

Introduction au Processus de Développement

La **qualité du processus de création** d'un logiciel influence directement la **qualité du produit final**.

👉 Autrement dit, pour obtenir un bon logiciel, il faut bien maîtriser la manière dont on le conçoit.

Qu'est-ce qu'un processus de développement ?

Un **processus de développement logiciel** regroupe toutes les activités nécessaires pour créer un logiciel de qualité.

Ses objectifs :

- Répartir clairement les **tâches et responsabilités** dans l'équipe.
- Livrer un **logiciel fiable** dans un **délai et un budget limités**.
- Définir les **bonnes pratiques de travail**.
- **Réduire les risques** et mieux prévoir le temps et les coûts.

Cycle de vie d'un logiciel :

Étude de faisabilité -> Spécification -> Organisation -> Conception -> Implémentation -> Tests -> Livraison -> Maintenance

Approches méthodologiques :

Une méthodologie = une démarche + notation + outils

Son but est de **faciliter le développement** et de **standardiser** la manière de travailler.

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)

Les modèles de développement :

♦ **Modèle en cascade (Développédanslesannées1970parW.Royce)**

Principe : on avance étape par étape.

On commence la suivante seulement quand la précédente est finie.

Avantages :

- Simple à appliquer.
- Bonne documentation.
- Le client connaît à l'avance le contenu et la date de livraison.

Inconvénients :

- Très rigide.
- Peu de contact avec le client après le début.
- Les erreurs sont découvertes tard, donc plus coûteuses.

Utilisation :

Quand les besoins sont **clairs et stables**.

♦ **Modèle en V**

Principe : chaque phase de conception est liée à une phase de test.

On prépare les validations dès le début.

Avantages :

- Réduit les risques d'erreurs.

- Meilleure conformité au cahier des charges.

Inconvénients :

- Trop formel et lourd à appliquer.
- Beaucoup de documents à produire.

Utilisation :

Quand le **client veut des validations fréquentes** et que les tests sont essentiels.

◆ **Modèle incrémental**

Principe : le logiciel est développé **par parties successives**.

Chaque partie (incrément) ajoute de nouvelles fonctions.

Avantages :

- Livraison plus rapide.
- Meilleure adaptation aux changements.
- Feedback régulier du client.

Inconvénients :

- Difficile à gérer.
- Intégration finale parfois complexe.

Utilisation :

Pour des projets **de taille moyenne**.

◆ **Modèle en spirale**

Principe : développé par **Boehm**, ce modèle met l'accent sur les **risques**.

Chaque boucle de la spirale correspond à une phase du projet :

1. Définir les objectifs.
2. Identifier les risques.
3. Développer et tester.
4. Préparer la phase suivante.

Avantages :

- Flexible et adaptable.
- Bonne gestion des risques.

Inconvénients :

- Coûteux et long à mettre en place.
- Nécessite une bonne expertise technique.

Utilisation :

Pour les **grands projets complexes** avec beaucoup d'incertitudes.