

**Équipe de développeurs :**

**El Bakkali Tamara Soufian, Konstantopoulos Alexios, Sanou Pap, Bernard Loïc, Haal Benoît, Nguyen Khanh, Manzer Ali, Piekarski Maciej**

**Client : Dallant Justin**

## **Planification du projet**

### **Itération 2 :**

Pour cette deuxième itération, il a été convenu de réaliser les histoires suivantes :

- Histoire 2 : terminer les tâches, 18 points
- Histoire 3 : Prioritisation des tâches, 15 points
- Histoire 8 : Statistiques, 25 points
- Ainsi que 12 points vers la refactorisation du code

Pour cette itération-ci, toutes les fonctionnalités demandées par le client ont été incorporées dans le programme. Comme la réalisation de l'histoire 8 s'est faite plus rapidement que prévu, l'équipe a décidé de déplacer les heures débloquées vers l'amélioration du code. Les détails spécifiques sur la distribution des points sur les différentes tâches complétées peuvent être consultés à partir du document « BurndownChart\_iteration2.ods ». Nous avons malheureusement dû dépasser de 2 points afin de conclure une dernière tâche.

Voici une liste des fonctionnalités qui ont été ajoutées au projet :

- Un utilisateur peut maintenant créer et modifier des tâches pour ses projets, et il peut également sélectionner une date de début et/ou une date de fin. Les tâches peuvent être triées en fonction du temps
- Un nouveau tab affichant les statistiques des projets :
  - L'utilisateur peut voir le temps restant à chacun des projets ainsi que la durée totale de celui-ci.
  - Le nombre de tâches du projet ainsi qu'un diagramme circulaire qui montre les tâches terminées et celles qui sont toujours en cours.
  - Une option pour regarder les statistiques globales : voir le temps total cumulé de tous les projets et les tâches terminées/ouvertes...
- L'utilisateur peut exporter ces statistiques dans un fichier .csv
  - Celui-ci a le choix entre exporter les statistiques d'un projet spécifique ou bien de tous ses projets.

- Le fichier .csv contient les informations suivantes :
- « Username, projet, temps estimé, nombre de tâches »

## **Motivation des choix pris lors de la conception :**

- Model-View-Controller (MVC) : Architecture classique pour la conception de projets visuels utilisant JavaFx. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet et améliore la clarté de l'architecture.
- SQLite : Base de données locales afin de stocker les utilisateurs ainsi que leurs projets. Les raisons principales sont que cette librairie est simple à comprendre et nous n'avons pas besoin de serveur distant.