

DOSSIER GESTION DE PROJET – ECORIDE

Projet réalisé dans le cadre de l'ECF – Développeur
Web Full Stack

Auteur : **Othmane Lecoeuvre**
Année : 2025/2026

1. Introduction

2. Méthodologie de gestion de projet

2.1 Approche utilisée

2.2 Découpage en User Stories

2.3 Utilisation de Trello (Kanban)

2.4 Gestion du versionnement Git

2.5 Suivi et planification

2.6 Design & Maquettage (Figma)

2.7 Déploiement

3. Organisation des livrables

4. Conclusion

1. Introduction

Le présent document décrit la **gestion de projet** mise en place pour le développement de l'application **EcoRide**, dans le cadre de l'Évaluation en Cours de Formation (ECF) du Graduate *Développeur Full Stack Web & Web Mobile* (Studi).

Il expose de manière structurée les méthodes, outils et processus utilisés pour mener à bien ce projet complexe, allant de la planification à la livraison finale.

2. Méthodologie de gestion de projet

2.1 Approche utilisée

La gestion de projet s'est appuyée sur une approche **Agile**, inspirée des principes Scrum, adaptée au contexte d'un projet individuel.

Objectifs :

- Découpage fonctionnel clair
- Adaptation rapide aux changements
- Visualisation continue de l'avancement
- Livraison par étapes
- Tests réguliers

Bien que Scrum soit pensé pour une équipe, ses principes ont été adaptés pour un développement solo.

J'ai également choisi de développer mon site dans un premier temps en local, sans base de données. La base de données a été rattachée en toute fin de projet. Avec le recul, je ferais autrement et je déployerai la base de données en même temps que la création des divers User Stories.

2.2 Découpage en User Stories

Le projet a été divisé en **13 User Stories** indépendantes :

1. US 1 : Page d'accueil
2. US 2 : Menu de l'application
3. US 3 : vue des covoiturages
4. US 4 : filtres des covoiturages
5. US 5 : vue détaillée d'un covoiturage

6. US 6 : Participer à un covoiturage
7. US 7 : Création de compte
8. US 8 : Espace Utilisateur
9. US 9 : Saisir un voyage
10. US 10 : historique des covoiturages
11. US 11 : Démarrer et arrêter un covoiturage
12. US 12 : Espace employé
13. US 13 : Espace administrateur

Chaque User Story :

- possède ses critères d'acceptation
- est développée sur une branche dédiée
- est testée avant d'être mergée

2.3 Utilisation de Trello (Kanban)

Un tableau Trello organisé en 5 colonnes :

- Backlog / Fonctionnalités prévues
- À développer / Sprint en cours
- En cours de dev
- Prêtes sur branche develop
- Fusionnées dans main

Chaque tâche possède :

- un titre clair
- une description
- des critères d'acceptation
- une checklist tech
- parfois une maquette / capture écran

Trello permettait :

- de visualiser l'état du projet
- d'éviter le multitâche
- de garantir une progression linéaire et fluide
- d'assurer la conformité avec l'ECF

2.4 Gestion du versionnement Git

Workflow professionnel adopté :

main ← develop ← feature/usX-nom-de-la-us

Règles strictes appliquées :

- Aucun développement directement sur **main**
- Une branche par User Story
- Messages de commits clairs
- Tests avant merge
- Suppression des branches après utilisation

Résultat :

- Un dépôt propre, lisible, structuré
- Une histoire claire et logique
- Une gestion des conflits inexistante

2.5 Suivi et planification

Le projet a suivi un rythme d'itération régulier :

- 1 User Story / phase
- tests fonctionnels
- validations croisées avec le cahier des charges

Cela a permis :

- une montée progressive en complexité
- un code stable en continu
- une adaptation rapide aux problèmes identifiés

2.6 Design & Maquettage (Figma)

Avant le développement :

- Création d'une **charte graphique complète**
- Élaboration de **maquettes desktop & mobile** pour toutes les pages clés
- Déclinaison UI :
 - boutons
 - champs
 - cartes
 - couleurs, typographies
 - états hover / focus / erreur

Ces maquettes ont servi de référence jusqu'à la fin du projet. Certaines fois entre la maquette et le rendu sur le site, je me suis aperçu qu'il fallait retravailler certaines choses, donc la maquette a servi de support de travail et mon design à évoluer au fil de l'eau.

2.7 Déploiement

Pour l'hébergement, le projet EcoRide utilise **Fly.io**, une solution simple et performante permettant le déploiement d'applications Web via Docker.

Ce choix a été motivé par la facilité d'intégration avec Git, le support natif de PHP et la possibilité d'automatiser le déploiement depuis la branche main.

3. Organisation des livrables

Le projet a été livré avec :

- Code source organisé
- Base de données MySQL + SQL d'import
- Documentation technique
- Manuel utilisateur
- Charte graphique PDF
- Dossier complet
- README.md

4. Conclusion

La gestion du projet EcoRide repose sur :

- une méthode claire
- une organisation stricte
- une planification cohérente
- des outils professionnels (Trello, Git, Figma)
- un versionnement rigoureux
- un découpage fonctionnel efficace