

## **Organisation de l'équipe**

Après trois semaines à travailler ensemble sur un projet de génie logiciel, nous essayons de prendre du recul sur ce qui a été notre organisation durant tout ce temps, afin de retenir ce qui nous a permis de bien fonctionner, et de trouver des moyens d'améliorer les aspects qui nous ont ajouté du temps de travail inutile. Le problème principal ici, fut que l'organisation générale a mis du temps à se mettre en place, principalement à cause du fait que personne ne se connaissait au premier jour de projet.

Etant donné que ce projet représente pour nous le premier travail d'envergure à réaliser, et que nous n'avons aucune expérience en gestion de projet, les premiers temps furent plutôt difficiles en termes d'organisation. Nous avons surtout du mal à prendre le temps pour identifier les tâches nécessaires pour parvenir à notre objectif de sprint, ainsi que d'estimer le temps nécessaire pour accomplir ces dernières. Ce défaut s'est un peu amélioré dans la suite du projet, mais nous avons encore des difficultés pour estimer correctement les temps des tâches à réaliser, ce qui nous a obligé à travailler en dehors du temps de travail prévu (les week-end notamment). Une organisation efficace dès le début du projet nous aurait permis de mieux répartir le temps passé sur les différentes étapes, même si nous avions dès le départ réfléchi à une organisation de projet.

En effet, nous avons, dès le premier jour, décidé de travailler en suivant la méthode Agile, qui veut que l'on se pose des objectifs à court terme, dans le but d'avoir dès que possible un produit fonctionnel, que l'on améliore ensuite. A chaque itération (appelée sprint, qui dure une semaine ici), on donne une version utilisable du produit qui contient les nouvelles fonctionnalités ajoutées durant celui-ci. Nous avons utilisé le logiciel tiers trello pour noter les différentes fonctionnalités à ajouter pour le sprint en cours, afin d'avoir une vue d'ensemble sur ce qui est fait et qui reste à faire. Cela nous a aussi permis de partager les tâches en fonction des capacités de chacun.

Peu après le début du projet, nous avons remarqué que les différents membres du groupe n'avaient pas la même aisance face aux problèmes que nous rencontrions (2 membres du groupe ne savaient quasiment pas programmer en java). Pour palier à ce soucis, nous avons eu l'idée de différencier les rôles entre ceux qui implantent les fonctionnalités, et ceux qui mettent en place les tests. De plus, lorsque c'était possible, nous avons utilisé le partage d'écran sur discord pour pouvoir coder à plusieurs, ce qui permettait quand on avait moins d'aisance de se faire aider par les autres membres de l'équipe. Cependant, cette manière de fonctionner a eu un coût : les fonctionnalités n'étaient pas testées au moment de leur production, ce qui a mené à de nombreux petits retards car nous devons constamment revenir sur d'anciennes fonctionnalités pour corriger des bugs découverts trop tardivement.

En effet, nous avons relevé que les tests étaient faits beaucoup trop tard pour être vraiment efficaces. La plupart du temps, les tests étaient produits plusieurs jours après avoir implémenté une fonctionnalité, et servaient plus comme tests de non-régression que vraiment comme de tests unitaires. Sinon, la plupart du temps, comme nous n'étions pas spécialisés par étape, cela nous a permis de travailler sur chaque étape l'une après l'autre, sans avoir de soucis liés au fait que certains attendaient le travail de la part d'autres membres de l'équipe pour continuer.

En conclusion, nous remarquons que notre principal défaut réside dans le manque de rigueur de l'application de la méthode agile, ainsi que le manque d'expérience quant à la quantification en temps des tâches à réaliser. Si nous avions à refaire un sprint aujourd'hui, nous passerions probablement plus de temps à planifier le sprint en question que ce que nous avons fait jusqu'ici, en nous appuyant sur les sprints précédents pour donner un délai raisonnable à la création d'une fonctionnalité, en commençant par créer les tests liés à celle-ci, pour ensuite se mettre à programmer.