

PLAN DE DÉVELOPPEMENT

| | |
|--------------|----------------------|
| Version : | 1.0 |
| Date : | 13/01/2017 |
| Rédigé par : | L'équipe SmartLogger |
| Relu par : | L'équipe SmartLogger |

HISTORIQUE DE LA DOCUMENTATION

| Version | Date | Modifications réalisées |
|---------|------------|---------------------------------|
| 0.1 | 14/12/2016 | Création du document |
| 1.0 | 13/01/17 | Restructuration et correctifs |
| 1.1 | 26/01/17 | Ajouts et correctif de contenus |

1. Documents applicables et de référence

- Le document de présentation client : SmartLogger.pdf
- Le document de spécification technique du besoin : STB.pdf
- Le document d'architecture du logiciel : DAL.pdf
- Le glossaire associé à la documentation : Glossaire.pdf
- Le diagramme Gantt associé au projet : SmartLogger_Gantt.png

2. Contexte du projet

Le projet SmartLogger provient d'un projet pédagogique émanant d'un client externe à l'université, dans un cadre pédagogique de première année de Master Informatique spécialisé Génie Logiciel, sur une période de 6 mois, avec 3 mois sur la documentation puis 3 mois sur le développement.

L'émetteur de ce projet est l'entreprise Saagie spécialisée dans le Big Data, représentée par M.Jonathan GERMOND. Il sera possible d'avoir un soutien technique au travers d'un UX Designer pour aider au développement de l'Interface Utilisateur. Notre professeur référent lors de ce projet est Mme Lina SOUALMIA.

L'objectif poursuivi du projet est de développer un système, permettant d'alerter l'utilisateur sur des données en provenance d'applicatifs défectueux dans l'optique de faciliter leurs correctifs, à l'aide d'un développement AGILE, afin d'assurer, si le produit n'est pas complété pleinement comme nous le souhaiterions, de permettre au client de l'utiliser et de le finaliser si besoin.

Les documents de référence pour le Plan de Développement sont la Spécification Technique du Besoin, ainsi que le Document d'Architecture Logiciel et le Cahier de Recettes.

3. Méthodologie de développement

3.1. Méthodologie AGILE

Afin d'organiser notre développement, nous avons décidé de suivre les principes des méthodes AGILE suivante : eXtreme Programming (XP) et Test Driver Development (TDD).

Nous avons choisi la méthode eXtreme Programming car, étant une petite équipe, suivre les principes de cette méthode, consistant notamment à travailler par pair, et faire tourner l'effectif sur l'ensemble des composants du projet permettra une connaissance bien plus précise de chacune des composantes du projet.

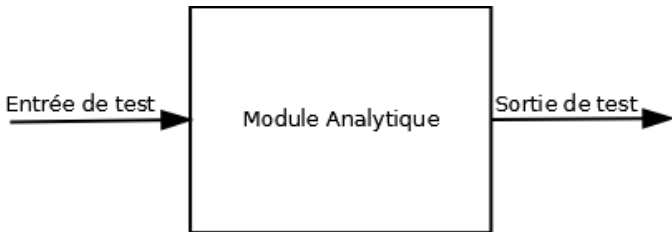
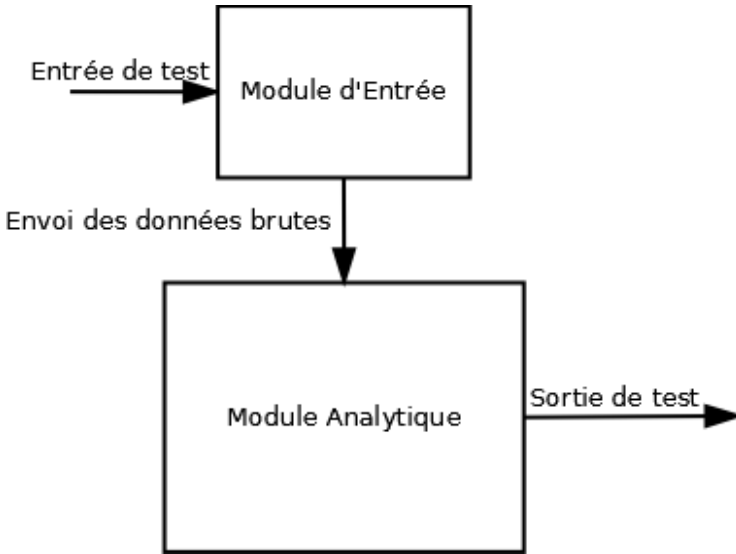
La méthodologie Test Driver Development consiste quant à elle à guider le développement via la rédaction préliminaire des tests afin de conduire les développeurs dans leurs phases de développement.

3.2. Conditions d'acceptation d'un livrable

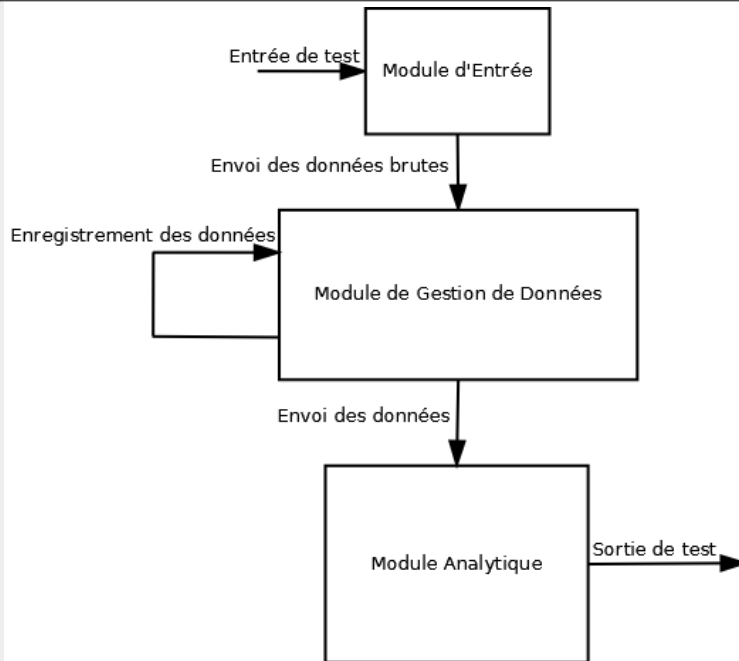
Afin qu'un livrable puisse être livré au client, celui-ci doit respecter un certain nombre de points, il doit d'une part passer l'ensemble des tests unitaires appliqués aux classes contenue dans le livrable, et d'autre part passer les tests d'intégration afin qu'il n'y est pas de régression au niveau du projet principal.

De plus, une fois le livrable envoyé au client, celui-ci à 3 jours (minimum) pour valider ou non le livrable et nous faire parvenir ces retours quant aux livrables.

3.3. Développement du projet

| Numéro de version de l'itération | Image Descriptive | Description |
|----------------------------------|--|--|
| 0.1 |  | L'itération 0.1 aura pour but de développer le module Analytique afin qu'il soit fonctionnel avec une entrée et une sortie de test afin d'effectuer les tests. |
| 0.2 |  | L'itération 0.2 aura pour but de développer le module d'Entrée afin qu'il soit fonctionnel avec une entrée de test adaptée au nouveau module, qui deviendra l'entrée de test définitive. La sortie restera la même, entre les deux modules s'effectueront des échanges du module d'Entrée vers le module Analytique de données brutes et conformes aux exigences du fonctionnement du module Analytique. |

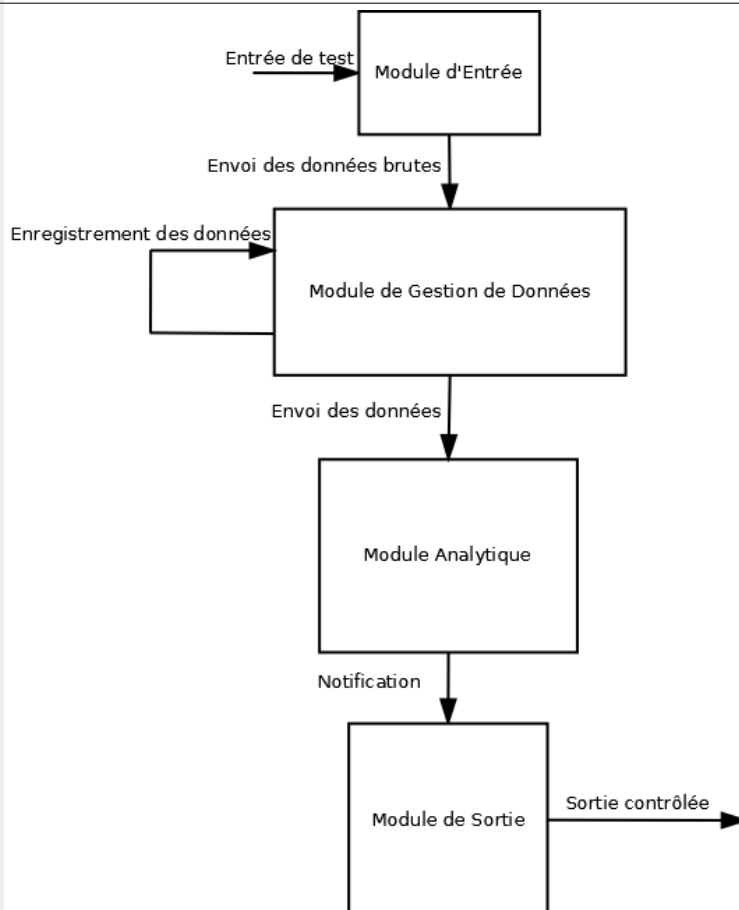
0.3



L'itération 0.3 aura pour but de développer le module de Gestion de Données afin qu'il soit fonctionnel et testable sur sa capacité à stocker les données en temps réel, le but étant de le rendre premièrement fonctionnel, si nécessaire, une version corrective viendra améliorer les performances du module. Le module d'Entrée enverra les données brutes au module de Gestion de Données qui les stockera puis les enverra au module Analytique.

Durée :
 1 semaine

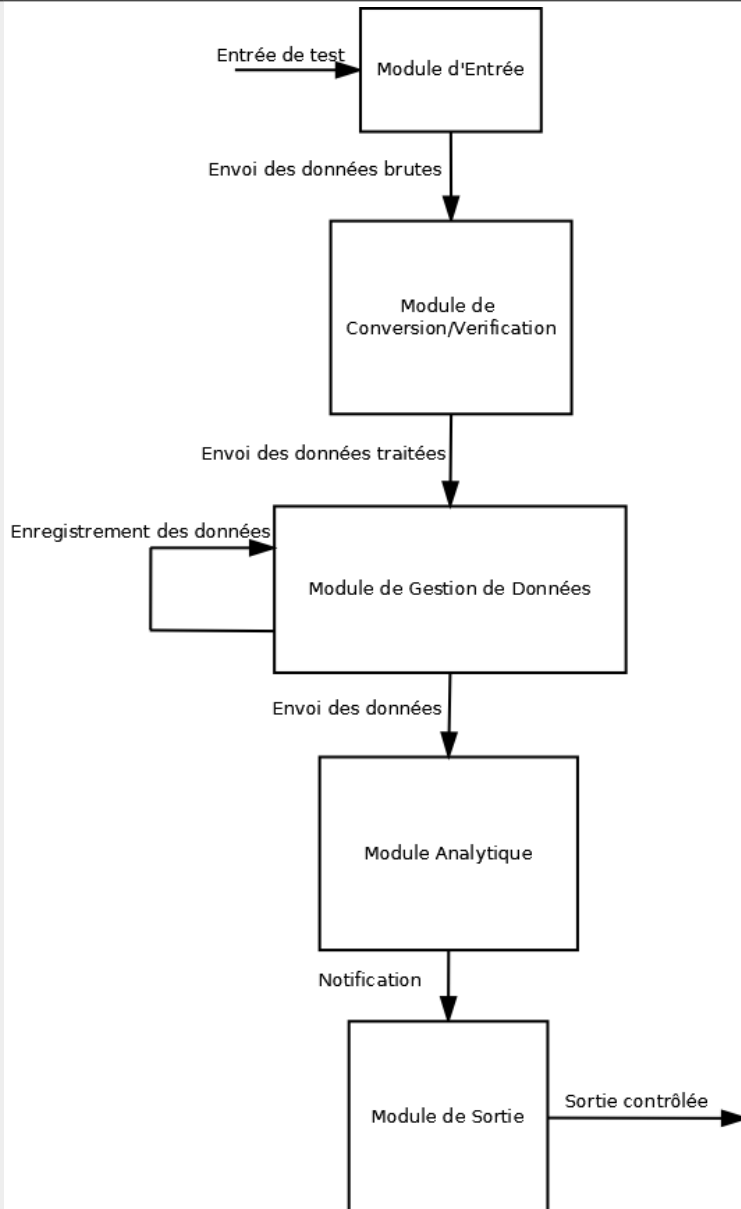
0.4



L'itération 0.4 aura pour but de développer le module de Sortie afin d'effectuer les tests sur des sorties réelles. Il sera aussi tenté d'apporter plusieurs APIs de sortie pour le module.

Durée :
 2 semaines

1.0

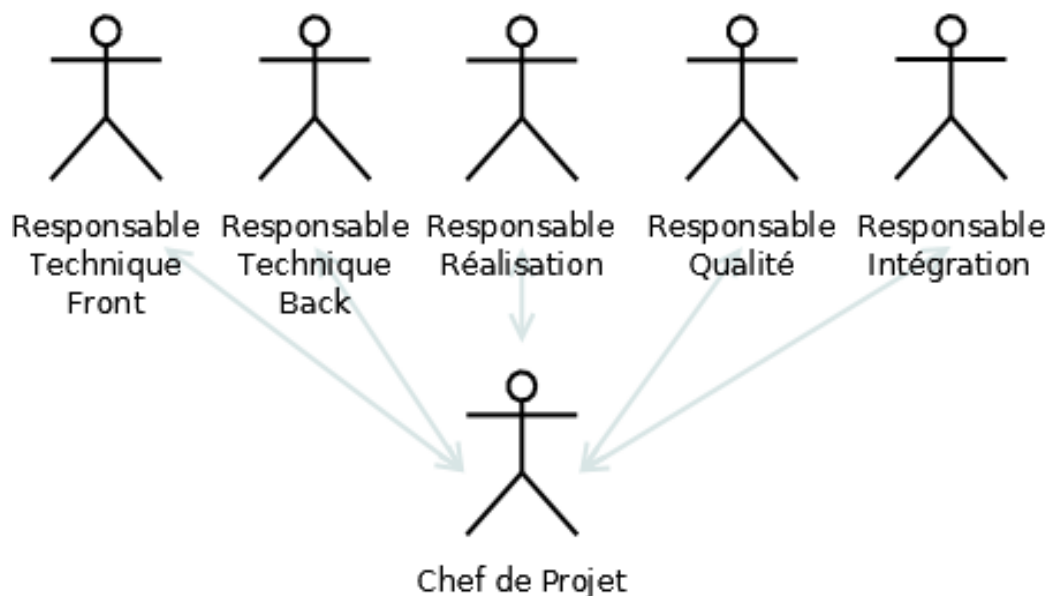


Le but de cette itération qui sera l'objectif final du projet (qui pourra être suivi d'autres itérations si le temps est suffisant, afin d'améliorer les performances et le traitement) aura pour but de finaliser le produit avec l'ajout du module de Conversion/Vérification et sera testé intégralement avant rendu au client.

Durée :
1 semaine

4. Organisation et responsabilités

Le mode d'organisation de l'équipe s'appuie sur le modèle suivant :



Chaque membre de l'équipe se verra affecter un poste valable pour l'intégralité du projet.

















Répartition des rôles

| Rôle | Nom de l'intervenant | Description | Heure de Travail |
|---------------------------------|----------------------|--|------------------|
| Chef de Projet | Franck Caron | Le rôle de chef de projet est de maintenir l'organisation de l'équipe et du projet tout au long de celui-ci, il est en contact direct avec le client et le professeur référent. | 5 |
| Responsable Technique Front-End | Khalid Rabou | Le rôle de responsable technique front-end est de collaborer avec les développeurs travaillant sur cette partie du projet afin d'assurer que les technologies sont au mieux utilisées. | 5 |
| Responsable Technique Back-End | Jordan Baudin | Le rôle de responsable technique back-end est de collaborer avec les développeurs travaillant sur cette partie du projet afin d'assurer que les technologies sont au mieux utilisées. | 5 |

| Rôle | Nom de l'intervenant | Description | Heure de Travail |
|-------------------------|----------------------|---|------------------|
| Responsable Réalisation | Nicolas Gille | Le rôle de responsable réalisation est s'assurer que chaque membre de l'équipe travaille sur les mêmes outils et les mêmes versions d'outils afin d'éviter tout problème dépendant de ceux-ci. | 5 |
| Responsable Qualité | Camille Leplumey | Le responsable qualité est chargé d'écrire et d'exécuter les tests lors des différentes itérations et d'effectuer un retour aux membres de l'équipe afin de corriger un quelconque problème. | 5 |
| Responsable Intégration | Grégoire Pommier | Le responsable intégration est chargé de s'assurer du fonctionnement des modules entre eux et de la connexion entre chacune des couches. | 5 |
| Responsable Client | Franck Caron | Le responsable client est chargé de s'assurer que le client est au courant de nos avancées et est à l'écoute du client dans le cas où celui-ci souhaiterait ajouter de nouvelles fonctionnalités. | 5 |

5. Évaluation du projet et dimensionnement des moyens

Listing organisationnel des tâches

| |  | Nom | Durée | Début | Fin | Prédécesseurs |
|----|---|---|--------|------------|------------|---------------|
| 1 |  | Installer Spark | 1j? | 30/01/2017 | 30/01/2017 | |
| 2 |  | Implanter l'interface ILearner | 1j? | 31/01/2017 | 31/01/2017 | 1 |
| 3 |  | Réaliser un type concret de Learner, basé sur des algorithmes Sp4j? | 4j? | 01/02/2017 | 06/02/2017 | 2 |
| 4 |  | Doter le module d'un sous-système d'entrée (pour fournir des don0.63j?) | 0.63j? | 07/02/2017 | 07/02/2017 | 3 |
| 5 |  | Associer un dispositif de sortie. | 1j? | 07/02/2017 | 07/02/2017 | 3 |
| 6 |  | Etalonner le module à l'aide d'un jeu initial | 1j? | 08/02/2017 | 08/02/2017 | 4,5 |
| 7 |  | Tester les performances des fonctionnalités de prédiction et d'anal1j? | 1j? | 09/02/2017 | 09/02/2017 | 6 |
| 8 |  | Réalisation des tests unitaires | 1j? | 10/02/2017 | 10/02/2017 | 7 |
| 9 |  | Rendu de l'itération 0.1 | 1j? | 13/02/2017 | 13/02/2017 | 8 |
| 10 |  | Implanter l'interface IEntryManager | 1j? | 14/02/2017 | 14/02/2017 | 9 |
| 11 |  | Réaliser un type concret LogManager | 2j? | 15/02/2017 | 16/02/2017 | 10 |
| 12 |  | Réalisation des tests unitaires | 1j? | 17/02/2017 | 17/02/2017 | 11 |
| 13 |  | Relier le module d'entrée au module de Learning | 1j? | 20/02/2017 | 20/02/2017 | 11,12 |
| 14 |  | Réalisation des tests d'intégrations | 2j? | 21/02/2017 | 22/02/2017 | 13 |
| 15 |  | Rendu de l'itération 0.2 | 1j? | 23/02/2017 | 23/02/2017 | 14 |
| 16 | | Implémenter la base de données | 1j? | 24/02/2017 | 24/02/2017 | 15 |
| 17 | | Implanter l'interface IHighwayToHell | 1j? | 27/02/2017 | 27/02/2017 | 16 |
| 18 | | Implémenter HighwayOperatorIO | 1j? | 28/02/2017 | 28/02/2017 | 17 |
| 19 | | Implémenter HighwayOperatorAnalyser | 1j? | 28/02/2017 | 28/02/2017 | 17 |
| 20 | | Réalisation des tests unitaires | 1j? | 01/03/2017 | 01/03/2017 | 17,18,19 |
| 21 | | Relier le module d'entrée au module de Base de données | 1j? | 01/03/2017 | 01/03/2017 | 16,17,18,19 |
| 22 | | Réalisation des tests d'intégrations | 1j? | 02/03/2017 | 02/03/2017 | 21 |
| 23 | | Rendu de l'itération 0.3 | 1j? | 03/03/2017 | 03/03/2017 | 22 |
| 24 | | Implanter l'interface IAlertManager | 1j? | 06/03/2017 | 06/03/2017 | 23 |
| 25 | | Réaliser un type concret de Alert | 1j? | 07/03/2017 | 07/03/2017 | 24 |
| 26 | | Adapter la sortie de test | 1j? | 08/03/2017 | 08/03/2017 | 25 |
| 27 | | Réalisation des tests unitaires | 1j? | 09/03/2017 | 09/03/2017 | 24,25,26 |
| 28 | | Réalisation des tests d'intégrations | 2j? | 10/03/2017 | 13/03/2017 | 27 |
| 29 | | Rendu de l'itération 0.4 | 1j? | 14/03/2017 | 14/03/2017 | 28 |
| 30 | | Implanter la classe DataConverter | 1j? | 15/03/2017 | 15/03/2017 | 29 |
| 31 | | Implanter la classe DataValidator | 1j? | 15/03/2017 | 15/03/2017 | 29 |
| 32 | | Implanter la classe DataCompletionist | 1j? | 15/03/2017 | 15/03/2017 | 29 |
| 33 | | Réalisations des test unitaires de chacunes des classes | 1j? | 16/03/2017 | 16/03/2017 | 30,31,32 |
| 34 | | Réalisations des tests d'intégrations | 1j? | 17/03/2017 | 17/03/2017 | 33 |
| 35 | | Rendu de l'itération 0.5 | 1j? | 20/03/2017 | 20/03/2017 | 34 |

Le GANTT est disponible dans le dossier documentaire du projet (cf SmartLogger_Gantt.png).