

Gabarit Cahier des charges

Ce document propose un exemple de structuration pour un cahier des charges. Ce n'est qu'un exemple général qui pourra être largement adapté en fonction des particularités de vos projets.

1. Présentation du projet

1.1. Contexte

Décrire l'environnement du projet : domaine, enjeux, problématique, utilisateurs ciblés, contexte académique ou organisationnel.

1.2. Objectifs

Résultats que le projet doit atteindre. Formuler des objectifs spécifiques, par exemple :

- Permettre à un utilisateur de...
- Réduire/optimiser...
- Assurer la sécurité de...

Spécifiez les critères de performance qui seront évalués. Par exemple, la rapidité, la fiabilité, la qualité de code, l'interface utilisateur, l'impact social, etc.

1.3. Périmètre du projet

Définir clairement ce qui fait partie du projet et ce qui n'en fait pas partie.

- Ce qui **est inclus** dans le projet. Par exemple, le développement de l'interface utilisateur, la création d'un backend, les tests de performance.
- Ce qui **est exclu** du projet. Par exemple, l'intégration avec des systèmes externes, certaines fonctionnalités avancées ou des processus non pertinents.

2. Description fonctionnelle

2.1. Besoins et exigences métiers

Décrire les besoins du client ou les exigences métiers auxquelles la solution doit répondre.

- **Problématique à résoudre** : Identification du problème ou du besoin à satisfaire. Par exemple, gérer une base de données, optimiser des algorithmes de tri, créer un site e-commerce.
- **Utilisateurs cibles** : Décrire les utilisateurs qui interagiront avec la solution. Par exemple, utilisateurs finaux, administrateurs, clients.
- **Scénarios d'utilisation** : Illustrations concrètes des actions attendues des utilisateurs. Vous pouvez joindre ou ajouter un diagramme de cas d'utilisation ici.

Par exemple, "L'utilisateur doit pouvoir créer un compte", "L'administrateur doit pouvoir générer des rapports".

2.2. Fonctionnalités principales

Liste des fonctionnalités que la solution doit absolument contenir. Chaque fonctionnalité doit être bien décrite. Par exemple, Gestion des utilisateurs, enregistrement des transactions, génération de rapports, etc.

2.3. Interface Utilisateur

Décrire les caractéristiques de l'interface graphique. Par exemple :

- **Ergonomie et design de l'interface** : Définir les attentes en termes d'interface. Par exemple, la simplicité, l'intuitivité, l'adaptation mobile, etc.
- **Technologies recommandées** : Les technologies pour l'interface utilisateur. Par exemple, le HTML avec CSS, React, etc.
- **Wireframes** : Ajout de schémas démontrant la base de l'interface graphique

2.4. Conditions d'utilisation

Décrire les **environnements** où la solution sera utilisée (PC, mobile, serveur, etc.). Spécifiez aussi des exemples de **limites d'utilisation** ou de contraintes liées à l'environnement.

3. Description technique

3.1. Technologies à utiliser

Spécifier toutes les technologies utilisées pour la réalisation du projet.

- Langages de programmation utilisé pour les différentes parties du projet.
- Frameworks, plateforme ou bibliothèques de code spécifiques recommandés ou exigés.
- Base de données utilisé.

3.2. Architecture du système

Spécifier, de façon générale, l'organisation du système que vous planifiez créer et de ses composants.

- Diagramme d'architecture, ou description textuelle de l'architecture du système. Spécifiez clairement le type d'architecture utilisé, comme le client-serveur, l'architecture microservices, etc.
- Schéma d'intégration si l'application doit interagir avec d'autres systèmes.

3.3. Sécurité

Spécifier les exigences de sécurité, comme les mesures à prendre concernant la sécurité des données, la gestion des identifiants, les communications cryptées, etc.

3.4. Performance et scalabilité

Spécifier les critères de performance et les exigences spécifiques sur la capacité du projet à supporter un nombre défini d'utilisateurs ou de données. Par exemple :

- Le système doit répondre en moins de 2 secondes.
- Le système doit supporter 1000 utilisateurs simultanés.

4. Planification et livrable

4.1. Phases du projet

Description et échéancier des différentes phases du projet. Il est important d'y ajouter les **dates principales du projet**. L'ajout d'un calendrier ou d'une forme de calendrier peut être intéressante ici.

4.2. Livrables attendus

Description de ce qui est attendu comme livrable. Ce que vous allez remettre au client. Le livrable peut contenir du code, des données, des bases de données, de la documentation, etc. Un livrable peut aussi contenir des services opérationnels, comme un serveur web fonctionnel déjà déployé avec les méthodes d'authentification pour y accéder.

5. Modalité de validation

Les procédures de validation à mettre en place pour s'assurer que le produit ou le service livré répond aux attentes, besoins et exigences du client. Exposez ici les procédures de validation qui seront mises en œuvre afin de garantir que le produit ou le service livré réponde de manière adéquate aux attentes, besoins et exigences spécifiés plus haut. Par exemple :

- Test d'acceptation utilisateur
- Vérification des fonctionnalités spécifiques
- Tests de performance

6. Conclusion

Mot de la fin pour conclure le cahier de charge. C'est l'endroit pour ajouter des informations qui ne vont pas dans les autres sections ou pour spécifier des informations supplémentaires.