

# Méthodes et outils projet

Formation

# PLAN DE COURS

**Chapitre 1** : Le phasing

**Chapitre 2** : Initialisation du projet

**Chapitre 3** : Définition du périmètre

**Chapitre 4** : Découpage

**Chapitre 5** : Gestion des risques

**Chapitre 6** : Planification

[Image par Freepik](#)

# Chapitre 1 : Le phasing

# Le Phasing dans la gestion de projet informatique

- Découpage logique du projet en étapes successives
- Chaque phase produit un ou plusieurs livrables
- Permet de :
  - Mieux planifier et contrôler le projet
  - Faciliter la communication entre les acteurs
  - Réduire les risques

# Les objectifs du phasing

- Structurer la démarche projet
- Donner de la visibilité à l'équipe et aux parties prenantes
- Identifier les points de validation (jalons)
- Préparer la planification et le suivi



# Les grandes phases d'un projet



# Le phasing

- Chaque phase se conclut par un jalon décisionnel qui décide de la suite
- Les instances de pilotage valident l'avancement
- Permet de maîtriser coûts, délais, qualité



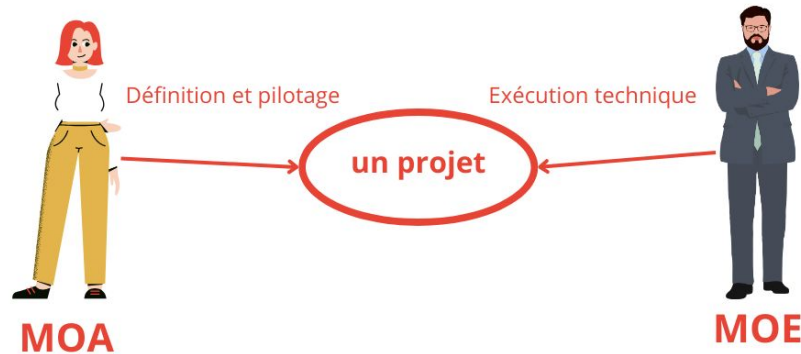
# Facteurs clés

- Phasing clairement défini et partagé
- Livrables intermédiaires validés
- Implication des parties prenantes
- Documentation claire
- Communication fluide entre MOA et MOE



# MOE et MOA dans un projet

- Maîtrise d'ouvrage (MOA) : définir les besoins et les objectifs métier du système, définir la direction du projet, son périmètre. (Partie client)
- Maîtrise d'oeuvre (MOE) : concevoir, développer et mettre en oeuvre la solution technique



# Chapitre 2 : initialisation du projet

# Rôle de la phase d'initialisation

- Première étape concrète du cycle de vie du projet
- Vise à **valider la faisabilité** et à **poser le cadre du projet**
- Principales actions :
  - Identifier les besoins et objectifs
  - Définir les acteurs et rôles
  - Évaluer les contraintes et risques initiaux
  - Produire la charte de projet



Étape cruciale qui évite les problèmes en aval si bien réalisée

# Objectifs de l'initialisation

1. Clarifier le **pourquoi** du projet
2. Aligner les **attentes des parties prenantes**
3. Identifier les **ressources et moyens** disponibles
4. Produire un **document de cadrage** (charte) validé par le sponsor



# Livrables attendus

Livrable	Contenu	Utilité
Charte de projet	Objectifs, périmètre initial, acteurs, budget, planning préliminaire	contrat entre MOA et MOE
Expression des besoins	Liste des exigences du client/utilisateur	Base pour le périmètre et la planification
Analyse de faisabilité	Étude technique, financière, organisationnelle	Vérifie la viabilité du projet

# Collecte des exigences

- Comprendre ce que veut vraiment le client/utilisateur
- 2 types d'exigences :
  - **Fonctionnelles** : ce que le système doit faire
  - **Non-fonctionnelles** : performances, sécurité, ergonomie
- Une exigence doit être **claire, mesurable, testable**

# Comment collecter les exigences

- **Entretiens individuels** (discussion avec les utilisateurs)
- **Ateliers** (réunions collaboratives)
- **Questionnaires** (information ciblée)
- **Observations / analyse** (via le travail réel ou des rapports existants)

# Catégorisation et priorisation

- **Catégoriser :**
  - Trier les exigences selon leur nature et importance
  - Must / Should / Could / Won't (Méthode MoSCoW)
- **Prioriser :**
  - Selon la valeur métier et le coût de mise en œuvre
  - Utiliser des matrices (effort/valeur)



# Catégoriser : classification F/NF/TC

<b>F – Fonctionnelle</b>	Ce que le système doit faire (fonctions, traitements, comportements attendus)	Authentification utilisateur, génération de rapports, envoi de notifications
<b>NF – Non-Fonctionnelle</b>	Qualités attendues du système, contraintes techniques, ergonomiques ou sécuritaires	Temps de réponse < 2 secondes, conformité RGPD
<b>TC – Technique / Contrainte</b>	Exigences liées à l'environnement technique, aux normes ou à l'infrastructure	Base de données SQL, intégration avec l'ERP existant, hébergement cloud

# Prioriser : méthode MoSCoW

M - Must have	Indispensable	Fonctionnalité critique sans laquelle le projet échoue
S - Should have	Important	Fonction souhaitée pour la version initiale mais non bloquante
C - Could have	Confort	Fonction optionnelle si le temps/budget le permet
W - Won't have	Exclu ou différé	Fonction volontairement mise de côté pour une version ultérieure

# Risques liés à une mauvaise initialisation

- Mauvaise compréhension du besoin
- Manque d'adhésion des parties prenantes
- Sous-estimation du budget ou du délai
- Livrables non conformes aux attentes

# Exercices

# Chapitre 3 : définition du périmètre

# Rôle de la définition du périmètre

- Précise ce qui sera livré (et ce qui ne le sera pas)
- Sert de base au découpage et à la planification
- Permet de limiter les dérives et clarifier les attentes
- Si le périmètre est mal défini des dépassements sont assurés

# Objectif de cette phase

- Décrire la solution envisagée
- Délimiter le **champ d'action** du projet
- Identifier les **livrables majeurs**
- Déterminer les **exclusions et hypothèses**
- Formaliser un **document de périmètre** validé par les parties prenantes



# Périmètre du projet vs périmètre de la solution

Aspect	Périmètre du projet	Périmètre de la solution
Définition	Ensemble des activités à réaliser	Ensemble des fonctionnalités / composants livrés
Exemple	Développement, tests, formation, déploiement	Logiciel de gestion des stocks, interface web, autre module
Responsabilité	Chef de projet	Équipe fonctionnelle et/ou technique



Périmètre projet = “comment” - périmètre solution = “quoi”



# Exploiter la charte de projet

- La **charte de projet**, issue de l'initialisation, fournit :
  - Les objectifs et indicateurs de réussite
  - Les acteurs et parties prenantes
  - Les contraintes (budget, délai, ressources)
  - Le contexte et les dépendances
- C'est un point de départ pour cadrer le périmètre

# Périmètre : différents livrables

- Documentaire (Cahier des charges fonctionnel, plan de tests)
- Technique (Application développée, base de données, interface)
- Organisationnel (Formation, documentation utilisateur...)
- Communication (supports de présentation, fiches projet...)



Chaque livrable identifié doit être daté et validé

# Périmètre : définir les exclusions

- Exclure les parties non couvertes de manière à éviter les malentendus
- “Ce projet **ne couvre pas** ... “
- Exemples :
  - Le développement mobile est hors périmètre
  - La migration des données historiques et à la charge du client

# Exemple d'outils pour définir le périmètre

- Cahier des charges (besoins et attentes)
- WBS préliminaire (aperçu du découpage)
- Matrice
- Carte mentale

# Exercices

# Chapitre 4 : découpage

# Rôle du découpage

- Transformer les livrables du périmètre en éléments concrets et pilotables
- Organiser le travail de manière logique
- Faciliter la planification et le suivi
- Clarifier les responsabilités
- Identifier les dépendances



# Pourquoi découper

1. Pour rendre le projet **compréhensible** et **structuré**
2. Pour **estimer coûts, durées** et **ressources**
3. Pour déléguer et **suivre plus facilement**
4. Pour **identifier les points de contrôle** (jalons)
5. Pour **éviter les oublis** ou **redondances**



# Les trois types de découpage

Type	Signification	Objectif
WBS	Découpage du travail à réaliser	Vue activités / tâches
PBS	Découpage du produit / livrables	Vue résultats attendus
OBS	Découper par organisation / responsabilités	Vue acteurs / ressources



Ces structures se complètent pour offrir une vision plus complète

# Le WBS (Work Breakdown Structure)

- Structure hiérarchique du **travail à accomplir**
- Chaque niveau correspond à un **degré de détail**
- Le dernier niveau (work package) = **unité de travail mesurable**

1. Développement du site e-commerce  
1.1 Analyse des besoins  
1.2 Conception  
1.3 Développement  
1.3.1 Front-end  
1.3.2 Back-end  
1.4 Tests  
1.5 Déploiement

# Le PBS (Product Breakdown Structure)

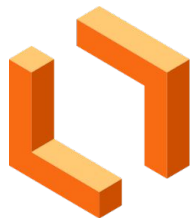
- Structure des livrables du projet
- Chaque élément représente un **composant de la solution finale**

- 1. Application e-commerce
  - 1.1 Interface utilisateur
  - 1.2 Base de données produits
  - 1.3 Module de paiement
  - 1.4 Tableau de bord administrateur

# Le OBS (Organizational Breakdown Structure)

- Structure hiérarchique des **acteurs impliqués**
- Permet de relier **chaque lot de travail à une équipe ou un rôle**

1. Direction de projet
  - 1.1 Chef de projet
  - 1.2 PMO
2. Équipe technique
  - 2.1 Développeur front-end
  - 2.2 Développeur back-end
3. Équipe fonctionnelle
  - 3.1 Analyste métier
  - 3.2 Testeur utilisateur



**Lucidchart**

# Méthodologie de découpage

1. Lister les livrables issus du périmètre en arborescence (PBS)
2. Identifier les activités nécessaires à chaque livrable
3. Regrouper les activités par lots logiques
4. Créer la structure hiérarchique (WBS)
5. Associer des responsables (OBS)



# Coordination et matrice RACI

- **Responsible** : réalise l'action
- **Accountable** : porte la responsabilité finale
- **Consulted** : donne une expertise
- **Informed** : est tenu informé



La matrice RACI permet de clarifier “qui fait quoi” sur chaque tâche

# Coordination et matrice RACI : exemple

Activité	Chef de projet	Dév	Testeur	Client
Cahier des charges	A		C	C
Développement du module	I	R		
Test de validation	I	C	R	C
Recette finale	A	I	C	R



# Exercices

# Chapitre 5 : gestion des risques

# Qu'est ce qu'un risque

- Événement incertain pouvant avoir un impact positif ou négatif sur le projet.
- Risque **négatif** : **menace** (ex. retard, panne, bug critique)
- Risque **positif** : **opportunité** (ex. gain de productivité, nouvelle technologie efficace)
- Peut concerner : le planning, le budget, la qualité, les ressources, ou la technique

# Pourquoi gérer les risques ?

- Anticiper les problèmes avant qu'ils ne surviennent
- Réduire les incertitudes et les effets de surprise
- Sécuriser les coûts, délais et la qualité
- Faciliter la prise de décision (pilotage par le risque)

# Analyse et positionnement du risque

- Objectif : évaluer la **gravité de chaque risque**
- **Probabilité** : chances que le risque se produise
- **Impact** : niveau de conséquence sur le projet
- Criticité =  $P \times I$

Impact / probabilité	Faible	Moyenne	Forte
Faible	Vert	Jaune	Orange
Moyen	Jaune	Orange	Rouge
Fort	Orange	Rouge	Rouge (très)



On pourra utiliser la matrice probabilité impact

# Traitement du risque

- Définir une **stratégie adaptée** à chaque risque

Stratégie	Description	Exemple
Éviter	Supprimer la cause du risque	Ne pas utiliser une techno instable
Réduire	Diminuer la probabilité ou l'impact	Ajouter des tests automatisés
Transférer	Déléguer à un tiers	Sous-traiter ou assurer le risque
Accepter	Assumer le risque avec plan B	Prévoir une marge de planning

# Plan d'action associé

Pour chaque risque majeur :

- Identifier un responsable du risque
- Décrire les actions préventives
- Prévoir les actions correctives
- Suivre l'évolution

Risque	Responsable	Action préventive	Plan B
Retard de développement	Chef de projet	Ajuster la charge, réunions hebdo	Ajout d'un renfort technique

# Exercices



# Chapitre 6 : planification

# Pourquoi planifier

- Donner une **vision temporelle** claire **du projet**
- Assurer la **cohérence** entre les tâches, ressources et livrables
- Permettre le **suivi de l'avancement**
- Identifier les dérives de délai ou de charge



# Les types de planification

- **Planification prévisionnelle** : plan initial validé avant lancement (cadre de référence)
- **Planification réelle** : plan réajusté selon l'avancement réel (suivi et pilotage)
- **Replanification** : ajustement après dérive ou imprévu (maintenir la maîtrise du projet)

# Étapes de la planification

1. Identifier les **tâches** issues du WBS
2. Déterminer leur **durée estimée**
3. Identifier les **dépendances** entre tâches
4. Affecter les **ressources humaines** et **matérielles**
5. Déterminer les **jalons** (points de validation)
6. Établir le **planning global** (Gantt / PERT)

# Focus sur Gantt

- Outil visuel de planification et de suivi
- Chaque tâche = une barre horizontale
- Axe horizontal = temps
- Axe vertical = tâches / lots de travail
- Permet de visualiser :
  - Début et fin de chaque tâche
  - Chevauchements et dépendances
  - État d'avancement
- ganttproject

# Planification prévisionnelle versus réelle

- Attention aux chemins critiques :
  - l'ensemble des tâches sans marge
  - Si une tâche critique est retardée c'est tout le projet qui l'est
- Les risques identifiés influent la planification :
  - Marges de sécurité (temps, budget)
  - Scénarios alternatifs (plans B)
  - Jalons de contrôle supplémentaires

# Exercices

# **TP : Validation des acquis**