

# Documentation de Gestion de Projet - EcoRide

---

Ce document explicite la méthodologie, l'organisation et les outils mis en œuvre pour mener à bien le développement de la plateforme de covoiturage **EcoRide**.

---

## 1. Méthodologie et Organisation

Le projet a été conduit selon une approche **Agile Scrum simplifiée**, favorisant un développement itératif et incrémental. Cette méthode permet de livrer rapidement des fonctionnalités opérationnelles (MVP) et d'ajuster le produit en continu.

### A. Découpage en User Stories (US)

Pour traduire le cahier des charges en tâches techniques réalisables, nous avons utilisé des **User Stories**. Chaque fonctionnalité répond à un besoin utilisateur précis.

*Exemples de User Stories traitées :*

- **En tant que visiteur**, je veux rechercher un covoiturage (US 1).
- **En tant que conducteur**, je veux publier un trajet avec mon véhicule (US 4).
- **En tant qu'employé**, je veux valider les avis laissés par les passagers (US 10).
- **En tant qu'administrateur**, je veux visualiser les statistiques de la plateforme (US 6).

### B. Planification et Phasage (Roadmap)

Pour garantir la couverture complète du cahier des charges, nous avons structuré le développement en **phases successives** plutôt qu'en priorités optionnelles :

#### 1. Phase 1 : Socle Technique & MVP (Minimum Viable Product)

- **Objectif** : Assurer les fonctions vitales de la plateforme.
- Fonctionnalités : Authentification, Recherche, Publication, Réservation via crédits, Gestion des rôles.

#### 2. Phase 2 : Enrichissement Fonctionnel

- **Objectif** : Implémenter la logique métier et les interactions utilisateurs.
- Fonctionnalités : Système d'avis et notation, Historique des trajets, Espace Employé (Validation).

#### 3. Phase 3 : Administration & Finalisation

- **Objectif** : Intégrer les fonctionnalités de pilotage et les bonus ergonomiques.
- Fonctionnalités : Gestion des litiges, Dashboard Admin (KPI & Graphiques des revenus), Filtres "Véhicule Écologique".

#### 4. Limites du périmètre (Hors Scope)

- Paiement réel par carte bancaire (simulé par crédits).
- Application mobile native (Responsive Web App privilégiée).

### C. Suivi de l'Avancement

Un tableau de suivi (type **Kanban**) a été utilisé pour piloter le projet. Il a permis de :

- Visualiser l'état d'avancement en temps réel (To Do / In Progress / Done).
- Identifier les blocages techniques.
- Valider chaque étape avant de passer à la suivante.

---

## 2. Outils de Développement et Collaboration

L'environnement technique a été choisi pour assurer robustesse, portabilité et facilité de maintenance.

### A. Gestion de Version (Git & GitHub)

- **Git** a été utilisé pour historiser toutes les modifications du code source.
- Adoption d'une stratégie de branches simplifiée (**Git Flow**) :
  - **main** : Code stable, prêt pour la production.
  - **develop** : Branche d'intégration des nouvelles fonctionnalités.
  - Commit atomiques avec messages clairs (ex: *"fix: correction bug affichage admin"*).

### B. Environnement de Développement (IDE)

- **Visual Studio Code** : Éditeur principal avec extensions PHP/HTML/CSS.
- **Serveur Local** : PHP Built-in Server (**php -S**) pour des tests rapides sans configuration lourde (Apache/Nginx).
- **Base de Données** : MySQL/MariaDB gérée via scripts SQL (**schema.sql**) pour garantir la reproductibilité.

### C. Déploiement Continu (Approche DevOps)

- Déploiement sur **Railway** (PaaS) pour rendre l'application accessible en ligne.
- Configuration via **Procfile** et **nixpacks.toml** pour automatiser la mise en production.
- Documentation de déploiement (**DOCUMENTATION\_DEPLOIEMENT.md**) pour faciliter la reprise du projet.

---

## 3. Gestion de la Qualité et des Risques

### A. Stratégie de Tests

- **Tests Unitaires (Manuels)** : Chaque fonctionnalité (ex: calcul du prix, filtres de recherche) a été testée individuellement lors de son développement.
- **Scénarios Utilisateurs (E2E)** : Des parcours complets (ex: "Inscription -> Recherche -> Réservation -> Avis") ont été validés pour s'assurer de la cohérence globale. Voir **MANUEL\_UTILISATION.md**.

### B. Gestion des Risques

Risque Identifié	Impact	Solution Mise en Place
------------------	--------	------------------------

---

Risque Identifié	Impact	Solution Mise en Place
Bug bloquant en production	Critique	Système de rollback Git (revenir au commit précédent).
Perte de données	Élevé	Scripts de sauvegarde ( <code>schema.sql</code> , <code>seed.php</code> ) pour restaurer la base rapidement.
Délais dépassés	Moyen	Priorisation des phases critiques (MVP) pour garantir une livraison fonctionnelle
Incompatibilité Mobile	Moyen	Design "Mobile First" avec CSS Responsive (Flexbox/Grid).