

Bilan individuel

1 Travailler en mode projet

Au cours du projet génie logiciel, aucune heure de travail n'a été imposé, ce qui demande de l'autonomie et une bonne organisation pour mener à bien ce projet. Il était donc nécessaire de réaliser une organisation efficace, dans le but de répartir les tâches équitablement à travers tous les membres de l'équipe. Avec le rôle de chef de projet qui m'a été attribué, j'ai dû m'investir un peu plus que le reste de l'équipe face à la gestion de l'organisation.

Pour vérifier l'avancement du projet, un planning était réalisé chaque semaine. La création de planning prévisionnel m'a permis de mieux suivre l'avancé du projet et de savoir si nous étions dans les temps. Pour m'assurer du bon déroulement du projet, je me suis investi dans l'ensemble des étapes, même celle dans lesquelles je ne travaillais pas, pour me permettre de comprendre le fonctionnement et mieux pouvoir estimer les temps de travail de chacun. Afin de prévenir les futurs rendus, je prévenais chacun des membres de l'équipe via nos moyens de communications et je les rencontrais chaque jour pour connaître l'avancement de leur partie.

Pour pouvoir gérer le projet proprement, j'ai appris à utiliser des outils externes, ainsi Trello nous permettait d'avoir un tableau numérique en commun sur lequel inscrire des tâches, et les placer dans différentes catégories. Cela nous a permis de communiquer sur l'avancement des fonctionnalités, en créant des catégories "*A faire*", "*En cours*" et "*Fini*". Cet outil permettait de savoir simplement ce qui restait à faire et donc d'augmenter la productivité. De plus, le fait de se réunir chaque matin, pour faire un point sur ce que chacun devait faire, les objectifs que l'on se fixe, et donc de devoir gérer son temps et connaître ses capacités face à ce que l'on s'était fixé.

A la suite du rendu intermédiaire, nous avons constater un manque dans la couverture de nos tests, j'ai donc appris à utiliser des outils externes tels que Cobertura pour améliorer la couverture du programme par les tests et cela efficacement.

2 Mettre en oeuvre des processus de validation

Pour permettre la validation de nouvelle implémentation rapidement pour tous, il était nécessaire de produire des test automatisés. Je me suis occupé de rédiger les différents scripts qui tests le programme, j'ai donc appris de nouvelle connaissance technique en langage Bash. De plus, pour faciliter l'extensibilité des tests, j'ai dû apprendre à découper les tests en sous problème, avec un unique script général appelant l'ensemble des tests dans le bon ordre. Il fallait aussi que ce soit facile à comprendre pour les autres membres de l'équipe, alors j'ai fais en sorte qu'ils n'aient pas à écrire de scripts mais uniquement à produire des fichiers sur lesquels appeler le script.

Pour chaque étape, il était associé une répertoire dans lequel étaient stockés des fichiers textes devant être testés. Chaque fois qu'une fonctionnalité était implémentée, plusieurs fichiers tests étaient réalisés avec l'utilisation de la fonctionnalité en question. Cela m'a appris à avoir des automatisme de test à chaque fois que je produis du code, de plus, si chaque membre réalise la même chose, on construit un grand jeu de tests qui sont pertinents car ils testent toutes les fonctionnalités.

Dans un travail de groupe, avec un dépôt partagé, la qualité de ce qu'on envoie est primordial, la possibilité de corrompre le projet est existante et doit donc être contré. En voulant éviter ce danger, j'ai mis en place une règle simple pour pouvoir envoyer sa version vers le dépôt, si le test global ne fonctionne pas sur la version actuelle alors c'est un danger et ce ne doit pas être ajouté. Le projet m'a mis dans des conditions où j'ai dû réfléchir pour mettre en place cette méthode qui a permis de garder un dépôt sain pour le projet, évitant que certains membre créer des bugs pour d'autres.