
DOSSIER DE PROJET

TITRE PROFESSIONNEL
CONCEPTEUR DÉVELOPPEUR
D'APPLICATIONS

Csaba Schnitchen

Octobre 2023



Table des matières

1	Liste des compétences du référentiel couvertes par le projet	2
2	Project Summary	3
3	Introduction	4
3.1	Présentation personnelle	4
3.2	Présentation de l'entreprise	4
3.2.1	La petite histoire de SuiviDeFlotte.net	6
4	Gestion de projet	7
4.1	Méthodologie Agile	9
4.2	Jira : L'outil essentiel pour la gestion de projet	11
4.3	Versioning et les environnements	13
4.4	Intégration Continue et Déploiement Continu	18
4.4.1	Le déploiement des paquets de projet Cores	19
4.4.2	Le déploiement du projet Pipeline documentaire	19
5	Cahier des charges	21
6	Spécifications fonctionnelles	22
7	Spécifications techniques	23
8	Réalisations	24
9	Présentation d'un jeu d'essai	25
10	Veille sur les vulnérabilités de sécurité	26
11	Description d'une situation de travail ayant nécessité des travaux de recherches	27
12	Conclusion	28
13	Remerciements	29
	Annexes	30
	Annexe A Jira : L'outil pour la gestion de projet	31

1 Liste des compétences du référentiel couvertes par le projet

Compétences professionnelles	Présentation orale	Dossier Projet	Dossier Professionnel
Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité			
Maquetter une application	X	X	
Développer une interface utilisateur de type desktop			X
Développer des composants d'accès aux données	X	X	
Développer la partie front-end d'une interface utilisateur web	X	X	
Développer la partie back-end d'une interface utilisateur web	X	X	
Concevoir et développer la persistance des données en intégrant les recommandations de sécurité			
Concevoir une base de données	X	X	
Mettre en place une base de données	X	X	
Développer des composants dans le langage d'une base de données	X	X	
Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité			
Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement	X	X	
Concevoir une application	X	X	
Développer des composants métier	X	X	
Construire une application organisée en couches	X	X	
Développer une application mobile			X
Préparer et exécuter les plans de tests d'une application	X	X	
Préparer et exécuter le déploiement d'une application	X	X	

2 Project Summary

Setipp is a French company, based in Tours, offering telecommunications, isolated worker safety and professional geolocation/fleet management solutions under the brand names Setipp, Beepiz and SuiviDeFlotte.net respectively.

It is within this company that I had the chance to do my work-study training program year to prepare myself for the degree of Application designer and developer.

At the company, I joined the SuiviDeFlotte.net geolocation web development team. My tutor and I, along with a colleague who joined us later, we worked on a completely new project, an API, called “Document Pipeline.” The purpose of the API was to process various types of documents coming from other services of the company, such as text files containing fuel purchase transactions or scanned invoices. It needed to extract and store the data from these documents in the database. The API had to be designed in a way that it could easily be expanded to handle processing of new file types in the future.

We developed the API using the Laravel framework and stored the data in several MariaDB databases. During the project, I also worked on transforming the web user interface for uploading files containing fuel purchase transactions within the geolocation/fleet management website, another Laravel project, to adapt its functionality to the new API. This interface utilized the Blade templating language and the Vue.js framework.

Within the geolocation development team, we followed agile principles, specifically applying the SCRUM methodology to organize our work, which allowed us to adapt flexibly to emerging needs.

3 Introduction

3.1 Présentation personnelle

Je m'appelle Csaba SCHNITCHEN et j'ai 46 ans. Je suis né en Hongrie, où j'ai obtenu un diplôme en biologie/écologie en 2002, suivi d'un doctorat en sciences de l'environnement en 2007. Jusqu'en 2020, j'ai travaillé en tant que professeur et chercheur dans deux universités. Au cours de mes recherches en écologie, j'ai découvert la programmation et j'ai utilisé les langages R et Python pour créer différents modèles écologiques et des graphiques pour des publications scientifiques. Par la suite, j'ai appliqué mes connaissances en Python pour créer un site web universitaire en utilisant le framework Django.

En 2020, j'ai déménagé définitivement en France. Ici, j'ai pris la décision de changer de profession et de me lancer dans le domaine de l'informatique. En 2022, j'ai suivi la formation Développeur web et web mobile au CEFIM, ce qui a considérablement approfondi mes connaissances en informatique et en programmation, acquises en autodidacte. Par la suite, j'ai poursuivi mes études en suivant la formation Concepteur développeur d'applications, toujours au CEFIM. Cette formation se déroulait en alternance. Pour cela, j'ai eu la chance de rejoindre Setipp, une société française, basée à Tours, proposant des solutions de télécommunications, de sécurité des travailleurs isolés et de géolocalisation professionnelle/gestion de flotte, sur un rythme de trois semaines en entreprise et une semaine CEFIM.

3.2 Présentation de l'entreprise

Setipp est une société française dont le siège est à Tours et qui possède également des bureaux à Lille, Paris, Strasbourg, Nantes, Lyon, Toulouse et Montpellier.

Setipp propose trois familles de services et de produits à ses clients sous trois marques : Setipp, Beepiz et SuiviDeFlotte.net. Setipp fournit des services de télécommunications aux entreprises. Beepiz propose des applications web et mobiles qui permettent aux travailleurs isolés de se protéger en toutes circonstances et d'accéder aux services d'urgence. SuiviDeFlotte.net propose des solutions professionnelles de géolocalisation et de gestion de parc. Leurs services en ligne incluent des fonctions telles qu'aider les conducteurs à économiser du carburant et aider les entreprises à décider quels véhicules valent la peine de passer à l'électrique.

Au sein de l'entreprise, j'ai intégré l'équipe de développement de la géolocalisation de SuiviDeFlotte.net tel qu'il apparaît dans l'organigramme de SuiviDeFlotte (Figure 3.1).

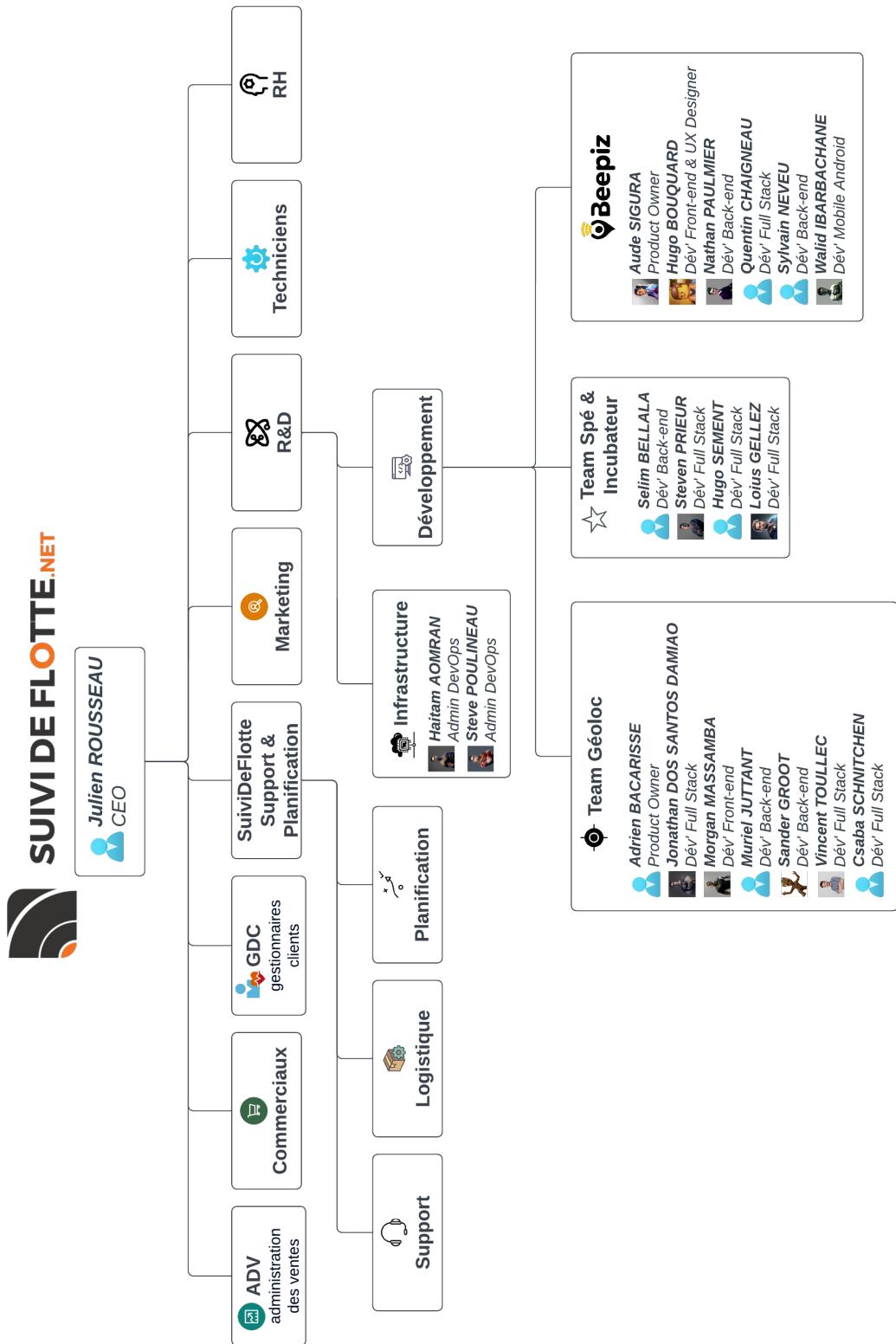


FIGURE 3.1 – Organigramme

3.2.1 La petite histoire de SuiviDeFlotte.net

SuiviDeFlotte.net, créée en 2001, est la branche de l'entreprise spécialisée dans la géolocalisation et la gestion de flottes de véhicules et d'objets connectés pour les entreprises. À l'origine de cette initiative se trouve Julien Rousseau, actuel Président Directeur Général, qui a remarqué que la gestion de la flotte automobile était le deuxième poste de dépense le plus important pour les entreprises. Pour optimiser ces parcs automobiles et améliorer la productivité des métiers nécessitant des interventions sur le terrain, il a envisagé d'appliquer les succès de la télécommunication aux véhicules.

Les bénéfices potentiels de la télématicque embarquée se sont avérés concluants, conduisant ainsi à la création officielle de SuiviDeFlotte.net, pionnière de la télématicque embarquée en France. Initialement centrée sur la géolocalisation des véhicules, l'entreprise a progressivement élargi ses services pour offrir des outils complets de gestion de flottes, incluant la géolocalisation, la gestion de parc et l'éco-conduite. Depuis ses débuts, SuiviDeFlotte.net propose ses services via une plateforme SaaS, permettant d'ajouter de nouvelles fonctionnalités aux utilisateurs sans installation ni maintenance.

Aujourd'hui, l'entreprise se focalise sur l'innovation, cherchant à répondre aux besoins de ses utilisateurs en proposant trois grandes mises à jour par an, intégrant plus de 100 nouvelles fonctionnalités chaque année. SuiviDeFlotte.net conçoit et commercialise des solutions clés en main de géolocalisation, écoconduite et gestion de flottes de véhicules (VL, VU, poids lourds), qui sont utilisées par 4000 entreprises, qu'il s'agisse de TPE, PME ou entités de grands groupes. Elle compte 50 collaborateurs, génère un chiffre d'affaires de 7 millions d'euros et consacre 25% de son effectif à la recherche et développement.

4 Gestión de proyecto

Pendant mon alternance, j'ai très vite compris que pour l'entreprise, il est important de maintenir son esprit innovant, de constamment générer davantage de valeur, et d'être à l'écoute tout en s'ajustant selon les exigences de sa clientèle. La politique d'innovation de l'entreprise repose sur les recommandations émanant à la fois de ses clients et de ses collaborateurs. Un comité dédié à l'innovation se rassemble hebdomadairement pour examiner les suggestions les plus récentes. Chaque concept est traité, trié et priorisé. Il s'agit d'un processus en plusieurs étapes qui implique également d'autres comités, comme l'illustre la Figure 4.1 et la Table 4.1. Par la suite, l'ensemble de ces propositions est transmis au département Recherche et Développement en vue de la création de nouvelles fonctionnalités. Tous les quatre mois, de nouvelles options viennent enrichir l'ensemble des services, désignées sous le terme "Editions".

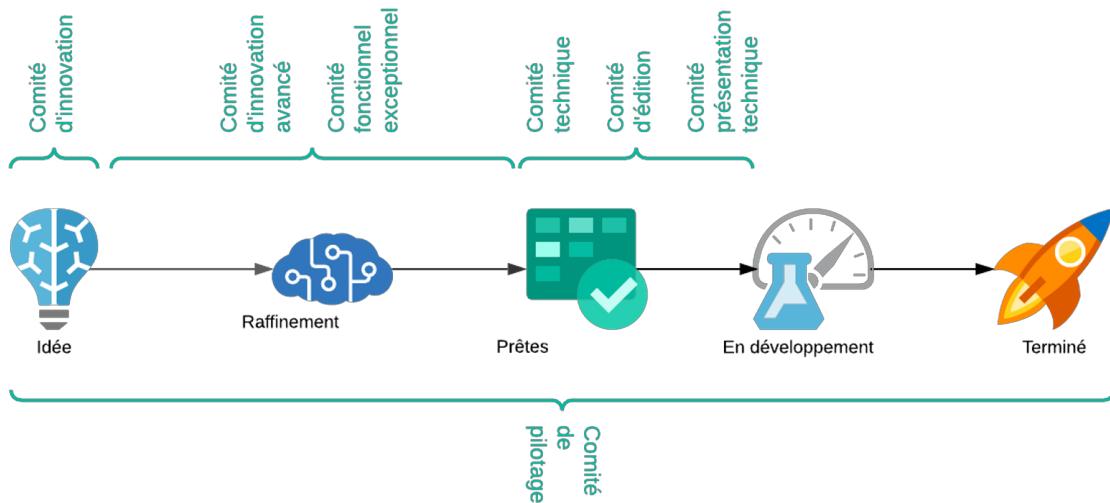


FIGURE 4.1 – La participation des différents comités au processus de traitement des idées d'innovation.

Une édition représente le fruit de quatre mois de travail de développement, cependant, son élaboration ne s'arrête pas là. Elle englobe la mise en place et la planification des activités de communication (marketing), la formation des équipes support et commerce, la rédaction de manuels et de tutoriels destinés aux clients, ainsi que la préparation des prochaines éditions à venir.

Table 4.1: Les caractéristiques des différents comités.

Nom du comité	Récur-rence	Objectif	Entrées	Sorties
Comité d'innovation	Hebdo-madaire	Première analyse des idées exprimées puis étude de l'intérêt de chaque idée.	Liste des idées	Idées clarifiées, note par niveau d'intérêt
Comité d'innovation avancé	Mensuel	Clarifier les spécifications fonctionnelles des idées exprimées au cours des comités d'innovations.	Ordre du jour des idées, déjà analysées	Réponses aux questions à l'ordre du jour, ajoutées aux backlog
Comité fonctionnel exceptionnel	Excep-tionnel	Le PO sollicite une partie prenante identifiée, qui devrait lui permettre de lever un certain nombre de questions autour d'une fonctionnalité.	Fonctionnalités, déjà analysées et comportant toujours des questions	Les réponses sont ajoutées aux fonctionnalités
Comité tech	En fonction des items en attente de chiffrage dans le backlog. Au moins hebdo-madaire.	Une présentation des items, validés fonctionnellement, pour faire ressortir un macro-chiffrage, réalisé par l'équipe d'expert.	Fonctionnalités prêtes du backlog produit	Macro-chiffrage ou questions fonctionnelles
Comité pilotage	Bimestriel	Donner une vision claire de l'avancement du travail réalisé pour les services concernés.	Indicateur clé de performance à partager	Adaptations à mettre en œuvre
Comité d'édition	Bimestriel	Établir une constitution d'édition à partir des idées prêtes. En déduire un objectif d'édition permettant de fédérer autour d'une réalisation.	Fonctionnalités prêtes, analyse du temps disponible faite à partir des paramètres vitesse / nombre de demandes courantes / dette technique ...	Liste des composants d'édition, objectif d'édition
Comité présentation technique / Kickoff Edition	En début d'édition	Une présentation des items embarqués dans l'édition, ainsi qu'un focus sur l'objectif d'édition.	L'objectif, le contenu d'édition	Le retour de l'équipe sur le contenu et l'objectif

4.1 Méthodologie Agile

L'équipe de développement de la géolocalisation de SuiviDeFlotte – comme les autres équipes de développement de l'entreprise – travaille selon la méthodologie agile SCRUM. Cette méthodologie est une approche de gestion de projet qui met l'accent sur la flexibilité, la collaboration et la livraison continue. Elle est largement utilisée dans le développement de logiciels et peut également être appliquée à d'autres domaines. SCRUM divise un projet en cycles appelés “itérations” ou “sprints” de courte durée, généralement de deux à quatre semaines, pendant lesquels une partie du travail est accomplie et livrée.

Les termes clés de la méthodologie SCRUM sont énumérés ci-dessous et illustrés dans la Figure 4.2.

Product Owner (Propriétaire du Produit) La personne responsable de définir et de prioriser les éléments du produit à développer. Le Propriétaire du Produit représente les besoins des utilisateurs et des parties prenantes.

Scrum Master (Maître de Scrum) Le facilitateur du processus SCRUM. Le Scrum Master s'assure que l'équipe suit les principes SCRUM, élimine les obstacles et favorise un environnement de travail efficace.

Équipe de Développement Le groupe de professionnels chargé de concevoir, développer, tester et livrer les éléments du produit à la fin de chaque sprint.

User Story (Histoire Utilisateur) Une Histoire Utilisateur est une courte description d'une fonctionnalité ou d'un aspect du produit, racontée du point de vue de l'utilisateur. Elle suit généralement le format “En tant que [utilisateur], je veux [action] afin de [objectif]”. Les Histoires Utilisateurs sont des éléments du Carnet de Produit et aident à définir les fonctionnalités du produit du point de vue de l'utilisateur.

Story Point (Point d'Histoire) Le Point d'Histoire est une unité relative utilisée pour estimer la complexité, l'effort et la taille des Histoires Utilisateurs ou des tâches de développement. Il n'a pas de valeur absolue, mais il sert à comparer la difficulté relative entre différentes Histoires Utilisateurs. Les équipes de développement attribuent des points d'histoire lors des estimations, ce qui les aide à planifier la quantité de travail qu'elles peuvent accomplir dans un sprint donné.

Epic (Épique) Un Épic est une unité de travail plus large que les Histoires Utilisateurs individuelles. Il représente généralement un ensemble de fonctionnalités, de tâches ou de travaux qui sont trop importants pour être traités dans un seul sprint. Les Épics sont souvent des objectifs à long terme qui sont décomposés en Histoires Utilisateurs plus petites et gérables. Ils aident à organiser et à structurer le développement du produit en regroupant des éléments liés autour d'un thème ou d'un objectif commun. Les Épics sont inclus dans le Carnet de Produit et sont priorisés en fonction de leur valeur pour l'utilisateur et du contexte global du projet.

SCRUM encourage la transparence, l'adaptabilité et la collaboration continue entre les membres de l'équipe et les parties prenantes, ce qui permet de s'adapter aux changements et de fournir rapidement de la valeur tout au long du projet.

Dans l'équipe Géoloc, le processus suit un calendrier de sprints de deux semaines (Figure A.4). Chaque cycle commence par un ensemble de réunions clés qui

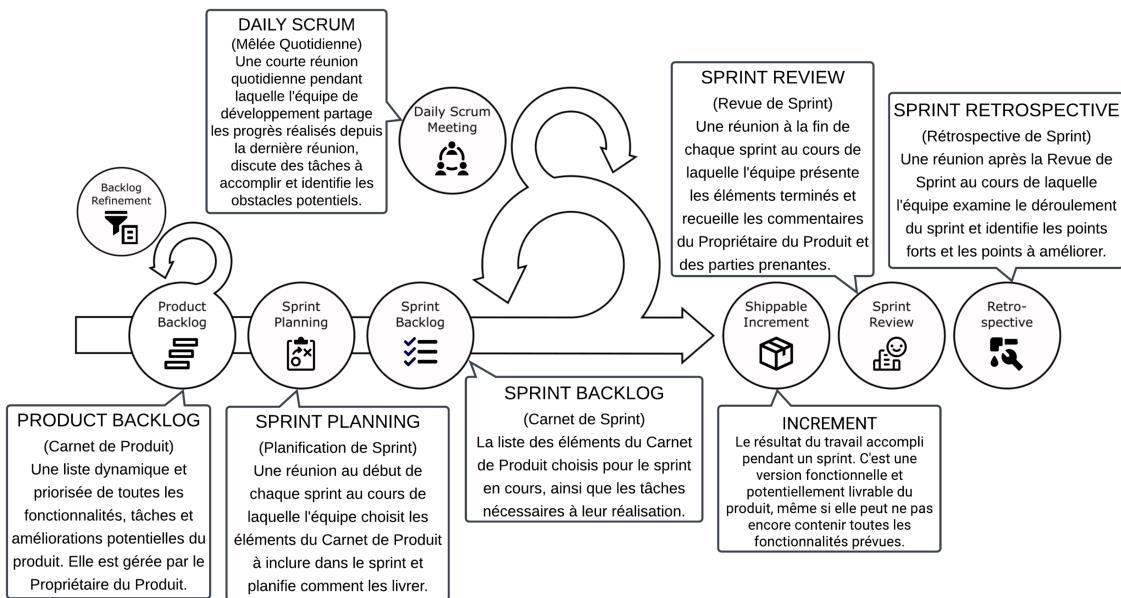


FIGURE 4.2 – Cérémonies SCRUM.

ont lieu tous les deuxièmes mardis. Cette journée englobe la Revue de Sprint, la Rétrospective de Sprint et la Planification de Sprint. Lors de la Revue de Sprint, les éléments achevés sont présentés au Propriétaire du Produit et aux parties prenantes, les retours sont recueillis et les priorités sont ajustées si nécessaire. La Rétrospective de Sprint offre l'opportunité à l'équipe de réfléchir aux succès et d'identifier les domaines à améliorer, favorisant une culture d'amélioration continue. Ensuite, la Planification de Sprint implique la sélection des Histoires Utilisateurs du Carnet de Produit à inclure dans le sprint à venir, en tenant compte de leur complexité et de leur priorité. Pour l'estimation de la complexité des tâches, des Points d'Histoire basés sur la séquence de Fibonacci (1, 2, 3, 5, 8, 13) sont utilisés. Cette approche aide à attribuer des valeurs relatives à différentes tâches et assure une évaluation cohérente de la charge de travail.

À la fin de chaque sprint, les nouveaux incréments sont également mis en production. Grâce à l'approche Agile et à la fréquence des mises en production, les modifications apportées à l'application sont rapidement portées à la connaissance de l'utilisateur.

Chaque jour ouvrable débute par une Mélée Quotidienne de 15 minutes, au cours de laquelle les progrès depuis la dernière réunion sont passés en revue, les tâches sont discutées et les obstacles potentiels sont identifiés. Ces réunions se déroulent généralement en ligne via Google Meet pour accueillir les collègues travaillant à distance et garantir la participation de tous les membres de l'équipe, peu importe leur emplacement. Par contre, pour maintenir la communication et la cohésion de l'équipe, les réunions du mardi de chaque deuxième semaine sont des réunions en personne. Cette pratique encourage les échanges directs, la collaboration étroite entre les membres de l'équipe et facilite la résolution rapide de tout problème ou obstacle pouvant survenir.

Actuellement, les rôles du Propriétaire du Produit et du Maître de Scrum ont été temporairement fusionnés en une seule personne. Cette configuration permet au Propriétaire du Produit de gérer les responsabilités du projet tout en facilitant

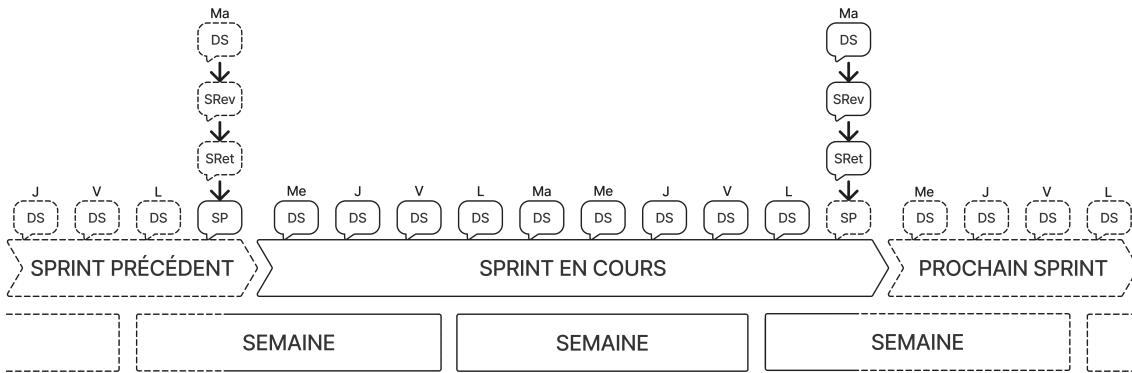


FIGURE 4.3 – Programme des sprints chez SuiviDeFlotte. Légende : L, Ma, Me, J, V – jours de la semaine ; DS – Daily Scrum ; SRev – Sprint Review ; Sret – Sprint Retrospective ; SP – Sprint Planning ; ligne pointillée – Sprint précédent ou suivant ; ligne continue – Sprint en cours.

le processus SCRUM, en maintenant une communication fluide avec l'équipe de développement.

En complément des cycles de deux semaines des Sprints, l'opération de l'entreprise comprend également un cycle plus long. En effet, comme je l'ai mentionné dans l'introduction de ce chapitre, l'entreprise publie une nouvelle version, appelée nouvelle Edition, de ses services en ligne tous les quatre mois, soit trois fois par an. Cela implique qu'il y a environ huit sprints pour que les développeurs puissent concevoir les nouvelles fonctionnalités des nouvelles éditions.

La Figure 4.4 résume les relations entre les comités et le processus des sprints. Comme on peut le voir, les tickets du Carnet de produit proviennent du Comité d'innovation, du Comité d'innovation avancée, du Comité fonctionnel exceptionnel, du Comité d'édition et du Comité d'innovation technique. Ensuite, le Comité d'édition sélectionne les tickets pour la prochaine Edition et le Comité de présentation technique les présente à l'équipe de développeurs. Une autre source de tickets dans le Backlog de Produit est l'équipe de support qui crée des tickets de bogues. Le traitement des bogues est illustré par la Figure 4.5. La section suivante explique comment ces tickets sont traités dans un outil appelé Jira.

4.2 Jira : L'outil essentiel pour la gestion quotidienne de projet

Au sein de l'entreprise, les équipes SCRUM se servent de Jira afin de consigner et de suivre tous les aspects de leur travail.

Jira est un outil essentiel pour les équipes SCRUM car il facilite la gestion complète du processus de développement agile. Il permet aux équipes de collaborer de manière efficace et de suivre chaque étape du cycle de vie du projet. Avec Jira, les équipes SCRUM peuvent créer, organiser et hiérarchiser leur Carnet de Produit en ajoutant des Histoires Utilisateurs, des tâches, des Épics et des bugs.

L'outil facilite la planification des sprints en permettant aux équipes de sélection-

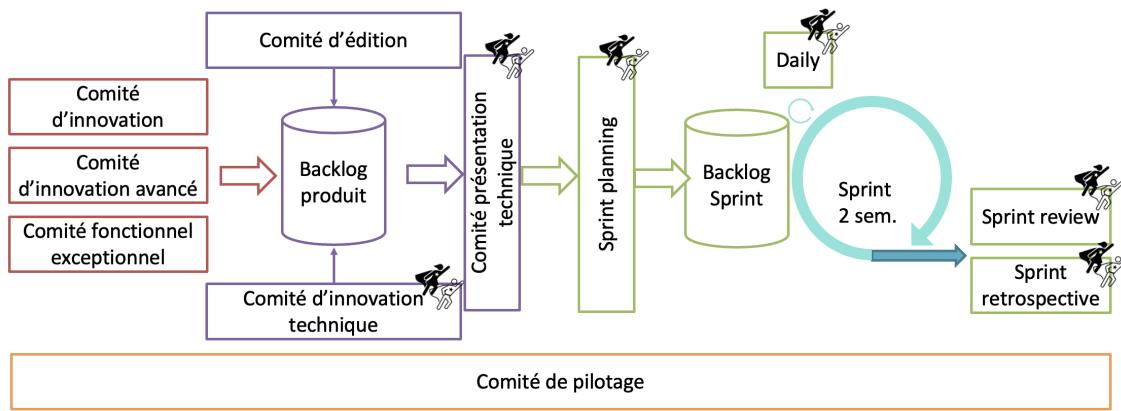


FIGURE 4.4 – Résumé des relations entre les comités et le processus des sprints.

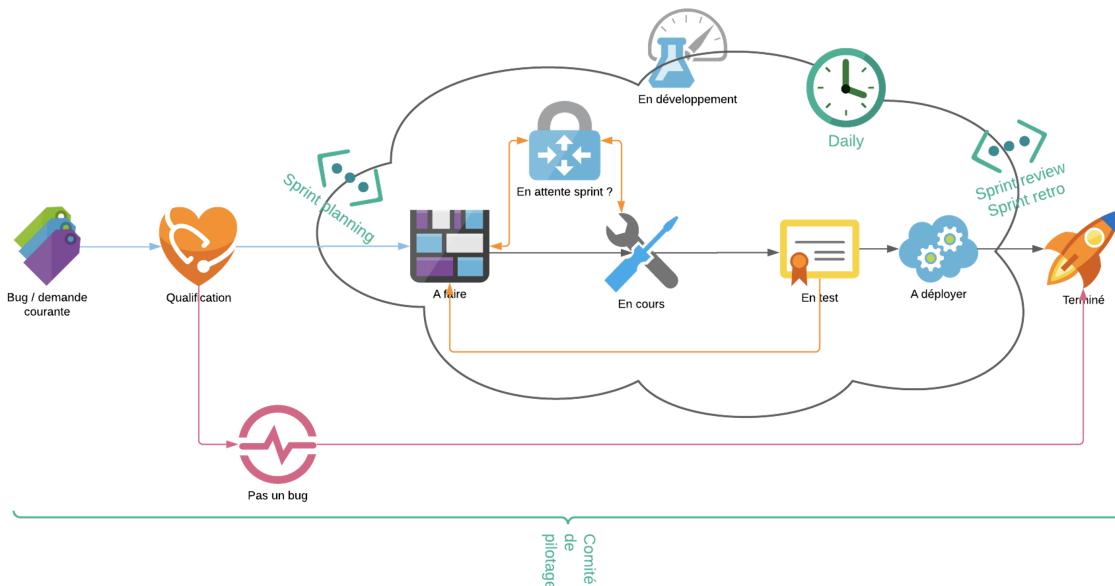


FIGURE 4.5 – Diagramme d'activité du traitement des bugs.

ner les éléments du Carnet de Produit à inclure dans chaque itération. Les équipes peuvent estimer la complexité des tâches à l'aide de Story Points et suivre leur progression au fil du temps.

Les fonctionnalités de suivi des tâches et des problèmes dans Jira aident les équipes à gérer leur travail quotidien. Chaque membre de l'équipe peut mettre à jour l'état de ses tâches, signaler les obstacles et collaborer de manière transparente avec les autres membres de l'équipe.

Les réunions SCRUM telles que la Mélée Quotidienne, la Revue de Sprint et la Rétrospective de Sprint peuvent être orchestrées efficacement grâce à Jira. L'outil permet de suivre les progrès, de partager les résultats et de documenter les réflexions pour chaque itération.

Les flux de travail (workflows) dans Jira constituent un mécanisme central pour orchestrer et suivre le déroulement des tâches et des projets. Ces flux définissent les étapes spécifiques à suivre pour qu'une tâche ou un problème progresse, de sa création à son achèvement. Les équipes peuvent personnaliser ces flux pour refléter leurs processus uniques, en définissant les transitions entre les étapes et en assignant

des responsabilités à différents membres de l'équipe. Les flux de travail de Jira contribuent à maintenir la transparence, à améliorer l'efficacité et à garantir que toutes les parties prenantes restent informées de l'avancement du travail. Grâce à cette fonctionnalité, les équipes peuvent gérer avec agilité et précision les tâches, tout en favorisant une collaboration fluide et une visibilité accrue sur les projets.

Chez SuiviDeFlotte, le diagramme d'activité présenté dans les Annexes, dans la Figure A.2 (page 32) est la base du flux de travail dans Jira. La partie supérieure gauche du diagramme illustre à nouveau le traitement des idées novatrices par les différents comités. En revanche, la partie supérieure droite présente le traitement des bogues (demandes courantes) qui proviennent de l'équipe de support. Ensuite, le propriétaire du produit sélectionne les tickets pour le prochain sprint à partir de l'ensemble des tickets prêts. Ce diagramme contient les étapes suivantes, qui sont utilisées comme étapes du flux de travail dans Jira : en raffinement, à faire, en cours, en test, à déployer, terminé, abandonné.

Un autre diagramme d'activité montre plus en détail le traitement des bogues (Annexes, Figure A.1, page 31). À partir de ce diagramme, nous pouvons ajouter quelques étapes supplémentaires à la liste des étapes du flux de travail : en attente, en attente support.

Le diagramme de flux de travail qui en résulte et qui est actuellement utilisé est présenté à la Figure 4.6.

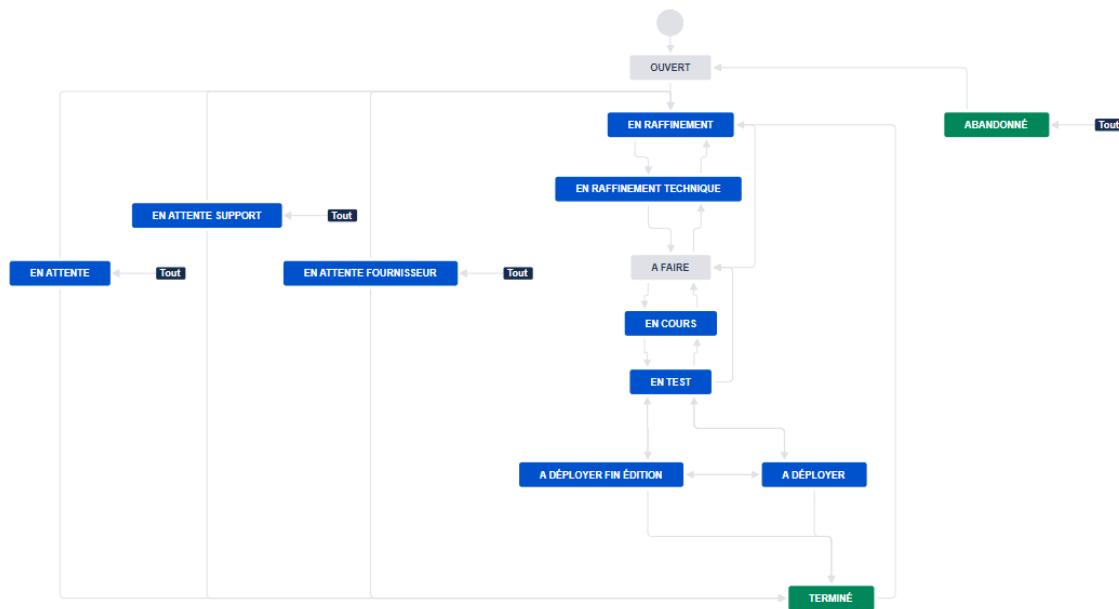


FIGURE 4.6 – Le flux de travail utilisé dans Jira chez SuiviDeFlotte.

Dans les Annexes, la Figure A.3 (page 33) et la Figure A.4 (page 33) présentent un exemple de Backlog de produit et de Backlog de sprint sous forme de tableau Kanban, et la Figure A.5 (page 34) donne un exemple d'un ticket.

4.3 Versioning et les environnements

Dans l'entreprise, les équipes de développement utilisent Git pour la gestion des versions et GitLab pour gérer le processus de développement.

Git est un système de gestion de version décentralisé largement utilisé dans le développement de logiciels. Il permet aux équipes de collaborer efficacement sur des projets en suivant les modifications apportées aux fichiers au fil du temps. Grâce à Git, les développeurs peuvent créer des branches pour travailler sur des fonctionnalités spécifiques ou des corrections de bugs sans perturber le code principal. Les commits, qui représentent des enregistrements de changements, sont la pierre angulaire de Git, permettant de garder une trace claire de l'évolution du code.

GitLab, quant à lui, est une plateforme de gestion de développement logiciel basée sur Git. Elle offre un environnement complet pour le cycle de vie du développement, de la planification à la surveillance. GitLab permet aux équipes de suivre les problèmes, de planifier les sprints, de gérer les demandes d'extraction et de créer des pipelines d'intégration continue pour automatiser les tests et le déploiement. En regroupant toutes ces fonctionnalités au même endroit, GitLab facilite la collaboration entre les membres de l'équipe et permet une gestion transparente et efficace des projets de développement.

Parmi ces nombreuses fonctionnalités, les équipes de DevOps et de développement n'utilisent cependant pas les fonctionnalités de gestion de projet et d'intégration continue/de livraison continue de GitLab, puisqu'elles utilisent Jira (comme nous l'avons vu dans la section précédente) et TeamCity (comme nous le verrons dans la section suivante) à ces fins. GitLab est donc principalement utilisé comme un hébergeur de dépôt de code et un outil de révision de code avec des fonctionnalités telles que les demandes de fusion (merge request).

La Figure 4.7 illustre les principes du versioning et les différents environnements utilisés pour héberger les différentes versions du code. La Table 4.2 apporte quelques compléments d'information.

Du point de vue des versions et des environnements, le processus de développement entre deux éditions est le suivant. Avant la sortie publique de la nouvelle édition, il y a toujours une sortie interne. La sortie interne est créée par la fusion de la branche `develop` dans la branche `master` avec la balise `{numéro d'édition}.0`. Après cela, une nouvelle branche de `release` est créée à partir de la branche `master` avec le nom `release/{numéro d'édition}`. Cette nouvelle branche `release` n'est pas touchée avant la sortie publique.

Entre la sortie interne et la sortie publique, des correctifs urgents (`hotfix`) peuvent être apportés. Lorsque le développeur travaille sur un `hotfix`, il crée d'abord une branche à partir de l'ancienne branche `release` avec un nom `hotfix/{bogue}`, il y dépose ses modifications en travaillant dans son environnement sandbox développeur, puis il crée une demande de fusion pour réintégrer cette branche dans l'ancienne branche `release`. Lorsqu'une décision positive est prise concernant la sortie publique, l'ancienne branche `release` est fusionnée dans les branches `master` et `develop`, l'ancienne branche `release` est clôturée (archivée), la branche `master` est fusionnée dans la nouvelle branche `release`, puis la nouvelle branche `release` est fusionnée dans la branche `develop`, enfin la branche `master` est mise en production dans l'environnement de production avec la balise `{numéro d'édition}.1`. Pendant la période entre la sortie interne et la sortie publique, les développeurs peuvent travailler sur de nouvelles fonctionnalités, mais celles-ci ne seront pas intégrées dans l'édition actuelle, elles ne pourront l'être que dans la prochaine.

Après la sortie publique de la nouvelle édition, les développeurs commencent à travailler sur la prochaine édition. Lorsqu'ils travaillent sur des bogues, des cor-

rectifs (`fix`), ils créent une branche à partir de la branche `release` avec un nom `fix/{bogue}`, ils travaillent dans leur environnement `sandbox`, et lorsqu'ils ont terminé, ils créent une demande de fusion pour fusionner leur branche dans la branche `release`. Lorsqu'ils travaillent sur des fonctionnalités, cela se passe de la même manière, sauf qu'ils créent leur branche à partir de la branche `develop` avec un nom `feat/{fonctionnalité}` et bien sûr ils créent la demande de fusion pour fusionner leur branche dans la branche `develop`. Au cours de cette période, les corrections peuvent être fusionnées dans les branches `master` et `develop` et elles peuvent être mises en production si nécessaire à la fin des sprints ou même au milieu de ceux-ci. Dans ce cas, une nouvelle subversion est ajoutée au commit de fusion sous la forme d'une balise (`{numéro d'édition}.{numéro d'increment}`).

Table 4.2: Les caractéristiques des différents environnements.

Environnement	Pour quels services	Branches	Type de comits	Déploiement TeamCity
sandbox développeur	Développeur	feat / {fonctionnalité} fix/{bogue} hotfix/{bogue}		—
recette-develop	Le PO vérifie les fonctionnalités	develop	feat	Auto
recette-releases	Le PO vérifie les corrections	releases/édition	fix, hotfix	Auto
recette-master	Autres services pour test (direction, marketing, commerce ...)	master	(env juste avant la prod, préproduction)	Manuel, section master dans TeamCity, demande via ticket MEP à DevOps
production		tag	Commit en production avec tag <code>{numéro d'édition}.{numéro d'increment}</code>	Manuel, section production dans TeamCity, demande via ticket MEP à DevOps

A ce stade, il est important de noter que le processus expliqué ci-dessus ne concerne pas qu'un seul projet, mais que, puisqu'il y a plusieurs projets chez SuiviDeFlotte, il les concerne tous. Cela signifie que chaque projet dispose de ses propres branches de `develop`, de `release` et `master`, ainsi que de ses propres environnements. Pour donner une vue d'ensemble des différents projets, la Figure 4.8 présente une image de l'architecture de projets chez SuiviDeFlotte et Beepiz.

Le nouveau projet sur lequel j'ai travaillé la plupart du temps pendant mon alternance est le projet Pipeline documentaire. Cependant, j'ai reçu de temps en temps des tickets de bogues provenant des projets Portail, Gestion de parc, API

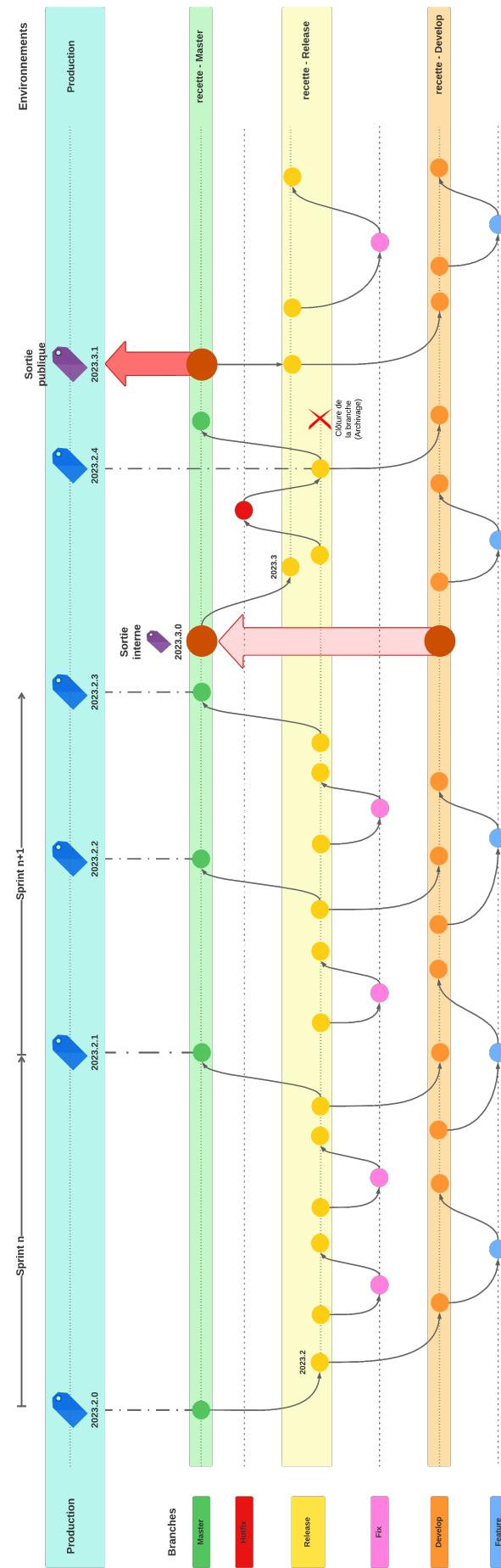


FIGURE 4.7 – Le schéma du versioning et les environnements.

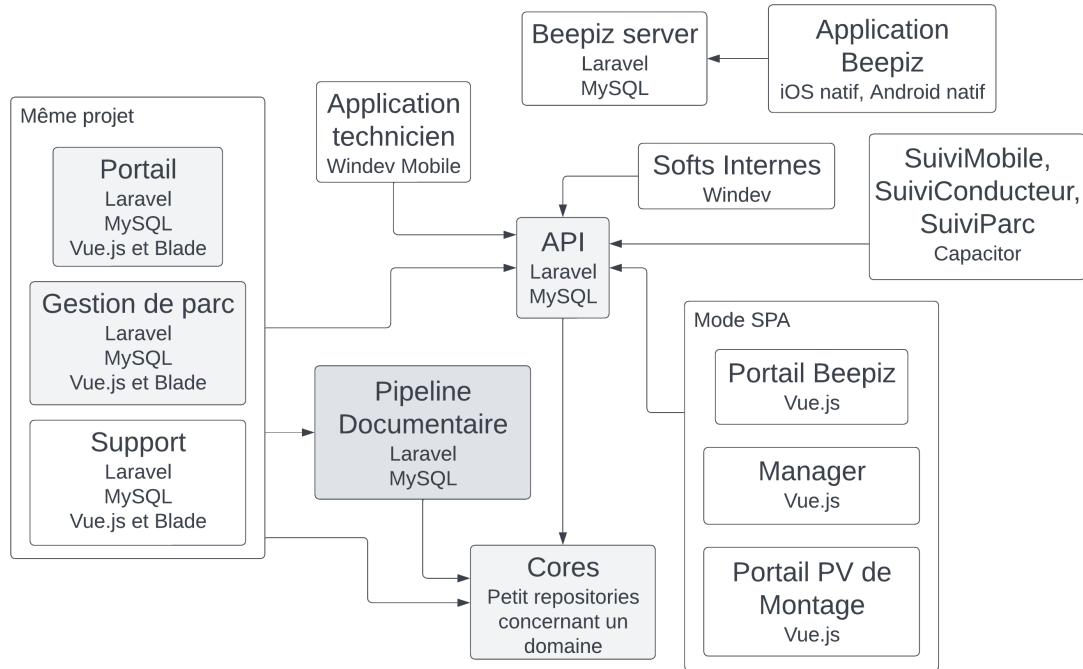


FIGURE 4.8 – La vue d’ensemble des différents projets chez SuiviDeFlotte et Beepiz. Le projet principal sur lequel j’ai travaillé pendant mon alternance était le projet de Pipeline documentaire, mais j’ai également travaillé sur les projets colorés en gris clair.

et Cores. De plus, j’ai travaillé sur une mission d’amélioration du frontend dans le projet Gestion de parc, qui était lié au projet Pipeline documentaire.

Le Portail, la Gestion du parc et le Support font partie du même projet Laravel qui est une application SaaS disponible pour les clients de SuiviDeFlotte. La partie Portail fournit les fonctionnalités de géolocalisation, la partie Gestion de parc fournit les fonctionnalités de gestion de parc comme son nom l’indique et la partie Support aide l’équipe de support dans son travail. L’API est un projet Laravel backend central qui fournit des données à d’autres projets à partir de la base de données. Le projet Cores implique un grand nombre de paquets PHP relativement petits qui sont stockés dans un dépôt de paquets privé créé par Satis.¹ Ces paquets contiennent des fonctionnalités communes qui peuvent être utiles dans plusieurs autres projets. Cependant, lorsque nous avons commencé à travailler sur le projet de Pipeline documentaire, nous avons commencé à travailler sur leur refactorisation et au cours de ce processus, nous avons créé un nouveau paquet (**core-redesing-share**) avec du code refactorisé provenant d’autres paquets et qui n’est pas encore utilisé, mais seulement par le Pipeline documentaire pour le moment.

1. Satis est un outil qui permet aux développeurs PHP de créer un dépôt de paquets privé pour les dépendances de leurs projets. Il offre un contrôle accru sur la distribution des paquets, une sécurité améliorée et des installations de paquets plus rapides, en créant un registre Composer statique qui peut être hébergé n’importe où (même via Docker, localement). Disponible sur <https://github.com/composer/satis>

4.4 Intégration Continue et Déploiement Continu

L'équipe Infrastructure/DevOps utilise TeamCity qui est une plateforme d'intégration continue et de livraison continue largement adoptée dans le domaine du développement logiciel. Elle permet aux équipes de développeurs de automatiser le processus de construction, de test et de déploiement de leurs applications. TeamCity offre un environnement convivial où les développeurs peuvent configurer des pipelines d'intégration continue en spécifiant les étapes nécessaires, telles que la compilation du code, les tests unitaires et les déploiements sur différents environnements.

Une caractéristique clé de TeamCity est sa capacité à détecter automatiquement les changements dans le code source et à déclencher des builds et des tests en conséquence. Cela permet aux équipes de détecter rapidement les problèmes et de s'assurer que le code reste stable et fonctionnel à chaque étape du développement.

En outre, TeamCity offre des fonctionnalités avancées telles que la gestion des agents de construction, la parallélisation des tâches, la gestion des paramètres de build et des rapports détaillés sur les résultats des tests. Cette plateforme aide les équipes à maintenir un processus de développement fluide et à garantir la qualité du code grâce à l'automatisation et à la surveillance continue.

Comme l'indique la Table 4.2 (page 15), le déploiement des branches `develop` et `release` dans les environnements `recette-develop` et `recette-releases` est automatique. Lors d'un commit ou d'un merge dans la branche `develop` ou `release`, TeamCity intercepte l'événement et lance automatiquement les actions ci-dessous :

- Compilation du code
- Exécution des tests unitaires
- Création des fichiers de publication
- Création du package de publication
- Upload de la mise à jour sur Lambda
- Création d'une version et déploiement sur l'environnement correspondant

Ces actions s'enchaînent sauf en cas d'erreur dans l'une d'elle. Dans ce cas, la build est en échec et les actions s'interrompent.

Pour la branche `master` et l'environnement de production, le processus de déploiement est lancé manuellement par l'équipe DevOps.

Pendant mon alternance, je n'ai pas eu accès au portail TeamCity, parce qu'il était géré et utilisé par l'équipe DevOps. En plus, je travaillais principalement sur un tout nouveau projet dont le déploiement n'était pas encore intégré dans TeamCity. Je ne l'ai donc pas vu de l'intérieur. Cependant, j'ai vu les résultats de son fonctionnement lorsque j'ai travaillé parfois sur d'autres projets (ex. Portail, API, Gestion de Parc, Cores) qui ont été déployés par TeamCity. Lorsque je travaillais sur des bogues ou des fonctionnalités et que ma branche était fusionnée dans la branche `release` ou `develop`, je voyais le message dans Slack (qui est la plateforme de communication utilisée par les équipes R&D) que le déploiement avait commencé et plus tard un autre message sur le succès ou l'échec du déploiement. Après un déploiement réussi, bien sûr, j'ai vu aussi que les modifications que j'ai faites ont été appliquées dans l'environnement `recette-releases` ou `recette-develop`.

4.4.1 Le déploiement des paquets de projet Cores

Le déploiement des paquets PHP dans le projet Cores est bien sûr différent du déploiement des autres projets. La procédure est la suivante :

1. **Le développeur met à jour le paquet dans GitLab** : Un développeur apporte des modifications et des mises à jour au code du paquet, éventuellement en corrigeant des bogues ou en ajoutant de nouvelles fonctionnalités. Il enregistre ces modifications dans le dépôt GitLab.
2. **Augmentation de la version du paquet Composer** : Dans le fichier `composer.json` du paquet, le développeur incrémente le numéro de version du paquet. Il s'agit d'une étape importante pour que Composer reconnaisse qu'une mise à jour a eu lieu.
3. **Notifier Satis** : Déclencher le processus de mise à jour pour Satis en exécutant le script créé à cet effet.
4. **Processus de mise à jour de Satis** : Le déclenchement du processus de mise à jour de Satis implique l'exécution d'une commande qui régénère les métadonnées statiques de Composer pour le dépôt Satis. Cette commande récupère les dernières informations du dépôt GitLab et met à jour les métadonnées en conséquence.
5. **Paquet mis à jour dans le dépôt Satis** : Le paquet mis à jour avec sa nouvelle version est maintenant inclus dans le dépôt Satis. Cela signifie que le paquet mis à jour est maintenant disponible pour l'installation via Composer à partir du dépôt privé Satis.
6. **Composer Update** : Les développeurs travaillant sur des projets qui dépendent de ce paquet mis à jour peuvent maintenant utiliser Composer pour mettre à jour leurs dépendances. Lorsqu'ils lancent `composer update` ou `composer require`, Composer vérifie le dépôt Satis, trouve la nouvelle version du paquetage mis à jour, le télécharge et l'installe dans leur projet.

Comme nous pouvons le constater, ce processus n'est pas encore automatisé à l'exception du fichier script pour notifier Satis. Dans ce processus manuel, il est de la responsabilité du développeur ou de l'équipe DevOps de notifier le Satis et de déclencher la mise à jour.

4.4.2 Le déploiement du projet Pipeline documentaire

Comme je l'ai mentionné précédemment, étant un nouveau projet, le déploiement du projet Pipeline documentaire n'est pas encore intégré dans TeamCity, il est fait de manière semi-automatique par l'équipe DevOps. Il peut être qualifié de semi-automatique car le projet fonctionne dans des conteneurs Docker² pendant le développement ainsi qu'en production, il a donc un `Dockerfile`,³ un fichier

2. Docker est une plateforme de virtualisation légère et portable qui permet de créer, déployer et exécuter des applications dans des conteneurs. Les conteneurs sont des environnements isolés qui regroupent tous les éléments nécessaires à l'exécution d'une application, tels que le code, les bibliothèques et les dépendances. Cela garantit une cohérence entre les environnements de développement, de test et de production, simplifiant ainsi le processus de déploiement.

3. Un `Dockerfile` est un fichier texte qui contient les instructions pour construire une image Docker. Une image Docker est un modèle d'environnement qui contient tous les composants né-

`docker-compose.yaml`⁴ et l'équipe DevOps a créé un fichier de script d'installation aussi.

cessaires pour exécuter une application. Le **Dockerfile** décrit les étapes pour installer les dépendances, copier les fichiers de l'application et configurer l'environnement. Une fois le **Dockerfile** créé, il peut être utilisé pour générer une image Docker prête à être exécutée dans un conteneur.

4. Le fichier `docker-compose.yaml` est un autre élément clé dans l'écosystème Docker. Il permet de définir et de gérer plusieurs conteneurs interconnectés en tant qu'application unique. Ce fichier décrit les services, les images, les ports exposés et les liens entre les conteneurs, facilitant ainsi le déploiement et la gestion d'applications complexes à plusieurs composants.

5 Cahier des charges

6 Spécifications fonctionnelles

7 Spécifications techniques

8 Réalisations

9 Présentation d'un jeu d'essai

10 Veille sur les vulnérabilités de sécurité

11 Description d'une situation de travail ayant nécessité des travaux de recherches

12 Conclusion

13 Remerciements

Annexes

A Jira : L'outil essentiel pour la gestion quotidienne de projet

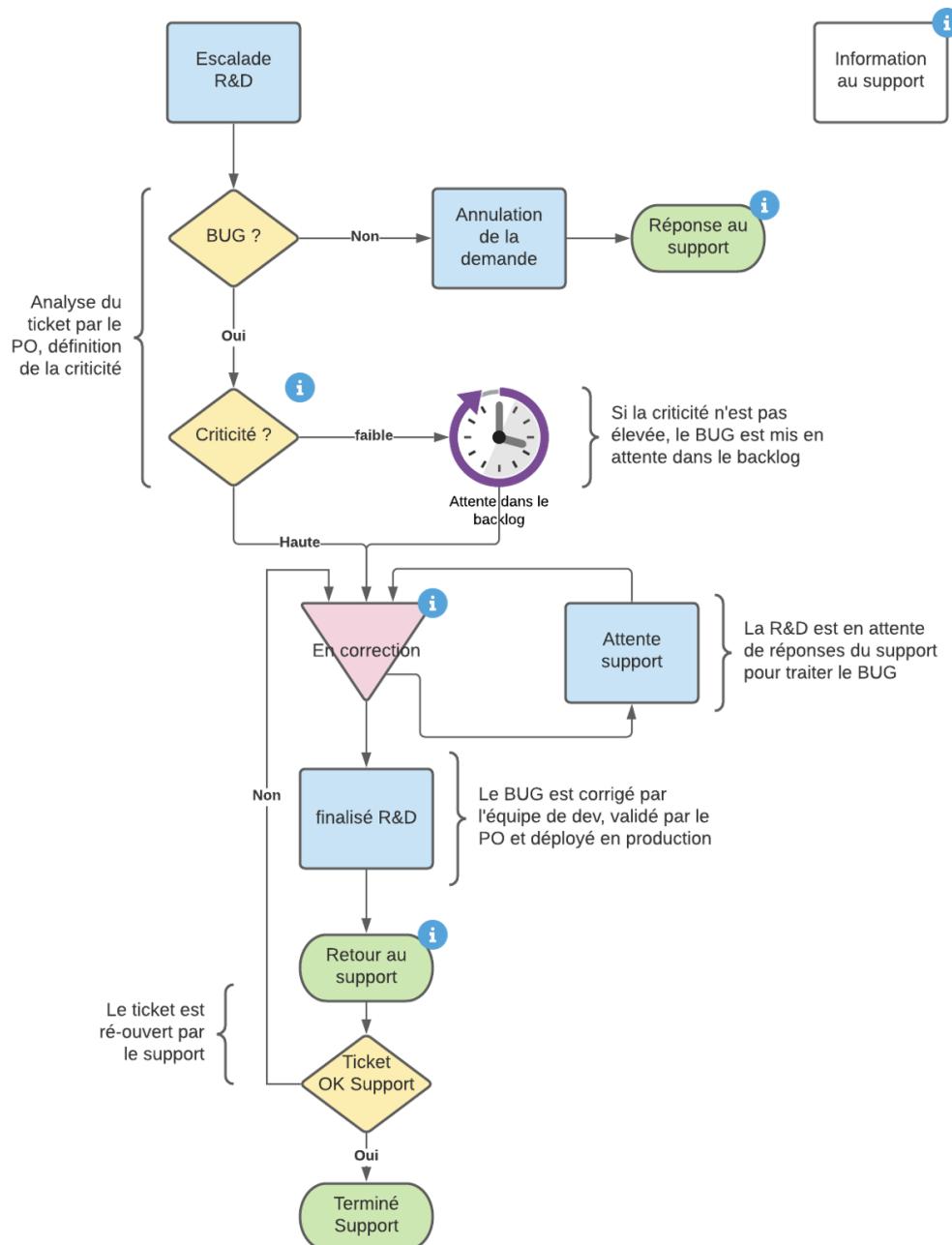


FIGURE A.1 – Diagramme d'activité du traitement des bogues.

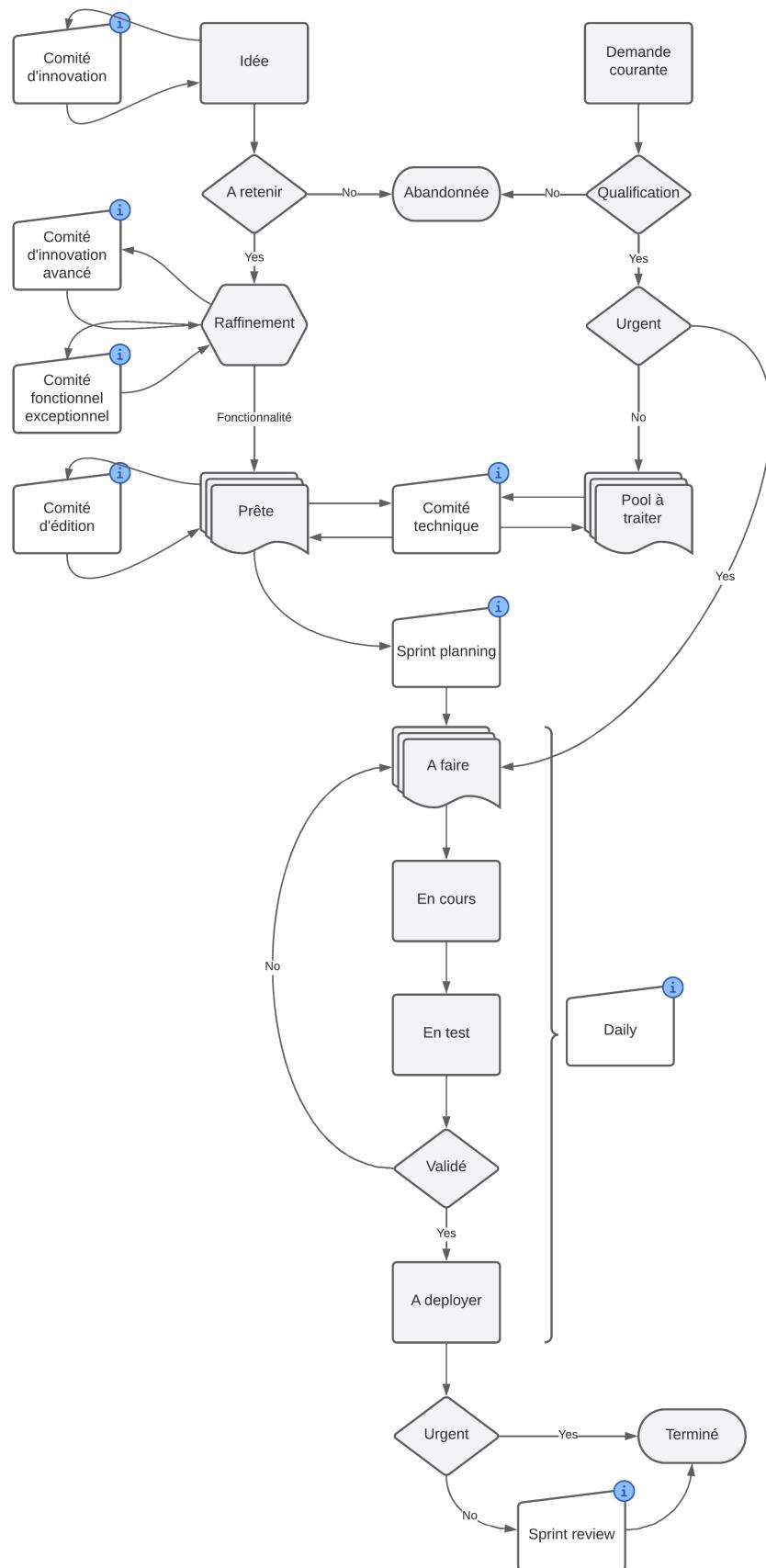


FIGURE A.2 – Diagramme d'activité du traitement des idées et des bogues (demandes courantes) entrants chez SuiviDeFlotte.

Tableau Kanban

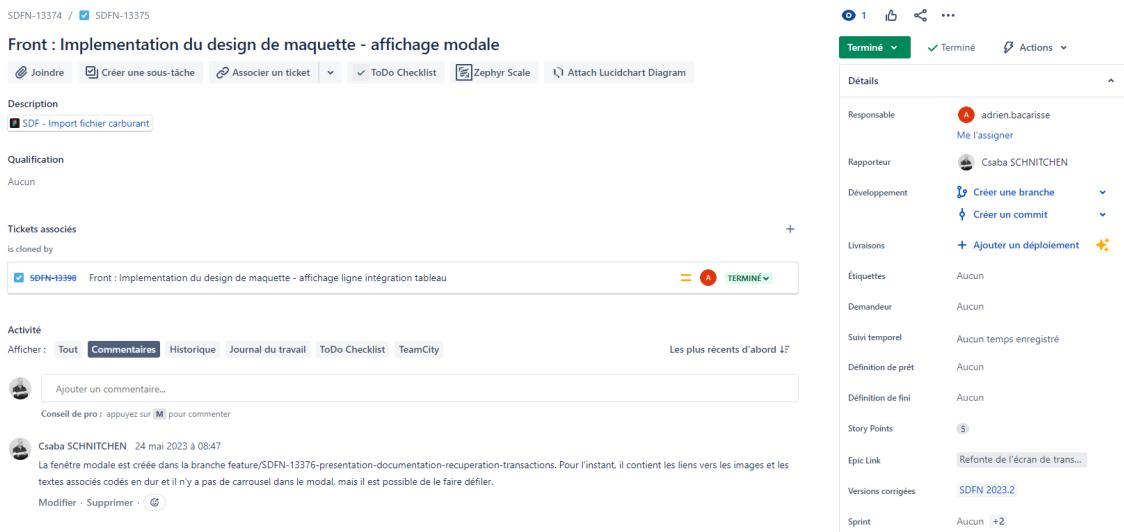
IDÉES	EN RAFFINEMENT	EN RAFFINEMENT TECHNIQUE	PRÈTES	EN COURS	EN ATTENTE	TERMINÉES
Zones de livraison d'un rayon de 10km Aucun Portail Géolocalisation 5 SDFN-381	Ajouter acces application en mode Badge identifiant au lieu de Immatriculation Aucun Portail Géolocalisation 5 SDFN-286	Gestion Cartes carburant Aucun Aucun 4 SDFN-12348	Mettre en ligne le guide d'intégration DVK Aucun 1 Aucun 5 SDFN-13565	Refacto : Création des projets de test RPC Aucun 3 Aucun 5 SDFN-13501	Analysier la problématique de traitement de la queue kuantic de la semaine du 26 - Aucun 1 Aucun 5 SDFN-12930	Etude refonte alertes de zones REFONTE - ALERTE DE ZONE SDFN 2023.3 5 Aucun 5 SDFN-12753
Gestion cuve de gasoil Aucun Portail Géolocalisation 5 SDFN-379	Alerte échéance Carte entreprise (Offre PL) Aucun Portail Géolocalisation 5 SDFN-272	Trouver l'origine des crash récurrents de php5-fpm sur le serveur de dev 2022.2 Aucun Aucun 5 SDFN-12303	Analyser les possibilité sur les POI avec l'api Mapotempo Aucun Aucun 5 SDFN-13265	Remplacer les appels par l'api de géocodage REFONTE DU MÉCANISME DE G... Aucun 5 Aucun 5 SDFN-13507	Les filtres doivent être dynamiques et se recharger selon les selections des autres EVOLUTIONS TCO Aucun 5 Portail Gestion de parc 5 SDFN-12260	Vérifier les utilisations de la table fournisseur legacy Aucun 5 Aucun 5 SDFN-13400
Mettre les conf de RC à niveau pour SuiviMission Aucun Aucun 5 SDFN-12502	Documents réglementaires de conduite Aucun Portail Gestion de parc 5 SDFN-383	Refonte de la gestion du paramètre vie privée d'un véhicule Aucun Aucun 5 SDFN-12151	Visualiser, dans le rapport journalier, le passage en ZFE - Partie Carte MISE EN PLACE DE LA GESTION ... Aucun Aucun 5 SDFN-12822	Utilisation de l'outil de reverse géocodage dans le reste de SDF REFONTE DU MÉCANISME DE G... SDFN 2023.3 1 Aucun 5 SDFN-13404	Interactivity tableau / diagramme EVOLUTIONS TCO Aucun 8 Portail Gestion de parc 5 SDFN-12243	Plan d'action de refonte générale back Aucun 8 Portail Gestion de parc 5 SDFN-13497
SI : Support technique attribution de ticket Aucun Aucun 5 SDFN-13496	Déclaration de sinistre via l'application mobile Aucun Portail Gestion de parc 5 SDFN-54	Détection coupure moteur Aucun Aucun 4 SDFN-12791	Vérification de fin de charge ZOE FG-490-EA Aucun Aucun 5 SDFN-12791	Pipelinedoc - Import dkv PIPELINE D'IMPORT Aucun Aucun 5 SDFN-13581	Scan plaque d'immat quick privacy	Analysé découpe BDD REFONTE TECHNIQUE - DÉCOU... Aucun 5 Aucun 5 SDFN-13586
Nom du groupe d'un véhicule sur la carte Aucun Portail Géolocalisation 5						

FIGURE A.3 – Le Backlog de produit sous forme de tableau Kanban dans Jira.

Sprint SDFN 2023.3 S5

A FAIRE	EN COURS	EN TEST	A DÉPLOYER
Mettre en ligne le guide d'intégration DVK Aucun 5 SDFN-13565	Refacto : Création des projets de test RPC Aucun 3 SDFN-13501	Remplacer les appels par l'api de géocodage REFONTE DU MÉCANISME DE GÉOCODAGE Aucun 5 SDFN-13507	Date de changement du ticket Jira Aucun 1 SDFN-13586
128393 Module carburant: souci sur transaction non remontées Aucun 1 SDFN-13525	Migration véhicules non ExVe sur Kinesis REFONTE CONNECTEURS - KINESIS SDFN 2023.2 3 SDFN-13396	Analyse trajet Kinesis et Kinesis ExVe non clôturé. REFONTE CONNECTEURS - KINESIS Aucun 3 SDFN-13533	11963 Import POI erreur SQL Aucun 3 SDFN-13395
Table yy_ww_trajets le champs fleet_vehicle_id ne doit pas être vide Aucun 5 SDFN-13546	Validation et augmentation charge ExVe Kinesis REFONTE CONNECTEURS - KINESIS SDFN 2023.2 3 SDFN-13397	pipelinedoc - endpoints de consultations des uploads, leurs états et erreurs PIPELINE D'IMPORT Aucun 3 SDFN-13583	Plan d'action de refonte générale back Aucun 8 SDFN-13497
Etude des librairies à utiliser REFONTE COUCHE AUTHENTICATION Aucun 5 SDFN-13549	128399 Rapport hebdomadaire non centrés Aucun 3 SDFN-13515	119631 Problème kilométrages semaine: total kilométrage semaine + weekend non correct Aucun 3 SDFN-13518	Lister les batch v4 à refaire en v5 SUPPRIMER LE SERVEUR DE BATCH DE TOURS Aucun 5 SDFN-13562
Préparation du docker REFONTE COUCHE AUTHENTICATION Aucun 5 SDFN-13550			

FIGURE A.4 – Le Backlog de sprint sous forme de tableau Kanban dans Jira.



The screenshot shows a Jira ticket page for ticket SDFN-13374. The ticket title is "Front : Implementation du design de maquette - affichage modale". The status is "Terminé" (Completed). The ticket has a responsible person, adrien.bacarisse, and a reporter, Csaba SCHNITCHEN. It includes sections for development (branch creation, commit creation), deployment (deployment addition), and other fields like labels, requester, and sprint. There are also sections for associated tickets and activity history.

Description

SDFN-13374 / SDFN-13375

Front : Implementation du design de maquette - affichage modale

Actions

- Joindre
- Créer une sous-tâche
- Associer un ticket
- ToDo Checklist
- Zephyr Scale
- Attach Lucidchart Diagram

Description

SDFN-13374 - Front : Implementation du design de maquette - affichage modale

Qualification

Aucun

Tickets associés

is cloned by

SDFN-13398 - Front : Implementation du design de maquette - affichage ligne intégration tableau

Activité

Afficher : Tout, Commentaires, Historique, Journal du travail, ToDo Checklist, TeamCity

Ajouter un commentaire...

Conseil de pro : appuyez sur M pour commenter

Csaba SCHNITCHEN - 24 mai 2023 à 08:47

La fenêtre modale est créée dans la branche feature/SDFN-13376-présentation-documentation-recuperation-transactions. Pour l'instant, il contient les liens vers les images et les textes associés codés en dur et il n'y a pas de carrousel dans le modal, mais il est possible de le faire dérouler.

Modifier · Supprimer ·

Détails

Responsable	adrien.bacarisse Me l'assigner
Rapporteur	Csaba SCHNITCHEN
Développement	Créer une branche · Créer un commit ·
Livrées	Ajouter un déploiement ·
Étiquettes	Aucun
Demandeur	Aucun
Suivi temporel	Aucun temps enregistré
Définition de prêt	Aucun
Définition de fini	Aucun
Story Points	5
Epic Link	Refonte de l'écran de trans...
Versions corrigées	SDFN 2023.2
Sprint	Aucun +2

FIGURE A.5 – Un exemple de ticket dans Jira pour le projet d'amélioration de la page d'importation des fichiers des transactions de carburant.