

# Plan de mise en œuvre du projet ERP selon la méthode Waterfall

Ce document décrit notre stratégie pour chaque phase du modèle Waterfall, les livrables associés et l'importance de chacun. L'objectif est d'assurer une exécution rigoureuse du projet tout en garantissant que chaque étape est clairement définie et comprise par toutes les parties prenantes.

## Phase 1 : Exigences système et logiciel

### Plan général de la phase

#### 1. Élicitation des exigences

##### Guide de l'élicitation

##### Les 6 étapes de la pyramide des problèmes

#### 2. Analyse des exigences

##### Organisation des exigences

##### Analyse de la faisabilité

##### Modélisation des exigences

##### Identification des conflits et ambiguïtés

##### Priorisation des exigences

#### 3. Spécification

### Livrables de la phase Exigences système et logiciel

## Phase 2 : Conception du système (Design)

## Phase 3 : Implémentation et Vérification (Implementation and Verification)

## Phase 4 : Déploiement et Maintenance (Deployment and Maintenance)

### Waterfall

### Conclusion

### Prochaines étapes

---

## Phase 1 : Exigences système et logiciel

### Plan général de la phase

Les exigences logicielles d'un système sont la description de ce que le système doit faire, du ou des services qu'il fournit et des contraintes qui pèsent sur son fonctionnement.



L'IEEE Standard Glossary of Software Engineering Terminology définit une exigence comme suit :

- Une condition ou une capacité nécessaire à un utilisateur pour résoudre un problème ou atteindre un objectif.
- Une condition ou une capacité qui doit être remplie ou possédée par un système ou un composant du système pour satisfaire un contrat, une norme, une spécification ou tout autre document formellement imposé.
- Une représentation documentée d'une condition ou d'une capacité comme dans 1 ou 2

Les activités liées au travail sur les exigences logicielles peuvent être divisées en trois grandes catégories : **l'élicitation**, **l'analyse** et la **spécification**.

## 1. Élicitation des exigences

*élicitation qui vient de elicio (« tirer de, faire sortir ; attirer ; tirer, arracher, exciter, provoquer, obtenir »)*

**L'élicitation** est la collecte et la découverte des besoins auprès des parties prenantes et d'autres sources. C'est la **première étape de l'élaboration des besoins**.

Le terme « élicitation » est utilisé pour souligner le fait que de bonnes exigences ne peuvent pas être simplement collectées auprès du client, comme l'indiquerait le terme « collecte des exigences ». L'élicitation des exigences n'est pas triviale car on ne peut jamais être sûr d'obtenir toutes les exigences de l'utilisateur et du client en leur demandant simplement ce que le système doit faire ou ne pas faire.

Les pratiques d'élicitation des exigences comprennent :

- les entretiens,
- les questionnaires,
- l'observation des utilisateurs,

- les ateliers,
- le brainstorming,
- les cas d'utilisation,
- les jeux de rôle,
- et le prototypage.

Le processus d'élicitation des exigences peut sembler simple : demander au client, aux utilisateurs et à d'autres personnes quels sont les objectifs du système ou du produit, ce qui doit être accompli, comment le système ou le produit s'inscrit dans les besoins de l'entreprise et, enfin, comment le système ou le produit doit être utilisé au quotidien. Cependant, des problèmes peuvent survenir et compliquer le processus, comme :

- **Problèmes de portée.** Les limites du système sont mal définies ou les clients/utilisateurs spécifient des détails techniques inutiles qui risquent d'embrouiller les objectifs généraux du système au lieu de les clarifier.
- **Problèmes de compréhension.** Les clients/utilisateurs ne sont pas tout à fait sûrs de ce dont ils ont besoin, comprennent mal les capacités et les limites de leur environnement informatique, ne comprennent pas entièrement le domaine du problème, ont des difficultés à communiquer leurs besoins à l'ingénieur système, omettent des informations considérées comme « évidentes », spécifient des exigences qui entrent en conflit avec les besoins d'autres clients/utilisateurs, ou spécifient des exigences qui sont ambiguës ou non testables.
- **Problèmes de volatilité.** Les exigences changent au fil du temps. Le taux de changement est parfois appelé le niveau de volatilité des exigences (*requirement volatility*).

La **qualité des exigences** peut être améliorée grâce aux approches suivantes :

- **Visualisation.** Utilisation d'outils qui favorisent une meilleure compréhension du produit final souhaité, tels que la visualisation et la simulation.
- **Langage cohérent.** Utiliser des définitions simples et cohérentes pour les exigences décrites en langage naturel et utiliser la terminologie commerciale qui prévaut dans l'entreprise.

- **Lignes directrices.** Suivre les lignes directrices de l'organisation qui décrivent les techniques de collecte et les types d'exigences à collecter. Ces lignes directrices sont ensuite utilisées de manière cohérente dans tous les projets.
- **Utilisation cohérente des modèles.** Produire un ensemble cohérent de modèles et de gabarits pour documenter les exigences.
- **Documentation des dépendances.** Documenter les dépendances et les interrelations entre les exigences.
- **Analyse des modifications.** Effectuer une analyse des causes profondes des modifications apportées aux exigences et prendre des mesures correctives.

## ▼ Guide de l'éllicitation

En 1997, Sommerville et Sawyer ont proposé un ensemble de lignes directrices pour l'éllicitation des exigences, afin de répondre à des préoccupations telles que celles identifiées par Christel et Kang.

- Évaluer la faisabilité commerciale et technique du système proposé
- Identifier les personnes qui aideront à spécifier les exigences et comprendre leurs biais organisationnels
- Définir l'environnement technique (par exemple, l'architecture informatique, le système d'exploitation, les besoins en télécommunications) dans lequel le système ou le produit sera placé.
- Identifier les « contraintes du domaine » (c'est-à-dire les caractéristiques de l'environnement commercial propres au domaine d'application) qui limitent la fonctionnalité ou les performances du système ou du produit à construire.
- Définir une ou plusieurs méthodes d'éllicitation des besoins (par exemple, entretiens, groupes de discussion, réunions d'équipe).
- Solliciter la participation de nombreuses personnes afin que les exigences soient définies à partir de différents points de vue ; veiller à identifier la justification de chaque exigence enregistrée.
- Identifier les exigences ambiguës comme candidates au prototypage.

- Créer des scénarios d'utilisation ou des cas d'utilisation pour aider les clients/utilisateurs à mieux identifier les exigences clés.

## ▼ Les 6 étapes de la pyramide des problèmes

En 2004, Goldsmith a proposé une « pyramide des problèmes » composée de « six étapes qui doivent être exécutées dans l'ordre » :

1. Identifier le problème, l'opportunité ou le défi réel
2. Identifier la (les) mesure(s) actuelle(s) qui montre(nt) que le problème est réel
3. Identifier la (les) mesure(s) de l'objectif pour montrer que le problème a été abordé et l'intérêt de l'atteindre
4. Identifier la ou les causes « actuelles » du problème, car ce sont les causes qui doivent être résolues, et non le problème lui-même.
5. Définir les « besoins » de l'entreprise qui doivent être satisfaits pour atteindre la ou les mesures de l'objectif.
6. Spécifier la conception d'un produit pour satisfaire les exigences réelles de l'entreprise.

Cependant, Goldsmith note que l'identification du problème réel « est extrêmement difficile ».

Sources : [https://en.wikipedia.org/wiki/Requirements\\_elicitation#cite\\_note-5](https://en.wikipedia.org/wiki/Requirements_elicitation#cite_note-5)

## 2. Analyse des exigences

L'analyse des exigences suit l'élicitation et se concentre sur la compréhension approfondie des exigences recueillies, leur organisation, leur classification, leur clarification, et leur priorisation. C'est ici qu'on identifie les incohérences ou les ambiguïtés et qu'on s'assure que les exigences sont viables techniquement et économiquement.

Les exigences doivent être **documentées, exploitables, mesurables, testables, traçables**, liées à des **besoins ou à des opportunités commerciales identifiés**, et définies à un **niveau de détail suffisant** pour la conception du système.

## Organisation des exigences

Classer les exigences par catégories : fonctionnelles (ce que le système doit faire) et non fonctionnelles (performances, sécurité, etc.).

## Analyse de la faisabilité

Vérifier que les exigences peuvent être mises en œuvre avec les technologies disponibles.

Évaluer les ressources nécessaires pour chaque exigence (temps, coûts).

Identifier les éventuelles contraintes réglementaires ou techniques.

## Modélisation des exigences

Créer des diagrammes de cas d'utilisation ou de scénarios pour illustrer comment le système répondra aux exigences.

Utiliser des diagrammes UML (Unified Modeling Language) pour représenter les relations entre les composants.

## Identification des conflits et ambiguïtés

Identifier les conflits entre les exigences (ex. : des besoins qui se contredisent ou qui sont techniquement incompatibles).

Clarifier les exigences ambiguës avec les parties prenantes pour éliminer les doutes.

## Priorisation des exigences

### Identifier et classer les parties prenantes.

Les types de parties prenantes sont les suivants :

- **Les parties prenantes primaires** : celles qui, en fin de compte, sont les plus touchées, positivement ou négativement, par les actions d'une organisation.
- **Les parties prenantes secondaires** : les « intermédiaires », c'est-à-dire les personnes ou les organisations indirectement touchées par les actions d'une organisation
- **Les parties prenantes tertiaires** : celles qui seront le moins touchées.
- **Parties prenantes clés** : celles qui exercent une influence significative sur l'organisation ou qui ont de l'importance au sein de celle-ci ; elles peuvent

également appartenir aux autres groupes.

Cela permet d'identifier :

- Intérêts des parties prenantes
- Risques et malentendus potentiels
- Mécanismes permettant d'influencer positivement les autres parties prenantes
- Personnes clés à informer sur le projet pendant la phase d'exécution
- Les parties prenantes négatives et leurs effets négatifs sur le projet

Source : [https://en.wikipedia.org/wiki/Stakeholder\\_analysis](https://en.wikipedia.org/wiki/Stakeholder_analysis)

**Collaborer avec les parties prenantes pour prioriser les fonctionnalités les plus cruciales.**

**Identifier les exigences qui peuvent être reportées ou réduites en fonction des contraintes de projet.**

### 3. Spécification

La spécification implique la représentation et le stockage des connaissances recueillies sur les exigences d'une manière persistante et bien organisée qui facilite une communication efficace et la gestion du changement. Les use cases, user stories, les exigences fonctionnelles et les modèles d'analyse visuelle sont des choix populaires pour la spécification des exigences.

## Livrables de la phase Exigences système et logiciel

Livrable	Description	Utilité
<b>Business Requirements Document (BRD)</b>	Document qui décrit les besoins et objectifs métier du projet, ainsi que les attentes des parties prenantes du point de vue business.	Assure que les objectifs stratégiques et les besoins de l'entreprise sont clairement définis et compris par toutes les parties prenantes, alignant le projet avec les objectifs métiers avant de passer aux détails techniques.
<b>Cahier des charges fonctionnel</b>	Document qui spécifie les fonctionnalités principales attendues du système du	Sert de base de discussion entre les parties prenantes métier et les équipes techniques pour définir les

	point de vue métier, sans entrer dans les détails techniques.	fonctionnalités du système. Aide à clarifier ce que le système doit accomplir pour répondre aux besoins métier.
<b>Document de spécifications des exigences (SRS)</b>	Document technique détaillant les exigences fonctionnelles et non fonctionnelles, ainsi que les contraintes techniques.	Fournit une référence technique complète pour les équipes de développement et de tests. Il permet de s'assurer que toutes les exigences (fonctionnelles et non fonctionnelles) sont couvertes et respectées lors de la conception et du développement du système.
<b>Diagrammes des cas d'utilisation</b>	Illustrations des interactions entre les utilisateurs et le système à travers des scénarios utilisateurs (Use Case Diagrams).	Clarifient les interactions entre les utilisateurs et le système, facilitant la compréhension des besoins fonctionnels et des comportements attendus du système.
<b>Diagrammes de flux de processus</b>	Visualisation des processus métiers et du flux de données dans le système, montrant les étapes et les décisions clés.	Aident à visualiser comment les processus d'affaires seront intégrés dans le système, et à clarifier les interactions complexes.
<b>Matrice de traçabilité des exigences (RTM)</b>	Tableau qui lie chaque exigence aux livrables correspondants (tests, composants logiciels, etc.), assurant la couverture complète des besoins.	Permet de suivre la conformité des exigences avec les composants du projet tout au long du cycle de vie du développement. Garantit que toutes les exigences sont prises en compte dans la conception, le développement, et les tests.
<b>Rapport d'analyse de faisabilité</b>	Évaluation de la faisabilité technique, temporelle et financière des exigences identifiées.	Permet de valider la viabilité des exigences avant de commencer la conception détaillée, en prenant en compte les contraintes de ressources et les risques.
<b>Plan de gestion des exigences</b>	Stratégie et processus de gestion des exigences tout au long du projet, incluant	Garantit une bonne gestion des exigences, en s'assurant qu'elles sont suivies, mises à jour, et contrôlées correctement tout au long du projet, ce



la gestion des modifications et des validations.	qui est essentiel pour la réussite globale.
--	---

## Phase 2 : Conception du système (Design)

### *Plan de la phase :*

Sur la base des exigences validées, nous allons élaborer l'architecture globale du système. Cela inclut la conception de l'architecture logicielle (choix des couches, des modules, des composants), la modélisation des bases de données, la définition des interfaces utilisateur et des API, ainsi que la sélection des technologies, langages de programmation et frameworks appropriés. Nous créerons également des prototypes pour certaines fonctionnalités clés afin de valider nos choix de conception avec le client avant de passer au développement complet.

### *Livrables :*

- **Document de conception logiciel (Software Design Document (SDD)) :** Ce document détaillera l'architecture du système, décrira les composants logiciels, les interfaces entre eux, les technologies utilisées et les normes de codage à respecter. Il servira de guide pour les développeurs lors de l'implémentation.

### Software Design Document (SDD) NovaFund

- **Modèles de données et schémas de base de données :** Ces documents représenteront les structures de données, les relations entre les différentes entités et les contraintes d'intégrité, ce qui est essentiel pour assurer la cohérence et l'efficacité du stockage des données.
- **Prototypes d'interfaces utilisateur :** Des maquettes visuelles ou des prototypes interactifs seront créés pour donner une idée concrète de l'apparence et de la convivialité du système, permettant au client de fournir des retours avant le développement.
- **Plan de test initial :** Nous définirons les stratégies de test, les types de tests à effectuer (unitaires, intégration, système, acceptation), ainsi que les critères d'acceptation pour garantir la qualité du produit final.

### *Importance des livrables :*

La phase de conception sert de plan directeur pour le développement. Une conception bien pensée assure que le système sera robuste, évolutif et aligné avec les besoins du client. Elle facilite également la répartition du travail entre les membres de l'équipe et permet d'estimer plus précisément les délais et les coûts. Les prototypes d'interface permettent d'impliquer le client et d'obtenir son adhésion sur l'aspect et la convivialité du système.

## **Phase 3 : Implémentation et Vérification (Implementation and Verification)**

### *Plan de la phase :*

Cette phase consiste à traduire la conception en code exécutable. Les développeurs seront assignés à des modules spécifiques en fonction de leurs compétences. Nous adopterons des méthodes de développement standardisées et utiliserons des outils de gestion de version pour assurer la cohérence du code. Parallèlement, nous organiserons un planning de tests rigoureux. Chaque développeur effectuera des tests unitaires sur son code, puis des tests d'intégration seront réalisés pour vérifier que les différents modules fonctionnent ensemble harmonieusement. Des revues de code régulières seront menées pour maintenir la qualité du code.

### *Livrables :*

- **Code source du système** : Le code complet de l'ERP, structuré selon l'architecture définie, bien commenté et conforme aux normes de codage établies.
- **Documentation technique** : Incluant des manuels de développeurs, des descriptions des API, et des guides pour faciliter la maintenance et les futures évolutions du système.
- **Rapports de tests** : Résultats détaillés des tests unitaires, d'intégration, et éventuellement des tests système préliminaires, incluant les bugs identifiés et les actions correctives entreprises.

- **Version bêta du logiciel :** Une version fonctionnelle de l'ERP, avec l'ensemble des fonctionnalités principales implémentées, prête pour les tests d'acceptation par le client.

#### *Importance des livrables :*

L'implémentation est le cœur du projet, où le travail précédemment planifié et conçu devient une réalité tangible. La qualité du code et des tests effectués détermine la fiabilité et la performance du système. La documentation technique est essentielle pour la maintenance future et pour tout nouveau membre rejoignant l'équipe. Les rapports de tests assurent la traçabilité des problèmes rencontrés et renforcent la confiance dans la qualité du produit.

## **Phase 4 : Déploiement et Maintenance (Deployment and Maintenance)**

#### *Plan de la phase :*

Après validation du système par le client, nous procéderons au déploiement de l'ERP dans l'environnement de production. Cela impliquera la configuration des serveurs, l'installation du logiciel, la migration des données existantes et la mise en place des mesures de sécurité nécessaires. Nous organiserons des sessions de formation pour les utilisateurs finaux et les administrateurs du système. Par la suite, nous assurerons la maintenance du système, en résolvant rapidement les éventuels problèmes qui pourraient survenir, en fournissant des mises à jour et en envisageant des améliorations basées sur les retours des utilisateurs.

#### *Livrables :*

- **Plan de déploiement détaillé :** Incluant les étapes spécifiques, les ressources nécessaires, les dates clés et les plans de contingence en cas de problème.
- **Système ERP opérationnel :** L'ERP pleinement fonctionnel, intégré dans l'environnement du client et prêt à être utilisé pour les opérations quotidiennes.
- **Guides d'utilisation et de formation :** Documentation destinée aux utilisateurs finaux pour les aider à utiliser efficacement le système, ainsi que des manuels pour les administrateurs pour la gestion quotidienne.

- **Plan de maintenance et de support** : Définissant les procédures pour signaler et résoudre les problèmes, les calendriers de maintenance préventive, et les modalités pour les mises à jour futures.
- **Rapports de satisfaction et de performance** : Collecte des retours des utilisateurs et des indicateurs de performance pour évaluer le succès du déploiement et identifier les axes d'amélioration.

*Importance des livrables :*

Le déploiement est la phase où le travail de toute l'équipe est finalement mis entre les mains du client. Un déploiement bien planifié et exécuté garantit une transition en douceur, minimisant les perturbations pour le client. La formation et les guides d'utilisation sont essentiels pour assurer une adoption rapide et efficace du système par les utilisateurs. La maintenance continue est cruciale pour la pérennité du système, pour maintenir la satisfaction du client et pour construire une relation de confiance à long terme.

## Waterfall

<https://kderec.monday.com/boards/1634569995/>

[Waterfall NovaFund.pdf](#)

	Q3 2024		Q4 2024			Q1 2025			Q2 2025			Q3 2025	
	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
<p>● Analyse des exigences (Requirement Analysis)</p> <p>Préparation des ateliers de recu 17 - 18 Sep</p> <p>Organisation des réunions avec 19 - 25 Sep</p> <p>Rédaction du document de spéc 26 Sep - 2 Oct</p> <p>Définition des cas d'utilisation e 3 - 8 Oct</p> <p>Création des diagrammes de flu 9 - 11 Oct</p> <p>Validation finale des exigences 14 - 15 Oct</p>			<p>■ Préparation des ateliers de recueil des besoins</p> <p>■ Organisation des réunions avec les parties prenant...</p> <p>■ Rédaction du document de spécifications des exigen...</p> <p>■ Définition des cas d'utilisation et des scénarios ...</p> <p>■ Création des diagrammes de flux de processus</p> <p>■ Validation finale des exigences</p>										
<p>● Conception du système (Design)</p> <p>Élaboration de l'architecture log 16 - 22 Oct</p> <p>Modélisation des bases de donn 23 - 28 Oct</p> <p>Création du document de conce 29 Oct - 5 Nov</p> <p>Conception des interfaces utilis 6 - 12 Nov</p> <p>Définition du plan de test initial 13 - 15 Nov</p> <p>Révision et validation de la conc 18 - 19 Nov</p>			<p>■ Élaboration de l'architecture logicielle</p> <p>■ Modélisation des bases de données</p> <p>■ Création du document de conception logicielle (SDD...</p> <p>■ Conception des interfaces utilisateur</p> <p>■ Définition du plan de test initial</p> <p>■ Révision et validation de la conception</p>										
<p>● Implémentation et Vérification (Implementation and Verification)</p> <p>Mise en place de l'environneme 20 - 28 Nov</p> <p>Implémentation du backend (AF 29 Nov - 19 Dec</p> <p>Implémentation du frontend (In 20 Dec, '24 - Jan 9, '25</p> <p>Intégration des modules Jan 27, '25 - Feb 7, '25</p> <p>Implémentation des fonctionnal Feb 26, '25 - Mar 11, '25</p> <p>Tests unitaires et d'intégration Mar 12, '25 - Apr 1, '25</p> <p>Préparation de la version bêta Apr 2, '25 - Apr 8, '25</p>				<p>■ Mise en place de l'environnement de développement</p> <p>■ Implémentation du backend (API et base de données)...</p> <p>■ Implémentation du frontend (Interface utilisateur)...</p> <p>■ Intégration des modules</p> <p>■ Implémentation des fonctionnalités de reporting et...</p> <p>■ Tests unitaires et d'intégration</p> <p>■ Préparation de la version bêta</p>									
<p>● Déploiement et Maintenance (Deployment and Maintenance)</p> <p>Préparation du déploiement Apr 9, '25 - Apr 15, '25</p> <p>Déploiement initial Apr 16, '25 - Apr 22, '25</p> <p>Tests post-déploiement Apr 23, '25 - Apr 29, '25</p> <p>Formation et lancement Apr 30, '25 - May 6, '25</p>									<p>■ Préparation du déploiement</p> <p>■ Déploiement initial</p> <p>■ Tests post-déploiement</p> <p>■ Formation et lancement</p>				

## Conclusion

Ce plan détaillé pour chaque phase du modèle Waterfall nous permettra de gérer le projet ERP de manière structurée et efficace. En définissant clairement les activités, les livrables et leur importance à chaque étape, nous assurons la transparence avec le client et au sein de l'équipe. Cela facilite également le suivi de l'avancement du projet et la gestion des risques. En suivant rigoureusement ce plan, nous maximiserons nos chances de livrer un produit de haute qualité qui répondra pleinement aux attentes du client, tout en respectant les délais et le budget établis.

## Prochaines étapes

Je propose que nous organisions une réunion de lancement avec l'équipe projet pour détailler ce plan et attribuer les responsabilités. Nous pourrions également planifier les premières interactions avec le client pour démarrer l'analyse des

exigences. Je reste à votre disposition pour discuter de ce plan et apporter les ajustements nécessaires.

**Merci de votre attention.**