

# GUIDE CAHIER DES CHARGES

Octobre 2024

Blog intéressant: <https://marclou.beehiiv.com/>

## 1. Page de garde

- **Nom du projet** : Titre précis et court du projet.
- **Date** : Date de création du document.
- **Version** : Numéro de version du document.
- **Responsables** : Nom(s) des responsables du projet (gestionnaire de projet, client, etc.).
- **Résumé** : Brève description du projet.

## 2. Introduction

- **Contexte** : Description du contexte dans lequel le projet a été initié (problème ou opportunité).
- **Objectifs du projet** : Définir les objectifs globaux (ce que le projet doit accomplir).
- **Périmètre du projet** : Préciser ce qui est inclus et ce qui est exclu du projet.

## 3. Présentation des acteurs

- **Client** : Nom et rôle du client (peut être une entreprise ou un département interne).
- **Utilisateurs finaux** : Décrire les profils des utilisateurs finaux (qui va utiliser la solution).
- **Équipe projet** : Définir les rôles des membres de l'équipe impliquée dans le projet (chef de projet, développeurs, testeurs, etc.).

## 4. Description du besoin

- **Problème à résoudre** : Détails sur le problème que le projet doit résoudre.
- **Opportunité à saisir** : Si applicable, préciser l'opportunité (nouveau marché, amélioration de processus, etc.).
- **Fonctionnalités principales attendues** : Lister et décrire brièvement les fonctionnalités clés que doit offrir le produit ou la solution finale.

## 5. Spécifications fonctionnelles

- **Fonctionnalités** : Détail des fonctionnalités que l'application ou le système doit fournir. Chaque fonctionnalité doit être décrite de manière précise et compréhensible par tous les acteurs du projet.
  - Par exemple :
    - **Gestion des utilisateurs** : Créer, lire, modifier, supprimer des utilisateurs.
    - **Authentification** : Système de login/mot de passe.
    - **Tableau de bord** : Visualisation des données en temps réel.
- **Scénarios d'utilisation (user stories)** : Décrire des scénarios typiques d'utilisation des fonctionnalités du système.
  - Exemples :
    - "En tant qu'administrateur, je souhaite pouvoir ajouter de nouveaux utilisateurs pour gérer l'accès à l'application."

- "En tant qu'utilisateur, je souhaite pouvoir modifier mes informations personnelles."

## 6. Spécifications techniques

- **Architecture logicielle** : Description de l'architecture du système (client-serveur, microservices, etc.).
- **Technologies utilisées** : Indiquer les technologies retenues (langages de programmation, frameworks, bases de données, etc.).
- **Exigences matérielles** : Si applicable, préciser les besoins en termes de matériel (serveurs, infrastructures, etc.).
- **Exigences en performance** : Indiquer les attentes de performance (temps de réponse, capacité de montée en charge, etc.).
- **Sécurité** : Décrire les mesures de sécurité à implémenter (authentification, protection des données, etc.).

## 7. Contraintes

- **Délais** : Dates clés, calendrier du projet, deadlines.
- **Budget** : Définition des ressources financières disponibles.
- **Contraintes réglementaires** : Préciser les réglementations à respecter (GDPR pour les données personnelles, etc.).
- **Contraintes techniques** : Compatibilité avec d'autres systèmes, limites technologiques, etc.

## 8. Risques

- **Identification des risques** : Lister les principaux risques qui pourraient affecter le projet (retards, problèmes techniques, etc.).
- **Plans de mitigation** : Décrire les solutions de contournement ou les stratégies de réduction des risques.

## 9. Organisation du projet

- **Rôles et responsabilités** : Détail des rôles de chaque acteur impliqué dans le projet.
- **Planning prévisionnel** : Plan détaillé avec les différentes étapes du projet (phase de conception, développement, tests, déploiement).
- **Livrables attendus** : Lister les documents et les éléments concrets à fournir tout au long du projet (maquettes, prototypes, rapports, etc.).

## 10. Processus de validation

- **Critères de validation** : Définir les critères d'acceptation du produit final (fonctionnalités, qualité, respect des délais).
- **Tests** : Détail des types de tests à réaliser (tests unitaires, tests fonctionnels, tests de charge, etc.).
- **Recette** : Processus formel de validation avec le client.

## 11. Maintenance et support

- **Plan de maintenance** : Décrire comment le produit sera maintenu après la livraison (correctifs, mises à jour).
- **SLA (Service Level Agreement)** : Niveau de service attendu après la livraison (temps de réponse, disponibilité, etc.).

## 12. Annexes

- Ajouter toutes les informations supplémentaires qui peuvent être utiles (glossaire, diagrammes UML, etc.).

### Conseils supplémentaires :

1. **Être précis** : Plus les spécifications sont claires et détaillées, plus il sera facile de gérer le projet.
2. **Impliquer les parties prenantes** : S'assurer que tous les acteurs du projet ont bien compris et validé les besoins et les contraintes.
3. **Réviser régulièrement** : Le cahier des charges n'est pas figé. Il peut être révisé si les besoins évoluent.

### Exemples d'outils pour créer un cahier des charges :

- Google Docs ou Microsoft Word pour rédiger.
- Diagrammes UML : Lucidchart, Microsoft Visio.
- Outils de gestion de projet : Trello, Asana, Jira.

# **Description du besoin**

Décrire le besoin dans un cahier des charges est essentiel pour garantir une compréhension claire et partagée des attentes du projet, afin d'assurer que toutes les parties prenantes travaillent vers un objectif commun et réaliste.

# 1. Outils pour la description du besoin

## 1.1 Analyse SWOT

L'analyse SWOT (Strengths, Weaknesses, Opportunities, Threats) permet d'évaluer les forces, faiblesses, opportunités, et menaces d'un projet ou produit. C'est un bon point de départ pour décrire le besoin, en prenant en compte les facteurs internes et externes.

SWOT	Description
<b>Forces</b> (interne)	Atouts internes du projet (ex. : expertise, innovation, ressources).
<b>Faiblesses</b> (interne)	Lacunes ou limitations (ex. : manque de budget, dépendance à des technologies).
<b>Opportunités</b> (externe)	Facteurs externes favorables (ex. : tendances du marché, nouvelles technologies).
<b>Menaces</b> (externe)	Facteurs externes défavorables (ex. : concurrence, risques économiques).

## 1.2 Modèle de Kano

Le modèle de Kano permet de catégoriser les besoins des utilisateurs en fonction de l'impact qu'ils ont sur la satisfaction. Cela aide à définir les priorités pour les fonctionnalités du produit ou du service.

Catégorie	Description
<b>Fonctionnalités de base</b>	Nécessaires pour que le produit fonctionne (si manquantes, elles créent de l'insatisfaction).
<b>Fonctionnalités de performance</b>	Proportionnelles à la satisfaction utilisateur (plus c'est performant, plus les utilisateurs sont satisfaits).
<b>Fonctionnalités attractives</b>	Non attendues, mais qui génèrent de la satisfaction si présentes (surprise positive).
<b>Indifférence</b>	Les utilisateurs ne remarquent pas vraiment la présence ou l'absence de ces fonctionnalités.

### 1.3 Matrice MoSCoW

La matrice MoSCoW aide à prioriser les besoins fonctionnels et non-fonctionnels d'un projet en fonction de leur importance.

Catégorie	Description
Must Have	Fonctionnalités essentielles, sans lesquelles le projet ne peut pas être un succès.
Should Have	Fonctionnalités importantes mais qui peuvent être reportées si nécessaire.
Could Have	Fonctionnalités souhaitées mais non critiques ; elles ajoutent de la valeur si incluses.
Won't Have	Fonctionnalités qui ne seront pas intégrées dans cette version ou phase du projet.

### 1.4 User Stories

Les User Stories permettent de décrire les besoins fonctionnels en se plaçant du point de vue de l'utilisateur. Elles sont particulièrement utiles pour définir les fonctionnalités en termes de cas d'utilisation.

User Story	Description
En tant que [type d'utilisateur]	Qui va utiliser la fonctionnalité.
Je veux [action ou fonctionnalité]	Ce que l'utilisateur veut accomplir.
Afin de [objectif]	Pourquoi l'utilisateur en a besoin.

### 1.5 Persona

Un persona est une représentation fictive d'un utilisateur type. Cela aide à comprendre les besoins spécifiques d'un groupe d'utilisateurs ciblés.

Nom du Persona	Description
Nom	Nom fictif du persona.
Âge	Âge moyen de ce type d'utilisateur.
Fonction	Ce que cette personne fait au quotidien (travail, loisirs, etc.).
Objectifs	Ce que cet utilisateur cherche à accomplir via le produit ou service.
Frustrations	Les problèmes ou frustrations qu'il rencontre dans des solutions existantes.



## 2. Outils pour l'étude de marché

### 2.1 Matrice PESTEL

La matrice PESTEL est un outil d'analyse des facteurs externes qui peuvent affecter un projet. Cela permet de mieux comprendre l'environnement macroéconomique du marché.

Facteur	Description
Politique	Réglementations, politiques gouvernementales, stabilité politique.
Économique	Situation économique actuelle (inflation, taux d'intérêt, croissance économique).
Sociologique	Tendances démographiques, comportements des consommateurs.
Technologique	Innovations, avancées technologiques, digitalisation.
Environnemental	Sensibilité environnementale, réglementation écologique.
Légal	Lois en vigueur, contraintes juridiques à respecter (réglementation locale, internationale).

### 2.2 Les 5 forces de Porter

Le modèle des 5 forces de Porter est un outil d'analyse concurrentielle. Il permet d'évaluer l'attractivité d'un marché et de mieux comprendre les dynamiques concurrentielles.

Force	Description
Rivalité entre concurrents	Intensité de la concurrence dans le secteur.
Menace des nouveaux entrants	Facilité ou difficulté pour de nouveaux acteurs d'entrer sur le marché.
Pouvoir de négociation des fournisseurs	Influence des fournisseurs sur les coûts et la qualité.
Pouvoir de négociation des clients	Influence des clients sur les prix et les caractéristiques du produit.
Menace des produits de substitution	Disponibilité de produits alternatifs pouvant remplacer ceux proposés.



## 2.3 Segmentation de marché

La segmentation de marché permet de diviser le marché en sous-groupes d'utilisateurs ayant des besoins ou comportements similaires.

Segment	Caractéristiques
Segment démographique	Âge, sexe, revenu, niveau d'éducation.
Segment géographique	Localisation, région, climat.
Segment psychographique	Style de vie, personnalité, valeurs.
Segment comportemental	Comportement d'achat, fidélité, utilisation du produit.

## 2.4 Analyse des concurrents

Analyser la concurrence permet de mieux comprendre qui sont les principaux acteurs du marché, leurs forces et leurs faiblesses, et comment se positionner par rapport à eux.

Concurrent	Produits ou services	Forces	Faiblesses	Positionnement sur le marché
Concurrent A	Produit X	Prix compétitif, bonne réputation	Manque d'innovation	Leader sur le segment B
Concurrent B	Service Y	Offre personnalisée	Temps de réponse long	Challenger avec potentiel

## 2.5 Étude des tendances du marché

L'étude des tendances permet de comprendre comment évolue le marché et d'anticiper les besoins futurs des consommateurs.

Tendance	Impact sur le projet ou produit
Digitalisation accrue	Nécessité de services en ligne ou de digitalisation de l'offre.
Croissance des solutions SaaS	Opportunité de développer des services cloud ou d'abonnement.
Préférence pour les produits écoresponsables	Intégrer des pratiques durables dans la chaîne de production.

### 3. Outils numériques pour organiser la description du besoin et l'étude de marché

1. **Google Sheets / Excel** : Très utiles pour créer des tableaux comme les matrices SWOT, PESTEL, MoSCoW, etc.
2. **Trello / Asana / Monday** : Pour organiser les tâches relatives à la recherche et à la formalisation des besoins.
3. **Miro / Lucidchart** : Pour créer des diagrammes visuels (persona, user stories, mind mapping pour définir les besoins).
4. **SurveyMonkey / Google Forms** : Pour recueillir des données directement auprès des utilisateurs ou des segments de marché.
5. **SEMrush / Google Trends** : Pour analyser les tendances du marché et la concurrence.

# **Spécification fonctionnelles**

Les spécifications fonctionnelles dans un cahier des charges permettent de détailler précisément le comportement attendu du système ou produit, assurant ainsi que chaque fonctionnalité réponde aux besoins définis tout en facilitant la validation et le développement efficace du projet.

## 1. Jira

- **Description** : Jira est un outil de gestion de projet très utilisé pour organiser et suivre les spécifications fonctionnelles, particulièrement dans des projets de développement agile. Il permet de créer des tickets pour chaque spécification fonctionnelle, de les lier à des user stories, et de suivre leur avancement avec des workflows personnalisés.
- **Avantages** :
  - Collaboration simplifiée entre les équipes (développeurs, testeurs, managers).
  - Organisation claire des tâches, des user stories et des spécifications sous forme de tickets.
  - Suivi du cycle de vie des spécifications (développement, test, validation).
- **Utilisation** : Créer des tickets pour chaque fonctionnalité, les assigner aux équipes, suivre l'évolution via des sprints et des tableaux Kanban.

## 2. Confluence

- **Description** : Confluence, souvent utilisé en complément de Jira, est un outil de documentation collaboratif. Il permet de rédiger les spécifications fonctionnelles de manière détaillée, sous forme de pages organisées en espaces de travail, et d'y ajouter des diagrammes, des captures d'écran et des commentaires.
- **Avantages** :
  - Centralisation des spécifications fonctionnelles et de la documentation du projet.
  - Intégration directe avec Jira (les spécifications documentées dans Confluence peuvent être liées à des tickets dans Jira).
  - Fonctionnalités de collaboration en temps réel (commentaires, révisions).
- **Utilisation** : Rédiger les spécifications fonctionnelles sous forme de pages, ajouter des annexes comme des diagrammes UML, et collaborer avec les parties prenantes.

## 3. Visual Paradigm

- **Description** : Visual Paradigm est un outil de modélisation et de conception qui permet de définir les spécifications fonctionnelles à travers des diagrammes UML (cas d'utilisation, diagrammes de séquence, diagrammes d'activités). Il est très utile pour visualiser et organiser les flux fonctionnels complexes.
- **Avantages** :
  - Création de diagrammes visuels pour représenter les interactions fonctionnelles.
  - Idéal pour des projets nécessitant des spécifications détaillées et bien structurées (UML, BPMN).
  - Peut générer automatiquement une documentation fonctionnelle à partir des diagrammes.

- **Utilisation** : Modéliser les processus, les interactions, et les fonctionnalités avec des diagrammes, puis générer des documents de spécifications à partir de ces modèles.

Ces trois outils offrent des fonctionnalités complémentaires : **Jira** pour le suivi des spécifications, **Confluence** pour la documentation détaillée et la collaboration, et **Visual Paradigm** pour la modélisation visuelle. Ils couvrent les différents besoins de gestion, documentation, et structuration des spécifications fonctionnelles.

# **Spécification techniques**

Les spécifications techniques dans un cahier des charges définissent les exigences technologiques et les contraintes à respecter, garantissant que les solutions choisies sont adaptées, viables et cohérentes avec l'infrastructure et les objectifs du projet.

## 1. Microsoft Visio / Lucidchart (Outils de modélisation technique)

- **Description** : Ces outils sont utilisés pour créer des diagrammes techniques détaillés, comme des **diagrammes UML**, des **diagrammes de flux de données**, des **architectures logicielles** et des **schémas de base de données**. Ils permettent de représenter visuellement les aspects techniques du projet, tels que l'architecture système ou le modèle de données.
- **Avantages** :
  - Visualisation claire des composants techniques (infrastructure, réseaux, bases de données, etc.).
  - Large bibliothèque de symboles pour créer des diagrammes techniques spécifiques.
  - Collaboration possible dans le cloud (Lucidchart) pour permettre des modifications en temps réel.
- **Utilisation** :
  - Créer des **diagrammes d'architecture logicielle** (par exemple, schéma client-serveur, microservices).
  - Modéliser des **diagrammes de classes UML** pour représenter les relations entre différents composants logiciels.
  - Dessiner des **flux de données** pour montrer comment les informations circulent entre les systèmes.

## 2. PlantUML (Outil de documentation technique automatisée)

- **Description** : PlantUML est un outil open-source permettant de créer des diagrammes UML à partir de simples lignes de texte. Il est particulièrement utile pour intégrer directement les diagrammes dans la documentation technique ou les outils de gestion de versions, comme Git. Cela permet d'automatiser la création et la mise à jour des diagrammes en fonction des modifications apportées au code ou aux spécifications.
- **Avantages** :
  - Intégration facile avec des outils de versionnage (GitHub, GitLab).
  - Génération rapide et automatisée de diagrammes techniques à partir de texte.
  - Idéal pour les équipes qui utilisent des pratiques DevOps et d'intégration continue (CI).
- **Utilisation** :
  - Rédiger des spécifications techniques sous forme de texte et les convertir automatiquement en diagrammes UML ou en schémas d'architecture.
  - Maintenir les diagrammes à jour avec le code source en temps réel.

## 3. Méthode 4+1 (Architecture logicielle)

- **Description** : La méthode **4+1** est une approche qui permet de décrire l'architecture d'un système à travers cinq vues distinctes : **vue logique**, **vue de développement**, **vue de processus**, **vue physique**, et **scénarios** (ou cas d'utilisation). C'est une méthode largement utilisée pour structurer les spécifications techniques dans les projets informatiques.

- **Avantages :**
  - Fournit une vue complète de l'architecture d'un système, à la fois du point de vue technique et fonctionnel.
  - Structure les spécifications techniques de manière cohérente en prenant en compte les différentes perspectives des équipes (développement, opérations, etc.).
  - Facilement adaptable à des projets complexes nécessitant une documentation technique détaillée.
- **Utilisation :**
  - **Vue logique** : Décrire la structure du logiciel (modèles de classes, interactions).
  - **Vue de développement** : Organiser le code source et les composants logiciels.
  - **Vue de processus** : Modéliser les processus d'exécution (concurrency, threads, etc.).
  - **Vue physique** : Décrire l'infrastructure matérielle et logicielle (serveurs, réseaux).
  - **Cas d'utilisation** : Scénarios typiques qui valident le bon fonctionnement de l'architecture.

**Visio / Lucidchart** : Outils de modélisation pour créer des diagrammes techniques détaillés comme UML, architecture système, schémas de flux.

**PlantUML** : Outil automatisé pour générer des diagrammes UML à partir de texte, idéal pour les projets avec des intégrations DevOps.

**Méthode 4+1** : Méthode pour structurer les spécifications techniques avec cinq vues différentes pour une vision complète de l'architecture.



# Risques & Contraintes

L'identification des risques et contraintes dans un cahier des charges permet d'anticiper les obstacles potentiels et de définir des stratégies d'atténuation, assurant ainsi une gestion proactive des imprévus pour maintenir le projet sur la bonne voie.

## 1. Matrice des risques (Risk Matrix)

La matrice des risques est un outil permettant d'évaluer les risques en fonction de deux critères : la **probabilité** d'occurrence et l'**impact** sur le projet. Cette matrice aide à prioriser les risques et à définir des actions pour les atténuer.

Risque	Probabilité	Impact	Niveau de risque	Plan de mitigation
Retard de livraison	Élevée	Élevé	Critique	Réévaluation du planning, ajout de ressources
Problème technique	Moyenne	Moyen	Modéré	Tests techniques approfondis avant déploiement
Manque de budget	Faible	Élevé	Élevé	Négociation avec les parties prenantes pour ajuster les coûts
Départ d'un développeur clé	Moyenne	Faible	Modéré	Documenter et partager les connaissances techniques

**Probabilité** : Fréquence à laquelle le risque peut se produire (Faible, Moyenne, Élevée).

**Impact** : Gravité des conséquences si le risque se matérialise (Faible, Moyen, Élevé).

**Niveau de risque** : Calculé en combinant la probabilité et l'impact. Cela aide à prioriser les actions à entreprendre.

**Plan de mitigation** : Actions spécifiques à mettre en place pour minimiser l'impact ou réduire la probabilité du risque.

## 2. Matrice des contraintes (Constraint Matrix)

La matrice des contraintes permet de lister et de surveiller les contraintes qui peuvent affecter le projet, qu'elles soient liées aux ressources, aux délais, au budget, ou à des facteurs techniques. Elle aide à anticiper les points bloquants et à trouver des solutions ou des ajustements.

Contrainte	Type	Impact sur le projet	Solution possible
Budget limité	Financière	Retards ou baisse de qualité	Priorisation des fonctionnalités
Délais serrés	Temporelle	Manque de tests ou de validations	Ajustement du périmètre du projet
Ressources humaines limitées	Ressources	Augmentation de la charge de travail des équipes	Outsourcing ou recrutement temporaire
Incompatibilité technique	Technique	Refonte d'une partie du système	Recherche d'une solution alternative

**Type** : Type de contrainte (Financière, Temporelle, Ressources, Technique).

**Impact sur le projet** : Décrire en quoi la contrainte affecte la progression ou la qualité du projet.

**Solution possible** : Propositions pour surmonter ou atténuer l'effet de la contrainte.

### **Outils numériques pour la gestion des risques et des contraintes :**

1. **Jira / Trello / Asana** : Ces outils de gestion de projet permettent de créer des tableaux de suivi des risques et des contraintes. Chaque risque ou contrainte peut être créé sous forme de "ticket" avec des champs personnalisés pour la probabilité, l'impact, les actions de mitigation, etc.
2. **Excel / Google Sheets** : Ces outils sont parfaits pour créer des matrices et des registres de risques et de contraintes avec des formules permettant de calculer automatiquement les niveaux de risque.

# **Organisation du projet et des parties prenantes**

L'organisation du projet et des parties prenantes dans un cahier des charges clarifie les rôles, responsabilités et processus de communication, assurant une coordination fluide et une prise de décision efficace tout au long du projet.

## 1. Matrice des parties prenantes (Stakeholder Matrix)

Cet outil permet de classer les parties prenantes en fonction de deux critères : leur **pouvoir** et leur **intérêt** dans le projet. La matrice permet d'adapter la stratégie de communication et d'implication pour chaque partie prenante.

Partie prenante	Pouvoir	Intérêt	Stratégie
Client principal	Élevé	Élevé	Collaborer et consulter
Utilisateurs finaux	Faible	Élevé	Informé régulièrement
Direction générale	Élevé	Faible	Gérer de près et informer à besoin
Fournisseur IT	Moyen	Moyen	Maintenir informé

**Pouvoir** : Influence qu'une partie prenante peut avoir sur le projet.

**Intérêt** : Niveau d'intérêt de la partie prenante concernant les résultats du projet.

**Stratégie** :

- **Gérer de près** : Pour ceux qui ont un pouvoir et un intérêt élevés.
- **Maintenir satisfait** : Ceux avec un pouvoir élevé mais un intérêt faible.
- **Informé régulièrement** : Ceux avec un intérêt élevé mais un faible pouvoir.
- **Surveiller** : Ceux avec peu de pouvoir et peu d'intérêt.

## 2. Matrice RACI (Responsable, Accountable, Consulted, Informed)

La matrice RACI clarifie les rôles et responsabilités de chaque partie prenante à chaque étape du projet. Cet outil est idéal pour s'assurer que tout le monde sait ce qu'il doit faire et comment il interagit avec les autres.

Tâche / Activité	Responsable (R)	Accountable (A)	Consulted (C)	Informed (I)
Définition des besoins	Chef de projet	Client	Utilisateurs	Direction
Développement	Développeurs	Chef de projet	Architecte	Client
Tests	Testeurs	Chef de projet	Utilisateurs	Direction
Déploiement	DevOps	Chef de projet	Administrateurs IT	Client, Utilisateurs

**Responsable (R)** : Ceux qui effectuent les tâches.

**Accountable (A)** : La personne qui est responsable ultime du succès de la tâche.

**Consulted (C) :** Ceux qui sont consultés et fournissent des avis ou des contributions à la tâche.

**Informed (I) :** Ceux qui sont informés de l'avancement et des résultats.

### 3. Analyse des parties prenantes (Stakeholder Analysis)

Ce tableau analyse les besoins, attentes et potentiels conflits entre les parties prenantes. Il permet de comprendre les positions de chacun, et d'adapter les stratégies de communication.

Partie prenante	Objectifs et attentes	Préoccupations	Stratégie d'engagement
Client principal	Respect des délais et du budget	Dépassements de coûts	Réunions hebdomadaires
Utilisateurs finaux	Facilité d'utilisation de l'application	Complexité de la solution	Démonstration et tests utilisateurs
Fournisseur IT	Livraison des composants techniques	Manque de spécifications précises	Spécifications détaillées régulières
Direction générale	ROI et impact financier positif	Retards de livraison	Rapports de performance trimestriels

### 5. Plan de communication des parties prenantes

Un tableau pour organiser les communications avec les différentes parties prenantes selon leur rôle dans le projet.

Partie prenante	Type de communication	Fréquence	Canal	Responsable
Client principal	Réunion de suivi projet	Hebdomadaire	En personne / Zoom	Chef de projet
Utilisateurs finaux	Mises à jour fonctionnelles	Bimensuel	Email / Portail web	Responsable des tests
Fournisseur IT	Statut des livrables	Mensuel	Email	Gestionnaire technique
Direction générale	Rapport financier	Trimestriel	Rapport écrit	Chef de projet

Ce tableau permet de s'assurer que la bonne information est délivrée à la bonne personne, au bon moment.

## **Outils numériques pour gérer les parties prenantes :**

- **Trello** : Pour créer des tableaux collaboratifs permettant de suivre les parties prenantes, leurs rôles et leur suivi.
- **Asana / Monday** : Outils de gestion de projet avec des fonctionnalités pour assigner des tâches et communiquer avec les parties prenantes.
- **Miro** : Outil de tableau blanc collaboratif pour créer des cartes de parties prenantes visuellement.
- **Microsoft Excel / Google Sheets** : Parfait pour créer des matrices RACI ou des tableaux d'analyse de parties prenantes.

# Product Market Fit & Go To Market

En somme, un **GTM** planifie comment amener le produit sur le marché de manière efficace, tandis que le **PMF** assure que ce produit a une réelle demande et qu'il satisfait bien son marché cible. Ces deux éléments sont cruciaux pour le succès commercial à long terme.

Importance du Go to Market (GTM)



Le Go to Market (GTM) est une stratégie essentielle pour introduire un produit ou un service sur le marché de manière efficace. Il définit la manière dont une entreprise positionne son offre, atteint ses clients cibles et génère des ventes. Un plan GTM bien conçu permet de :

1. **Minimiser les risques d'échec** : Il anticipe les obstacles et les opportunités, réduisant ainsi les incertitudes liées au lancement.
2. **Optimiser les ressources** : Un plan structuré permet d'allouer correctement les ressources humaines, financières et technologiques.
3. **Accélérer le retour sur investissement (ROI)** : En ciblant les bons clients avec les bons messages, un GTM maximise les chances de succès commercial dès les premiers mois.
4. **Différencier le produit** : Il met en avant ce qui rend l'offre unique sur le marché, aidant à se démarquer de la concurrence.

### Comment définir un Go to Market ?

1. **Cible** : Identifier les segments de marché précis qui bénéficieront le plus du produit ou service (personas).
2. **Positionnement** : Définir la proposition de valeur et pourquoi elle est pertinente par rapport à la concurrence.
3. **Canaux de distribution** : Choisir les meilleurs moyens pour atteindre ces segments (en ligne, direct, retail, etc.).
4. **Stratégie de tarification** : Déterminer un modèle tarifaire qui reflète la valeur perçue tout en restant compétitif.
5. **Plan marketing** : Développer une stratégie de communication (publicité, réseaux sociaux, etc.) pour sensibiliser et attirer des clients.
6. **Objectifs et KPIs** : Fixer des objectifs clairs pour mesurer le succès (ventes, croissance des utilisateurs, parts de marché).

### Outils pour le Go to Market (GTM)

1. **Business Model Canvas** : Un outil visuel permettant de cartographier rapidement les éléments clés du modèle d'affaires, comme la proposition de valeur, les segments de clientèle, les canaux de distribution, et les flux de revenus. Il aide à structurer le GTM de manière claire.
2. **Buyer Persona Templates** (ex. : **Xtensio**, **HubSpot**) : Ces outils permettent de créer des profils de clients cibles, en se basant sur des données démographiques, comportementales, et psychographiques. Ils aident à mieux comprendre les segments de clientèle à cibler dans le GTM.
3. **SWOT**: Un cadre classique pour identifier les forces, faiblesses, opportunités et menaces. Utilisé dans la phase GTM, il aide à évaluer les aspects externes et internes qui peuvent influencer le succès du produit sur le marché.
4. **Trello / Asana** : Des outils de gestion de projet pour organiser toutes les étapes du lancement du produit, suivre les tâches, et coordonner les équipes. Ils sont particulièrement utiles pour l'exécution du plan GTM.
5. **Google Trends** et **SEMrush** : Outils pour analyser les tendances de recherche et la demande autour des mots-clés liés à votre produit. Cela permet d'ajuster votre stratégie de positionnement et de comprendre ce que les clients recherchent activement.
6. **Mixpanel** ou **Google Analytics** : Pour suivre le comportement des utilisateurs sur vos plateformes, ces outils aident à évaluer quels canaux de communication et de distribution fonctionnent le mieux, vous permettant d'affiner votre GTM.

## Importance du Product Market Fit (PMF)

Le Product Market Fit (PMF) est le point où un produit répond parfaitement aux besoins du marché cible. C'est un indicateur clé de la viabilité d'un produit, car il montre que la demande est suffisamment forte pour soutenir une croissance durable. Un bon PMF apporte :

1. **Croissance accélérée** : Une demande forte génère des ventes rapides et une adoption naturelle.
2. **Satisfaction et fidélité client** : Un produit qui résout efficacement un problème pousse les clients à revenir et à recommander l'offre.
3. **Réduction des coûts marketing** : Quand le produit trouve sa place, il devient plus facile de convaincre de nouveaux clients, réduisant ainsi les efforts et les coûts de promotion.
4. **Attraction des investisseurs** : Un produit avec un bon PMF a de meilleures chances d'attirer des financements car il prouve la validation du marché.

### Comment définir le Product Market Fit ?

1. **Compréhension du problème client** : Connaître les besoins spécifiques et les douleurs du marché cible.
2. **Proposition de valeur claire** : Montrer en quoi le produit répond précisément à ces besoins de manière supérieure à la concurrence.
3. **Feedback utilisateur** : Recueillir des retours réguliers des utilisateurs pour ajuster le produit, notamment via des tests ou des pilotes.
4. **Indicateurs d'adoption** : Mesurer l'engagement des utilisateurs, les taux de rétention, et la satisfaction pour évaluer si le produit correspond aux attentes du marché.
5. **Adaptabilité** : Être prêt à pivoter si le produit n'obtient pas la traction escomptée. Le PMF est souvent atteint après des itérations ou améliorations.

### Outils pour le Product Market Fit (PMF)

1. **NPS (Net Promoter Score)** : Un outil de mesure de la satisfaction client. Le NPS vous permet de déterminer si vos utilisateurs recommanderaient votre produit à d'autres. Un score élevé est souvent un signe de Product Market Fit.
2. **Survey Tools** (ex. : **Typeform**, **SurveyMonkey**, **GForms**) : Ils permettent de collecter des feedbacks qualitatifs et quantitatifs auprès de vos utilisateurs. Les sondages peuvent aider à comprendre si le produit répond à leurs attentes et quels aspects doivent être améliorés pour atteindre le PMF.
3. **Customer Development Framework (Steve Blank)** : C'est une méthode qui encourage à interagir directement avec les clients dès le développement initial du produit. L'objectif est de valider les hypothèses sur les besoins du marché et de construire le produit en itérant en fonction des retours.
4. **Cohort Analysis** : Disponible dans des outils comme **Google Analytics**, **Mixpanel** ou **Amplitude**, cette méthode permet d'analyser les comportements de groupes spécifiques d'utilisateurs (cohortes) sur une période donnée. Cela vous aide à mesurer la rétention client et à voir si votre produit trouve de l'adhésion dans le marché.
5. **Pirate Metrics (AARRR)** : Cet outil popularisé par Dave McClure analyse cinq étapes critiques du cycle de vie utilisateur : Acquisition, Activation, Rétention, Referral et Revenue. Il est utilisé pour identifier à quel moment les utilisateurs décrochent et pour ajuster le produit afin d'atteindre le PMF.
6. **Lean Startup Methodology (Build-Measure-Learn)** : Cette approche repose sur des cycles rapides d'itérations. L'objectif est de construire un **MVP (Minimum Viable Product)**, de le tester rapidement auprès du marché cible, de mesurer les retours, puis d'apprendre des résultats pour améliorer le produit.
7. **Productboard** : Un outil de gestion de produit qui aide à centraliser les retours utilisateurs, prioriser les fonctionnalités, et aligner l'équipe sur la feuille de route du produit. Il est très utile pour identifier ce qui est important pour les utilisateurs afin d'atteindre le Product Market Fit.
8. **Churn Rate Analysis** : Le taux de désabonnement ou d'abandon est un indicateur clé pour le PMF. Si un nombre élevé d'utilisateurs abandonnent votre produit, cela peut indiquer que le produit n'atteint pas encore le PMF.