

## PLAN

Chapitre 1 : Introduction aux processus de développement  
Chapitre 2 : Le Processus de développement Unifié (UP)  
Chapitre 3 : Les méthodes Agiles  
Chapitre 4 : Le framework Scrum  
Chapitre 5 : Etude de cas avec Scrum  
Chapitre 6 : DevOps

### Chapitre 1

## Introduction au Processus de Développement

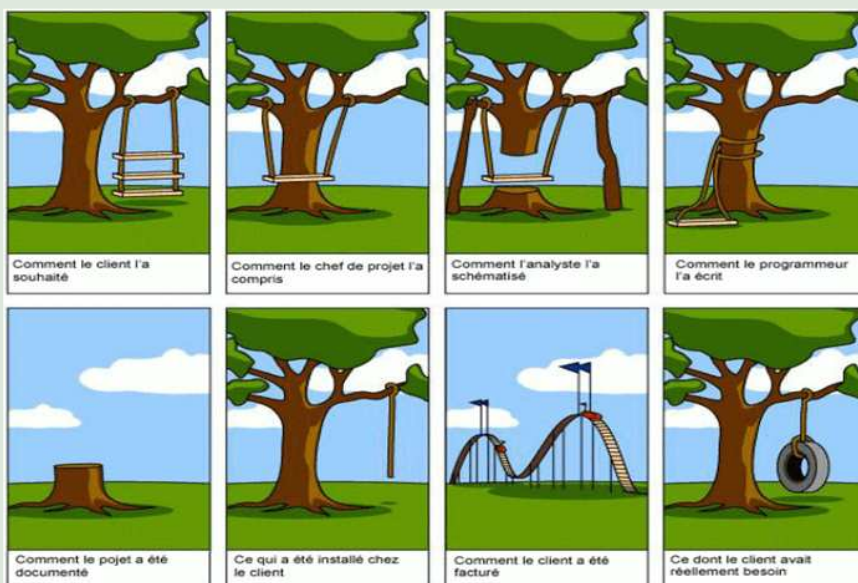
# 1

## Introduction au Processus de Développement

### PLAN

- 1- Processus de développement
- 2- Approches méthodologiques
  - 2.1- Modèle en cascades
  - 2.2- Modèle en V
  - 2.3- Modèle incrémental
  - 2.4- Modèle en spirale
- 3- Evaluation des approches

## Introduction



## Introduction

« La qualité du processus de fabrication est garante de la qualité du produit »

➔ Pour obtenir un logiciel de qualité, il faut en maîtriser le processus d'élaboration

## 1- Processus de développement

- Un processus est une série de tâches qui aboutissent à un objectif.
- Un processus définit qui fait quoi, quand et comment.
- Un processus de développement logiciel est un ensemble (structuré) d'activités qui visent à produire un logiciel de qualité.

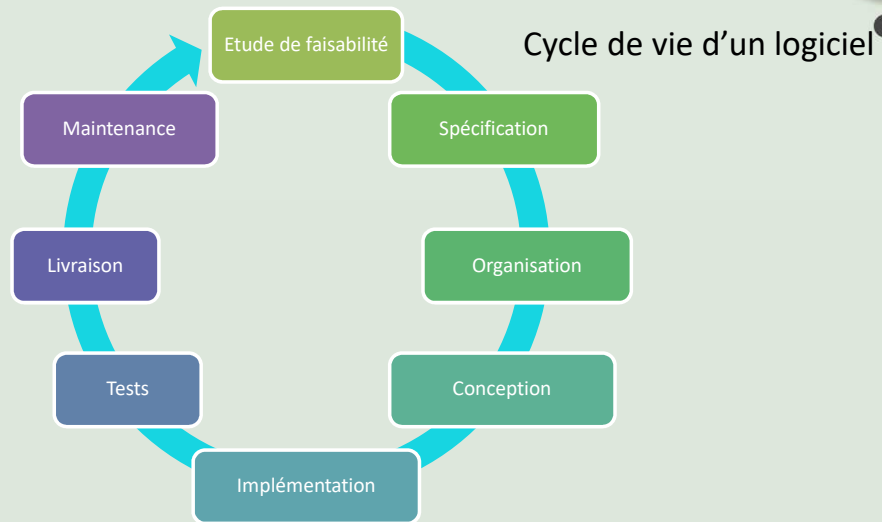
# 1- Processus de développement

- Ses objectifs sont de :
  - Fournir une approche pour assigner les tâches et les responsabilités au sein d'une organisation.
  - Garantir la livraison d'un logiciel de haute qualité avec un temps et budget limité.
  - Décrire les meilleures méthodes de travail
  - Réduire les risques et améliorer les prévisions

# 1- Processus de développement

- N'importe quel processus doit comporter les étapes principales du cycle de vie d'un logiciel.
- Le but du découpage en étapes est de :
  - Maîtriser les risques.
  - Maîtriser au mieux les délais et les coûts.
  - Obtenir une qualité conforme aux exigences.

# 1- Processus de développement



# 2- Approches méthodologiques

Une méthodologie  
=  
une démarche + notation + outils

- Intérêt :
  - Faciliter le développement de solutions
  - Standardiser le travail

## 2- Approches méthodologiques

13

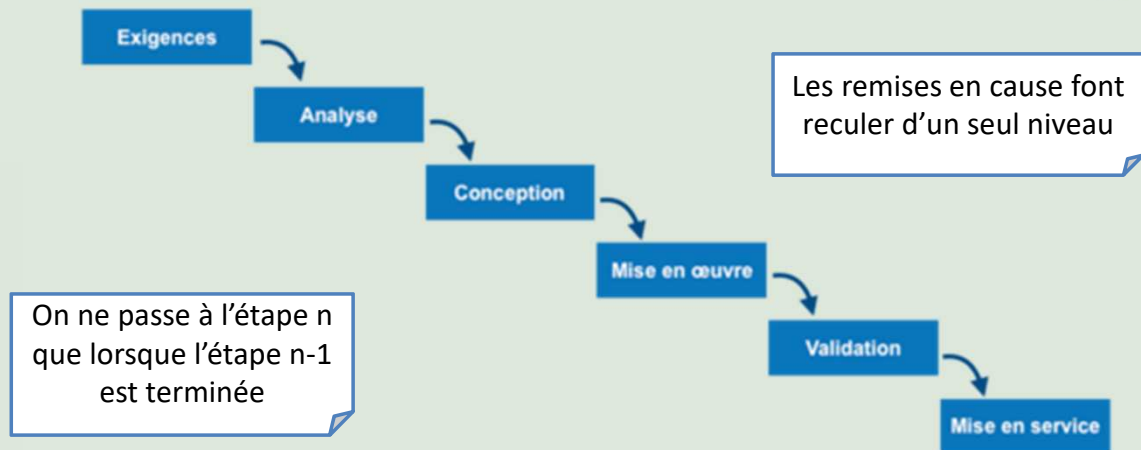
- Les principales méthodologies existantes en Génie Logiciel sont :
  - L'approche cartésienne (70')
  - L'approche systémique (80')
  - L'approche Orientée Objet (90')
  - L'approche Agile (2000)

## 2- Approches méthodologiques

14

- Plusieurs modèles de méthodes ont vu le jour afin d'organiser le cycle de vie d'un logiciel de manière qui facilite son développement.
- Les plus connus sont :
  - Modèle en cascades (waterfall)
  - Modèle en V
  - Modèle incrémental
  - Modèle en spirale

## 2.1- Modèle en cascade



## 2.1- Modèle en cascade

### Principe

- Développé dans les années 1970 par W.Royce.
- Inspiré par les méthodes classiques de l'ingénierie.
- Il se base sur 2 idées fondamentales :
  - Une étape ne peut pas être débutée avant que la précédente ne soit achevée. ⇨ Des documents sont produits pour concrétiser la réalisation de chaque phase.
  - La modification d'une étape du projet a un impact important sur les étapes suivantes.



## 2.1- Modèle en cascade

### Avantages

- Simple à mettre en œuvre.
- Bien documenté à chaque phase.
- Le planning est établi à l'avance → le maître d'ouvrage (client) sait précisément ce qui va lui être livré et quand.

## 2.1- Modèle en cascade

### Inconvénients

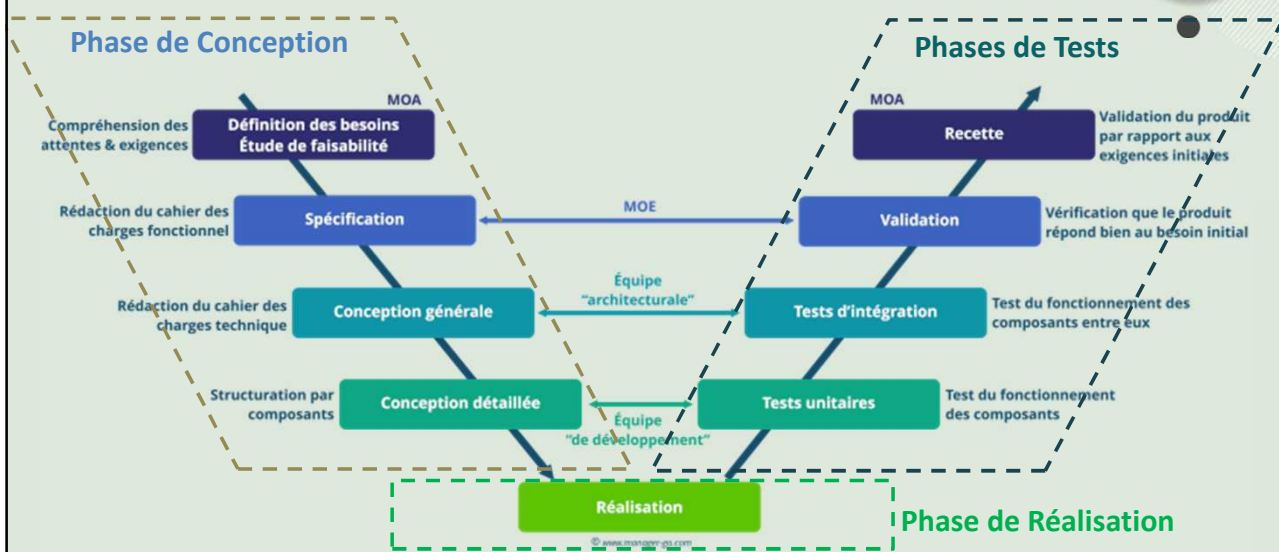
- Le client est exclu dès la phase de spécification.
- La rédaction des documents de validation de chaque phase demande beaucoup de travail.
- Rigide : il est difficile de s'adapter aux besoins nouveaux ou modifiés des clients
- Les tests sont prévus tardivement → un changement tardif coûte cher.

## 2.1- Modèle en cascade

### Dans quels cas l'utiliser ?

- Il est adapté dans le cas où les besoins sont clairement identifiés et stables → Très rare !!
- Il est principalement utilisé dans les grands projets où les systèmes sont développés sur plusieurs sites ;
  - Dans ce cas, le modèle en cascade facilite la planification du projet.

## 2.2- Modèle en V



## 2.2- Modèle en V

### Principe

- Dérivé du cycle de vie en cascade, il a été mis au point en 1980, puis référencé en 1990 par Forsberg et Mooz.
- La représentation en V est plus proche de la réalité, le processus de développement n'est pas parfaitement séquentiel.
- La caractéristique principale de ce processus est que chaque étape de conception fonctionne en binôme avec une phase de test ( validation ).

## 2.2- Modèle en V

### Avantages

- Il permet d'anticiper sur les phases ultérieures de développement du produit.
- Les phases de conception permettent de limiter les risques et dérives pendant les phases de tests.  
⇒ Diminuer les risques d'erreur.
- Le respect du cahier de charge est mieux garanti.

## 2.2- Modèle en V

### Inconvénients

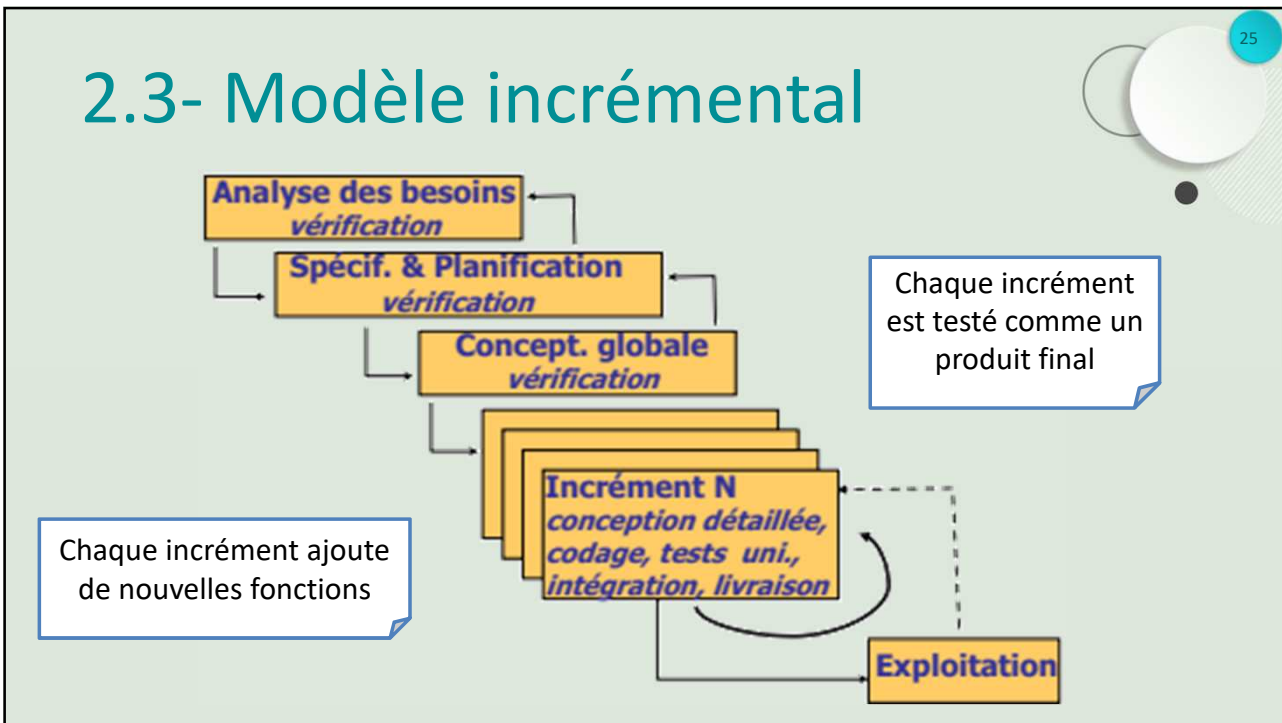
- Il est rigide : en cas de mauvaise maîtrise du périmètre, il faut refaire la phase de conception.
- Documentation importante : chaque phase produit des documents parfois en doublons.
- Organisation lourde : le modèle fait intervenir plusieurs responsables.
- Son application réelle est très difficile.

## 2.2- Modèle en V

### Dans quels cas l'utiliser ?

- Il est souvent utilisé lors des projets où :
  - Le client est exigeant et demande souvent des validations (tests)
  - Il y a parfois quelques changements dans les besoins au cours du projet.
  - Les tests sont une priorité

## 2.3- Modèle incrémental



## 2.3- Modèle incrémental

### Principe

- Diviser le projet en incréments, tel qu'un **incrément** est une sous partie fonctionnelle cohérente du produit final.
- Un logiciel noyau est d'abord développé puis des incréments sont successivement développés et intégrés.



## 2.3- Modèle incrémental

### Avantages

- Il est possible d'avoir des feedbacks réguliers du client :
  - Garantir la satisfaction du client.
  - S'adapter facilement aux changements des exigences
- Une première version du système est fournie rapidement → Livraison plus rapide par modules.

## 2.3- Modèle incrémental

### Inconvénients

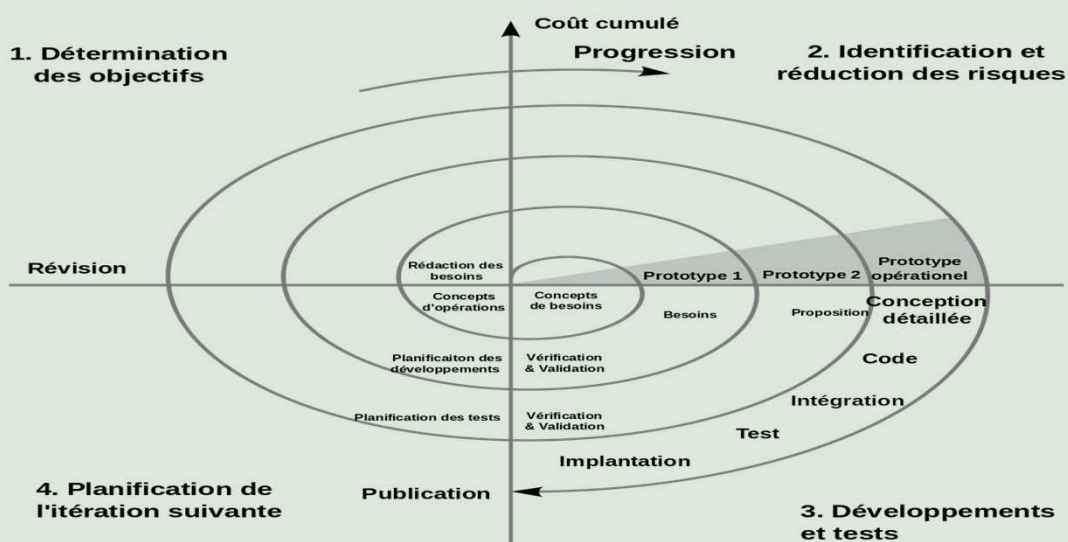
- Difficile à gérer :
  - Le processus n'est pas visible dès le début.
  - Il faut attendre longtemps avant de voir le résultat final.
- Intégration est parfois trop complexe
  - Elle peut conduire à un résultat dysfonctionnel.
  - Le coût en termes de tests et de validation du produit final peuvent devenir très importants.

## 2.3- Modèle incrémental

### Dans quels cas l'utiliser ?

- Peut être utilisé quand il n'y a pas assez de ressources disponibles pour une livraison à temps.
- Adapté aux projets de taille moyenne (inférieur à 100'000 lignes de code)

## 2.4- Modèle en spirale



## 2.4- Modèle en spirale

### Principe

- Modèle itératif, proposé par Boehm en 1968, et qui met l'accent sur l'analyse des risques.
- Le cycle de vie est représenté à l'aide d'une spirale telle que chaque boucle de la spirale se déroule en 4 étapes :
  - Définition des objectifs, alternatives, contraintes (basé sur la boucle précédente)
  - Evaluation des risques et plan de gestion
  - Développement et vérification de la solution retenue
  - Tests et validation puis planification de la phase suivante

## 2.4- Modèle en spirale

### Avantages

- Cette approche continue et régulière minimise les risques d'échec.
- S'adapte à n'importe quel nombre de changements, qui peuvent se produire.
- L'estimation des coûts devient facile à chaque boucle de la spirale.



## 2.4- Modèle en spirale

### Inconvénients

- Modèle long et coûteux à mettre en place.
- Il nécessite des compétences dans l'évaluation des incertitudes et des risques associés au projet et leur réduction.

## 2.4- Modèle en spirale

### Dans quels cas l'utiliser ?

- Convient aux projets de grandes envergures.
- Adapté à des projets innovants où les risques sont élevés et les enjeux importants.

## 3- Evaluation des approches

- Il n'y a pas qu'un seul processus et il n'existe pas de processus standard pour tout projet.
- Plusieurs approches existent mais chacune présente des avantages et des inconvénients.
- Certaines ne sont plus utilisées mais sont quand même enseignées à titre informatif.
- Dans la plupart des projets, deux ou plus d'un modèle sont utilisés.

1

**FIN**

Introduction aux processus de développement