Introduction à la méthode

Version PPT, pour UML

Bertrand LIAUDET

USAGE du document : des notions et des explications

En bleu, certains mots-clés et certaines phrases clés :

La méthode c'est la suite des opérations mises en œuvre pour arriver à un but.

SOMMAIRE

SOMMAIRE 1

INTRODUCTION A LA METHODE 3

1. Généralités sur la méthode 3

Notions 3

La méthode (slides 1 et 2) 4

Définition pratique 4

Les objectifs centraux de toute méthode 4

Le trio impossible : Vite - Bien - pas Cher 5

Conclusion 5

Difficulté de la bonne compréhension du cahier des charges (slides 3 et 4) 6

2. Développement d'une application : les 3 distinctions capitales 8

Notions 8

Application, Système, Système d'information (slide 5) 9

Application 9

Système 9

Système d'information 9

Développement d'une application = conception + réalisation (slide 6) 10

1è distinction : développement = conception + réalisation 10

Conception d'une application = fonctionnelle + technique (slide 7) 11

2è distinction : conception = fonctionnelle + technique 11

MOA et MOE / Build the right system et build the system right (slide8) 12

2 points de vue sur une application – données + traitements (slide 9) 13

3è distinction : application = les données et les traitements 13

L'analyse des données – « Données métier » (slide 10) 14

Cahier des charges (slide 11) 15

ArchitectureS (slide 12) 16

3. Le cycle en V 17

Notions 17

Cycle en V - Tests unitaire et techniques / recette : (slide 13) 18

Relations entre analyse et tests : logique du cycle en V (slide 14) 19

Documentation logicielle (slide 15) 20

Edition mai 2025

INTRODUCTION A LA METHODE

Il est facile de décrire la méthode encore que
son application exige à coup sûr savoir et pratique.
La méthode est dénuée de sens tant qu'elle est déconnectée de la réalisation concrète.

USAGE du document :
En bleu, certains mots-clés et certaines phrases clé :
La méthode c'est la suite des opérations mises en œuvre pour arriver à un but.

1. Généralités sur la méthode

Notions

- 1) La méthode
- 2) Les objectifs centraux de toute méthode
- 3) Le trio impossible : Vite - Bien - pas Cher
- 4) Difficulté de la bonne compréhension du cahier des charges

La méthode (slides 1 et 2)

Définition pratique

La méthode c'est la **suite des opérations mises en œuvre pour arriver à un but**

Les objectifs centraux de toute méthode

⇒ **Répondre à la demande (c'est-à-dire aux besoins = le cahier des charges)**

⇒ Bien comprendre les besoins

⇒ Aider à les exprimer

*Obtenir le résultat souhaité, c'est chercher la meilleure **qualité fonctionnelle** (= le « bien » externe)*

⇒ Répondre au **meilleur coût**

⇒ Répondre avec les **meilleurs délais**

⇒ Répondre et **faciliter les évolutions**

*Faciliter les évolution, c'est chercher la meilleures **qualité technique** qui permette ça (= le « bien » interne)*

Le trio impossible : Vite - Bien - pas Cher

- ⇒ Vite et Bien => Cher
- ⇒ Vite et pas Cher => pas Bien
- ⇒ Bien et pas Cher => pas Vite

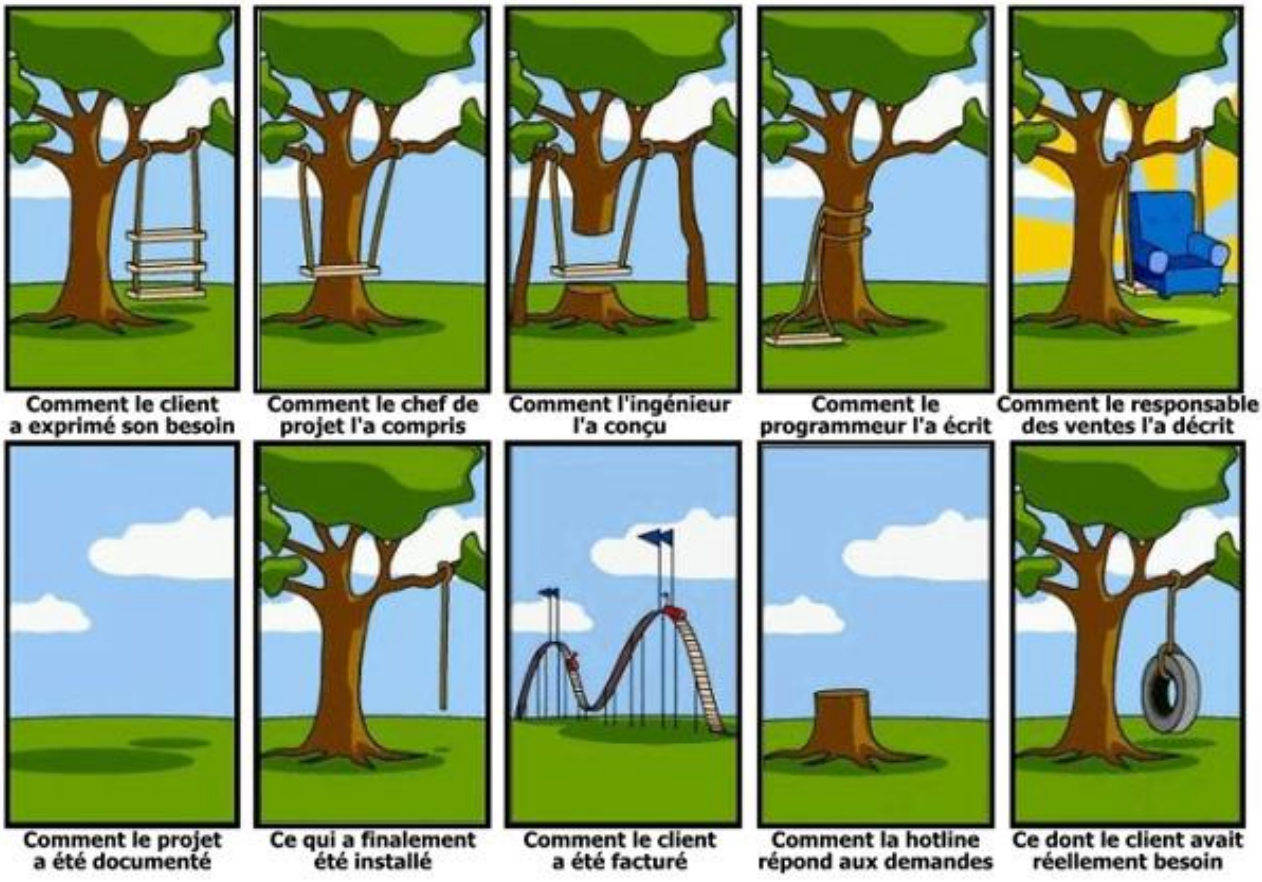
Conclusion

- ⇒ La méthode s'adapte le plus souvent aux contraintes de temps et d'argent.

Difficulté de la bonne compréhension du cahier des charges (slides 3 et 4)

- Le cahier des charges est un document qui décrit ce qu'il y a à faire.
  - ⇒ Bien comprendre le cahier des charges est central pour bien répondre aux besoins : c'est loin d'être simple.
  - ⇒ Les dessins ci-dessous, classiques, montrent cette difficulté.
- Ces dessins font apparaître différents acteurs du projet :
  - ⇒ client,
  - ⇒ chef de projet,
  - ⇒ ingénieur,
  - ⇒ programmeur,
  - ⇒ commercial,
  - ⇒ hotline,
  - ⇒ utilisateur.
- Il montre **3 problèmes classiques** :
  - ⇒ A qui s'adresse ce que l'on fait ?
  - ⇒ Comment est-ce compris ?
  - ⇒ Comment est-ce traduit ?

- Les **4 premiers dessins** et le **dernier** sont les plus classiques.



2. Développement d'une application : les 3 distinctions capitales

Notions

- 5) Application, Système, Système d'information
- 6) Développement d'une application = conception + réalisation
- 7) Conception d'une application = fonctionnelle + technique
- 8) Fonctionnel – MOA – externe – Quoi - Build the right system
- 9) Technique - MOE – interne – Comment - Build the system right
- 10) 2 points de vue sur une application – données + traitements
- 11) L'analyse des données et les données métier
- 12) Notion de cahier des charges
- 13) Les architectures

Application, Système, Système d'information (slide 5)

Application

- Une application c'est un logiciel ou un programme informatique. On parle aussi de « système ».

Système

- Un système est un ensemble d'éléments interconnectés qui fonctionnent ensemble pour atteindre un objectif commun.
- Concrètement, un système, pour nous, c'est une application (à noter qu'un serveur, c'est un programme sans interface utilisateur).

Système d'information

- Système utilisé pour **collecter**, **stocker**, **mettre à jour**, **traiter** et **diffuser** des informations afin de :
  - ⇒ 1) rendre des services aux **utilisateurs**. C'est la partie **transactionnelle**.
  - ⇒ 2) soutenir la prise de décision par les **décideurs**. C'est la partie **décisionnelle**.

Développement d'une application = conception + réalisation (slide 6)

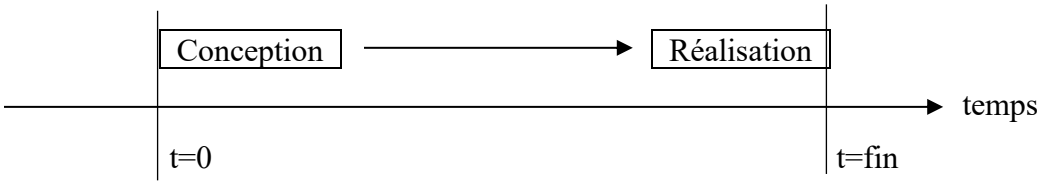
1è distinction : développement = conception + réalisation

- **Développer** une application, c'est **coder le programme informatique** de l'application pour qu'elle fonctionne.
- 1è distinction : les 2 parties du développement d'une application sont : **la Conception** et **la Réalisation.**
- Pour tout projet : 2 étapes

  La conception

  La réalisation

- La conception, c'est la première étape de la réalisation : ça consiste à réfléchir à ce qu'on va faire et à s'organiser avant de commencer la réalisation. Pour ça, on va comprendre ce qu'il y a à faire et décrire comment on va le faire.
- La réalisation d'une application informatique consiste à coder. C'est réaliser concrètement ce qu'il y a à faire.
- La distinction entre la conception et la réalisation permet d'**organiser la division du travail**.
- **Le projet se déroule dans le temps :** il commence avec la conception, il se termine avec la réalisation.



Conception d'une application = fonctionnelle + technique (slide 7)

2è distinction : conception = fonctionnelle + technique

⇒ Conception d'une application = **l'analyse fonctionnelle** et **l'analyse technique**.

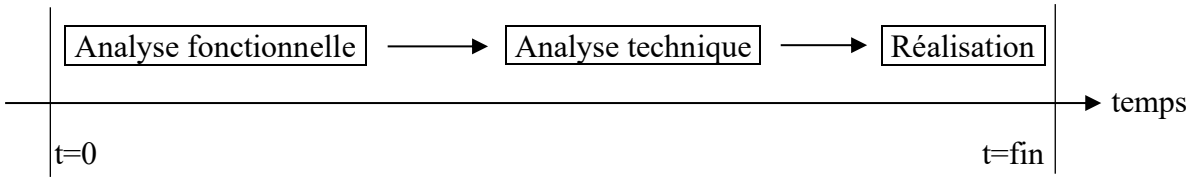
⇒ On peut aussi dire : **le pourquoi** et **le comment**.

Pour toute conception : 2 parties

Le fonctionnel = l' externe = le quoi, le pourquoi

Le technique = l' interne = l'organique = le comment

- **L'analyse fonctionnelle** (ou « **le fonctionnel** ») c'est l'analyse des « fonctionnalités » de l'application (on dit aussi les services ou les usages). C'est ce qui est visible donc externe. Ça répond globalement à la question : que fait le logiciel ?
- Plus précisément : l'analyse fonctionnelle répond aux questions : que fait le système : **QUOI**, **POUR QUI** comme utilisateur, **AVEC QUI** comme récepteur, **AVEC QUOI** comme autres systèmes avec qui notre application interagit. Les qui et les quoi sont ici extérieurs à l'application.
- **L'analyse technique** (ou « **le technique** »), c'est l'analyse de ce qu'on va mettre en œuvre pour la réalisation en répondant aux attentes de l'analyse fonctionnelle. C'est l'organisation du code. C'est ce qui est interne, comme les organes (on parle aussi d'analyse organique). Ça répond globalement à la question : comment est fait le logiciel ?
- **Le projet se déroule dans le temps** : il commence avec l'analyse fonctionnelle, puis l'analyse technique et enfin il se termine avec la réalisation.



MOA et MOE / Build the right system et build the system right (slide8)

Analyse fonctionnelle (= Le fonctionnel)	**Analyse technique (= Le technique)**
EXTERNE	**INTERNE**
Le **POURQUOI** POUR QUI – AVEC QUOI – AVEC QUI	Le **COMMENT**
Point de vue de l'utilisateur et/ou du maître d'ouvrage (**MOA**) : et/ou du client	**Point de vue de l'informaticien** et/ou du maître d'œuvre (**MOE**) : et/ou de celui qui réalise le logiciel
Build the right system	*Build the system right*

- **Pour l'utilisateur, ce qui compte** : c'est l'usage du système (à quoi sert le système).
- **Pour l'informaticien, ce qui compte :** c'est l'architecture interne du système (comment est fait le système)
- **L'analyse fonctionnelle** garantit qu'on va **faire ce qui est demandé** : répondre aux exigences du client.
- L'analyse technique garantit que ce qu'on va faire, on va techniquement bien le faire.

2 points de vue sur une application – données + traitements (slide 9)

3è distinction : application = les données et les traitements

- On peut voir une application informatique comme
  - **un gestionnaire des d'informations** (information = donnée)
  - **un fournisseur des services** (service = fonctionnalité = traitement)

- La dernière distinction capitale dans une application, c'est donc celle entre :
  - les **données** **: l'analyse des données UML sera le MCD-Diagramme des classes métier.**
  - les **traitements** **: l'analyse des traitements UML sera le Diagrammes de Use-Cases.**

L'analyse des données – « Données métier » (slide 10)

- **Les données peuvent être analysées pour elle-même** :
    - indépendamment des traitements qu'on leur appliquera, en tant que réseau de données interconnectées.
- Les données interviennent :
    - dans la base de données (la BD)
    - dans les classes de la programmation objet.
- Les données métier :
    - D'un point de vue fonctionnel : ce sont les **données utilisées par les utilisateurs**.
    - D'un point de vue technique : ce sont les données qui interviendront à la fois **dans la BD et dans les classes.**

⇨ On va d'abord analyser ces « données métier »

Cahier des charges (slide 11)

- Le cahier des charges décrit **ce qu'il y a à faire**.
  - ⇒ Ce qu'il y a un faire ce sont les « **exigences** » ou « **requirement** » : c'est **le quoi**.
  - ⇒ Ce qu'il y a à faire, ce sont aussi les « **spécifications** » : plutôt le **comment**.
- Cahier des charges **en anglais** se dit couramment : « **specifications** ».
- Le cahier des charges le plus courant est celui qui décrit le projet au début, plutôt le quoi que le comment.
- Toutefois, le cahier des charges s'applique à tous les niveaux =>
  - ⇒ Il y a des exigences fonctionnelles : il peut y avoir un cahier des charges fonctionnelles ou des spécifications fonctionnelles
  - ⇒ Il y a des exigences techniques : il peut y avoir un cahier des charges techniques ou des spécifications techniques

=> conclusion : il y a du quoi et du comment à tous les niveaux !

ArchitectureS (slide 12)

- L'architecture c'est l'organisation des composants du projets.
- Ça intervient à tous les niveaux => Il y a DES architectures.
  - ⇒ Architecture logiciel (organisation générale du code),
  - ⇒ Architecture des classes,
  - ⇒ Architecture MVC,
  - ⇒ Architecture Client-Serveur,
  - ⇒ Achitecture WEB,
  - ⇒ Architecture du cycle en V,
  - ⇒ Archtecture des cas d'utilisation,
  - ⇒ Architecture des tâches du projet (planification, Gantt, Pert, …)
  - ⇒ Etc.

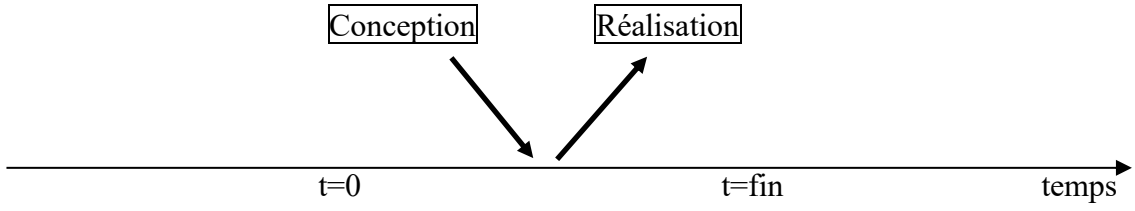
3. Le cycle en V

Notions

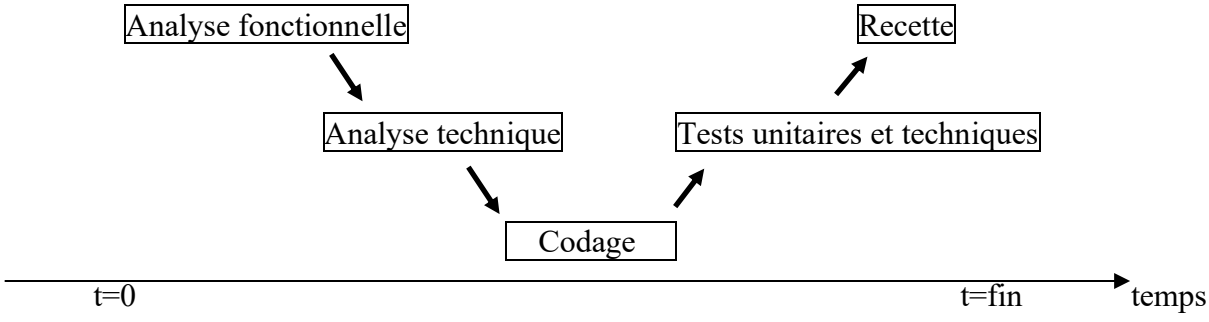
- 14) Cycle en V, tests unitaires, tests d'intégration, recette
- 15) Relation entre analyse et tests
- 16) Documentation logicielle : document d'analyse fonctionnelle, document de recette.
- 17) Caractéristiques d'un document technique : plan et document partagé

Cycle en V - Tests unitaire et techniques / recette : (slide 13)

- Le cycle en V c'est une méthode classique de développement du logiciel.
- La conception et la réalisation forment les deux branches du cycle en V :



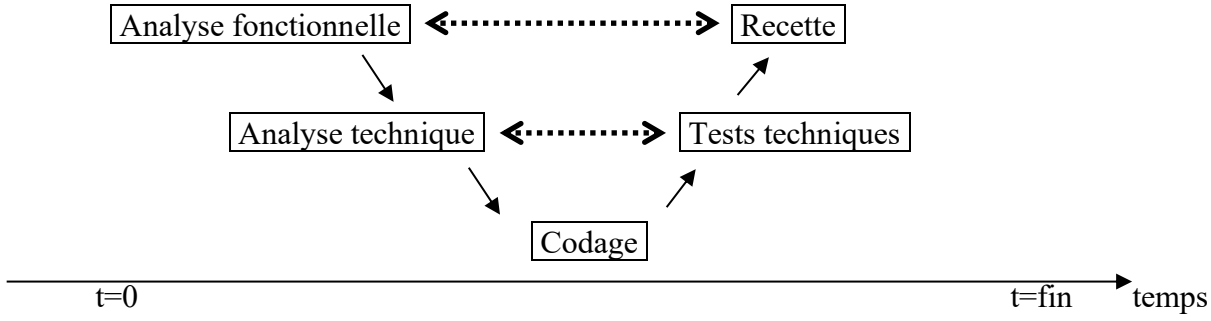
- **Conception** = Analyse fonctionnelle + Analyse technique
- **Réalisation** = codage + tests techniques + tests fonctionnelles (= recette)



- **Tests technique** = les tests unitaires (en général, des classes de POO) + tests d'intégration des différents sous-systèmes.

Relations entre analyse et tests : logique du cycle en V (slide 14)

- A chaque étape d'analyse, on peut concevoir l'étape de test.
- C'est le lien entre les étapes de chaque branche qui justifie la forme en V :



- Cette méthode permet de simuler le système au fur et à mesure de l'analyse.
- Cette méthode permet de simuler le les composants techniques du système au fur et à mesure de l'analyse.
- Cette méthode permet aussi, en cas de problème de test (tests techniques ou recette), de revenir facilement à la partie de la conception à laquelle le problème correspond.

Documentation logicielle (slide 15)

- L'étape d'analyse fonctionnelle consiste à écrire :
  - ⇒ un **document d'analyse fonctionnelle (avec les schémas UML)**
  - ⇒ un **document de recette**.
- C'est particulièrement sur cette étape qu'on va travailler dans ce cours :
  - ⇒ L'analyse fonctionnelle sera divisée en plusieurs étapes (les données, les usages, le MVC, la simulation pour la recette).
- On peut ajouter une étape d'analyse MVC.

- Pour produire les documents, on utilise des logiciels variables selon les attentes :
  - ⇒ Traitement de texte classique : **Word,** etc.
  - ⇒ Traitement de texte partagé : **Google docs,** etc.
  - ⇒ Traitement de texte scientifique : LaTex
  - ⇒ Texte avec calcul Python : NoteBook

  - ⇒ Les usages dépendent des entreprises et des applications réalisées.

Caractéristiques d'un document technique (slide 16)

- Un document technique est un document qu'on doit pouvoir lire « en diagonale » :
  - ⇒ En utilisant la **table des matières** pour accéder aux chapitres qui nous intéressent et qu'on lira plus ou moins complètement.
  - ⇒ En utilisant le **volet de navigation** qui relie table des matière et contenu si on a une version pdf. Ca s'utilise directement dans le navigateur.
  - ⇒ En utilisant la **recherche directe de mots** si on a une version électronique du document pour accéder aux thèmes qui nous intéressent et qu'on lira plus ou moins complètement.

- Table des matières :
  - ⇒ Il faut maitriser la gestion de la table des matières, quel que soit l'outil utilisé.
  - ⇒ La table des matières c'est le plan d'organisation du texte : elle est fondamentale !
  - ⇒ Aujourd'hui, les « pdf » sur accessible sur internet avec Chrome (par exemple) permettent de circuler dans la table des matières : c'est à utiliser !

- Document partagé :
  - ⇒ Selon les environnements, on peut aussi utiliser des outils de partage de documents et de gestion de version (google docs)