

CENTRALE  
L Y O N

# UML et le Développement de Logiciels

**Les spécifications :**

- Principes
- UML

BTD/UML-DL 1

CENTRALE  
L Y O N

# Cycle en V

Bertrand DAVID :  
UML et le Développement de Logiciels

The diagram illustrates the V-model software development cycle. It consists of nine numbered phases arranged in a V-shape. The left side of the V represents the development phases, and the right side represents the testing phases. The bottom of the V is the coding phase. Horizontal arrows labeled 'Est en relation' (is related to) connect corresponding phases on both sides of the V, indicating that testing phases are directly related to their corresponding development phases. The phases are: 1. Analyse des besoins, 2. Spécification, 3. Conception globale, 4. Conception détaillée, 5. Codage, 6. Tests unitaires, 7. Tests d'intégration, 8. Tests du système, and 9. Tests d'acceptation.

BTD/UML-DL 2

CENTRALE  
L Y O N

## Spécifications

Bertrand DAVID :  
UML et le Développement de Logiciels

- Traduire le Cahier des Charges dans un langage plus formalisé.
- Vérifier la cohérence des informations.
- Ce dossier est réalisé par l'équipe de développement

- Elaborer le dossier de spécifications
- Rédiger le manuel provisoire d'utilisation et d'exploitation
- Rédiger le dossier de validation

BTD/UML-DL

3

CENTRALE  
L Y O N

## Questionner

Bertrand DAVID :  
UML et le Développement de Logiciels

- Première étape critique pour l'équipe de développement  
L'étape doit répondre aux questions suivantes :

- QUOI ? Identifier le besoin et produire les spécifications
- POURQUOI ? Comprendre le contexte du problème
- COMMENT ? Identifier des contraintes
- QUAND ? Etablir le planning et comprendre les relations entre activités
- OU ? Situer les flux d'information
- QUI ? Identifier les parties concernées

BTD/UML-DL

4

CENTRALE  
L Y O N

# Analyser

Bertrand DAVID :  
UML et le Développement de Logiciels

- **borner le contexte du problème**
- **identifier les éléments constitutifs, selon la méthode utilisée :**
  - objets / entités
  - attributs / propriétés
  - fonctions / actions
  - liens / relations
- **construire un modèle**
- **Modélisation est une activité fondamentale du Génie Logiciel.**
- **La plupart de méthodes d'analyse - spécification consiste en construction de modèle.**

BTD/UML-DL 5

CENTRALE  
L Y O N

# Modéliser

Bertrand DAVID :  
UML et le Développement de Logiciels

- **modéliser quoi ?**

activités - données - dialogues
- **modéliser comment ?**

formalismes - schémas - outils

Objectif découvrir la structure et le comportement (signification, sémantique)

Pour la structure : constitution d'un réseau de liens entre éléments identifiés (activités, données, utilisateurs, ...)

BTD/UML-DL 6

CENTRALE  
L Y O N

## S 'appropriier

- L'objectif principal est de comprendre pour pouvoir communiquer
- L'appropriation du problème par l'équipe de développement

Bertrand DAVID :  
UML et le Développement de Logiciels

BTD/UML-DL

7

CENTRALE  
L Y O N

## Approche Système

- Système de décision (pilotage)
- Système d'information
- Système opérant

Bertrand DAVID :  
UML et le Développement de Logiciels

BTD/UML-DL

8

CENTRALE  
L Y O N

Bertrand DAVID :  
UML et le Développement de Logiciels

## Plan type du dossier de spécifications

Objectif : Réécrire le Cahier des Charges en le rendant plus complet et plus homogène.

1. Introduction
2. Objectifs et contraintes (de développement et d'exploitation)
3. Spécifications générales (délimiter le projet)
4. Spécifications détaillées
5. Spécifications de l'environnement

BTD/UML-DL 9

CENTRALE  
L Y O N

Bertrand DAVID :  
UML et le Développement de Logiciels

## Moyens d'expression formelle (plus ou moins) :

- SADT
- DFD : Diagrammes de flux de données
- E-A : Modèle Entité Association
- UML
  
- Notre choix : UML

BTD/UML-DL 10

CENTRALE  
L Y O N

## UML (Unified Modeling Language)

Bertrand DAVID :  
UML et le Développement de Logiciels

**Un formalisme issu des méthodes :**

- OMT - James Rumbaugh,
- Booch - Grady Booch,
- OOSE - Ivar Jacobson

Objectif global, proposer un support complet comportant :

- Processus et cycle de vie
- Méthodes
- Formalismes et outils

UML constitue le premier pas : formalisme unifié basé sur une présentation triaxiale d'un objet

- Les cas d'utilisation : le savoir-faire
- La description : propriétés, associations
- La dynamique : états, événements

BTD/UML-DL 11

CENTRALE  
L Y O N

## Modéliser avec UML

Bertrand DAVID :  
UML et le Développement de Logiciels

- **UML ne définit pas le processus d'élaboration des modèles**
- **Mais suggère 3 démarches**
  - Travailler de façon itérative et incrémentale : l'idée est de bâtir un prototype et de l'améliorer
  - Exprimer les besoins utilisateurs : les utilisateurs guident la réalisation du modèle (validation des livrables)
  - Se centrer sur l'architecture : en utilisant les différentes vues de UML proposées par Ph. Kruchten (voir ci-après)

BTD/UML-DL 12

CENTRALE  
L Y O N

Bertrand DAVID :  
UML et le Développement de Logiciels

## 4 +1 vues centrées sur l'architecture

- 5 vues différentes :
  - La vue logique se concentre sur l'abstraction et l'encapsulation (classes, stéréotypes, diagramme de classes)
  - La vue implémentation est une vue de bas niveau composée de modules logiciels
  - La vue des processus utilisée dans un environnement multitâches
  - La vue déploiement est utilisée dans un environnement distribué pour représenter l'architecture et la topologie
  - La vue des cas d'utilisation ou de scénarios est une vue qui sert de guide à tous les autres

BTD/UML-DL 13

CENTRALE  
L Y O N

Bertrand DAVID :  
UML et le Développement de Logiciels

## UML : la vue logique

- Décomposition orientée objets
  - Les objets, les classes
  - Relations entre objets et comportement des objets
    - Diagramme d'activités
    - Diagramme d'états
    - Diagramme de collaboration
    - Diagramme de classes
    - Diagramme de dépendance de dossiers logiques
  - Structuration sous forme de dossier / répertoire (ou package)

BTD/UML-DL 14

CENTRALE  
L Y O N

Bertrand DAVID :  
UML et le Développement de Logiciels

## UML : la vue des processus

- Décomposition en tâches / Processus
- Organisation en groupe de processus
- Les interactions entre processus
- Synchronisation et communication parallèle
- Différents types de communication
  - Asynchrone
  - Synchrone
  - Implicite, etc.
- Les paramètres clés
  - Disponibilité, fiabilité
  - Intégrité, performance
  - Contrôle, synchronisation

BTD/UML-DL 15

CENTRALE  
L Y O N

Bertrand DAVID :  
UML et le Développement de Logiciels

## UML : La vue d'implémentation ou de composants

- Décomposition statique des modules et sous systèmes
  - L'allocation des éléments de modélisation dans des modules (fichiers sources, exécutions, etc.)
  - L'organisation des composants : la distribution du code, les dépendances entre les composants
  - Les contraintes de développement (bibliothèques expertes ...)
  - Organisation complexe
    - Les modules en « sous-systèmes »
    - Dépendance entre les interfaces de sous-modules et d'autres sous-systèmes ou modules

BTD/UML-DL 16

CENTRALE  
L Y O N

Bertrand DAVID :  
UML et le Développement de Logiciels

## UML : La vue de déploiement

- **Description des ressources matérielles : logicielles en environnement distribué**
  - La disposition et nature physique des matériels, ainsi que leurs performances
  - L'implémentation des modules principaux sur les nœuds du réseau
  - Les exigences en terme de performances (temps de réponse, tolérance aux fautes et pannes,...)
  - Les paramètres clés
    - Disponibilité, faibilité, performance
    - Contrôle, intégrité, synchronisation
    - Installation, maintenance

BTD/UML-DL

17

CENTRALE  
L Y O N

Bertrand DAVID :  
UML et le Développement de Logiciels

## UML : La vue des scénarios ou cas d'utilisation

- Sert de cadre ou guide aux autres vues
- Définit les besoins des clients du système
- Contient les cas d'utilisation et les acteurs
- Contribue à définir un modèle d'architecture pertinent et cohérent
- Unifie les quatre autres vues
- Permet d'identifier les interfaces critiques
- Permet un regroupement logique des cas d'utilisation

BTD/UML-DL

18

CENTRALE  
L Y O N

## Composants du formalisme UML

- Diagramme de classes
- Diagramme d'objets
- Diagramme de cas d'utilisation
- Diagramme d'états
- Diagramme de séquences
- Diagramme d'activités
- Diagramme de collaboration
- Diagramme de composants
- Diagramme de déploiement

Voir présentation suivante

BTD/UML-DL

19