

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.

Bertrand DAVID : Génie Logiciel

BTD/GL/Intro 2

CENTRALE
L Y O N

Deux facettes importantes pour la réalisation de logiciels

Bertrand DAVID : Génie Logiciel

- **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

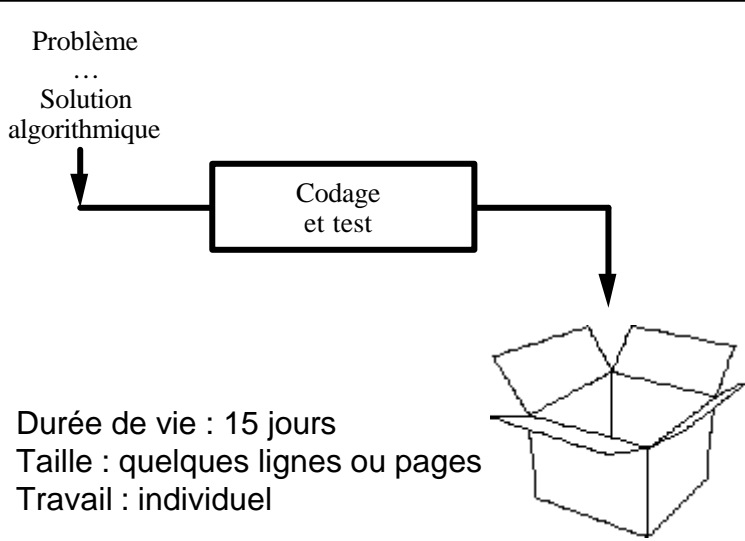
Réalisations des TP

Bertrand DAVID : Génie Logiciel

Problème
...
Solution algorithmique

Codage et test

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



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

Bertrand DAVID : Génie Logiciel

BTD/GL/Intro

5

CENTRALE
L Y O N

Quelques indications sur l'évolution de l'informatique

- Matériels

Performances: Graph showing an upward curve over time.

Prix: Graph showing a downward curve over time.

Parc: Graph showing an upward curve over time.

Bertrand DAVID : Génie Logiciel

BTD/GL/Intro

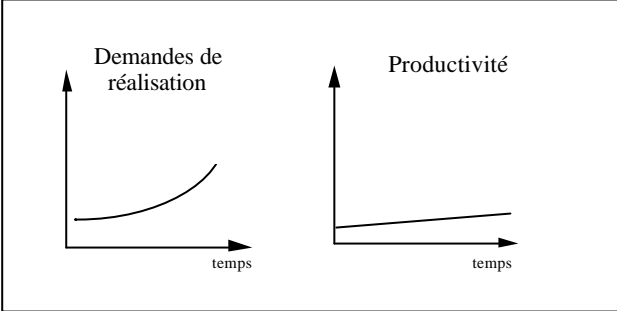
6

CENTRALE
L Y O N

Quelques indications sur l'évolution de l'informatique

Bertrand DAVID : Génie Logiciel

- Logiciels



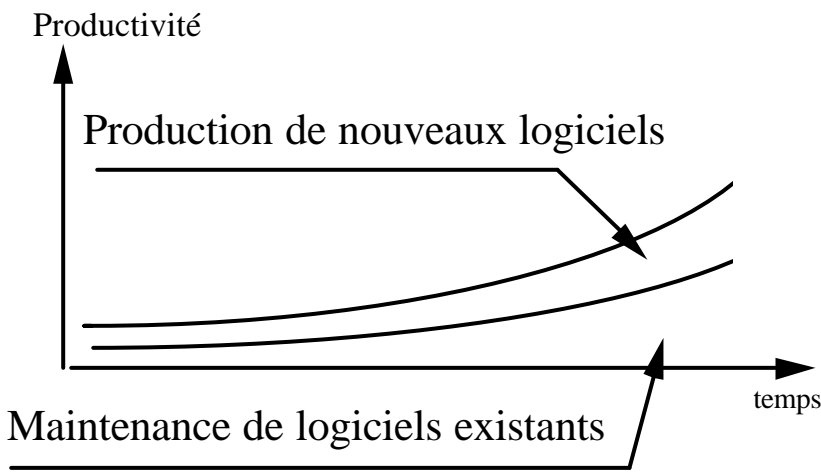
Ceci explique en partie la pénurie actuelle en informaticiens

BTD/GL/Intro 7

CENTRALE
L Y O N

Quelques indications sur l'évolution de l'informatique

Bertrand DAVID : Génie Logiciel



Productivité

Production de nouveaux logiciels

Maintenance de logiciels existants

temps

BTD/GL/Intro 8

CENTRALE
L Y O N

Quelques données chiffrées

Bertrand DAVID : Génie Logiciel

- **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

Bertrand DAVID : Génie Logiciel

- **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

Bertrand DAVID : Génie Logiciel

- 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

Bertrand DAVID : 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

Bertrand DAVID : Génie Logiciel

- **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

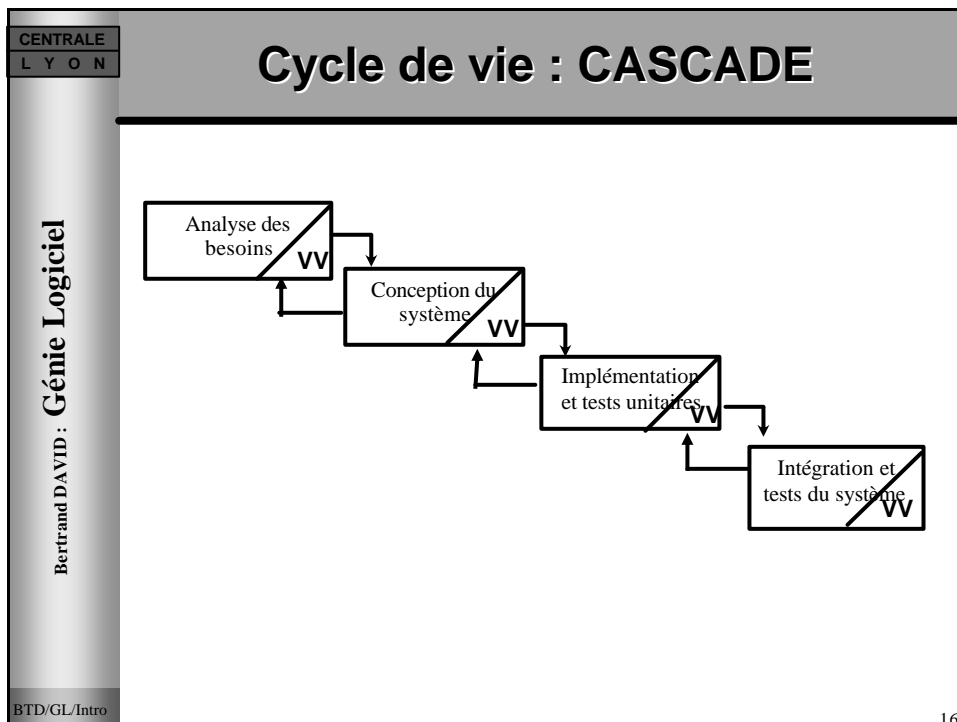
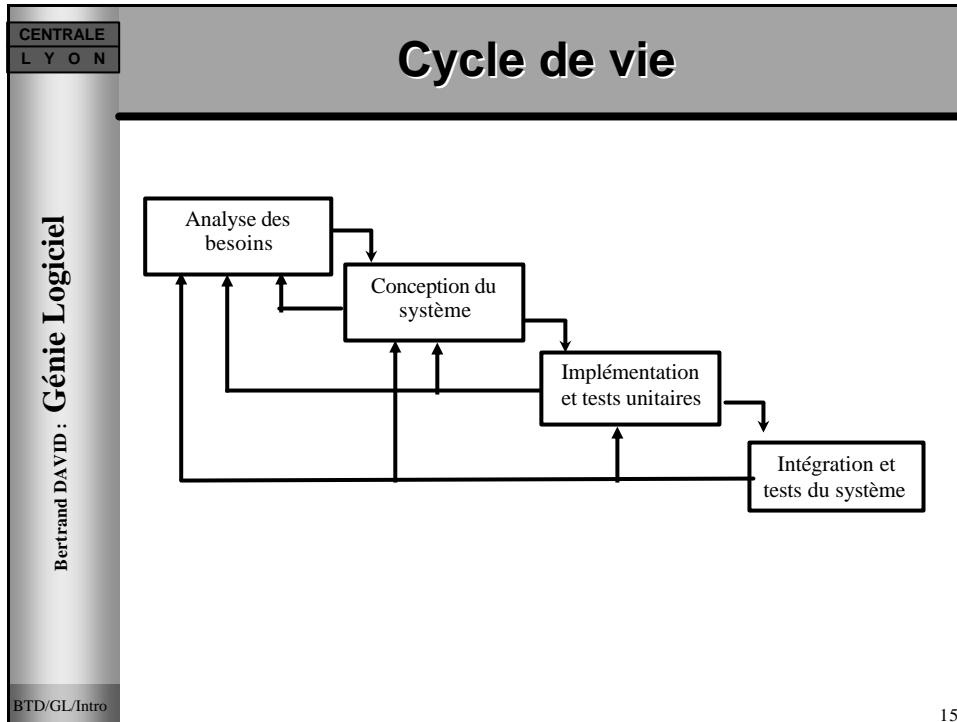
CENTRALE
L Y O N

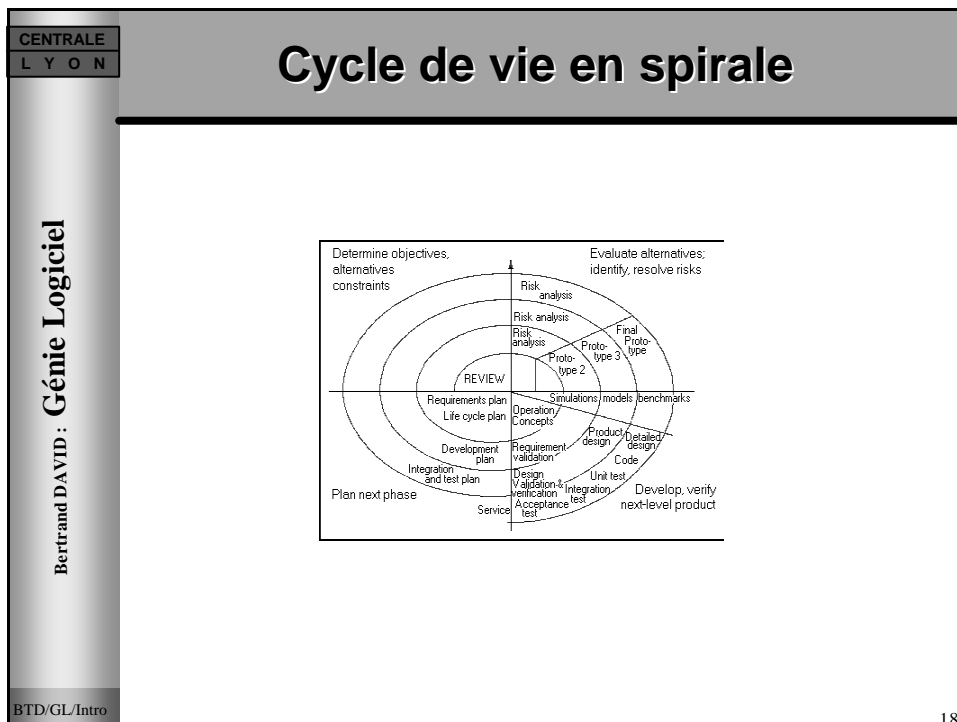
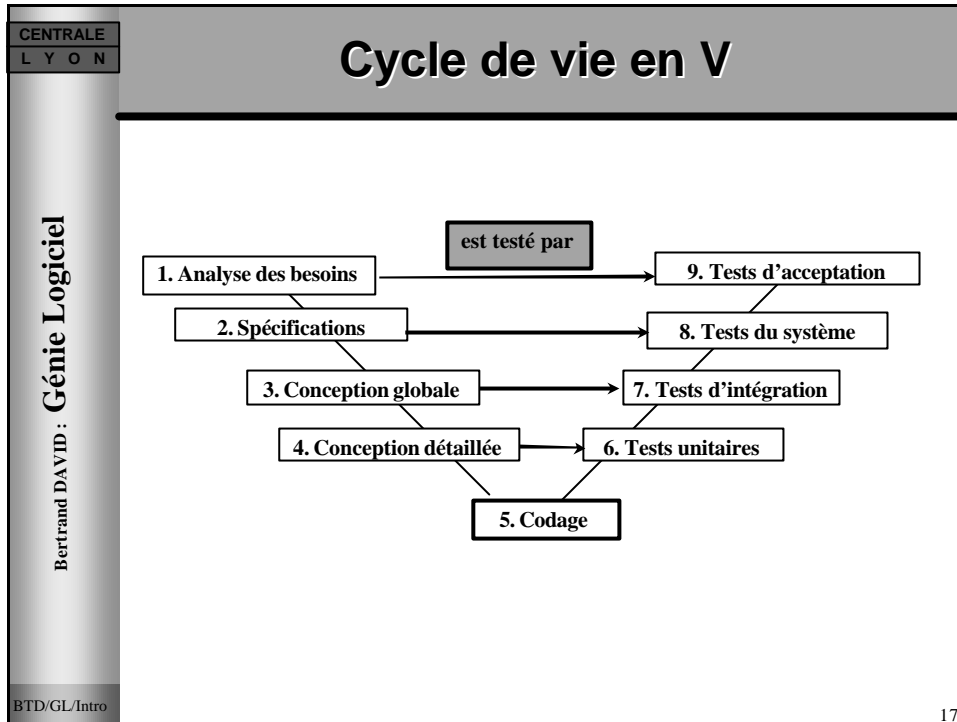
Gestion de projet logiciel

Bertrand DAVID : Génie 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 LYON

Approches de développement (1/3)

• Approche classique

Approche classique

```
graph TD; A[SA/SADT] --> B[Diag d'Archi]; B --> C[LP]; A --- S[Spécifications]; B --- Co[Conception]; C --- P[Programmation];
```

Spécifications

Conception

Programmation

BTD/GL/Intro

19

CENTRALE LYON

Approches de développement (2/3)

• Approches mixtes

Approche classique Approche Objet

```
graph TD; A[SA/SADT] --> B[COO]; C[LP] --> B; B --> D[POO]; A --- S[Spécifications]; B --- Co[Conception]; D --- P[Programmation];
```

Spécifications

Conception

Programmation

BTD/GL/Intro

20

CENTRALE
L Y O N

Approches de développement (3/3)

Bertrand DAVID : Génie Logiciel

- Approche tout objet

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

BTD/GL/Intro 21

CENTRALE
L Y O N

Synthèse des approches

Bertrand DAVID : Génie Logiciel

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

BTD/GL/Intro 22

CENTRALE
L Y O N

Développement de logiciels (1/3)

Espace du problème

Données de départ Résultats souhaités

Processus de transformation

Spécifications

Solution adhoc

Espace de résolution

Bertrand DAVID : Génie Logiciel

BTD/GL/Intro

23

The diagram illustrates the software development process. It is divided into two main horizontal sections by a thick line. The top section is labeled 'Espace du problème' (Problem Space) and contains a box for 'Données de départ' (Starting Data) on the left and 'Résultats souhaités' (Desired Results) on the right. A box labeled 'Processus de transformation' (Transformation Process) is positioned above a horizontal line. Two circles are connected by a right-pointing arrow, with a box labeled 'Spécifications' (Specifications) to their right. The bottom section is labeled 'Espace de résolution' (Resolution Space) and contains a box labeled 'Solution adhoc' (Ad-hoc Solution). Two circles are connected by a right-pointing arrow, with a box labeled 'Solution adhoc' to their right. Vertical arrows connect the circles between the two spaces: a downward arrow from the top-left circle to the bottom-left circle, and an upward arrow from the bottom-right circle to the top-right circle.

CENTRALE
L Y O N

Développement de logiciels (2/3)

Espace du problème

Données de départ Résultats souhaités

Processus de transformation

Spécifications

Solution générique

Espace de résolution

Bertrand DAVID : Génie Logiciel

BTD/GL/Intro

24

The diagram illustrates the software development process. It is divided into two main horizontal sections by a thick line. The top section is labeled 'Espace du problème' (Problem Space) and contains a box for 'Données de départ' (Starting Data) on the left and 'Résultats souhaités' (Desired Results) on the right. A box labeled 'Processus de transformation' (Transformation Process) is positioned above a horizontal line. Two circles are connected by a right-pointing arrow, with a box labeled 'Spécifications' (Specifications) to their right. The bottom section is labeled 'Espace de résolution' (Resolution Space) and contains a box labeled 'Solution générique' (Generic Solution). Two circles are connected by a right-pointing arrow, with a box labeled 'Solution générique' to their right. Vertical arrows connect the circles between the two spaces: a downward arrow from the top-left circle to the bottom-left circle, and an upward arrow from the bottom-right circle to the top-right circle.

CENTRALE LYON

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/GL/Intro

25

The diagram illustrates the software development process as a transformation. It is divided into two main horizontal sections by a thick line. The top section is labeled 'Espace du problème' (Problem Space) and contains boxes for 'Données de départ' (Starting data) and 'Résultats souhaités' (Desired results). Below these is a box for 'Processus de transformation' (Transformation process), which is represented by two circles connected by a horizontal arrow pointing right. To the right of this process is a box for 'Spécifications' (Specifications). The bottom section is labeled 'Espace de résolution' (Resolution Space) and contains a box for 'Solution objet' (Object solution). This section also features two circles connected by a horizontal arrow pointing right. Vertical lines connect the top and bottom circles, indicating a mapping or derivation between the problem space and the resolution space.

CENTRALE LYON

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/GL/Intro

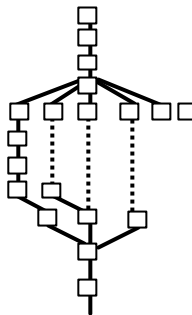
26

This diagram illustrates the synthesis phase of software development. It follows the same layout as slide 25, with 'Espace du problème' at the top and 'Espace de résolution' at the bottom. The 'Processus de transformation' is shown with two circles and a horizontal arrow. However, in this diagram, there are four such pairs of circles, one above the other. Each pair is connected by a horizontal arrow pointing right. Vertical lines connect the top and bottom circles of each pair. To the right of these pairs are boxes for 'Spécifications', 'Solution objet', 'Solution adhoc', and 'Solution générique', indicating different levels or types of solutions derived from the same starting data and desired results.

CENTRALE
L Y O N

Cycle de vie : Des unités de manipulation

Bertrand DAVID : Génie Logiciel



BTD/GL/Intro

27

CENTRALE
L Y O N

L'état de l'art

Bertrand DAVID : Génie Logiciel

- **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

28

CENTRALE
L Y O N

L'état de l'art (suite)

Bertrand DAVID : Génie Logiciel

- 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
 - Patterns
 - Framework
 - Utilisation massive de Java
 - composants JavaBeans, EntrepriseJavaBeans

BTD/GL/Intro 29

CENTRALE
L Y O N

Le défi de la qualité

Bertrand DAVID : Génie Logiciel

- 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

Mais aussi

- Utilité, utilisabilité, acceptabilité

BTD/GL/Intro 30

CENTRALE L Y O N	<h2>Qualité par le langage ADA</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none">● 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	31

CENTRALE L Y O N	<h2>Qualité par le langage ADA (suite)</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none">● 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	32

CENTRALE L Y O N	Qualité par la Conception et Programmation par Objets
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none">● 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	33