



## Mise en oeuvre de Waterfall

Maintenant que le **PDG a validé** l'utilisation de la méthode **Waterfall** pour le développement du logiciel **ERP**, voici comment je vais organiser chaque phase du projet.

### 1. Analyse des Exigences

**Objectif** : Collecter et analyser toutes les exigences fonctionnelles et non fonctionnelles du projet, en collaboration avec les parties prenantes.

a. **Livrable** : Cahier des charges

- i. Ce document est crucial car il sert de référence tout au long du projet. Il décrit clairement ce que le logiciel doit accomplir et quelles sont les contraintes à respecter. Toute ambiguïté dans cette phase pourrait entraîner des retards ou des erreurs coûteuses à corriger par la suite.
- ii. Importance : Garantir que tous les besoins sont compris et validés avant de passer à la phase de conception.

### 2. Conception du Système

**Objectif** : Créer un plan technique détaillé du logiciel basé sur les exigences définies.

a. **Livrable** : Documentation de conception.

- i. Ce document inclut les diagrammes d'architecture, les modèles de données, ainsi que les spécifications techniques des composants du système. Il est essentiel pour s'assurer que l'architecture du logiciel est adaptée aux exigences et qu'elle pourra être mise en œuvre efficacement.
- ii. Importance : S'assurer que toutes les décisions techniques sont prises en amont, minimisant ainsi les risques de retards ou de problèmes lors de l'implémentation.

### 3. Implémentation et Planification des Tests

**Objectif** : Développer le logiciel en suivant la documentation de conception tout en planifiant et exécutant des tests continus.



- a. **Livrables** : Code source et rapport de tests.
  - i. Le code source est le livrable principal de cette phase. Parallèlement, la planification des tests doit être effectuée pour garantir que chaque module est testé indépendamment (tests unitaires) et que le système dans son ensemble fonctionne correctement (tests d'intégration).
  - ii. Importance : La qualité du développement est garantie par une exécution rigoureuse des tests, ce qui permet d'identifier les erreurs à chaque étape et d'assurer que le produit répond aux attentes.

#### 4. Déploiement et Maintenance

**Objectif** : Assurer la mise en production du logiciel et préparer un plan de maintenance pour la gestion des corrections et des futures mises à jour.

- a. **Livrables** : Plan de déploiement et plan de maintenance.
  - i. Le plan de déploiement couvre les stratégies de livraison du logiciel, les configurations nécessaires et les formations éventuelles pour les utilisateurs. Le plan de maintenance inclut les procédures pour corriger les bugs et introduire des mises à jour.
  - ii. Importance : Le déploiement doit se dérouler sans accroc pour garantir la satisfaction des utilisateurs finaux. La maintenance est également cruciale pour assurer la pérennité du système et répondre aux besoins évolutifs.

### Conclusion

Chaque phase du modèle Waterfall produit des livrables essentiels qui permettent d'assurer la qualité, la traçabilité et la bonne progression du projet ERP.

Leur importance réside dans leur capacité à structurer le projet, à réduire les risques et à garantir une livraison conforme aux attentes du PDG et des parties prenantes.



## Premiers cycles Waterfall pour un projet ERP

### 1. Analyse des Exigences

**Objectif** : Comprendre les besoins fonctionnels et non fonctionnels du système ERP à développer.

a. **Exemple d'activités** :

- i. Réunions avec les parties prenantes (équipe de gestion, utilisateurs finaux, responsables informatiques) pour identifier les besoins clés.
- ii. Analyse des processus métiers actuels (comptabilité, gestion des stocks, ressources humaines, etc.) pour déterminer comment l'ERP peut les améliorer ou les automatiser.
- iii. Identification des contraintes techniques, telles que l'intégration avec les systèmes existants (CRM, solutions de paie, etc.).

b. **Livrable** : Cahier des charges (document de spécifications des exigences).

- i. Contient les exigences fonctionnelles (modules nécessaires, comme la gestion des ressources humaines, la comptabilité, etc.) et non fonctionnelles (performances, sécurité, compatibilité).
- ii. Importance : Ce document définit les fondations du projet. Tout développement futur doit s'y référer pour éviter des divergences avec les attentes initiales.

### 2. Conception du Système (Design)

**Objectif** : Créer un plan technique basé sur les exigences définies lors de la première phase.

a. **Exemple d'activités** :

- i. Définition de l'architecture technique de l'ERP (choix de l'architecture modulaire, microservices ou monolithique).
- ii. Élaboration de modèles de base de données pour les différents modules (gestion des utilisateurs, gestion des stocks, etc.).



- iii. Conception des interfaces utilisateurs (UI/UX) pour garantir une navigation simple et intuitive à travers les différents modules de l'ERP.
- iv. Choix des technologies (langages de programmation, frameworks, bases de données) en fonction des besoins du projet.
- b. **Livrable** : Documentation de conception.
  - i. Comprend des diagrammes d'architecture (ex. diagramme d'entités-relation pour la base de données, diagramme de composants pour les modules du système).
  - ii. Importance : Assure que le projet est techniquement réalisable et que les développeurs disposent d'une feuille de route claire pour l'implémentation.

### 3. Implémentation et Planification des Tests (Implementation and Verification)

**Objectif** : Développer le système ERP tout en assurant un plan de tests rigoureux.

- a. **Exemple d'activités** :
  - i. Développement de modules séparés, par exemple, un module de gestion des stocks, un module de comptabilité, un module de ressources humaines.
  - ii. Tests unitaires sur chaque module pour vérifier que chaque fonction fonctionne correctement (exemple : test d'ajout de produits au stock dans le module de gestion des stocks).
  - iii. Mise en place de tests d'intégration entre les modules (exemple : tester la synchronisation entre le module de comptabilité et le module de gestion des stocks).
- b. **Livrable** : Code source fonctionnel et rapport de tests
  - i. Importance : La phase d'implémentation doit être soigneusement réalisée pour éviter des retards ou des bugs coûteux à corriger en fin de cycle. Les tests



permettent de valider la robustesse de chaque module et du système dans son ensemble.

## Conclusion

En conclusion, cet exemple illustre comment les premières phases du modèle Waterfall s'appliquent efficacement au développement d'un ERP.

Il met en évidence l'importance cruciale d'une documentation claire, d'une planification rigoureuse et de tests réguliers pour garantir une progression fluide du projet, tout en minimisant les risques d'erreurs ou de retards.