



1 Analyser le contexte du projet et identifier les utilisateurs cibles.

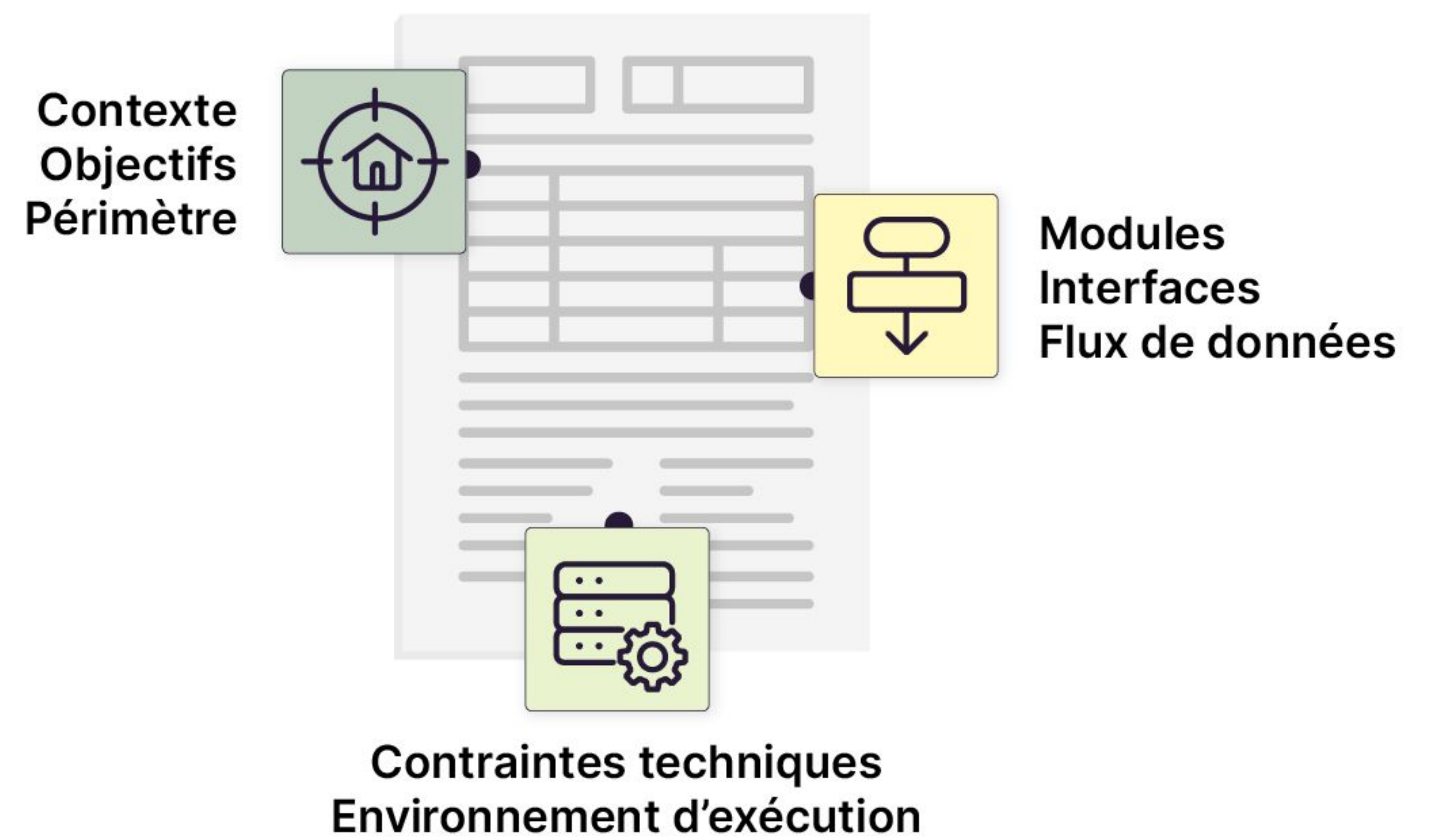
2 Recueillir les besoins métier via interviews et User Stories.

3 Formaliser les exigences fonctionnelles et non fonctionnelles.

4 Rédiger des spécifications techniques claires et détaillées.

5 Maintenir et adapter les spécifications tout au long du projet.

Spécifications techniques



Bonnes pratiques 👍

- ✅ Clarifier les objectifs, contraintes et profils utilisateurs.
- ✅ Utiliser des User Stories avec critères d'acceptation mesurables.
- ✅ Employer un langage clair et non ambigu dans les exigences.
- ✅ Hiérarchiser les besoins avec la méthode MoSCoW.
- ✅ Documenter chaque exigence avec des cas d'usage précis.
- ✅ Mettre à jour les spécifications après chaque changement.
- ✅ Vérifier la traçabilité entre besoins, specs et tests.
- ✅ Centraliser la documentation et l'adapter au cadre Agile.

Erreurs classiques 🙅

- ❌ Définir des exigences vagues ou non mesurables.
- ❌ Confondre exigences fonctionnelles et non fonctionnelles.
- ❌ Utiliser un vocabulaire subjectif comme « intuitif » ou « rapide ».
- ❌ Oublier d'impliquer les développeurs lors des validations.
- ❌ Négliger la documentation des changements (changelog).
- ❌ Laisser les spécifications devenir obsolètes ou non partagées.
- ❌ Sur-spécifier les détails d'implémentation internes.
- ❌ Ignorer les dépendances techniques entre fonctionnalités.

Définitions 🔍

User Story

Description concise d'un besoin utilisateur, souvent rédigée comme « En tant que [rôle], je veux [objectif], afin de [bénéfice] ».

Exigence fonctionnelle

Ce que le système doit faire (ex. : créer une tâche).

Exigence non fonctionnelle

Comment le système doit le faire (ex. : temps de réponse < 2 sec).

Backlog produit

Liste priorisée de toutes les fonctionnalités et exigences à développer.

Critère d'acceptation

Condition mesurable permettant de valider qu'une fonctionnalité est conforme.