## Résultats obtenus

Pendant ma mission, j’ai travaillé sur différentes tâches. Cette partie synthétise le travail effectué ainsi que les résultats obtenus afin de les replacer par rapport aux objectifs initiaux qui avaient été fixés

### Mise en œuvre de l’approche Domain-driven design

Ce premier travail qui avait avant tout un rôle introductif vis-à-vis de l’environnement logiciel et technique de Coheris a conduit à mettre en œuvre la nouvelle architecture DDD avec le design pattern CQRS et Event-Sourcing. La structure d’un module nouvelle offre Coheris permet plus de flexibilité lors de l’ajout des nouvelles fonctionnalités. De plus, il permet la séparation des rôles technique en encapsulant chacun dans une couche applicative.

De même, les différentes modes de déploiements nous permettent selon les besoins du client de choisir qu’elle approche de déploiement faudra le mettre en œuvre. En oubliant pas la possibilité de lancer le module soit en mode on-premise soit en mode multi-tenant.

### Planificateur de tache de hautes disponibilités

Le module scheduler était une véritable formation sur le Domain-driven design. Il m’a permis de mieux comprendre le principe de l’architecture des modules étudiés dans la première partie des travaux. En effet, mon travail qui s’est ´étalé sur 2 mois a permis d’apporter un système de planification de tache déployable sur différent mode. En oubliant pas l’apport de l’intégration du Framework Quartz dans la solution. Ce dernier nous apporté plusieurs avantage par rapport au scheduler de CRM Coheris : La gestion des retards, le multithreading, mise mode cluster, la persistance des tâches. Le module scheduler semble donc répondre aux attentes.

### Intégration de la Textmining dans le module social

Le module social DDD, comme on l’appel, est la nouvelle génération du connecteur social existant. Coheris souhaite le vendre comme étant une application indépendante à la fin de l’année 2015. Mon travail a d’une part consisté de mettre en place la structure de ce module en se basant sur l’approche DDD et de préparer la migration du connecteur socail CRM.

D’autre part, j’ai eu en charge de créer les fonctionnalités nécessaires d’abord pour intégrer SPAD RealTime dans le module et ensuite créer les interfaces de Tests qu’elle nécessite la solution.

Cette partie technique a été précédé d’une étude complète de l’existant connecteur social afin d’extraire les fonctionnalités que j’ai besoin pour bien répondre aux besoins

Enfin, mon travail a ainsi permis de mettre en place une stratégie pour tester le textmining sur les posts et les commentaires Facebook.

## Difficultés rencontrées

Cette partie a pour but de présenter les éléments qui ont pu ralentir ma progression dans le travail. Je vais principalement aborder des points techniques. D’abord, je présenterai les structures des projets java et le système de build adopté. Ensuite, j’attirerai l’attention sur les difficultés rencontrées pour tester les algorithmes de textmining de SPAD RealTime.

### Compilation des modules

*Structure d’un module DDD des projets dans*

En utilisant la plateforme JAVA EE de coheris, et après une étude faite sur la nouvelle architecture, l’équipe R&D CRM sont mis en accord que chaque couche du module (chaque rôle de l’approche DDD) est encapsulée d’un projet java déployer après comme une archive jar.

Il faut prendre en compte aussi que chaque couche de module dépend de de la même couche dans le module core. C’est-à-dire, si on prend par exemple la couche module-application, elle dépend de la couche core-application. La couche module-infrastructure dépend de la couche core-infrastructure et ainsi de suite. De plus, chaque module a son Tronc de développement dans le gestionnaire de version SVN. Le fait d’ajouter une fonctionnalité ou d’ajouter une classe dans le core, il doit être d’abord compilé et encapsulé dans un jar et après publier sur artifactory[]. Pour le module scheduler ou le social, et avant de lancer son build, il faudra s’assurer qu’on dépend de la bonne version du module core. Sinon, il faudra refaire toute l’enchaînement que je l’ai décrit pour le module core.

*Tester le module*

Pour pouvoir tester les fonctionnalités d’un module, on doit mettre en œuvre une autre approche d’encapsulation pour assurer les points d’entrée vers ce dernier soit par son api REST, soit par son IHM. Alors, un projet web dynamique (war) sous le nom « desktop-application » assure l’encapsulation des modules développé. Ce qui m’a permis de tester les mises à jour des messages facebook dans le module social avec une tache planifié créé à partir de l’IHM du module scheduler. Le problème ici apparait lorsqu’on fait une modification dans le code du module core et dans le module social par exemple. En effet, on ne peut pas utiliser la modification de code à chaud. Il faudra suivre l’enchainnment suivante :

1.compiler et publier le module core,

2.compiler le module social

3.républier les nouvelle jar dans le serveur d’application

De plus, j’étais obligé de migrer tous les pages JSF de chaque module de son couche gui vers le desktop-application pour que je puisse assurer la modification à chaud du code HTML. A la fin des tests, il faut les ré-migrer vers la couche gui de son module initiale.

### Tester l’efficacité de l’algorithme de Textmining

Malheureusement, je n’ai pas arrivé à tester l’efficacité des algorithmes de Textemining SPAD RealTime sur les messages Facebook avec le model « bayesien motif ». En effet, les développements sont encore en cours sur le serveur SPAD jusqu’à maintenant. Ainsi, les tests demande beaucoup de temps qu’il a étais planifié. Il faudra d’abord apprendre l’algorithme m par des milliers de message bien qualifier et les mettre dans un fichier l’amorçage au démarrage du serveur. A titre illustratif, le code relatif à ce dernier fichier est le suivant :

target text

demande je veux

demande j'aimerai

opportunity Vous ne faite plus rien pour les anniversaires?! Je n’ai même pas eu de mail avec une ptite offre

intervention moi non plus ..rien depuis longtemps pour ""anniversaire"" c bien dommage

lead Merci à toutes je suis vraiment étonnée ! Et merci à Yves rocher pour ce super concours !

….

On constate, il faut alimenter ce ficher pour qu’à chaque démarrage du serveur soit prêt à donner des bons résultats de qualification. Il faudra que toute l’équipe R&D CRM assiste à l’alimentation de ce fichier par des messages bien qualifiés pendant des jours !