

# Sprint 1

## I. Revue de Sprint

- Présentation des parties “Utilisateurs” et “Problèmes” ;
- Spring Security : intéressant à utiliser, pour créer les utilisateurs et se loguer ;
- Sur Travis CI : Rajouter d’autres JDK pour tester les plateformes, mais les différentes versions de Frameworks Grails ne sont pas aussi importants à tester ;

Comme le travis.yml est présent sur toutes les branches, on peut lancer les tests unitaires sur toutes les branches, tandis que l’intégration est faite surtout sur le master

- Scénarios : Pour la revue, intéressant d’arriver avec 2 ou 3 scénarios qu’on présente et qui illustrent les avancées du sprint ;
- Couverture de code : important d’utiliser Cobertura (et Codenarc), et de savoir quelle est notre stratégie, quelle pourcentage on décide de couvrir et par rapport à quelles raisons ou contexte ;
- Stories : peut-être mieux les découper en tâches ;
- Agile : aspects intéressant à dire dans les livrables à l’écrit et à l’oral (rédaction backlog etc.) ;

**Bilan** : 8 stories terminées pour une valeur ajoutée totale de **11** et une difficulté de **8**.

**Vélocité** : 8.

## II. Rétrospective

Trois points à améliorer :

- Couverture de code (Cobertura) ;
- Productivité (Pair Programming) ;
- Utilisation de Git (Branches).

Idée de workflow idéal :

- On développe sur une branche (individuelle) ;
- A chaque avancée importante, on fusionne sur la branche Develop, envoyé sur Travis (et tests unitaires) ;
- A chaque fin de sprint, on pousse de la branche Develop à la branche Master, envoyé sur Travis (et tests intégration), puis déployé sur Heroku.