

INSA LYON
DÉPARTEMENT INFORMATIQUE

PLD AGILE : Optimod'Lyon

Rapport de fin de projet

Auteur : H4302
Hazim ASRI
Nihal BOUTADGHART
Jassir HABBA
Ana MARTIN
Junior NOUKAM
Simon PERRET

Professeure :
Mme. LAFOREST

Table des matières

1	Introduction	0
2	Troisième livraison	1
2.1	Introduction de la dernière livraison	1
2.2	Développement des fonctionnalités	1
2.3	Planning des tâches	1
2.4	Conclusion	2
3	Rétrospective du projet	4
3.1	Rétrospective de groupe	4
3.1.1	Réussites du projet	4
3.1.2	Difficultés rencontrées	4
3.2	Point de vue personnel	4
3.3	Conclusions et suggestions	5

1 Introduction

Ce document constitue le livrable final du PLD Agile. Ce projet s'inscrit dans une démarche de développement orienté objet tout en appliquant la méthodologie agile Scrum.

Ce rapport est structuré en deux grandes parties : la description détaillée de la troisième et dernière livraison, et une rétrospective sur le projet dans son ensemble. Nous y aborderons les fonctionnalités implémentées, la planification des tâches, les réussites, ainsi que les difficultés rencontrées tout au long du dernier sprint.

2 Troisième livraison

2.1 Introduction de la dernière livraison

Ce document constitue le livrable de la troisième et dernière itération de notre projet. Lors de cette phase, nous nous sommes principalement concentrés sur l'amélioration et l'ajout de fonctionnalités précédemment développées et la préparation à la clôture du projet.

Lors des deux premières itérations, nous avons identifié certaines erreurs. Grâce aux conclusions tirées de ces phases, nous avons ajusté notre méthode de travail pour éviter leur répétition.

Durant cette itération, la conception et l'architecture corrigées lors de la deuxième itération n'ont pas été modifiées. Cependant, nous avons amélioré des modules critiques, tels que :

- La sauvegarde des tournées dans des fichiers
- Le calcul des tournées optimales
- La rédaction d'une documentation utilisateur et technique complète.

2.2 Développement des fonctionnalités

Voici les fonctionnalités développées au cours de cette phase :

- **Modification d'une demande de livraison** : Possibilité de modifier les points de collecte et de livraisons
- **Configuration de plusieurs coursiers**
- **Sauvegarde dynamique d'un tour**
- **Amélioration de l'UI** : flèche, numéro d'identification des livraison et durée d'un tour
- **Intersections bidirectionnelles**
- **Documentation technique**

2.3 Planning des tâches

Suite à la deuxième itération, nous avons conclu que les rôles n'avaient pas été correctement définis avant l'itération. Voici alors les rôles des membres de l'équipe qui leur ont été attribués en fonction des tâches qu'ils ont effectués lors des deux premières itérations :

Membre	Rôle
Ana	tests & intégration
Hazim	Scrum Master & développeur fullstack
Jassir	développeur fullstack et fileManager
Junior	développeur fullstack et tsp
Nihal	tests & documentation
Simon	Chef de projet & développeur backend

TABLE 1 – Rôles des membres de l'équipe pour la troisième livraison

Voici donc ci-dessous le tableau qui détaille les tâches réalisées, leur durée estimée, la difficulté et le membre assigné (voir tableau 2). Lors de cette itération, nous avons ajouté la colonne "Coût" qui indique la difficulté estimée de chaque tâche selon une échelle de 1 à 3, où 1 correspond à une tâche simple et 3 à une tâche particulièrement complexe nécessitant une expertise technique ou une coordination importante. Cette colonne que nous avons négligé lors des 2 premières itérations nous a permis de mieux répartir les tâches en fonction des compétences de chacun.

Les tâches de cette dernière itération ont été définies en tenant compte des besoins fonctionnels restants, des améliorations techniques nécessaires, ainsi que de la documentation et des tests finaux. Une attention particulière a été portée à la correction des bugs et à l'amélioration de l'interface utilisateur. Le tableau ci-dessous détaille les tâches accomplies, leur durée estimée, ainsi que la personne ou l'équipe chargée de leur réalisation.

Nous avons également passé du temps à merge les branches de chacun pour avoir une branche main propre et prête pour la release.

Description de la tâche	Durée	Coût	Assigné
Résout le bug lorsqu'on charge un mauvais type de fichier	10 minutes	1	Junior
Calcul de tour multiples pour n coursiers	1 heure	1	Junior
Chargement d'une carte avec les intersections bi-directionnelles	30 minutes	1	Junior
Bannière de succès ou d'erreur	45 minutes	1	Hazim
Documentation du CI/CD	30 minutes	1	Ana
Documentation des tests de front	30 minutes	1	Nihal
Documentation du technique du code	2 heures	1	Nihal & Ana
Release first version	15 minutes	1	Hazim
Documentation de la dernière itération	2 heure	1	Simon
Numérotation des marqueurs sur la carte	10 minutes	1	Junior
Implémentation au front de la modification d'une demande de livraison	3 heures	2	Simon
Restore d'un tour depuis un fichier	4 heures	2	Jassir
Affiche la durée d'un tour	15 minutes	2	Hazim
Drag and drop les marqueurs de carte	2 heures	3	Hazim
Ajout de flèches pour indiquer le sens de déplacement	∅	3	Hazim
Merge final de toutes les branches	6	3	Hazim & Jassir & Simon & Junior

TABLE 2 – Planning des tâches de la troisième livraison

Cette itération a montré que la clarification des rôles et la planification détaillée des tâches ont permis de gagner en efficacité. La gestion agile des priorités a permis de résoudre rapidement les problèmes critiques et d'intégrer de nouvelles fonctionnalités, tout en respectant les délais fixés. Cependant, des points d'amélioration subsistent, notamment pour mieux anticiper les conflits lors des merges ou pour optimiser la gestion des tests.

Concernant les flèches pour indiquer le sens de déplacement, nous n'avons pas pu les implémenter comme prévu, car la bibliothèque nécessaire n'était pas disponible dans la version de NetBeans que nous utilisons. Nous avons remplacé cette fonctionnalité par l'affichage d'index sur les marqueurs, permettant une visualisation claire et fonctionnelle de l'ordre des livraisons.

2.4 Conclusion

Cette dernière livraison nous a permis de stabiliser les fonctionnalités clés de l'application et d'assurer sa conformité avec les besoins spécifiés. La rédaction d'une documentation technique complète constitue également une étape essentielle pour faciliter la prise en main future de l'application.

Les conclusions tirées de ce sprint, résumées dans les deux tableaux ci-dessous, mettent en lumière à la fois les points positifs (voir tableau 3) qui ont favorisé notre réussite et les points négatifs, accompagnés d'axes d'amélioration pour des projets futurs (voir tableau 4).

Point positifs
Nous avons appris de nos précédentes erreurs pour ne pas les reproduire
Nous avons pris le temps de merge avant la dernière séance pour ne pas avoir de conflits au dernier moment
Nous avons été très réactif lors du développement si quelqu'un avait un problème
Meilleure répartition des tâches que précédemment en fonction des compétences et envie de chacun
Nous avons estimé le coût de chaque tâche et leur priorité
Bon respect de l'architecture choisie
Nous avons rendu les tâches plus atomiques
Formation de certains membres de l'équipe au front

TABLE 3 – Points positifs de la troisième livraison

Point négatifs	Axe d'amélioration
Désaccords sur la conception du frontend	Nous aurions dû mettre au clair dans un document disponible pour tout le monde
Manque encore de communication pour les changements majeurs	Pourquoi pas mettre un serveur discord en place pour les changements majeurs
Commit pas assez réguliers	rendre les commits plus atomiques et pousser sa branche régulièrement
Fichier ajouté au commit qui ne devrait pas	Éviter les "git add ." et vérifier les fichiers ajoutés
Convention de nommage de fonctions, variables, branches... non explicite	Préciser ce type d'information dans CONTRIBUTING.md
Prise de risque de l'attribution de tâche critique au dernier moment	Nous pouvons améliorer encore la distribution des tâches

TABLE 4 – Points négatifs de la troisième livraison

3 Rétrospective du projet

3.1 Rétrospective de groupe

Voici une rétrospective de groupe sur le projet dans son ensemble :

3.1.1 Réussites du projet

Voici les points forts identifiés au cours du projet :

- Le repo GitHub bien organisé et documenté, avec des branches distinctes pour chaque fonctionnalité
- Le choix de la technologie Java pour le backend et React pour le frontend, qui ont permis une bonne séparation des préoccupations
- Le choix de l'IDE Netbeans avec lequel nous étions familier et pour lequel nous n'avons donc pas eu à nous former
- Une bonne ambiance de travail et une communication ouverte au sein de l'équipe
- Le fait que le projet nous ait tous intéressés et que nous avons tous appris de nouvelles choses

3.1.2 Difficultés rencontrées

Malgré les réussites, nous avons rencontré certaines difficultés :

- Convention de nommage : Nous avons constaté un manque d'uniformité dans les conventions de nommage des fichiers, classes, et variables, ce qui a parfois compliqué la lecture du code. Nous aurions pu mettre en place un guide de style codifié dès le début du projet.
- Version trop ancienne de Netbeans : L'utilisation d'une version obsolète de NetBeans a entraîné des problèmes de compatibilité avec certaines bibliothèques, ce qui a ralenti le développement. Nous pourrions notamment utiliser une version plus récente ou envisager de migrer vers un IDE plus moderne, comme IntelliJ IDEA.
- Pas assez de temps sur la conception : La phase de conception a été réduite, ce qui a entraîné des ajustements fréquents et des difficultés d'intégration lors des itérations. Nous pourrions donc allouer davantage de temps sur la conception initiale pour anticiper les problématiques potentielles, par exemple en utilisant des outils comme des diagrammes UML pour valider les choix avant le développement.
- Méthodologie Scrum : L'application des principes Scrum a parfois été approximative, notamment en ce qui concerne la gestion des sprints et le suivi des tâches. Nous avons par exemple lors du dernier sprint ajouté le coût des tâches pour mieux les répartir en fonction des compétences de chacun.
- Communication : Les échanges au sein de l'équipe n'ont pas toujours été fluides, notamment lors de modifications importantes ou de décisions impactant plusieurs membres. Il serait préférable d'organiser des points réguliers pour synchroniser l'équipe et utiliser une plateforme centralisée de communication, comme Discord, pour éviter les malentendus.
- IHM : L'interface utilisateur n'avait pas été suffisamment définie en amont, ce qui a conduit à des ajustements imprévus et des incohérences dans son développement. Nous aurions dû préparer des maquettes détaillées dès la phase de conception pour avoir une vision claire des fonctionnalités et de l'organisation visuelle de l'IHM, en impliquant toute l'équipe dans la validation.

Même si les difficultés semblent nombreuses, l'ensemble de l'équipe considère le projet comme une réussite. En effet, il semble plus facile à la fin du projet de voir les points négatifs que les points positifs.

3.2 Point de vue personnel

Cette partie a pour objectif de permettre à tout le monde de faire un retour d'expérience du PLD :

- **Ana** : Even if I am an exchange student and some knowledges I did not have it , I reckon the development of the projet went pretty well. At the same time, I consider that the work environment is quite comfortable and it has been quite easy and pleasant to work with them.
- **Hazim** : En assumant le rôle de Scrum Master, j'ai appris à attribuer les issues, mais surtout à découper le travail de manière à garantir une répartition équitable tout en veillant à ce que le projet ne soit pas retardé. Cela n'a pas toujours été évident au début, car certaines tâches peuvent bloquer le démarrage d'autres tâches. Cependant, grâce à une communication régulière avec l'équipe, nous avons réussi à livrer le projet dans les délais impartis.
- **Jassir** :
- **Junior** :
- **Nihal** :
- **Simon** : En tant que chef de projet, cette expérience m'a permis de développer des compétences en gestion de projet agile, en communication et en leadership. J'ai appris à coordonner une équipe, à adapter les

plannings en fonction des imprévus, et à maintenir une documentation claire et précise tout au long du projet.

3.3 Conclusions et suggestions

Ce projet a été une expérience riche et formatrice pour l'ensemble de l'équipe. Bien que nous ayons rencontré des difficultés techniques, comme les limitations liées à certains outils ou le manque de temps initialement alloué à la conception, nous avons su nous adapter en trouvant des solutions pragmatiques et en capitalisant sur les forces de chacun.

La méthodologie agile, bien qu'imparfaitement appliquée, nous a permis d'itérer rapidement et de corriger nos erreurs au fil des sprints. L'intégration des feedbacks des premières itérations a clairement amélioré la qualité de notre travail lors de cette dernière phase, avec une meilleure répartition des tâches et une planification plus aboutie.

Nous terminons ce projet avec un livrable fonctionnel, documenté, et conforme au cahier des charges initial. Les apprentissages réalisés dans la méthodologie agile, le développement "fullstack", et la collaboration en équipe seront des atouts indispensables pour nos futurs projets. Ce projet a également permis de montrer à tout le monde l'importance de la communication et de la documentation dans un projet de développement logiciel ainsi que l'utilisation propre d'un système de versioning.

Pour de futurs projets similaires, voici quelques recommandations :

- Mieux planifier les estimations de temps avec des méthodes comme Planning Poker pour dès le départ échanger sur les difficultés de chaque tâche
- Renforcer les compétences techniques de l'équipe par des formations préalables dès le début du projet sans attendre que le besoin se fasse sentir
- Maintenir une communication constante et documenter tout changement structurel dans le projet
- S'assurer d'une répartition équitable et pertinente des tâches en fonction des compétences des membres et de leurs envies