
      COO --- C[Conception]
      POO --- P[Programmation]
    
```

BTD/IC/GL 29

CENTRALE LYON

Bertrand DAVID : Génie Logiciel

Synthèse des approches

```

    graph TD
      subgraph Approche_classique [Approche classique]
        S1((SA/SADT))
        S2((Diag d'Archi))
        S3((LP))
      end
      subgraph Approche_Objet [Approche Objet]
        AOO((AOO))
        COO((COO))
        POO((POO))
      end
      S1 -.-> AOO
      S2 -.-> COO
      S3 -.-> POO
      AOO --> COO
      COO --> POO
      AOO --- S[Spécifications]
      COO --- C[Conception]
      POO --- P[Programmation]
    
```

BTD/IC/GL 30

CENTRALE
L Y O N

Développement de logiciels (1/3)

Données de départ

Résultats souhaités

Processus de transformation

Spécifications

Solution adhoc

Espace du problème

Espace de résolution

Bertrand DAVID : Génie Logiciel

BTD/IC/GL

31

CENTRALE
L Y O N

Développement de logiciels (2/3)

Données de départ

Résultats souhaités

Processus de transformation

Spécifications

Solution générique

Espace du problème

Espace de résolution

Bertrand DAVID : Génie Logiciel

BTD/IC/GL

32

CENTRALE
L Y O N

Développement de logiciels (3/3)

Espace du problème

Données de départ

Résultats souhaités

Processus de transformation

Spécifications

Solution objet

Espace de résolution

Bertrand DAVID : Génie Logiciel

BTD/IC/GL

33

CENTRALE
L Y O N

Développement de logiciels (synthèse)

Espace du problème

Données de départ

Résultats souhaités

Processus de transformation

Spécifications

Solution objet

Solution adhoc

Solution générique

Espace de résolution

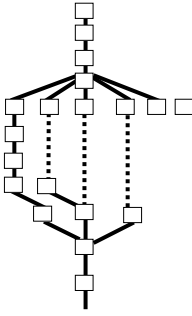
Bertrand DAVID : Génie Logiciel

BTD/IC/GL

34

CENTRALE
L Y O N

Cycle de vie : Des unités de manipulation



Bertrand DAVID : Génie Logiciel

BTD/IC/GL

35

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

Bertrand DAVID : Génie Logiciel

BTD/IC/GL

36

CENTRALE L Y O N	<h2>L'état de l'art (suite)</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> ● Dans les années 1990 : <ul style="list-style-type: none"> → Approche objet : <ul style="list-style-type: none"> ➢ Langages basés et orientés objets ➢ Architectures orientées objets ➢ Spécification objet ➢ Réutilisation ➢ Composants ● Dans les années 2000 : <ul style="list-style-type: none"> → Unification des méthodes : <ul style="list-style-type: none"> ➢ Processus unifié (RUP) ➢ UML ➢ Patterns ➢ Framework → Utilisation massive de Java <ul style="list-style-type: none"> ➢ composants JavaBeans, EntrepriseJavaBeans → Approche .NET <ul style="list-style-type: none"> ➢ Environnement, langages (x#),... → Open source (shareware, freeware,) → MDA : Model-Driven Architecture → IDM : Ingénierie Dirigée par les Modèles
BTD/IC/GL	37

CENTRALE L Y O N	<h2>Principaux aspects à traiter</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> ● <u>Langages</u> : Supports d'écriture de programmes ● <u>Méthodes</u> : Démarches proposées pour les différentes phases du développement ● <u>Processus</u> : Organisation globale des activités ● <u>Modèles</u> : Formalisations des produits et processus ● <u>Formalismes</u> : Expressions utilisées pour décrire les produits et processus ● <u>Composants et patterns</u> : supports de la réutilisation ● <u>Architecture</u> : Organisations des composants en squelettes (frameworks) réutilisables ● <u>Outils et Environnements</u> : Instrumentation des processus
BTD/IC/GL	38

CENTRALE L Y O N	<h1>Programme</h1>
Bertrand DAVID : Génie Logiciel	<p><u>Cours :</u></p> <ol style="list-style-type: none"> 1 Introduction au GL 2 Langages et LOO 3 UML1 et UML 2 4 OCL, RUP 5 Patterns, Programmation Orientée Aspect 6 MDA et Métamodélisation <p><u>Deux séances de TD :</u></p> <p>Tâches en ADA : simulateur de multiprocesseur Programmation Java, multithreading Mise en œuvre du simulateur du client-serveur en Java</p>
BTD/IC/GL	39

CENTRALE L Y O N	<h1>Organisation</h1>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> ● Organisation <ul style="list-style-type: none"> → 20 heures en tout → 6x2h de cours → 2x4h de BE → Cours : différents aspects du Génie Logiciel → Bureaux d'Etudes : application du cours à un cas concret → Modélisation en UML & programmation en Java ● Evaluation : <ul style="list-style-type: none"> → Note de BE basée sur le compte-rendu et la démo → Note du Cours : test sous forme de QCM
BTD/IC/GL	40