

## PLAN DE DÉVELOPPEMENT

|                     |                      |
|---------------------|----------------------|
| <b>Version :</b>    | 2.0                  |
| <b>Date :</b>       | 07/02/2017           |
| <b>Rédigé par :</b> | L'équipe SmartLogger |
| <b>Relu par :</b>   | 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              |
| 2.0     | 07/02/17   | Refonte des cycles d'itération et du Gantt |

## **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).

Nous avons choisi l'eXtreme Programming car la taille de notre équipe s'y apprêtait parfaitement, du fait que les principes de cette méthode consistent notamment à travailler en binôme. De plus, effectuer des rotations permet à toute l'équipe de travailler sur l'ensemble des composants du projet, ce qui fournit à chacun, une connaissance bien plus précise sur les points principaux du projet.

#### **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

Notre projet suivant la méthodologie AGILE, nous avons décidé de partitionner le développement selon les itérations suivantes :

| Itération n°1   | Durée : 6 semaines  | Version 0.1 |
|---|---|-------------|
| Schéma d'avancement   | Description   |             |
| <pre>graph TD; A[Entrée de test] --&gt; B[Module d'Entrée]; B -- "Envoi des données" --&gt; C[Module Analytique]; C -- "Notification" --&gt; D[Module de Sortie]; D -- "Sortie contrôlée" --&gt; E[ ]</pre> | <p>La première itération aura pour but de fournir les fonctionnalités minimales attendues par le projet :</p> <ul style="list-style-type: none"><li>• Accepter un log issu d'une requête HTTP externe</li><li>• Effectuer une analyse prédictive pour déterminer sa criticité</li><li>• Alerter le personnel de maintenance le cas échéant.</li></ul> <p>La production issue de cette version disposera alors du mécanisme réel en termes de modules (Analyse + Entrée + Sortie), mais seulement du minimum requis en termes de fonctionnalités afin de pouvoir assurer son rôle.</p> <p>Ainsi, seule l'alerte par mail sera implémentée, aucun pré-traitement sur les données ne sera effectué, et ce livrable ne sera pas doté de persistance (i.e, sera livré sans sa mémoire : le module de gestion des données).</p> |             |
| Date de livraison prévue : 13/03/2017   |   |             |

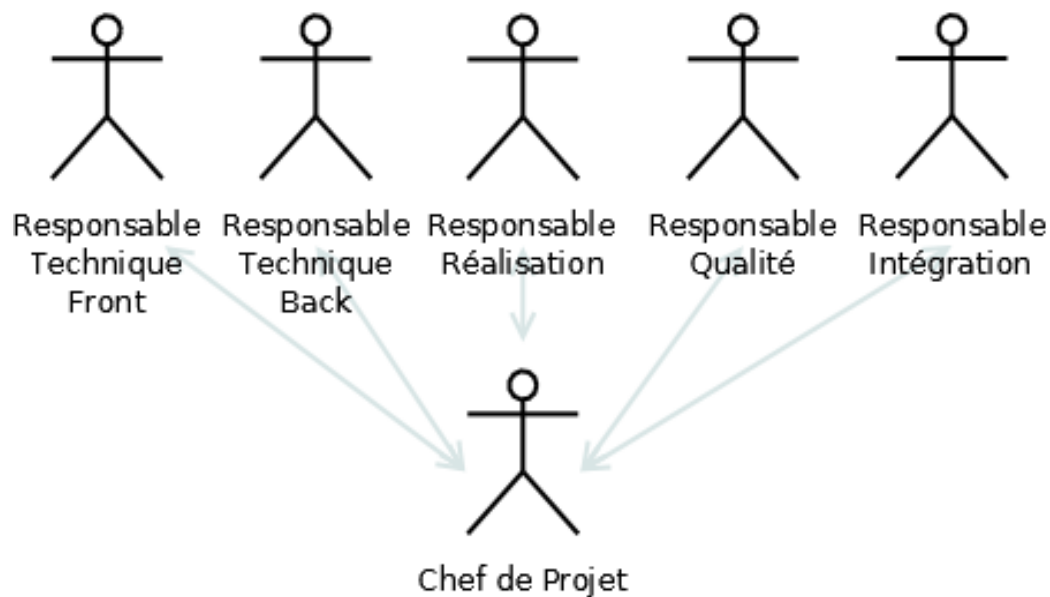
| Itération n°2  |  | Durée : 3 semaines   | Version 0.2 |
|--|--|--|-------------|
| Schéma d'avancement  |  | Description  |             |
| <pre>graph TD; A[Entrée de test] --&gt; B[Module d'Entrée]; B -- "Envoi des données brutes" --&gt; C[Module de Gestion de Données]; D[Enregistrement des données] --&gt; C; C -- "Envoi des données" --&gt; E[Module Analytique]; E -- "Notification" --&gt; F[Module de Sortie]; F -- "Sortie contrôlée" --&gt; G[ ];</pre> |  | <p>La deuxième itération consistera à fournir une capacité de persistance au système, en implantant le système de gestion de données. L'impact de cet apport est double :</p> <ul style="list-style-type: none"><li>• Conserver l'ensemble des logs traités et leurs résultats.</li><li>• Faciliter la transition des données brutes du module d'entrée au module analytique.</li></ul> <p>De cette manière, le modèle de traitement utilisé par le module d'analyse sera dans la capacité de conserver son comportement bien au-delà de l'utilisation simple : Le but étant que la version 0.2 puisse tourner en temps réel, et reprendre rapidement son activité après un arrêt.</p> |             |
| Date de livraison prévue : 03/04/2017  |  |  |             |

| Itération n°3   | Durée : 2 semaines  | Version 1.0 |
|---|---|-------------|
| Schéma d'avancement   | Description   |             |
| <pre> graph TD     A[Entrée de test] --&gt; B[Module d'Entrée]     B -- "Envoi des données brutes" --&gt; C[Module de Conversion/Verification]     C -- "Envoi des données traitées" --&gt; D[Module de Gestion de Données]     D -- "Sauvegarde des données" --&gt; D     D -- "Envoi des données" --&gt; E[Module Analytique]     E -- "Notification" --&gt; F[Module de Sortie]     F -- "Sortie contrôlée" --&gt; G[ ]           </pre> | <p>Cette troisième itération devra se clôturer par la production du système final attendu.</p> <p>Elle se décompose en 2 phases :</p> <p><u>Phase n°1</u> : Consistera dans l'ensemble à produire le module de conversion/vérification dont l'utilité sera de sécuriser l'application des données malveillantes, et de conserver les données inutilisables dans le cas d'une éventuelle ré-exploitation dans le futur.</p> <p><u>Phase n°2</u> : Une phase de test et d'optimisation : l'équipe s'assurera de l'utilisabilité du système et de la présence de toutes les fonctionnalités clients requises.</p> <p>Cette itération dispose d'un temps de développement moindre par rapport aux itérations précédentes, ce qui permettra de consacrer davantage de temps aux retours du client.</p> |             |
| Date de livraison prévue : 17/04/2017   |   |             |

Selon le temps de développement restant à la sortie de la version 1.0, il pourra être envisageable de préparer une version 1.1, en ajoutant des fonctionnalités supplémentaires qui seront définies par le client lui-même, après avoir utilisé l'applicatif. Cela constituerait alors une quatrième itération, dont le détail ne pourrait être décrit à ce jour, mais qui consisterait en une succession d'ajouts mineurs, de correctifs et d'optimisations diverses.

#### 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  | Heures 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                |