



sopra  steria

**Pierrick GIULIANI**  
Promotion 2014 - 2017  
Département Informatique & Gestion

**Sopra Steria**  
1231 Avenue Du Mondial 98,  
34090 Montpellier

---

## STAGE DE FIN D'ÉTUDE

Département Informatique & Gestion effectué du 15 février au  
13 août 2017

Développement d'évolutions en PHP pour l'application Facteurs  
Services Plus



Votre atout proximité

---

### Tuteur école :

Lysiane BUISSON-LOPEZ  
lysiane.lopez@umontpellier.fr  
04-67-14-46-56/48-70

### Tuteur entreprise :

Mathieu LE-GUIRRIEC  
mathieu.leguirriec@soprasteria.com  
06-37-28-39-39

Année 2016-2017



## Remerciements

Je tiens à remercier toutes les personnes qui m'ont permis de mener à bien ce stage de par leur implication dans celui-ci,

M. LE-GUIRRIEC mon tuteur d'entreprise pour sa confiance, sa disponibilité, son implications ainsi que ses conseils tout au long du stage.

M. FRAPPEREAU mon parrain au sein de Sopra Steria pour son écoute, sa patience, ses conseils et sa disponibilité.

M. KUHNEN leader technique du projet pour ses conseils, son écoute, son aide ainsi que pour la bonne ambiance.

Mme. BUISSON-LOPEZ ma tutrice enseignante pour sa disponibilité, ses conseils, son écoute ainsi que pour le travail de relecture.

L'équipe Française (Mme. PROUST, M. KUHNEN, M. LE-GUIRRIEC, M. FRAPPEREAU, M. GONZALES et M. CHAMBON) pour leur intégration, leur implication ainsi que le cadre de travail agréable.

L'équipe Barcelonaise (Mme. DELFIEU, M. JOSEP, M. MATHIOU, M. ELBOUDJAY et M. LEGIDO) pour leur implication, même à plusieurs centaines de kilomètres.

Ainsi que ceux que j'ai côtoyés de près ou de loin tout au long du stage.

# Table des matières

<b>1</b>	<b>Présentation de l'entreprise</b>	<b>2</b>
<b>2</b>	<b>Le projet</b>	<b>5</b>
2.1	La Poste . . . . .	5
2.2	Facteurs Services Plus ( <i>FS+</i> ) . . . . .	5
2.3	Aspect fonctionnel . . . . .	6
2.3.1	Diagramme d'utilisation . . . . .	7
2.3.2	Fonctionnalités offertes . . . . .	7
2.3.2.1	Prise de commande via IHM <sup>1</sup> . . . . .	8
2.3.2.2	Prise de commande via CSV . . . . .	10
2.3.2.3	Suivi de commandes . . . . .	10
2.4	Aspect technique . . . . .	11
2.4.1	Les modules . . . . .	11
2.4.2	Le module U1 . . . . .	13
2.4.2.1	Framework Zack et Zend . . . . .	13
2.4.2.2	Zack Manager . . . . .	13
2.4.2.3	Designs pattern et architectures principales . . . . .	14
2.4.3	Git . . . . .	15
2.4.4	Web-services et partenaires . . . . .	17
<b>3</b>	<b>Conduite du projet</b>	<b>19</b>
3.1	Présentation des équipes . . . . .	19
3.1.1	France . . . . .	19
3.1.2	Barcelone . . . . .	19
3.2	Gestion de projet . . . . .	20
3.2.1	SCRUM . . . . .	20
3.2.1.1	Rôles . . . . .	20
3.2.1.2	Evénements . . . . .	21
3.3	Organisation propre à Sopra Steria . . . . .	22
3.3.1	V1 . . . . .	22
3.3.2	P1 et CRA . . . . .	22
3.3.3	JIRA . . . . .	23
3.3.4	Sentinelle du MCO <sup>2</sup> . . . . .	23
<b>4</b>	<b>Partie technique</b>	<b>25</b>
4.1	Déroulement d'une évolution . . . . .	25
4.1.1	Initialisation . . . . .	25
4.1.2	Analyse et développement . . . . .	27
4.1.3	Plan de test et mise à jour JIRA . . . . .	28
4.2	Déroulement d'une anomalie . . . . .	29
4.3	Déroulement de la VNR . . . . .	30

---

1. Interface **H**omme **M**achine

2. **M**aintient en **C**ondition **O**opérationnelle

## Table des figures

1	Logos des entreprises . . . . .	2
2	Chiffres clés . . . . .	2
3	Secteurs de marché, implantations et activités de Sopra Steria . . . . .	3
4	Agence de Sopra Steria à Montpellier . . . . .	4
5	Logo Facteurs Services Plus (FS+) . . . . .	4
6	Logo La Poste . . . . .	5
7	Différents services par FS+ . . . . .	6
8	Use-case d'utilisation . . . . .	7
9	Options offertes pour un client . . . . .	8
10	Prise de commande via IHM 1 . . . . .	9
11	Prise de commande via IHM 2 . . . . .	9
12	Prise de commande via CSV 1 . . . . .	10
13	Prise de commande via CSV 2 . . . . .	10
14	Suivi de commandes . . . . .	11
15	Logos des technologies . . . . .	11
16	Interactions des modules . . . . .	12
17	Logo Zack . . . . .	13
18	Logo Zend . . . . .	13
19	Zack Manager . . . . .	14
20	Git flow . . . . .	16
21	Template d'un commit . . . . .	17
22	Ensemble des partenaires . . . . .	18
23	Diagramme d'explication d'un Sprint ( <i>méthode SCRUM</i> ) . . . . .	20
24	Cartes scrum . . . . .	21
25	Aperçu d'un tableau JIRA pour le sprint 16 . . . . .	23
26	Exemple d'une JIRA pour une de mes évolutions . . . . .	26
27	Ajout de la balise "passage" dans un fichier XSD . . . . .	27
28	Ajout de la balise "passage" dans les types . . . . .	27
29	Ajout de la variable application pour REMX . . . . .	28
30	Utilisation de la variable REMX dans le code . . . . .	28
31	Exemple d'une JIRA pour une correction d'anomalie . . . . .	29
32	Code existant de l'anomalie . . . . .	29
33	Code modifier de l'anomalie . . . . .	30
34	Scénario de VNR dans HPQC . . . . .	31

## Introduction

Dans le cadre du stage qui s'inscrit dans la dernière année de formation à Polytech Montpellier en **I**nformatique et **G**estion (*IG*), j'ai travaillé au sein du groupe Sopra Steria Montpellier qui est une **E**ntreprise de **S**ervices du **N**umérique (*ESN*) et plus particulièrement sur le projet **F**acteurs **S**ervice **P**lus (*FS+*), aussi appelé **N**ouveaux **S**ervices de **P**roximité (*NSP*) pour le compte de **La Poste**.

Face à la chute d'activité liée à la baisse du volume de courrier, La Poste propose une nouvelle gamme de services permettant de fortifier leurs activités : Nouveaux Services Plus.

Les facteurs assurent désormais bien d'autres services que la distribution et la collecte du courrier avec par exemple la relève de compteur, la livraison de course à domicile, des visites de prévention pour les personnes âgées ou fragiles... Tout cela dans un but de simplifier la vie des gens et des entreprises.

En tout, plus de 600 services sont proposés 6 jours sur 7 grâce aux 80 000 facteurs.

Le choix de mon stage s'est basé sur l'envie de découvrir le processus de développement ainsi que les bonnes pratiques de gestion de projet au sein d'une grande société telle que Sopra Steria. Il m'a permis de découvrir de l'intérieur le déroulement d'un projet conséquent, Facteurs Services Plus.

Ce rapport présentera l'ensemble du travail qui a été réalisé au cours du stage en commençant par une présentation de l'entreprise ainsi que du projet dans un deuxième temps. Puis, je détaillerai la stratégie de conduite de projet au sein d'une troisième partie. En avant dernière partie, je parlerai plus techniquement des tâches réalisées et je clôturerai ce rapport par un bilan du travail accompli.

# 1 Présentation de l'entreprise

Sopra Steria est une **E**ntreprise de **S**ervice du **N**umérique (*ESN*, ou bien une *Société de Services en Ingénierie Informatique (SSII)*). Cette entreprise est née d'une fusion en janvier 2015 des entreprises **Sopra** et **Steria** créées respectivement en 1968 et 1969.



FIGURE 1 – Logos des entreprises

Sopra Steria est l'une des plus importantes ESN de France et d'Europe avec notamment de nombreux secteurs différents, tels que le secteur public, les services financiers, l'aéronautique, la défense, les assurances, les transports, ... (*cf. Figure 3a*).

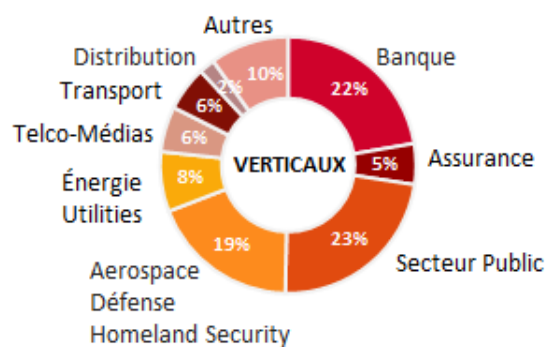
Elle compte aujourd'hui environ 40 000 collaborateurs échelonnés dans plus de 20 pays (*cf. Figure 3b*) et affiche un chiffre d'affaires de 3,7 milliards d'euros en 2016.



FIGURE 2 – Chiffres clés

Elle propose aujourd'hui des offres variées (*cf. Figure 3c*) telles que :

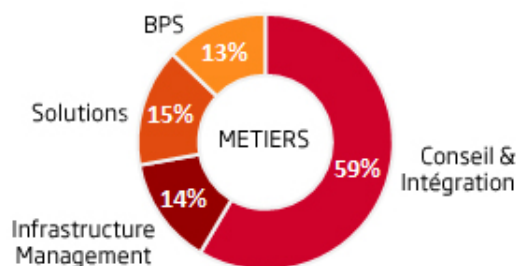
- **Conseil** grâce à sa filiale *Sopra Steria Consulting* permettant d'aider les grandes entreprises et organismes publics à franchir de nouvelles étapes dans leur développement et leur compétitivité
- **Solution métier** afin de construire des solutions performantes en harmonie avec la politique de développement des entreprises, notamment dans trois domaines : la banque de détail, l'immobilier et les **R**essources **H**umaines (*RH*)
- **Infrastructure & Management (IM)** afin d'aider les entreprises à adapter leurs infrastructures en adéquation avec leurs exigences métiers, notamment avec la mise en place de Cloud ainsi que de Datacenter
- **Business Process Services (BPS)** pour la conception de solution dans l'objectif d'externaliser des fonctions clés pour plus d'efficacité, d'agilité et de compétitivité



(a) Secteurs de marché de Sopra Steria



(b) Implantation de Sopra Steria



(c) Répartition des activités de Sopra Steria

FIGURE 3 – Secteurs de marché, implantations et activités de Sopra Steria

Sopra Steria coopère avec de nombreux clients que l'on peut considérer comme de "grands comptes", dont notamment *EDF*, *Vinci*, *SNCF*, *Carrefour*, *la RATP*, *Conforama* et *La Poste*.

## Sopra Steria Montpellier

C'est au sein de l'agence de Montpellier que j'ai été accueilli. Celle-ci fait partie de la division sud-Est du groupe, elle est dirigée par Mme. *Isabelle Reix Di Biase* et compte environ 180 collaborateurs pour 30 sous-traitants.

Elle est implantée dans le parc d'activités du millénaire (*où je me situe*) ainsi qu'au Parc Club. Les deux sites sont seulement éloignés d'environ 700 mètres ce qui facilite l'échange et la cohésion.





FIGURE 4 – Agence de Sopra Steria à Montpellier

**La Poste** est un client majeur de Sopra Steria, c'est en effet le client le plus important de l'agence de Montpellier (*environ 60% du chiffre d'affaires en 2016*). C'est justement sur une application pour le groupe de **La Poste** que j'ai travaillé durant mon stage, **Facteurs Service Plus** (*FS+*).



FIGURE 5 – Logo Facteurs Services Plus (FS+)

## 2 Le projet

Au sein de cette partie, je vais vous faire une présentation rapide de La Poste et du projet FS+ de manière générale ainsi que leurs aspects fonctionnels et techniques.

### 2.1 La Poste

La Poste est une entreprise française qui se divise en plusieurs branches principales :

- courrier, colis et express
- banque, assurance et opérateur de téléphonie mobile
- fournisseur de services numériques et de solutions commerciales
- commerce en ligne
- collecte et vente de données personnelles



FIGURE 6 – Logo La Poste

Durant mon stage, j'ai travaillé au sein du projet *MESSI* (qui permet à La Poste d'externaliser ses applications) se rattachant à la branche *Courrier, colis et express* permettant d'assurer la maintenance corrective et l'évolution des **S**ystèmes d'**I**nformation (*SI*)<sup>3</sup> de La Poste. Le projet que j'ai intégré est Facteurs Services Plus.

### 2.2 Facteurs Services Plus (FS+)

La Poste fait face depuis plusieurs années à une baisse de volume du courrier et des colis due aux développements des communications numériques (*Mails, SMS, ...*) et prévoit une forte baisse du volume de courrier entre 2007 et 2020.

Ainsi, afin de faire face à leurs coûts et de tirer parti du nombre important de facteurs à disposition (*environs 80 000*) et d'un contact 6 jours sur 7 avec la population, l'entreprise a donc mis en place de nombreux nouveaux services, dont les **N**ouveaux **S**ervices de **P**roximité (*NSP*), qui nous intéressent ici.

L'application FS+ est une des briques du SI La Poste, permettant la prise et la gestion de commandes mais également la facturation de ces nouveaux services.

Ces nouveaux services sont très variés :

---

3. **S**ystème d'**I**nformation correspondant à un ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information.

- **Proxi contact et data** : collecte d'informations utiles auprès de personnes ayant donné leur accord (cf. 1 et 2)
- **Proxi vigie** : action sociale et visite auprès de personnes âgées ou fragilisées (cf. 3)
- **Proxi diag** : diagnostic et constat, prise de photos dans les lieux publics pour surveiller l'état du mobilier public, relève des index de consommation d'énergie, ... (cf. 4)
- **Proxi green** : collecte et recyclage de produits usagés, par exemple les capsules de café usagées (pour Nespresso, un client important de La Poste) ou de papiers de bureau (cf. 5)
- **Proxi equip** : installation d'équipements ainsi que prise en main de ces outils (cf. 6)
- **Proxi course** (également appelé **Porteo**) : collecte et livraison de proximité à domicile (cf. 7)



FIGURE 7 – Différents services par FS+

Ces nouveaux services sont à destination des entreprises et collectivités, qui souscrivent à un ensemble de prestations, afin d'en faire bénéficier leurs clients, particuliers ou sociétés.

## 2.3 Aspect fonctionnel

Je vais ici vous présenter le fonctionnement principal de l'application à l'aide d'un diagramme général et de quelques captures d'écran afin que vous compreniez qui sont les utilisateurs (*par exemple : mairies, hopitaux, magasins, boulangers, particuliers, ...*) et que vous découvriez les principales fonctionnalités.

### 2.3.1 Diagramme d'utilisation

Comme on peut l'observer sur la figure ci-dessous, les utilisateurs ont deux choix principaux sur l'application :

- passer une commande afin de profiter d'un des services cités précédemment
- voir l'avancée de leurs commandes déjà passées

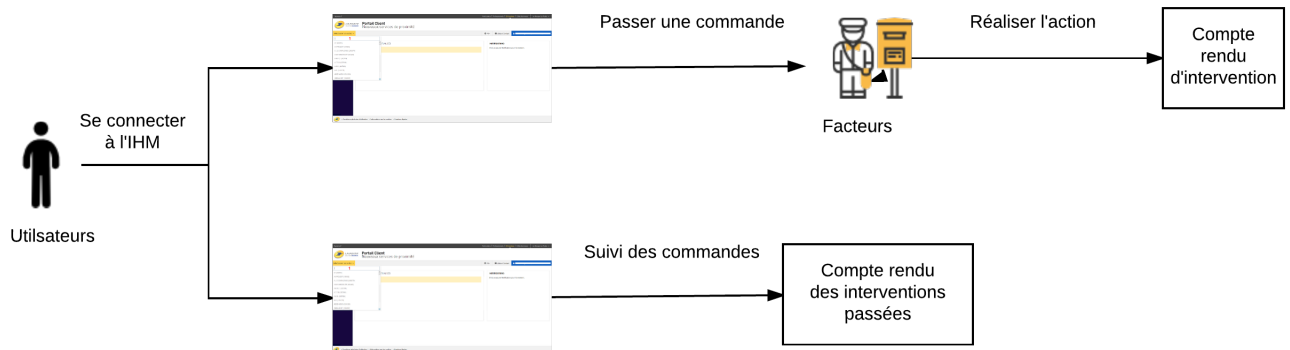


FIGURE 8 – Use-case d'utilisation

Les utilisateurs peuvent être de différentes natures, cela peut par exemple être un centre commercial comme *Carrefour* qui a besoin d'un prestataire pour faire livrer les courses à domicile, un boulanger qui voudrait faire livrer son pain à domicile, la mairie qui organise des actions de préventions pour les personnes âgées et qui a besoin de personnes assurant des visites quotidiennes, ...

### 2.3.2 Fonctionnalités offertes

Lorsque l'utilisateur se connecte, il arrive sur une page avec l'ensemble des fonctionnalités offertes par l'application (cf. 1), il a également une vue avec l'ensemble de ses services (cf. 2). À l'aide du bouton "*commander*", il est possible de passer une commande unitaire depuis l'interface (cf. 3), ou d'envoyer un fichier au format CSV (cf. 4) contenant les informations de plusieurs commandes.

Il a aussi la possibilité d'accéder au suivi de réalisation des commandes (cf. 5) qui ont déjà été passés afin d'en voir les informations. Il a également accès aux missions (cf. 6) qui ont été réalisées. Il pourra exporter les informations de ces dernières au format csv.

La différence entre la commande et la mission réside dans le fait que la commande correspond à ce que veut le client (*en terme de finalité, une livraison par exemple*) alors que les missions correspondent aux actions qui doivent être réalisées (*par exemple, lorsque que Carrefour souhaite livrer des courses chez quelqu'un, il passe une commande de livraison. Celle-ci va alors générer 3 missions : une mission de récupération des courses par le postier, une mission de livraison et une mission de retour à Carrefour*).

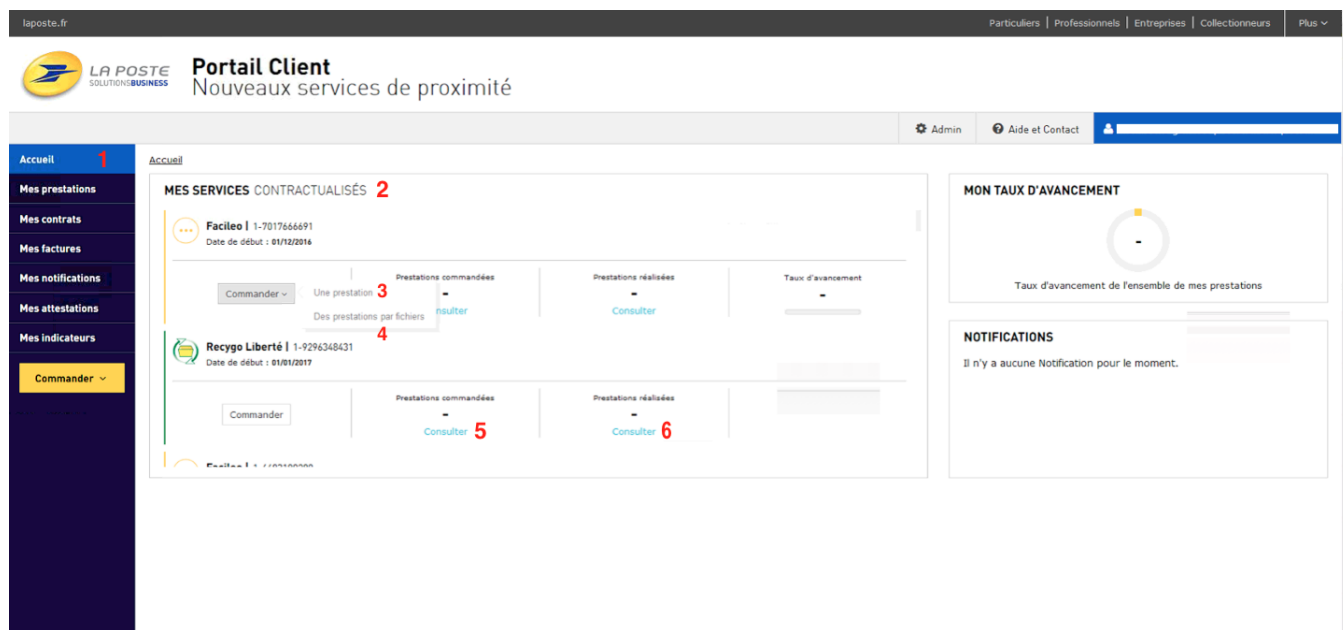


FIGURE 9 – Options offertes pour un client

#### 2.3.2.1 Prise de commande via IHM<sup>4</sup>

Lorsque l'utilisateur clique sur le bouton "Une prestation" (cf. Figure 9 numéro 3) il arrive sur les pages ci-dessous (cf. Figure 10 et 11) qui lui permet de commander une prestation à l'aide d'un formulaire.

Choix du contrat / service      Bénéficiaire      Dates souhaitées      Détails de la prestation      Confirmation

1 — 2 — 3 — 4 — 5

Facileo - 1-7017666691 | COLLECTE ENV T LA BANQUE POSTALE ASSURANCE IARD

**BÉNÉFICIAIRE**

Nom Prénom ou raison sociale \*  
PGI rapport

Lieu de remise

Référence bénéficiaire \*  
PGI rapport

N° d'appartement - Etage - Couloir - Escalier

Adresse \*  
1 IMPASSE UN 62740 FOUQUIERES LES LENS

Indication d'accès au bénéficiaire

Indications complémentaires

\* Champs obligatoires

Précédent    Suivant

FIGURE 10 – Prise de commande via IHM 1

LA POSTE SOLUTIONS BUSINESS    Portail Client    Nouveaux services de proximité

Admin    Aide et Contact    [User Profile]

Accueil    Mes prestations    Mes contrats    Mes factures    Mes notifications    Mes attestations    Mes indicateurs    Commander

Accueil > Mes commandes

**COMMANDER UNE PRESTATION**

Choix du contrat / service      Bénéficiaire      Dates souhaitées      Détails de la prestation      Confirmation

1 — 2 — 3 — 4 — 5

Votre sélection :

Facileo - 1-7017666691 | COLLECTE ENV T LA BANQUE POSTALE ASSURANCE IARD

**DATES SOUHAITÉES**

Date de début \*  
08/06/2017

\* Champs obligatoires

Précédent    Suivant

Conditions générales d'utilisation    Informations sur les cookies    Mentions légales

FIGURE 11 – Prise de commande via IHM 2

Ces formulaires diffèrent en fonction du service choisi (*Proxi course, proxi vigie, proxi contact, ...*).

### 2.3.2.2 Prise de commande via CSV

En revanche, lorsque l'utilisateur clique sur le bouton “*Des prestations par fichier*” (cf. Figure 9 numéro 4) il arrive sur la page ci-dessous (cf. Figure 12) qui lui permet de télécharger le format du fichier CSV.

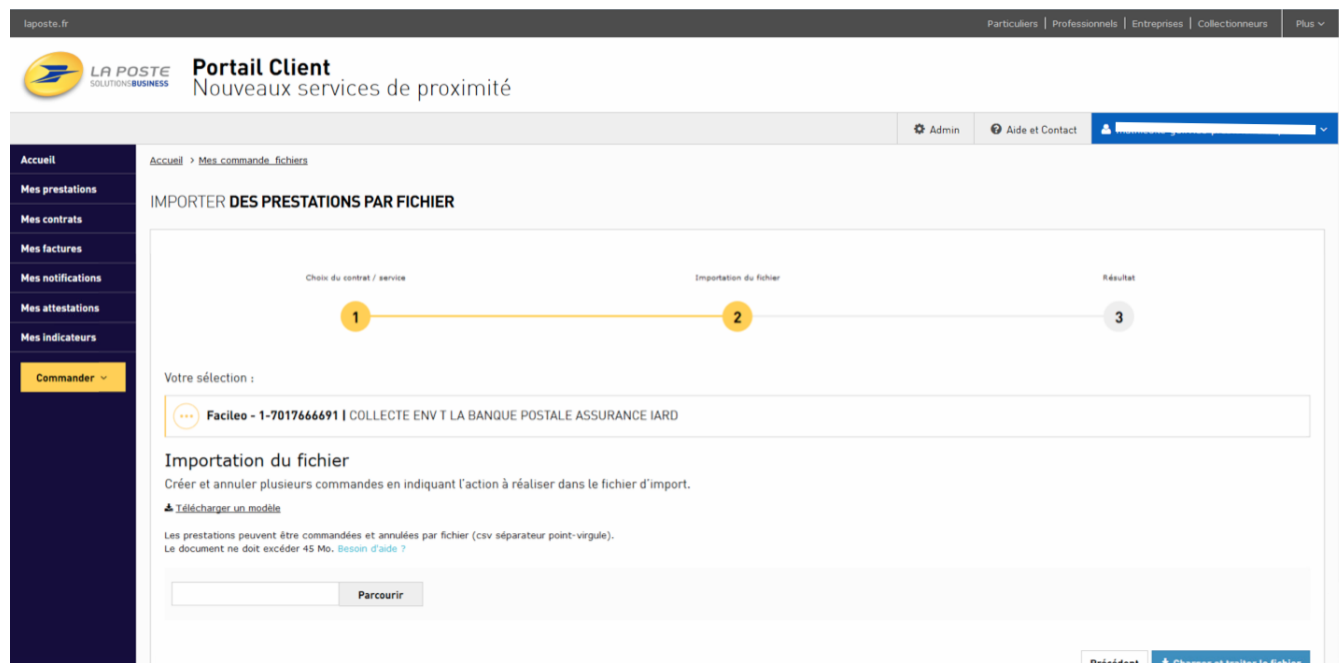


FIGURE 12 – Prise de commande via CSV 1

La particularité de ce canal de prise de commande correspond au fait que chaque ligne du fichier csv (cf. Figure 13) va ensuite générer une commande. Il est donc plus aisé d'utiliser ce moyen si l'utilisateur veut réaliser plusieurs commandes, comme c'est le cas parfois avec plusieurs milliers de lignes dans un même fichier csv...

exemple_1-701766691-faciles_ctact_tqo.csv [Lecture seule] - Microsoft Excel											
<div><div>FichierAccueilInsertionMise en pageFormulesDonnéesRévisionAffichageCompléments</div><div><div>Calibre</div><div>11</div><div>A</div><div>B</div><div>C</div><div>D</div><div>E</div><div>F</div><div>G</div><div>H</div><div>I</div><div>J</div><div>K</div><div>L</div><div>M</div><div>N</div><div>O</div><div>P</div><div>Q</div><div>R</div><div>S</div><div>T</div><div>U</div><div>V</div><div>W</div><div>X</div><div>Y</div><div>Z</div></div><div><div>Copier</div><div>Copier</div><div>Coller</div><div>Reproduire la mise en forme</div><div>Presses-papiers</div><div>Police</div><div>Alignement</div><div>Nombre</div><div>Mise en forme conditionnelle</div><div>Mettre sous forme de tableau</div><div>Style</div><div>Normal</div><div>Insatisfaisant</div><div>Neutre</div><div>Satisfaisant</div><div>Avertissement</div><div>Calcul</div><div>Insérer</div><div>Supprimer</div><div>Format</div><div>Somme automatique</div><div>Remplissage</div><div>Effacer</div><div>Trier et Rechercher</div><div>Filtrer et sélectionner</div><div>Édition</div></div></div>											
E18	A	B	C	D	E	F	G	H	I	J	K
1	Action	Date de début souhaitée	Référence Bénéficiaire	Nom Prénom ou Raison sociale	Lieu de remise N° ou Boîte aux lettres - Couloir-Escalier	Numéro et libellé de voie	Complément commune ou service postal	Code postal	Localité	Indication d'accès au bénéficiaire	Indications compl
2	CREER ANNULER JJ/MM/AAAA										

FIGURE 13 – Prise de commande via CSV 2

### 2.3.2.3 Suivi de commandes

Pour finir cette première présentation des fonctionnalités majeures de l'application, il est possible de suivre les commandes passées depuis le site web.

En effet, lorsque l'utilisateur clique sur les boutons “*consulter*” (cf. Figure 9 numéro 5) il arrive sur un récapitulatif des commandes qu'il a pu passer comme on peut le voir sur la figure ci-dessous.

## MES PRESTATIONS

Suivi de commande

Suivi de réalisation

Rechercher mes commandes

Exporter

Annuler

Filtres en cours : 1-6986230711

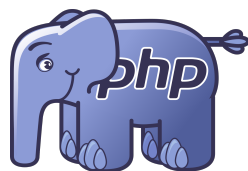
Recherche

<input type="checkbox"/>	N° de contrat	Service	Date de début	Bénéficiaire	Réf. Bénéficiaire	CP	Statut	<a href="#">Filtrer</a>
<input type="checkbox"/>	1-6986230711	LIVRAISON A LA PORTE	03/03/2017	MASLOW LIVP M01	MASLOW_LIVP_M01	29200	Terminé	
<input type="checkbox"/>	1-6986230711	LIVRAISON A LA PORTE	03/03/2017	MASLOW LIVP M02	MASLOW_LIVP_M02	29200	Terminé	
<input type="checkbox"/>	1-6986230711	LIVRAISON A LA PORTE	06/03/2017	MASLOW LIVP M03	MASLOW_LIVP_M03	29200	Terminé	
<input type="checkbox"/>	1-6986230711	LIVRAISON A LA PORTE	06/03/2017	MASLOW LIVP M04	MASLOW_LIVP_M04	29200	Terminé	

FIGURE 14 – Suivi de commandes

## 2.4 Aspect technique

Le projet fait partie du pôle “FS+” de l’entreprise, et utilise comme langage principale le *PHP* (cf. Figure 15a) en version 5.X. L’application utilise également *Oracle* (cf. Figure 15b) pour le stockage en base de données ainsi que du *XML* et du *JSON* pour tout ce qui est transfert de données. Pour ce qui est de la gestion des versions, le projet utilise *Git* (cf. Figure 15c) chez Sopra Steria et *SVN* à La Poste (Etant donné que je n’ai pas travaillé avec *SVN*, je ne rentrerai pas dans les détails pour ce logiciel par la suite).



(a) Logo PHP



(b) Logo Oracle



(c) Logo Git

FIGURE 15 – Logos des technologies

### 2.4.1 Les modules

Facteurs Services Plus est séparé en huit modules, déployés sur des machines différentes pour les environnements de production :

- **U0** : base de données **Oracle**, permettant de stocker les données locales de l’application (paramétrages de produits, commandes en cours d’importations, ...)



- **U1** : code métier de l'application
- **U2** : base de données **MySQL Drupal** permettant de stocker les utilisateurs et le paramétrage technique de Drupal
- **U3** : IHM (*back-office*) basé sur le framework **Drupal**. Le front-office est maintenant géré par NSP
- **U4** : proxy Frontal CMS Internet permettant l'accès depuis l'extérieur
- **U5** : proxy Frontal CMS Back Office permettant l'accès depuis les locaux de La Poste
- **U6** : **Web Service (WS)** de prise de commande depuis un SI client
- **U7** : batchs métier permettant d'automatiser certains traitements (*chargement de données en base de données, facturation des prestations, ...*)

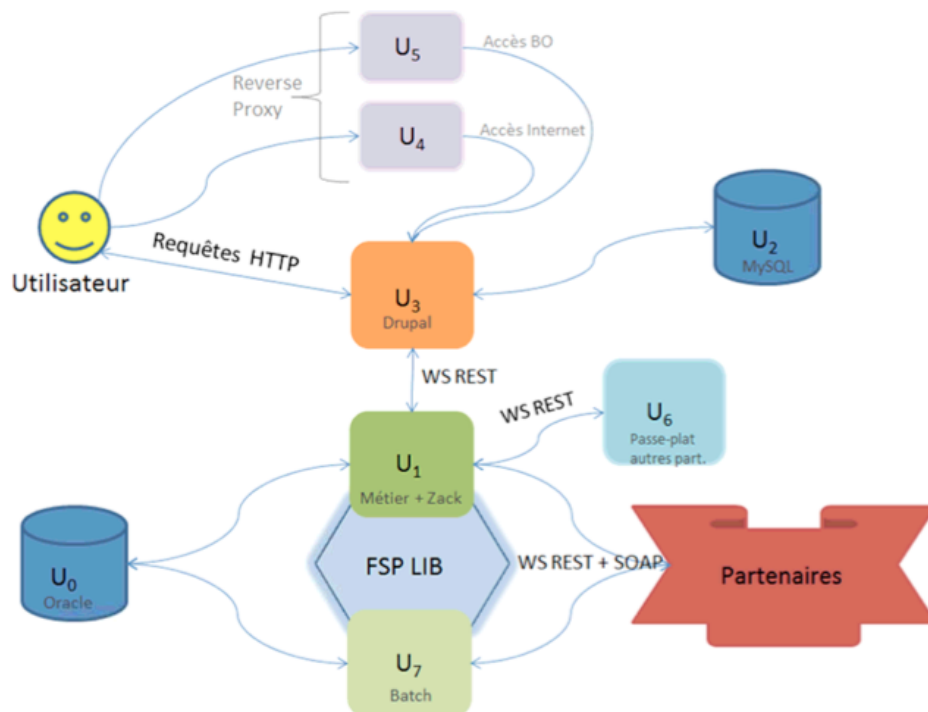


FIGURE 16 – Interactions des modules

Il faut savoir qu'en plus de ces modules, il existe un environnement bouchonné nommé **UA** qui nous permet (*ndlr : l'équipe de développeurs*) de pouvoir travailler en local à Sopra Steria. En effet, nous ne sommes pas situés au sein des locaux La Poste. De ce fait, nous n'avons pas les accès nécessaires afin de pouvoir interroger les différents partenaires avec lesquels l'application échange en permanence (*cf. 2.4.4 Web-services et partenaires*).

Une solution d'externalisation a donc été mise en place (*dans le cadre de MESSI*) en début de projet afin de reproduire au plus proche le comportement initial côté La Poste, c'est ce que l'on appelle les "*bouchons*". Ce sont des fichiers au format *XML* qui simulent l'ensemble des flux comme si nous travaillions en direct avec nos partenaires.

Cette solution permet aux développeurs de pouvoir avancer sans problème. Cependant je pense que ce n'est pas la solution optimale et qu'il faudrait avoir accès aux vrais partenaires, car

durant mon stage j'ai plusieurs fois rencontré des soucis de cohérences de données (*par exemple la balise contenant le numéro de téléphone du client n'était plus à la même place dans le XML, il a donc fallu changer la structure du fichier...*).

En effet, j'ai à plusieurs reprises mis en place des évolutions qui fonctionnaient en environnement bouchonné, mais pas en environnement connecté... Ces problèmes sont dus au fait que nos partenaires modifient des données sans que nous en soyons avertis ou que nous ne répercutons pas les changements en local... De ce fait, j'ai parfois dû corriger (*et donc perdre du temps*) mon code et l'environnement bouchonné afin que cela fonctionne correctement sur les deux environnements.

Au cours de mon stage j'ai principalement réalisé des évolutions et corrections d'anomalies sur le code métier de l'application, c'est-à-dire **le module U1** que nous allons voir plus en détail dans la partie suivante.

## 2.4.2 Le module U1

Comme spécifié dans les parties précédentes, le module U1 correspond au coeur même du projet. Je vais ici vous présenter quelques éléments clés de celui-ci.

### 2.4.2.1 Framework Zack et Zend

Le projet utilise un framework nommé Zack. C'est un framework PHP développé par La Poste qui est très utilisé pour la partie back-end des projets internes au groupe. Il est une surcouche du framework Zend qui est lui OpenSource et orienté objet.



FIGURE 17 – Logo Zack



FIGURE 18 – Logo Zend

Le framework Zack conserve la plupart des fonctionnalités de Zend, comme le patron **Modèle-Vue-Contrôleur (MVC)** ou le **Data Access Object (DAO)** utilisé sur FS+ tout en rajoutant une couche d'abstractions et de spécifications de certaines fonctionnalités afin de simplifier le développement.

Il permet également une gestion des traitements automatisés (*batches*) et simplifie également les étapes de déploiement et de configuration en les automatisant au maximum.

### 2.4.2.2 Zack Manager

Le framework Zack est fourni avec un manager (*qui se loge dans la barre de tâche du système, cf. Figure 19*) et qui permet de réaliser différentes actions, comme le build total du projet, la réinitialisation de la base de données, la génération de classe PHP à partir de données stockées en base, ...

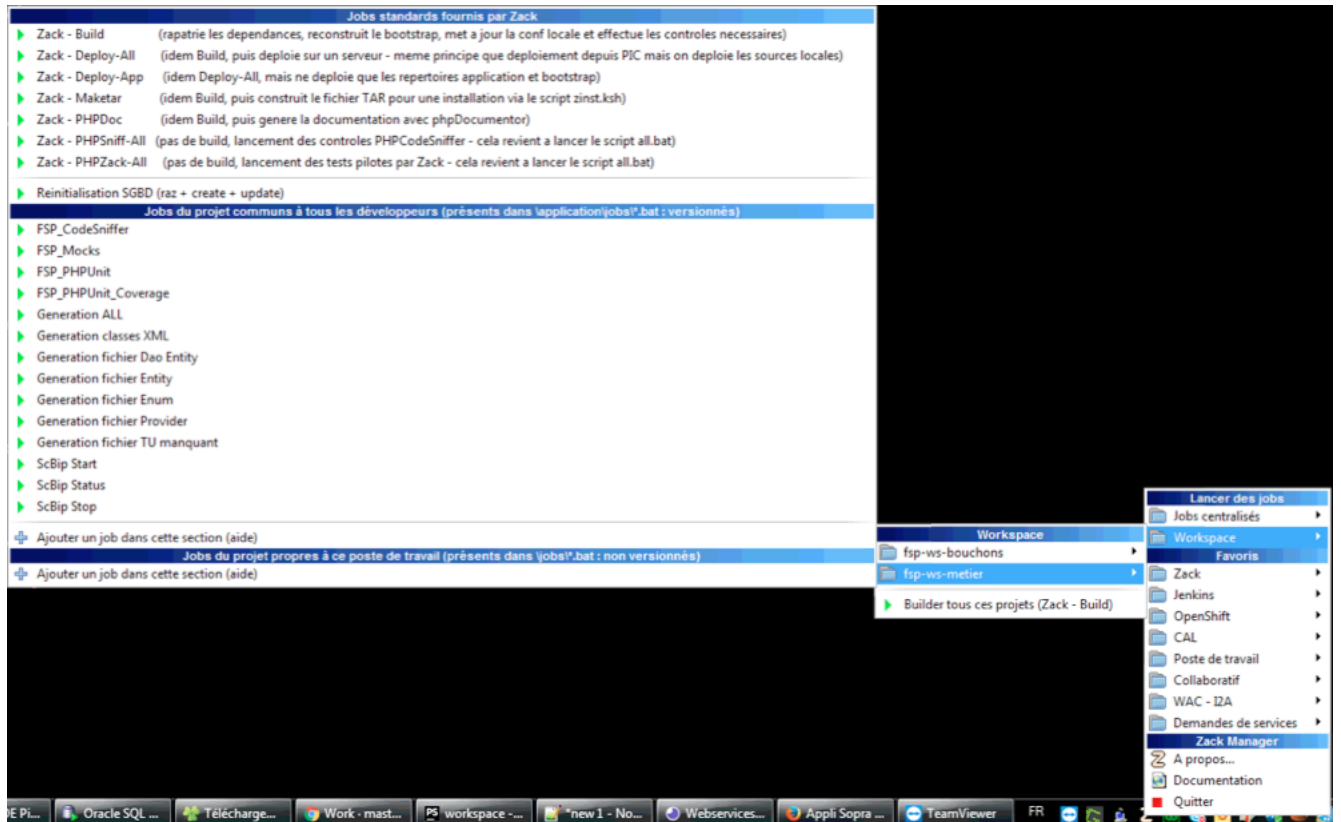


FIGURE 19 – Zack Manager

Cet outil est simple d'utilisation mais très utile, je m'en sers de manière quotidienne afin de régénérer certaines classes du code PHP (*comme les énumérations ou bien les DAO*) lorsque je rajoute ou modifie des champs en base de données lors de mes différentes tâches, par exemple. Cela me fait gagner du temps car je n'ai pas besoin de lancer des scripts à la main pour réaliser toutes ces opérations fastidieuses.

### 2.4.2.3 Designs pattern et architectures principales

#### Modèle-Vue-Contrôleur (MVC)

L'architecture MVC est utilisée par Zack car elle permet de séparer en trois parties le code afin de le rendre plus compréhensible et maintenable, mais également car ce n'est pas FS+ qui gère le front-end de l'application, mais une autre équipe (NSP). Ayant déjà utilisé cette architecture de découpage, cela m'a permis dès le début de pouvoir comprendre plus facilement la logique du découpage, et par la suite, cibler dans quelles classes je devais apporter les évolutions ou corrections.

#### Data Access Object (DAO)

Le patron DAO (*objet d'accès aux données*) est un patron de conception qui permet de regrouper l'accès aux données persistantes dans des classes à part, plutôt que les mélanger aux classes métiers. Ainsi, ce patron permet de faire abstraction de la façon dont les données sont stockées au niveau des objets métiers.

## Services

L'application utilise également des **services** qui lui permet de faire le lien entre les objets métiers et les objets DAO.

Durant mon stage, j'ai du créer plusieurs méthodes au sein de différents services lorsque ces dernières (*ndkr : les méthodes*) pouvaient être amenées à être réutilisées à d'autres endroits dans le code par la suite.

## Data Transfer Object (DTO)

Le patron DTO est un patron de conception qui permet d'harmoniser une donnée. En effet, l'application communique avec de nombreux partenaires (*cf. 2.4.4 Web-services et partenaires*), de ce fait, nous recevons des données sous différents formats (*JSON, XML, formulaires internet, ...*).

Ce design pattern permet une factorisation et une adaptation rapide, il suffit simplement de créer une méthode spécifique en entrée pour chaque partenaire afin d'avoir en sorties des objets identiques et pouvoir utiliser des méthodes plus générales ensuite.

L'avantage de son utilisation réside dans le fait que si l'application travaille avec un nouveau partenaire, il suffit simplement de créer une méthode d'harmonisation de la donnée et le traitement sera ensuite identique.

### 2.4.3 Git

Le logiciel *Git* est utilisé sur l'application afin de gérer les versions de développement. En effet, chaque version correspond à une branche précise, par exemple la branche **develop** correspond au développement effectué en *local* qui est ensuite envoyé sur des branches de **release** qui nous permet d'assurer des tests en environnements connectés (*avec les appels aux vrais partenaires et non plus les bouchons*), ...

Il a été mis en place un flow à respecter par chaque développeur sur le Git afin d'assurer une efficacité, une cohérence des données et d'éviter au maximum les conflits.

Voici quelques règles à respecter en fonction de ce que nous devons faire sur l'application :

- - Si c'est une évolution, nous devons partir de la branche **develop** et créer une branche **fsppor-xxxx** correspondant au numéro de l'évolution
- - Si c'est un correctif d'une ancienne branche, il faut partir de la branche **fsppor-xxxx** en question
- - Si c'est un correctif de bug détecté en environnement connecté, alors il faut partir de la branche **release-x** correspondants à la version de l'application sur laquelle le bug a été détecté et créer une **fsppor-xxxx**
- ...

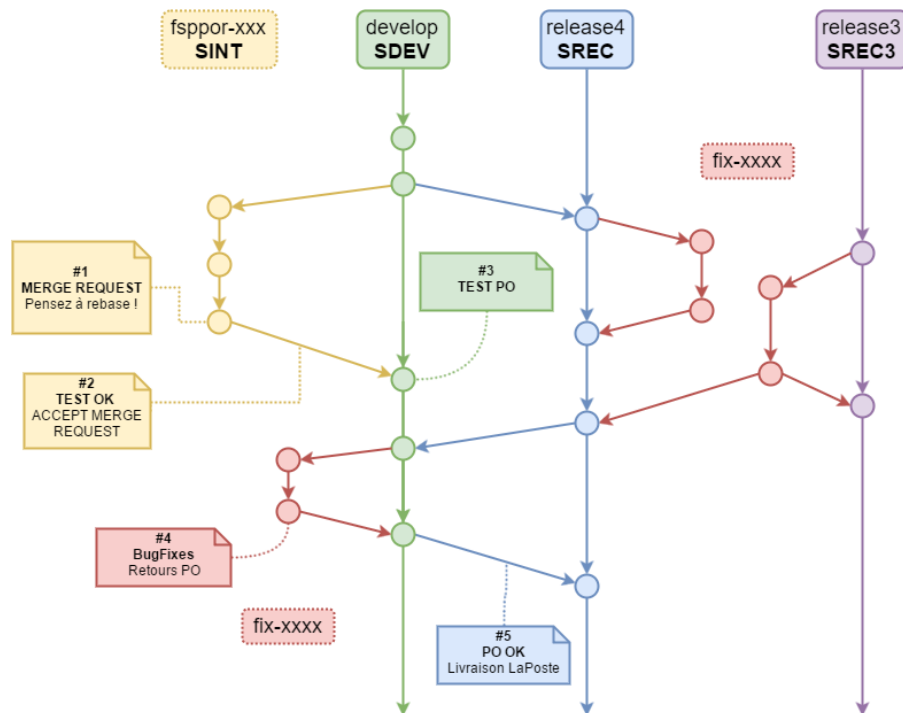


FIGURE 20 – Git flow

Tout au long du stage, j'ai pu m'appuyer sur les précieux conseils des autres développeurs de mon équipe afin de monter en compétences sur Git. En effet, cet outil est très puissant et est utilisé dans toutes les entreprises aujourd'hui.

J'avais déjà eu l'occasion de l'utiliser sur de petits projets durant ma scolarité, mais c'est durant ce stage que je me suis rendu compte de la puissance qu'il pouvait offrir. Il permet notamment une gestion des branches très pointues afin de ne (presque) jamais entrer en conflit avec les autres membres, il est très rapide car il ne recharge que les dernières modifications (*contrairement à SVN qui lui recharge toutes les modifications à chaque action*).

De plus, grâce au système de *merge request* mis en place sur le projet, j'ai appris et compris l'intérêt de faire des *revues de code*<sup>5</sup>. En effet, lorsque j'ai fini un développement je crée une *merge request* qui va permettre à un sachant (*membre de l'équipe ayant une bonne connaissance du projet*) d'analyser mon code et de me faire un retour si celui-ci ne respecte pas certaines normes. Une fois ce retour fait, le testeur peut alors tester et valider ma merge request afin d'envoyer le code sur la branche principale du git. Ce principe de merge request permet d'avoir du code propre et optimisé.

Concernant les messages de commit, un code est à respecter afin de pouvoir s'y retrouver rapidement par la suite comme le montre la figure ci-dessous.

5. Examenation du code (*code review*) afin de s'assurer du respect d'un ensemble de règles de programmation.



FIGURE 21 – Template d'un commit

J'ai également compris l'intérêt d'avoir des messages clairs et précis pour les commit. Cela permet de comprendre en quelques mots l'évolution ou la correction mais également de faire des recherches optimales dans les branches.

#### 2.4.4 Web-services et partenaires

Un web-service est un concept permettant de mettre en place un moyen de communication entre deux applications via des méthodes HTTP avec des paramètres définis et qui renvoient des données.

Il existe deux catégories différentes de web-services :

- **SOAP** : c'est un protocole qui définit les informations sur le message, sa destination, ainsi que le format des données
- **REST** : c'est une architecture et non un protocole qui met en place des requêtes simples basées sur le protocole HTTP, permettant l'appel d'une méthode via l'URL

Chacune des méthodes présente des indépendances. En effet, les WS REST sont plus légers et présentent plus d'indépendance entre le client et le serveur, mais utilisent seulement le protocole HTTP ce qui limite un peu son utilisation. . . En revanche, le protocole SOAP lui est plus ouvert, mais est plus lourd. . .

Un grand nombre de données gérées par l'application FS+ ne sont pas stockées en interne dans la base de données, mais sont envoyées et récupérées depuis des partenaires via ces web-services. Ces partenaires font partie du SI La Poste et servent par exemple pour la gestion des prestations (*Gesprest*), des contrats (*Meridien*), des produits (*Cabestan*), la facturation (*REMX*), . . .

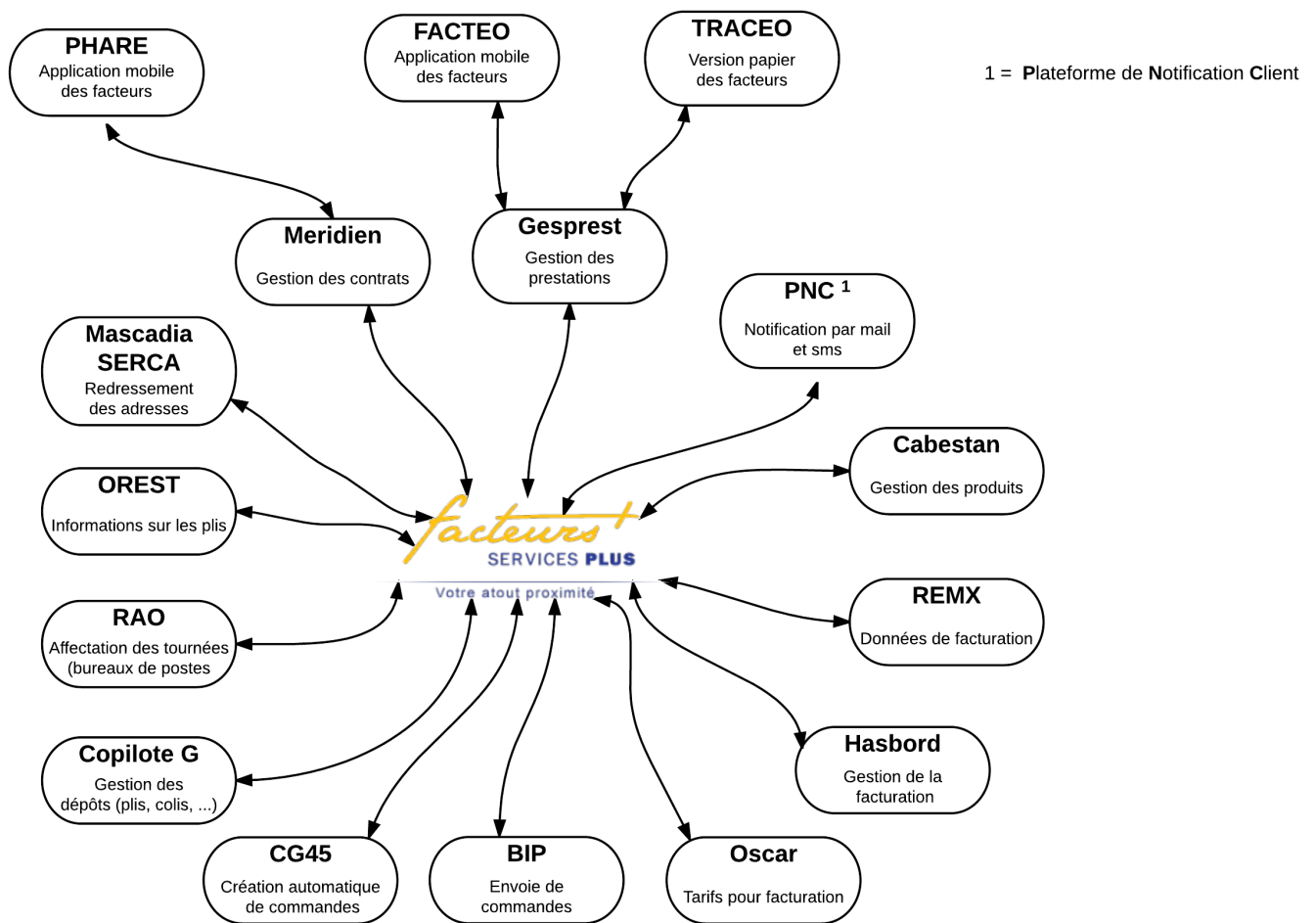


FIGURE 22 – Ensemble des partenaires

On peut observer sur l'image ci-dessus que le nombre de partenaires pour l'application FS+ est important et qu'elle se trouve au milieu, ce qui entraîne beaucoup de communication. Il est donc important que ces dernières soient le plus efficace possible, c'est-à-dire avoir un code optimisé et utiliser le type de WS (*SOAP ou REST*) offrant les meilleures performances en adéquation avec la demande.

## 3 Conduite du projet

Au cours de cette partie, je vais vous présenter l'équipe qui a la particularité d'avoir un aspect *nearshore*<sup>6</sup>, ainsi que les méthodes de gestion de projet appliquées au projet.

### 3.1 Présentation des équipes

L'équipe du projet FS+ est intégrée en deux parties. Il y a d'une part l'équipe *française* que j'ai intégrée et l'équipe *barcelonaise*. Cette particularité *nearshore* présente un aspect avantageux d'un point de vue financier pour Sopra Steria mais je trouve personnellement que cela est un frein pour les membres de chaque équipe...

En effet, les horaires espagnols sont décalés de nos horaires françaises (*par exemple : les pauses déjeuner en France se font entre 12 et 14h alors qu'en Espagne entre 13h30 et 15h30, les 'horaires intensifs' adoptés par les espagnols durant tout l'été leur permettant de faire 8h - 15h alors que nous faisons ici 9h - 18h, ...*), même si je pense que cela peut être un atout pour Sopra Steria vis-à-vis d'une disponibilité plus élargie, cela peut rendre parfois le dialogue plus difficile et retarder les tâches. Cela peut être une bonne chose de scinder les équipes en deux, mais je pense qu'une communication irréprochable doit être mise en place, ce n'est malheureusement pas toujours le cas.

#### 3.1.1 France

Dès mon arrivée dans les locaux de l'entreprise j'ai été accueilli et intégré au sein de l'équipe *française* à Montpellier sans aucun problème. Tous les membres du projet n'ont pas hésité à me porter de l'attention, me donner des conseils, me proposer des formations durant mes premiers jours sur différents aspects importants du projet, ... . J'ai eu l'opportunité d'intégrer l'équipe en même temps qu'un membre de l'équipe espagnole (*présent dans les locaux Français pour une formation*) ce qui nous a permis de nous entraider et de faire connaissance.

À mon arrivée, le projet avait eu une forte activité durant les mois précédents portant le nombre de développeurs à sept, le nombre de fonctionnels à trois, un chef de projet adjoint, un directeur de surveillance et un responsable commercial.

Cependant, l'activité a vite été restreinte du fait que La Poste n'ait plus assez de budget pour réaliser de nouvelles évolutions. Nous ne sommes aujourd'hui plus que trois développeurs, un fonctionnel, un chef de projet, un chef de projet adjoint, un directeur de surveillance et un responsable commercial.

La nouvelle fut difficile à comprendre pour moi dans un premier temps, en effet, cela faisait seulement quelques semaines que j'étais présent et que le projet se déroulait sans problèmes et on nous annonce une restriction budgétaire et une très forte diminution des équipes et de l'activité... Cependant, mes chefs n'ont pas hésité à me rassurer dès le début afin que cela ne me gêne pas pour la suite de mon stage.

#### 3.1.2 Barcelone

À mon arrivée, l'équipe barcelonaise était composée de quatre développeurs, un fonctionnel ainsi qu'un chef de projet. Cependant, avec la restriction budgétaire ils sont maintenant trois développeurs et un fonctionnel.

---

6. Délocaliser une activité économique, dans une autre région du même pays ou dans un pays proche.



## 3.2 Gestion de projet

### 3.2.1 SCRUM

La gestion du projet se base sur la méthode *SCRUM* (qui est une méthode agile<sup>7</sup>) qui offre un cadre de travail permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible. La méthode se base sur des *Sprint* qui durent ici trois semaines embarquant chacun un certain nombre de fonctionnalités chiffrées en **Story Point (SP)** et au cours duquel une version “*terminée*” et utilisable peut être livrée.



FIGURE 23 – Diagramme d’explication d’un Sprint (*méthode SCRUM*)

Cette méthode se veut *légère, simple à comprendre et difficile à maîtriser* et repose sur trois piliers principaux :

- **Transparence** : on communique tout au client, que ce soit l’avancement positif/négatif, les problèmes, ... Cela permet d’avoir un meilleur rebondissement par la suite
- **Inspection** : on vérifie de façon quotidienne l’avancée et les écarts possibles avec les objectifs définis au préalable
- **Adaptation** : on s’adapte en fonction de la demande, des retours, des inspections, ... afin d’être le plus efficace possible

#### 3.2.1.1 Rôles

L’équipe SCRUM comporte trois rôles principaux, à savoir :

- **Product Owner (PO)** : propriétaire du produit, c’est lui qui définit ce que l’équipe de développeur doit faire
- **L’équipe de développement**
- **Scrum master** : responsable de la bonne compréhension et mise en pratique de la méthode Scrum. C’est le *maître de mêlée*

---

7. Groupes de pratiques de pilotage et de réalisation de projets.

En plus des rôles ci-dessus, un nouveau rôle a été défini pour le projet, celui de **Product Owner Proxi** (*POP, interne à Sopra Steria*), qui joue le rôle de fonctionnel au sein de l'équipe, et qui fait le lien avec le PO côté client.

Il faut également savoir qu'il n'y a plus de scrum master aujourd'hui sur le projet, c'est à nous (*ndlr : toute l'équipe*) de respecter les règles et de s'autoresponsabiliser, pratique difficile à respecter et qui je trouve pose parfois des problèmes...

### 3.2.1.2 Événements

Durant chaque Sprint s'inscrit un certain nombre d'événements permettant d'inspecter et d'adapter le travail nécessaire au bon déroulement de celui-ci. Nous allons voir plus en détail les principaux événements :

#### Sprint Planning :

Réunion précédant le début du sprint afin de définir le travail futur à élaborer. La charge de travail dépend de la vitesse/capacité de l'équipe de développement.

#### Mêlée quotidienne :

La mêlée quotidienne (*aussi appelée Daily Scrum*) est un événement limité à environ 15 minutes au cours duquel l'équipe de développement synchronise ses activités (*avancement, difficultés, ...*) et crée un plan pour les prochaines 24 heures.

En début de projet, j'avais une certaine appréhension de ces réunions quotidiennes, car je pensais que c'était un endroit pour *rendre des comptes* et où il fallait montrer que l'on avance toujours... Cependant ce n'est pas du tout cela, c'est un moment d'échange bref, mais qui permet de se faire aider si besoin ou de résoudre les problèmes urgents.

#### Poker planning :

Réunion qui regroupe l'équipe de développeurs ainsi que les Product Owner Proxi (*POP*) lorsqu'il y a de nouvelles fonctionnalités à chiffrer.

Durant cet événement, les POP présentent les évolutions attendues par le client à l'équipe de développeurs qui analyse et définit la complexité (*du développement et des tests à réaliser*) au travers de cartes scrum.

0	1/2	1	2	3	5
0	1/2	1	2	3	5
8	13	20	40	100	?
8	13	20	40	100	?

FIGURE 24 – Cartes scrum

Nous nous basons sur la complexité de la carte à valeur "1" qui correspond à l'insertion d'un nouvel élément dans une liste déroulante (*en prenant en compte le développement et les tests en partant du stockage en base de données jusqu'à l'affichage dans l'IHM*). Si jamais il y a des désaccords sur les points attribués, les personnes les plus

écartées de la moyenne sont amenées à justifier leurs choix afin de trouver le meilleur compromis.

J'ai agréablement été surpris de l'importance que me porta l'équipe dès le premier poker planning, en effet, même si je n'avais presque encore aucune connaissance du projet techniquement, les membres de l'équipe prenaient en compte mon chiffrage et n'hésitais pas à m'expliquer clairement lorsque j'avais des questions.

### **Revue de sprint :**

La revue de sprint (*également appelée Sprint Review*) est tenue en fin de sprint afin d'inspecter l'incrément réalisé et adapter le Product Backlog<sup>8</sup> si nécessaire.

### **Rétrospective de sprint :**

La rétrospective de Sprint (*aussi appelé Sprint Retrospective*) est une occasion pour l'équipe Scrum de s'inspecter et de créer un plan d'amélioration qui sera mis en place au cours du Sprint suivant.

De manière générale, la méthode SCRUM n'est pas suivie à la lettre sur le projet (*je n'ai par exemple pas assisté aux réunions de revue de sprint et de rétrospective*), premièrement car il n'y a plus de scrum master ce qui ne facilite pas les bonnes pratiques, mais également au vu du manque de budget, nous n'avons plus le temps nécessaire pour faire chacune des réunions citées ci-dessus.

Je pense que cela est une erreur, car il est nécessaire que l'équipe soit bien encadrée, surtout avec l'aspect nearshore. Malgré cela, cette première approche pratique de la méthode SCRUM reste efficace et m'a permis de comprendre le fondement et l'utilité de celle-ci.

## **3.3 Organisation propre à Sopra Steria**

Sopra Steria possède également des processus de gestion et d'organisation propre à l'entreprise :

### **3.3.1 V1**

Réunions hebdomadaires avec toute l'équipe afin de discuter précisément du projet dans son intégralité avec par exemple les prochains rendez-vous importants avec le client, les congés/entrées/sorties des membres de l'équipe, l'état des anciens sprint (*livré, problèmes, retours, ...*) avec l'historique de qui a fait quoi, ...

Cette réunion ne me paraissait pas primordiale en début de projet, cependant, après avoir pris du recul, je pense que c'est un élément clé. En effet, cela permet de faire un suivi hebdomadaire des problèmes et de la charge de travail qu'il reste jusqu'à la fin du sprint, d'échanger sur la nécessité de l'apport d'un nouveau collaborateur sur le projet ... C'est un moment propice à l'échange et qui permet de faire avancer tout le groupe.

### **3.3.2 P1 et CRA**

Le P1 et le **C**ompte **R**endu d'**A**ctivité (*CRA*) permettent de renseigner les tâches réalisées ainsi que le temps passé sur celles-ci.

Le P1 permet notamment aux chefs de projets de faire des statistiques sur la productivité et de réajuster le budget ou les ressources de l'équipe alors que le CRA permet lui d'être rémunéré.

---

8. Ensemble des items sélectionnés pour le Sprint.

### 3.3.3 JIRA

Jira est un logiciel qui est représenté sous forme de tableau ici et qui permet de réaliser la gestion, le suivi de projet mais également un système de suivi de bugs et de gestion des incidents. Ce tableau comporte six colonnes :

- **À faire** : ce sont les tâches (*nommée FSPPOR ici*) qui n'ont pas été commencées pour le sprint en cours
- **En cours** : lorsqu'un développeur commence à travailler sur une tâche, il doit la déplacer dans cette colonne afin que plusieurs développeurs ne travaillent pas dessus en simultanée
- **À tester par pair** : une fois le développement réalisé, la tâche passe dans cette troisième colonne afin qu'un autre développeur teste (*en local*) et s'assure que tout fonctionne
- **À recetter par POP** : une fois le test par pair validé, c'est au tour des POP de réaliser le test en environnement connecté
- **À recetter métier** : si le précédent test est validé, c'est au tour du métier (*celui à qui est destiné l'application*) de tester et valider que l'évolution correspond à ce qu'il souhaitait initialement
- **Terminé** : cette dernière colonne permet de s'assurer que toute la chaîne est valide

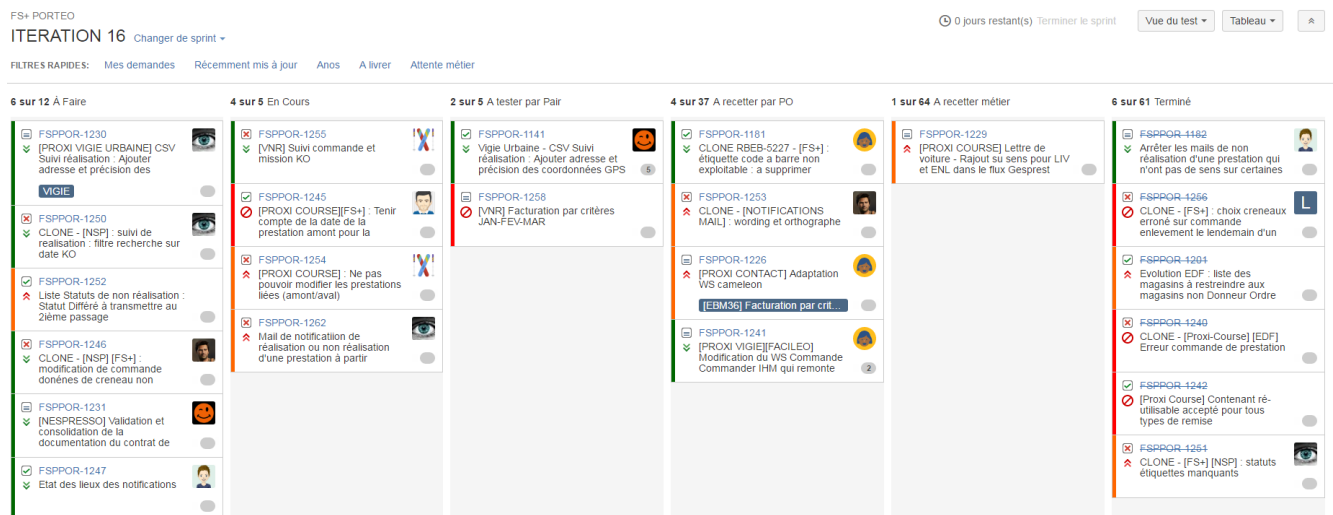


FIGURE 25 – Aperçu d'un tableau JIRA pour le sprint 16

Chaque sprint possède son propre tableau afin de pouvoir suivre l'avancée de ce dernier.

C'est un outil très puissant que j'aurais utilisé avec plaisir durant mes années d'études sur mes différents projets afin d'optimiser leurs suivis.

### 3.3.4 Sentinelle du MCO<sup>9</sup>

Le MCO est la correction et l'assistance aux clients lors de problèmes sur l'application en production. Par exemple, lorsqu'un bug est trouvé, que certaines parties de l'application ne fonctionnent plus, ce sont les équipes en charge du MCO qui s'occupent des corrections.

9. Maintient en Condition Opérationnelle

Le MCO est donc une partie critique du cycle de vie de l'application, et une réactivité importante est nécessaire afin de s'assurer du bon fonctionnement de l'application au quotidien.

De ce fait, chaque semaine et à tour de rôle, nous définissons (*lors du V1, cf. 3.3.1*) qui sera responsable de la prise en charge et de la répartition des tickets MCO. Cette répartition doit être assurée de 8h30 à 18h30, nous définissons donc une personne pour le matin et une autre pour l'après-midi.

Il est parfois difficile d'assurer le rôle de sentinelle, en effet, lorsque je suis sur une tâche précise (*évolution, anomalies, audio, ...*) et que je me focalise dessus pour la résoudre au plus vite, il faut souvent regarder les mails et faire attention de ne rien laisser passer sous peine d'avoir des pénalités car la correction de problèmes en production doit se faire le plus vite possible afin de ne pas impacter les utilisateurs.

## 4 Partie technique

Au cours de cette dernière partie, je vais rentrer plus en détail sur des tâches qui remplissent mes journées avec notamment le déroulement d'une évolution, une correction d'anomalie et une phase de **V**érification de **N**on **R**égression (*VNR*).

Durant mes quatre mois de stage, j'ai été amené à gérer des évolutions (*4 mineures et 6 majeures*), des anomalies (*3 mineures et 5 majeures*) mais également des retours de MCO (*2 mineurs et 1 majeur*). J'ai également pu réaliser bon nombre de tests (*5 mineurs et 9 majeurs*) me permettant d'en apprendre les bonnes pratiques ainsi que trois phases de VNR.

Grâce à mon implication et mon sérieux sur le projet, j'ai très vite été considéré comme un membre à part entière de l'équipe ce qui m'a permis de pouvoir faire des évolutions, des corrections d'anomalies, des retours de MCO ou bien des tests, peu importe leurs criticités. Ces différents travaux diversifiés m'ont permis de monter en compétence que ce soit d'un point de vue technique, fonctionnel ou bien vise à vis des tests.

### 4.1 Déroulement d'une évolution

#### 4.1.1 Initialisation

La première étape d'une évolution se situe dans le Backlog (*ensemble des items sélectionnés pour le sprint*) que nous chiffrons lors des Poker Planning. Une fois le chiffrage réalisé, elle apparaît alors dans le tableau JIRA avec toutes les informations la concernant, notamment les spécifications fonctionnelles (*cf. Figure 26*) que nous (*ndlr : les développeurs*) devons suivre.



## Motif de non Réalisation : adaptations

Modifier

Commentaire

Attribuer

Suite

Recetter

Rouvrir la demande

Fermer la demande

## Informations

Type: **1** ☒ User Story  
 Priorité: **2** ☒ Mineur  
 Affecte la/les version(s): Aucune  
 Composants: Aucune  
 Étiquettes: Aucune  
 Story Points: **3**  
 Sprint: ITERATION 15 **4**

Etat: **LIVRÉE**  
 Résolution: Corrigé  
 Version(s) corrigée(s): FS+ 27 avril 2017

Description **5**

FS+ n'est pas synchro avec les partenaires GESPREST/FACTEO/REMX suite à la [FSPOR-904](#).

Il convient de faire les aménagements suivants :

- transmettre systématiquement les types 1, 2 et 3
- transmettre la balise passage pour les types 1 et 2
- remettre l'ancien libellé pour le type 3 et rendre paramétrable le switch vers les nouveaux libellés pour Proxi Course (variable applicative "Ancien REMX" à 1, puis qui passera à 0 ou suppression de la variable applicative

Si le produit Cabestan est paramétré Avec l'option 2ème passage **6**

```
<ul class="ListeStatuts">
<li type="1" passage="1">Différé Raison La Poste</li>
<li type="1" passage="1">Différé Raison Exceptionnelle</li>
<li type="2" passage="2">Raison La Poste</li>
<li type="2" passage="2">Raison Exceptionnelle</li>
Si "Ancien REMX" à 1 ou Produit Non PORTEO alors
<li type="3">Anomalie du fait du bénéficiaire</li>
Sinon si prestation principale
<li type="3">Anomalie du fait du destinataire</li>
Sinon si prestation secondaire
<li type="3">Anomalie du fait du commerçant</li>
</ul>
```

Si le produit Cabestan est paramétré Sans l'option 2ème passage

```
<ul class="ListeStatuts">
<li type="1">Différé Raison La Poste</li>
<li type="1">Différé Raison Exceptionnelle</li>
<li type="2">Raison La Poste</li>
<li type="2">Raison Exceptionnelle</li>
Si "Ancien REMX" à 1 ou Produit Non PORTEO alors
<li type="3">Anomalie du fait du bénéficiaire</li>
Sinon si prestation principale
<li type="3">Anomalie du fait du destinataire</li>
Sinon si prestation secondaire
<li type="3">Anomalie du fait du commerçant</li>
</ul>
```

FIGURE 26 – Exemple d'une JIRA pour une de mes évolutions

On observe dans la partie haute de l'image ci-dessous que c'est une évolution (cf. 1, "User Story") de type mineure (cf. 2), qu'elle a une complexité de 3 (cf. 3), c'est-à-dire que le développement et les tests doivent prendre environ trois/quatre jours et qu'elle est affectée au sprint 15 (cf. 4).

La partie basse de l'image (cf. 5 description) correspond aux spécifications. Il est décrit ici que nos données et celles de nos partenaires (Gespret, Facteo et REMX) ne sont pas synchronisées, il faut donc que je fasse les modifications suivantes :

- transmettre systématiquement les types 1, 2 et 3
- transmettre la balise passage pour les types 1 et 2
- remettre l'ancien libellé pour le type 3 et rendre paramétrable le switch vers les nouveaux libellés pour Proxi Course (variable applicative "Ancien REMX" à 1, puis qui passera à 0 ou suppression de la variable applicative

La partie encore en dessous (cf. 6) correspond à des exemples de ce que nous devons envoyer à nos partenaires une fois l'évolution effectuée.

### 4.1.2 Analyse et développement

J'ai dans un premier temps fait une recherche dans le code afin de trouver où se situe l'existant pour les types de motifs de non-réalisation (*liste des statuts*) afin de réaliser une analyse et comprendre comment fonctionne le code afin d'y apporter les modifications nécessaires. C'est lors de cette phase d'analyse que je me suis rendu compte que la balise "*passage*" n'avait jamais été utilisée dans l'application et par conséquent n'était pas récupérée dans l'objet que j'allai devoir manipuler. De ce fait, j'ai donc commencé par faire remonter l'information en ajoutant cette balise dans le fichier XSD<sup>10</sup> (cf. Figure 27).

```
<xsd:element name="ListeStatuts" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="li" minOccurs="1" maxOccurs="5">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="type" type="xsd:integer"/>
              <xsd:attribute name="passage" type="xsd:integer"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

FIGURE 27 – Ajout de la balise "passage" dans un fichier XSD

Une fois cette première étape faite, j'ai fait en sorte de rajouter cette balise "*passage*" pour les types 1 et 2 de la manière suivante :

```
964 foreach ($type1 as $type) {
965     $statut =
966         new Fsp_Model_Xml_Gesprest_Commande_FacteurServicesPlus_DonneesComplementaires_ListeStatuts_Li();
967     $statut->setType(1);
968     $statut->setValue($type);
969     // Si le produit Cabestan est paramétré Avec l'option "2ème passage" on lui rajoute l'attribut "passage=1"
970     if ($complementProduit->getSecondPassage() != 0) {
971         $statut->setPassage(1); 1
972     }
973     $listeStatuts[] = $statut;
974 }
```

FIGURE 28 – Ajout de la balise "passage" dans les types

Comme on peut le voir sur la figure ci-dessus, j'ai rajouté les lignes 969, 970, 971 et 972 afin que la balise prenne la valeur 1 (cf. 1) si l'option 2<sup>e</sup> *passage* est activée. J'ai fait de même pour le type 2, seulement cette fois avec "2" à la ligne 971.

Pour finir cette évolution, notre partenaire REMX n'était pas prêt à recevoir les nouveaux libellés que nous utilisons sur l'application. Il a donc fallu que je crée une variable applicative<sup>11</sup> afin de pouvoir terminer l'évolution mais ne pas impacter notre partenaire.

10. Langage de description de format de document XML permettant de définir la structure et le type de contenu d'un document XML. Cette définition permet notamment de vérifier la validité de ce document.

11. Variable utilisée dans l'application afin de définir certaines règles et qui sont modifiables par des administrateurs.



L'ajout d'une variable applicative se fait via une requête SQL qui permet d'insérer une nouvelle ligne en base de données, comme ci-dessous :

```
DECLARE
    tmp NUMBER(3,1);
BEGIN
    SELECT COUNT(VAR_CODE) INTO tmp FROM T_SYS_VARIABLE_VAR WHERE (VAR_CODE = 'Ancien_Remx');
    if tmp = 0 then
        INSERT INTO T_SYS_VARIABLE_VAR (VAR_CODE, VAR_LIBELLE, VAR_TYPE, VAR_VALEUR, VAR_EDITABLE)
        VALUES ('Ancien_Remx', 'Permet d'avoir l'ancien libellé type 3 pour les motif de non réalisation',
        'Numerique', '1', '0');
    end if;
END;
```

FIGURE 29 – Ajout de la variable application pour REMX

Une vérification est faite aux lignes 4 et 5 afin de ne pas insérer cette ligne si elle est déjà présente en base de données, cela permet d'éviter les erreurs SQL lors de la mise en production si le script est joué plusieurs fois.

La modification de cette variable applicative se fait via la partie administrateur à l'aide d'un formulaire.

Je récupère ensuite la valeur de cette variable applicative afin de savoir si je dois mettre les anciens ou les nouveaux libellés comme dans l'exemple de code ci-dessous :

```
1006 $serviceComplementProduit = Fsp_Service_ServiceProvider::getInstance()->getComplementProduit();
1007 // On récupère la valeur de la var. applicative pour savoir si REMX est prêt à recevoir les nouveaux libelles
1008 // 1 = REMX n'est pas prêt à recevoir les nouveaux libelles
1009 // 0 = REMX est prêt
1010 $ancienRemx = intval(Fsp_Dao_DaoProvider::getInstance()->getDbVariableApplicative()->getByCode(
1011     Fsp_Enum_Param_CodeVariableEnum::ANCIEN_REMX
1012 )->getValeur());
1013
1014 if ($ancienRemx === 1 || !$serviceComplementProduit->isPorteo($complementProduit)) {
1015     $statut->setValue(Fsp_Enum_ComplementContrat_listeStatuts::ANO_BENEF);
1016 } else {
1017     if (in_array($complementProduit->getXml()->getTypeProduit(), $sanoDest)) {
1018         $statut->setValue(Fsp_Enum_ComplementContrat_listeStatuts::ANO_DEST);
1019     } elseif (in_array($complementProduit->getXml()->getTypeProduit(), $sanoCom)) {
1020         $statut->setValue(Fsp_Enum_ComplementContrat_listeStatuts::ANO_COM);
1021     }
1022 }
```

FIGURE 30 – Utilisation de la variable REMX dans le code

#### 4.1.3 Plan de test et mise à jour JIRA

Une fois le développement de l'évolution terminé, il faut rédiger un plan de test afin qu'un autre développeur sache comment tester. Cette fiche se fait via un fichier au format markdown<sup>12</sup>, à l'intérieur de celle-ci j'ai notamment précisé les prérequis : la branche sur laquelle il devra se placer afin de récupérer mon code, la génération de la variable applicative et le paramétrage du produit. Ensuite, il devra créer une commande et vérifier que dans le retour les balises correspondent bien aux attentes.

Toutes ces informations sont détaillées au maximum afin de faciliter le test par quelqu'un qui n'aurait aucune connaissance du projet.

12. Langage de balisage léger offrant une syntaxe facile à lire et à écrire.



De ce fait, il m'a simplement fallut rajouter ce test en amont pour ne plus rencontrer ce problème comme sur la figure ci-dessous :

```
'schedule_date' => !is_null($this->getDatePlanification()) ? $this->getDatePlanification()->format(Fsp_Tools_Date::FORMAT_ATOM) : null,
```

FIGURE 33 – Code modifier de l'anomalie

Ensuite le processus ne change pas, j'ai pushé ma branche, j'ai fait la fiche de test et déplacé la tâche dans la colonne "A tester par paire" afin que cela puisse être validé et déployé.

### 4.3 Déroulement de la VNR

La phase de VNR se fait quelques jours avant (*généralement une semaine*) la **Mise En Production (MEP)**<sup>13</sup>. Elle permet de tester tout le périmètre qui va être livré afin de desceller le plus d'anomalies en amont.

Durant celle-ci, j'utilise **HP Quality Center (HPQC)** qui est une solution logicielle de gestion de qualité proposée par HP permettant d'assurer notamment la gestion des tests dans notre cas. Des scénarios sont créés par les POP me permettant de savoir précisément ce que je dois tester (*cf. Figure 34*).

---

13. Etape où on livre les évolutions d'un sprint aux utilisateurs finaux.

Manual Runner: Test Set 01-01 - Prise de commande WS, Test [1]01-01 - Prise de commande WS **1**

Etape	Statut	Date d'exécution	Heure...
Step 1 Prise de c...	No Run	14/06/2017	13:10:25
Step 2 Prise de c...	No Run	14/06/2017	13:10:25
Step 3 Prise de...	No Run	14/06/2017	13:10:25

Description Commentaires

Faire une prise de commande par WS sur une offre Ponctuelle type PORTEO non EDF (Geodis 1768199 1-6812046111 code cabestan 49011) sur SREC3 connecté

Résultat attendu: **4**

Récupérer l'id de la commande et vérifier le flux transmis dans GESPREST (les informations de la commande sont bien reportées dans le flux GESPREST):

- Vérifier que la commande prestation principale est en Modification (balise titre) et que l'id de la commande se retrouve bien dans le numero de depot
- Vérifier que les prestations liées sont bien en creation (balise titre)
- L'enchainement est bien celui de la commande (ex: livraison-collecte-retour)
- Numero d'ordre chrono (2 por principale, 1 pour amont et 3 pour aval si m'me jour que la principale sinon 3)
- contenants correspondent a la commande
- Date et creneaux aussi (voir dans complement contrat pour le detail des creneaux)
- Vérifier le status des etiquettes
- Nature marchandise correspond au parametrage dans le coe cabestan
- Lieu de passage=type de remise (a la porte/BAL/sur le lieu de travail/Point relais)
- les balises code renate et code ROC sont bien

Résultat obtenu: **5**

FIGURE 34 – Scénario de VNR dans HPQC

Sur ce scénario, on observe dans le titre (*cf. 1*) que je teste ici la *prise de commande WS* et qu'il y a trois tests à réaliser (*cf. 2*). L'information au milieu de la figure (*cf. 3*) correspond au jeu de données que je dois utiliser. Le cadre en bas à gauche (*cf. 4*) correspond aux différentes étapes à réaliser et celui à droite aux résultats obtenus *cf. 5*).

Si jamais il y a un problème, je dois créer un *defect*, c'est-à-dire une anomalie via HPQC et la faire remonter dans notre JIRA afin qu'un développeur puisse la traiter (*cf. 4.2 Déroulement d'une anomalie*).

## Bilan

Le stage de dernière année a été pour moi une expérience enrichissante, c'est la première fois que j'en réalisais un au sein d'une ESN. Cette première expérience m'a permis de voir comment était organisée et fonctionnait une entreprise multinationale comme celle-ci, j'ai notamment pu m'imprégner des process ainsi que des méthodologies de travail.

Après avoir étudié durant mes années à Polytech les méthodes agiles j'ai également pu mettre en pratique grâce à la méthode SCRUM et prendre conscience de l'intérêt de celle-ci sur un projet conséquent comme FS+ afin de maintenir un rythme important.

La bonne ambiance constante durant le stage ainsi que la cohésion au sein de l'équipe m'ont permis de m'épanouir et de progresser, d'un point de vue technique mais aussi organisationnel. Le fait de comprendre rapidement, et sans rencontrer de grosses difficultés, les concepts inhérents, ainsi que la confiance que m'ont donné les autres membres de l'équipe, m'ont apporté une grande motivation, ainsi qu'une forte implication dans mon travail, jusqu'à devenir une ressource clé du projet comme les autres membres de l'équipe.

Ce stage m'a également permis de me rendre compte que la documentation était vraiment importante, que ce soit au niveau du code ou au niveau fonctionnel sinon cela peut faire perdre énormément de temps. De plus, j'ai également appris qu'il ne fallait pas avoir une confiance totale envers les partenaires ou aux données et qu'il faut systématiquement sécuriser ce que l'on fait. La constante multi communication avec les partenaires m'ont vraiment fait prendre conscience de ces aspects.

Au-delà du cadre purement pédagogique, mon stage ne se résume pas à la réalisation technique des tâches, il m'a aussi donné l'occasion de me développer tant sur le plan personnel que professionnel, et a constitué un véritable tremplin vers le monde de l'entreprise. Une proposition de CDI de la part de Sopra Steria m'a été faite et je l'ai acceptée.

## Webographie

### **Initiation aux pratiques scrum**

<http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-fr.pdf>

### **La Poste - Facteurs Services Plus**

<http://www.laposte.fr/entreprise/a-la-une/evenements-conferences/evenement-facteurs-ser>

<https://www.laposte.fr/entreprise/services-proximite/pourquoi-faire-appel-aux-facteurs>

### **Documentation PHP**

<http://php.net/docs.php>

### **Git**

<https://www.atlassian.com/git/tutorials>

<https://git-scm.com/documentation>

### **Sopra Steria - Développement durable**

<https://www.soprasteria.com/fr/groupe/responsabilite-d-entreprise/responsabilit%C3%A9-%C3%A9conomique/d%C3%A9veloppement-durable-notre-offre-pour-les-entreprises>

Ainsi que des documents propres au projet, non accessible depuis l'extérieur de Sopra Steria.

## Annexe 1 : Développement durable

Sopra Steria met en place plusieurs moyens de sensibilisation au développement durable pour leurs collaborateurs. Au sein de l'agence de Montpellier, il y a des affiches dans les couloirs, les toilettes ainsi que dans la salle de pause sur le gaspillage d'eau, l'utilisation des lumières et de la clim de façon abusive. Des bacs de recyclage ont également été mis en place, pour les capsules de café, les gobelets et le papier.

De plus, Sopra Steria s'engage à rembourser à hauteur de 50% les transports en commun (*à Montpellier, le tram*) afin de réduire la pollution liée aux véhicules.

Sopra Steria est engagée dans une politique de Responsabilité d'Entreprise volontariste. C'est dans ce cadre que le groupe a fait l'acquisition de 1 000 certificats « Water Benefit Certificates » en 2015 (*soit l'équivalent d'un million de litres d'eau potable*) permettant de développer des projets d'approvisionnement, de purification et de conservation d'eau lorsque l'accès y est très limité, notamment en Inde où elle possède une de ses antennes.

Le groupe a également fait le choix de limiter ses émissions de gaz à effet de serre et d'être carbone neutre pour ses Data Centers et ses voyages d'affaires. Le déploiement de la solution « PowerPlusTM » en Inde fait partie du programme de compensation carbone des Data Centers du groupe Sopra Steria. Cette initiative lui permet de contribuer aux investissements dans les énergies renouvelables, indispensables pour « décarboner » le réseau électrique local.

Ces actions sont dans la lignée de Steria, très engagées dans le respect de l'environnement et la lutte contre le changement climatique, et qui était devenue en 2013 la première entreprise française à atteindre le score maximal de 100A au CDP (*Carbon Disclosure Project*).





## Résumé

Dans le cadre du stage qui s'inscrit dans la dernière année de formation à Polytech Montpellier en **I**nformatique et **G**estion (*IG*), j'ai travaillé au sein du groupe Sopra Steria Montpellier qui est une **E**ntreprise de **S**ervices du **N**umérique (*ESN*) et plus particulièrement sur le projet **F**acteurs **S**ervice **P**lus (*FS+*) pour le compte de La Poste.

Les différents travaux réalisés au cours du stage se sont principalement appuyés sur le langage PHP pour le code métier, l'utilisation de web services pour la communication entre les différents prestataires ainsi que la manipulation de fichier au format JSON et XML.

**Mots clés :** PHP, Web-services, Partenaires, Facteurs Services Plus, La Poste.

## Abstract

As part of my internship in the last year of training at Polytech Montpellier in Computer Science and Management, I worked within the Sopra Steria Montpellier Group, which is an IT Services & Software Engineering and particularly on the “**F**acteurs **S**ervice **P**lus” (*FS+*) project for La Poste.

The different work carried out during the internship was mainly based on the PHP language for the code, the use of web-services for communication between the various partners, and the manipulation of the file in JSON and XML format.

**Key words :** PHP, Web-services, Partners, Facteurs Services Plus, La Poste.