

# BE – SIT 213

## **Quelle organisation adopter pour respecter les délais ?**

- Mise en place d'un Taiga (Kanban et gestion de projet) pour créer les lots de tâches et/ou de sous-tâches à réaliser et les assigner à des utilisateurs
  - L'objectif étant d'assigner des sous tâches par membre du groupe. Une sous-tâche peut être une classe à implémenter, ou un test sur une classe.

## **Quels critères de qualité (forme, contenu, cohérence, tests...) utiliser pour les livrables, et comment en garantir le respect à chaque itération ?**

Nous utilisons Github Action, permettant de générer un livrable conforme et d'effectuer des tests JUnit à des fins de validation du code poussé.

## **Comment assurer la qualité des documentations (Javadoc notamment)**

De manière globale, un commit sera appliqué dès lors que trois personnes auront accepté ledit commit. Une relecture croisée permettra d'assurer la conformité et la qualité des documentations.

## **Comment synchroniser le travail des développeurs ? Quels critères / règles de commits s'imposent (la compilation passe, les tests passent, le code est relu...) ?**

Chaque développeur travaillera sur des briques unitaires définies à l'avance. Il travaillera sur une branche dédiée à l'ajout ou la modification ciblé, et une fois celui-ci terminé, il effectuera un *merge request*. Ce *merge request* devra être relu par deux autres personnes pour qu'il soit accepté. Github Action permettra de s'assurer de la compilation conforme (via des tests JUnit).

## **Faut-il mobiliser des outils de test (JUnit, Emma...) ? Si oui, selon quelles modalités ?**

Des tests JUnit sont d'ores et déjà mis en place. L'objectif est d'en ajouter également par la suite.

Cependant en vue de la suite, il sera très certainement bien plus compliqué d'implémenter des tests unitaires sur des systèmes dans lesquels on aura intégré du bruit. Il faudra très certainement s'approcher de la théorie mathématique afin de savoir si ce qu'on simule est proche de la réalité théorique.

## **Quelles parties peut-on vraiment tester avec un programme de test ?**

## **Groupe B1 – Belkhir, Icard, Alegre, Fuchs, Achille**

Chaque élément individuel de la chaîne de transmission peut être testé (le comportement est purement déterministe dès lors qu'il n'y a pas de bruit). Une approche probabiliste sera nécessaire lorsque les problématiques de bruit seront prises en compte dans le projet.

### **Comment améliorer la testabilité ? / Comment organiser l'équipe spécifiquement pour les tests (p. ex. développeurs et testeurs différents) ?**

Les personnes en charge des tests réaliseront dans un premier temps des tests en boîtes noires, et feront les reviews. Ils pourront ensuite effectuer les tests en boîtes blanches pour s'assurer que tous les cas de tests ont été réalisés.

### **Comment s'assurer de la non-régression à chaque nouvelle itération du projet ?**

Pour s'assurer de la non-régression, il faudra de nouveau effectuer les tests qui ont été réalisés pour les étapes précédentes.

### **Quelle confiance accorder aux résultats de simulation, et comment accroître cette confiance ?**

La confiance accordée aux résultats de simulation doit être nuancée par les résultats des cours, et donc soumise à interprétation. On tâchera donc de réfléchir à la cohérence de la simulation avec la théorie des télécommunications.

### **Comment comparer des simulations lorsqu'elles reposent sur des comportements aléatoires (message, bruit...) ?**

Nous pouvons supposer que le processus stochastique respecte la notion d'ergodicité, il faut donc vérifier pour une réalisation que la moyenne statistique soit égale à la moyenne temporelle du processus (calculée via l'espérance).