RAPPORT ORGANISATION PROJET SIT213

Auteurs :

- Bartoli Mathieu
- Dumestre Lucas
- Francis Ludovic
- Gueye Oulimata
- Hug de Larauze Sébastien

Date :

16/09/2019

Destinataires :

Equipe pédagogique FIP 2A

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Organisation choisie :

Nous avons choisi de nous répartir les tâches de code uniformément. Chaque membre du groupe s'occupe donc d'un périmètre de développement défini. Nous utilisons un certain nombre d'applications et de services. Chaque membre a installé l'ensemble de ces services pour une meilleure collaboration et homogénéité du code.

Nous utilisons le logiciel GitKrakenGlo pour l'organisation et la répartition des tâches. Ce logiciel est similaire à Trello. Il comporte des regroupements de tâches en fonction de leur avancement (une partie « tâches à faire », « tâches en cours » et « tâches finies »). Il permet également d'instaurer des dates limites de tâches permettant ainsi de respecter les délais souhaitaient par le client.

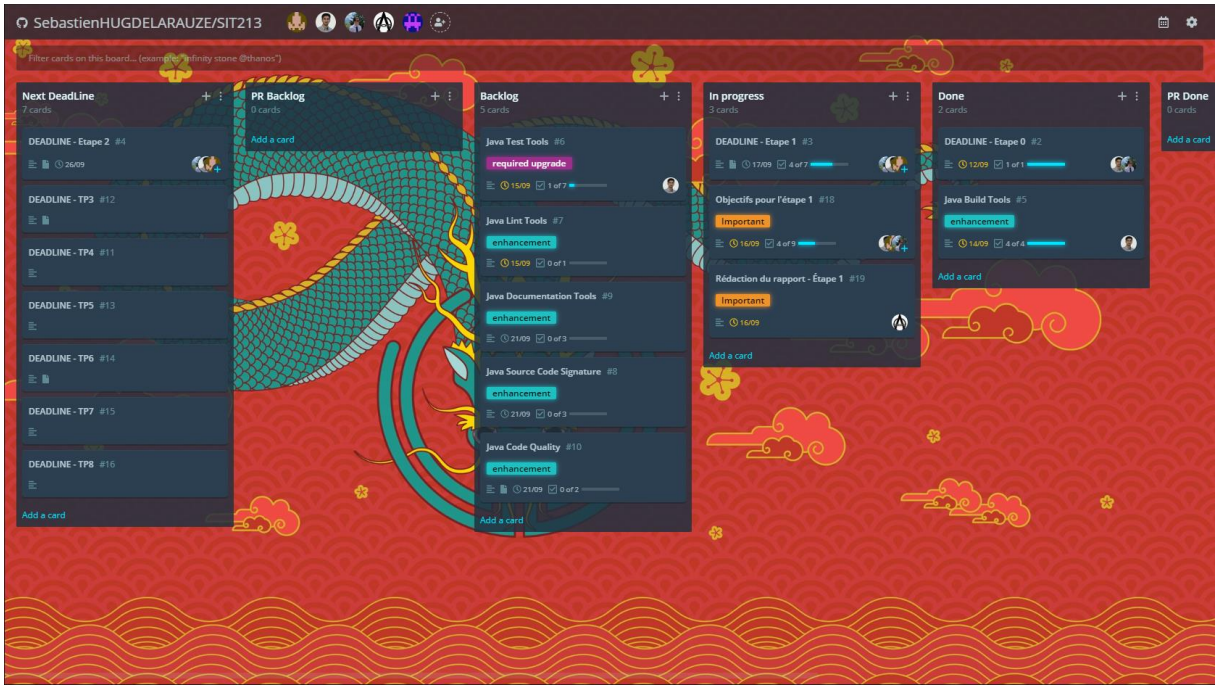

Figure 1 Interface GitKrakenGlo

Nous avons instauré des rôles à chaque membre pour permettre d'éviter les conflits dans la réalisation du projet. De plus, chaque rôle est déterminé en fonction des compétences fortes du membre. Ainsi chacun est acteur du projet.

Rôles attribués :

- Dumestre Lucas est le chef de projet (suivi de l'état d'avancement de chaque membre, apport d'une aide si besoin)
- Hug de Larauze Sebastien supervise (techniquement) la structure d'organisation du projet. Il gère l'utilisation du répertoire git et la cohérence du regroupement du travail effectué par tous les membres. Il fait également des synthèses entre chaque étape pour remonter ces informations à l'équipe de documentation. Ainsi que la cohérence du livrable.
- Francis Ludovic et Bartoli Mathieu sont les responsables de la rédaction des documentations et des rapports d'états.
- Gueye Oulimata spécifie, développe les tests à réaliser et vérifie que chacun corresponde bien au cahier des charges souhaitait.

Chaque membre travaille en parallèle sur une partie du développement du code.

Critères de qualité choisis :

TravisCI : notre service d'intégration continue

Pour évaluer la qualité du code, nous nous aidons d'un service TravisCI (similaire à Jenkins). Ce service est disponible en ligne. Cela rend alors le service accessible à tous les membres.

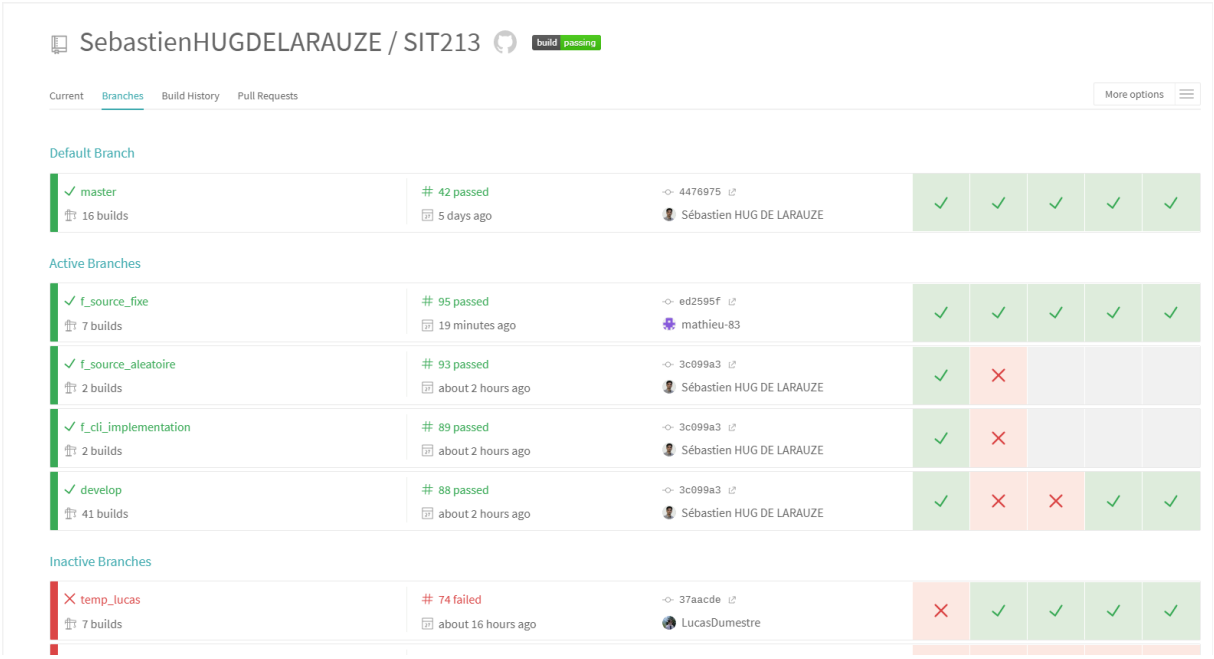

Figure 2 Interface TravisCI

Il nous permet l'intégration continue de notre code tout en le testant dans son ensemble.

Il nous permet de tester chaque commit en réalisant les tests unitaires et le déploiement (pour le moment) puis réalisera les tests d'intégrations, de qualité de code, de documentations et de syntaxe.

Si les tests sont validés, le service crée alors une archive correspondant directement au cahier des charges client. Cela apporte un gain de temps non négligeable et assure un livrable client propre.

Javadocs

Pour les documentations Javadocs, nous utiliserons le générateur de Javadoc. Si le temps nous le permet, nous utiliserons des outils de relecture syntaxique lors de la génération des documentations.

Ant : un outil d'automatisation des tâches

Nous utilisons également un outil d'automatisation des tâches nommé Ant. Il permet d'exécuter une multitude de tâches spécifiques aux librairies java telles que la compilation, l'exécution et la création de la Javadoc. L'automatisation des tâches permet d'assurer la qualité des livrables et éviter l'oubli d'actions importantes.

Git : gestionnaire de version et outil de synchronisation

Nous utilisons le gestionnaire de version git car c'est un outil populaire, maintenu à jour et pratique dans son utilisation. La résolution des problèmes liés à cet outil est donc facilitée de par sa large utilisation dans le domaine du code. De plus, de multiples services sont rattachés à ce gestionnaire de version tels que des logiciels de gestion de projet.