

1. Contexte et Objectifs du Projet

1.1. Contexte général

Ce projet constitue l'évaluation principale du module « Processus de Développement Logiciel » (50% de la note finale). Il vise à mettre en application l'ensemble des compétences acquises : choix et justification d'un processus de développement, application des pratiques agiles, production d'artefacts, qualité du code et intégration continue.

Le domaine métier retenu est celui d'une agence de location de voitures disposant de plusieurs points de service. L'agence souhaite se doter d'un système ERP intégré couvrant l'ensemble de ses opérations : gestion du parc automobile, suivi des locations, gestion financière, et relation client.

1.2. Objectifs pédagogiques

- Appliquer un processus de développement complet (de l'incubation à la livraison) sur un projet réel
- Justifier le choix du cycle de vie et des pratiques méthodologiques adoptées
- Produire les artefacts attendus : Product Backlog, User Stories, Sprint Backlog, Burndown Chart, documentation technique
- Développer un logiciel fonctionnel respectant les principes de Clean Code et d'architecture logicielle
- Mettre en place un pipeline CI/CD avec gestion de versions (Git workflow)
- Travailler en équipe selon les cérémonies SCRUM (ou le processus choisi et justifié)

1.3. Organisation des équipes

- Chaque équipe doit développer les deux plateformes décrites ci-dessous
- Attribution des rôles obligatoire : Scrum Master, Product Owner, Développeurs (avec rotation possible)
- Un dépôt Git unique par équipe (GitHub, GitLab ou équivalent) avec historique des contributions

2. Description des Deux Plateformes

Le système ERP est décomposé en deux plateformes complémentaires. Chaque équipe doit développer les deux plateformes, qui partagent la même base de données et communiquent entre elles.

PLATEFORME A — Back-Office (Administration & Gestion Interne)

Destinée aux employés et gestionnaires de l'agence. Cette plateforme constitue le cœur opérationnel du système ERP.

2.1. Plateforme A — Back-Office (Administration)

Module A1 : Gestion du Parc Automobile

- Fiche technique complète de chaque véhicule : immatriculation, marque, modèle, date de mise en circulation, kilométrage, description de l'entretien, date du dernier entretien, prochain vidange
- Suivi des dates d'expiration : assurance, contrôle technique, amortissement
- Gestion des statuts : disponible, en location, en maintenance, transféré, hors service
- Gestion des disponibilités en temps réel avec calendrier visuel

Module A2 : Gestion des Locations

- Enregistrement complet d'une location : client, véhicule, dates de début/fin, tarif appliqué, caution
- Historique complet des locations par véhicule et par client
- Calcul automatique des coûts de location en fonction de la durée, la catégorie du véhicule et le type de client
- Gestion des prolongations et retours anticipés
- Gestion des contrats de location (génération PDF)

Module A3 : Transfert Inter-Agences (Offshoring)

- Suivi des mouvements de véhicules entre les différentes agences
- Enregistrement des informations de transfert : agence d'origine, agence de destination, dates, coûts associés
- Traçabilité complète de l'historique des transferts

Module A4 : Gestion Financière

- Gestion des paiements : enregistrement des factures, modalités de paiement (espèces, virement, chèque), date de paiement, trésorerie (banque ou caisse)
- Gestion des charges des véhicules : maintenance, réparation, prélèvements (crédits), assurance
- Gestion des charges fixes et variables de la société : personnel, loyer, téléphone, bureautique, CNSS, TVA, impôts, dons, autres

- Suivi mensuel des entrées (chiffre d'affaires par véhicule)
- État de rapprochement mensuel : vision globale du CA et du résultat (bénéfice ou perte) calculé sur la base des entrées moins les charges par véhicule et les charges générales
- Synthèse annuelle : état de rapprochement mensuel sur les 12 mois de l'année pour chaque véhicule et pour la société

Module A5 : Notifications et Alertes

- Notifications automatiques pour les dates d'expiration : assurance, contrôle technique
- Alertes sur les réservations à venir et les retours prévus
- Tableau de bord avec indicateurs clés (KPI)

Module A6 : Rapports et Statistiques

- Génération de rapports selon des critères multiples : marque, durée, type de client, période
- Statistiques de performance : taux d'occupation, CA par véhicule, rentabilité
- Export des rapports (PDF ou Excel)

PLATEFORME B — Portail Client (Front-Office)

Destinée aux clients de l'agence, cette plateforme offre une expérience de réservation en ligne et un espace personnel pour chaque client.

2.2. Plateforme B — Portail Client (Front-Office)

Module B1 : Catalogue et Réservation en Ligne

- Consultation du catalogue des véhicules disponibles avec filtres : catégorie, marque, prix, dates, agence
- Fiche détaillée de chaque véhicule avec photos, caractéristiques et tarification
- Système de réservation en ligne avec sélection des dates, de l'agence de retrait et de restitution
- Calcul automatique du coût estimé avant confirmation
- Confirmation de réservation avec récapitulatif envoyé par email

Module B2 : Espace Client Personnel

- Inscription et authentification sécurisée
- Tableau de bord personnel : réservations en cours, historique des locations passées
- Consultation des factures et de l'état des paiements
- Modification ou annulation de réservation (selon les conditions)
- Gestion du profil : informations personnelles, permis de conduire, préférences

Module B3 : Notifications Client

- Rappel de réservation à venir (email ou notification in-app)
- Notification de confirmation de retour du véhicule
- Alerte en cas de dépassement de la durée de location

Module B4 : Avis et Évaluations

- Possibilité pour le client de laisser un avis après une location
- Système de notation par étoiles sur le véhicule et le service
- Affichage des avis sur les fiches véhicules (visible par les autres clients)

3. Exigences Techniques

3.1. Architecture et Technologies

Composant	Exigences
Architecture	Architecture en couches (Présentation, Métier, Accès aux données) avec séparation claire des responsabilités
Back-end	Libre choix technologique (Spring Boot, Django, Node.js, Laravel, .NET, ou autre) — à justifier
Front-end	Libre choix (React, Angular, Vue.js, ou autre) — interface responsive et ergonomique
Base de données	Libre choix (MySQL, PostgreSQL, MongoDB, ou autre) — base unique partagée entre les deux plateformes
Gestion de versions	Git obligatoire avec workflow structuré (branching, pull requests, revue de code)
CI/CD	Pipeline d'intégration continue recommandé (GitHub Actions, GitLab CI, ou équivalent)
Qualité du code	Respect des principes Clean Code, nommage cohérent, pas de code mort, commentaires pertinents

3.2. Exigences non fonctionnelles

- **Sécurité** : authentification et autorisation (gestion des rôles : administrateur, employé, client)
- **Performance** : temps de réponse acceptable pour les opérations courantes
- **Ergonomie** : interface intuitive et responsive (desktop et mobile)
- **Maintenabilité** : code modulaire, respect du principe de séparation des préoccupations
- **Fiabilité** : validation des données côté client et côté serveur

4. Livrables Attendus

Chaque équipe devra remettre l'ensemble des livrables suivants lors de la semaine des examens :

N°	Livrable	Description
1	Code source	Dépôt Git avec historique complet des commits, branches et pull requests
2	Product Backlog	Liste priorisée des User Stories avec critères d'acceptation et estimation en Story Points
3	Sprint Backlog & Burndown	Planification de chaque sprint avec suivi de l'avancement (Burndown Chart)
4	Documentation technique	Diagrammes UML (cas d'utilisation, architecture du système, modèle de données)

5	Justification du processus	Document argumentant le choix du cycle de vie et des pratiques adoptées (2–3 pages)
6	Rapport de sprint	Compte-rendu de chaque sprint : objectifs, réalisations, obstacles, rétrospective
7	Démonstration	Présentation orale avec démo en direct des deux plateformes fonctionnelles

5. Grille d'Évaluation

Le projet représente 50% de la note finale du module. La note sera attribuée selon la grille suivante :

Critère	Pondération	Éléments évalués
Processus et méthodologie	20%	Justification du processus, artefacts produits, cérémonies
Qualité fonctionnelle	25%	Couverture des modules, fonctionnalités opérationnelles, cohérence métier
Qualité technique et architecture	20%	Architecture, Clean Code, séparation des couches, patterns utilisés
Pratiques DevOps et Git	10%	Historique Git, workflow, CI/CD, tests
Travail d'équipe et collaboration	10%	Répartition équitable, contributions Git, rétrospectives
Présentation et démonstration	15%	Clarté de la présentation, démo fonctionnelle, réponses aux questions
TOTAL	100%	

6. Planning et Jalons

Le projet s'étend de la semaine 3 (après la présentation du Processus Unifié et de l'architecture) jusqu'à la semaine des examens. Les jalons suivants sont à respecter :

Semaine	Jalon	Livrable attendu
Semaine 2	Formation des équipes	Fiche projet validée (composition de l'équipe, sujet confirmé)
Semaine 3	Lancement du projet	Justification du processus choisi, Product Backlog initial, architecture préliminaire
Semaine 4	Simulation SCRUM	Sprint 1 planifié et lancé
Semaine 5	Suivi Intermédiaire	Démo Sprint 1, planification Sprint 2, Burndown Chart

Semaine 7	Point d'avancement final	Code quasi final, documentation à jour, préparation de la soutenance
Examens	Soutenance finale	Remise de tous les livrables + présentation orale + démonstration en direct

7. Règles et Consignes

7.1. Règles impératives

- Tout plagiat ou copie de code sans attribution entraînera la note de 0/20 pour l'ensemble de l'équipe concernée.
- L'utilisation d'outils d'IA générative (ChatGPT, Copilot, etc.) est autorisée comme outil d'aide, mais chaque étudiant doit être capable d'expliquer intégralement le code qu'il soumet. Un oral individuel pourra être organisé en cas de doute.
- Chaque membre de l'équipe doit contribuer de manière significative et vérifiable (commités Git individuels obligatoires).
- Les retards de livraison seront pénalisés : -2 points par jour de retard.

7.2. Liberté méthodologique

Les équipes sont libres de choisir, adapter ou combiner des processus de développement (SCRUM, XP, Kanban, Processus Unifié, ou hybride) à condition de justifier leur choix de manière argumentée dans le livrable n°5. Cette justification doit s'appuyer sur le contexte du projet, la taille de l'équipe, les contraintes de délai, et les concepts étudiés en cours.

7.3. Présentation finale

- Durée : 20 minutes de présentation + 10 minutes de questions
- Contenu attendu : présentation du processus suivi, démo en direct des deux plateformes, leçons apprises, rétrospective générale
- Tous les membres de l'équipe doivent participer activement à la présentation
- Des questions individuelles pourront être posées pour vérifier la compréhension de chaque étudiant

8. Priorisation des Fonctionnalités

Afin de guider les équipes dans la planification de leurs sprints, les fonctionnalités sont classées selon la méthode MoSCoW :

Priorité	Fonctionnalités
Must Have	Gestion du parc automobile (A1), Gestion des locations (A2), Gestion financière de base (A4 partie I), Catalogue et réservation (B1), Espace client (B2), Authentification

Should Have	Transfert inter-agences (A3), Notifications (A5 & B3), Rapports basiques (A6), Paiements complets (A4)
Could Have	Avis et évaluations (B4), Synthèse annuelle complète, Export PDF/Excel, Tableau de bord avancé
Won't Have	Paiement en ligne intégré, Application mobile native, Gestion multi-devises

Cette priorisation est indicative. Les équipes sont encouragées à l'adapter dans leur Product Backlog en justifiant tout changement. L'objectif est de livrer un produit minimum viable (MVP) fonctionnel couvrant au minimum les fonctionnalités « Must Have ».

Bon courage et bon développement !