

Bilan individuel de compétences

Pocreau Pierre

gl31

Le projet génie logiciel est le premier projet conséquent et complet auquel j'ai participé. De par sa nature, et sa durée, il nécessite la mise en place de méthodes et d'outils de gestion de projet et de travail en équipe.

La première difficulté fut celle de s'imprégner du sujet, de la documentation et du squelette de code qui nous est déjà fourni. La documentation est conséquente, et il n'était pas possible d'avoir une compréhension complète des tâches à effectuer avant de commencer une partie du projet, cela rend aussi difficile la mise en place d'un planning précis. Face au squelette du code, on peut se trouver dans une sorte de paralysie, on n'ose pas modifier ce qui est fait, on ne sait pas si des ajouts sont nécessaires, ou bien si le code déjà présent devrait être suffisant. Avec le recul, je passerai plus de temps à réfléchir à la cohérence globale des éléments que l'on ajoute, par exemple pour la gestion des registres, où notre classe de gestion n'est pas très satisfaisante.

Grâce au projet, je suis désormais très à l'aise avec les outils de développements comme git, ou maven. Ce sont des outils qui s'apprennent par la pratique, et après un mois à les utiliser quotidiennement je ne crains plus les conflits de fusion ou la compilation d'un projet. J'ai aussi appris l'importance d'un bon IDE (Intellij dans mon cas).

Le travail en équipe et la répartition des tâches ont été essentiels pendant le projet, le travail à distance n'a pas rendu la chose aisée, mais l'on s'est bien débrouillé et la séparation du travail s'est faite de manière naturelle. J'ai personnellement travaillé sur les trois parties A, B et C, j'avais donc une compréhension globale du fonctionnement du compilateur, du parser jusqu'à la génération de code. Ainsi, je pouvais discuter avec tous les membres de l'équipe qui travaillaient directement sur le compilateur, faire du code review, aider à déboguer, discuter d'une fonctionnalité...

Cependant, j'ai trouvé difficile d'échanger à distance pour prendre des décisions au niveau de l'architecture du code. La pression des rendus a fait que finalement nous résolvions les problèmes auxquels nous étions confrontés avec la première idée que nous avions, sans qu'elle soit réellement cohérente avec l'ensemble de la structure du problème. Nous sommes repassés plusieurs fois sur des parties du code pour en changer la structure.

Enfin, j'ai appris l'importance d'une base de test solide. Je n'ai pas beaucoup participé à la rédaction des tests, mais j'en ai grandement bénéficié lors du travail sur les différentes parties. J'ai compris que les tests étaient indispensables, à la fois pour s'assurer que le code fonctionne, et pour effectuer des tests de régression lorsque l'on ajoute des fonctionnalités ou lorsque l'on corrige des erreurs.

J'ai pu apprendre de nombreuses choses durant ce projet, des éléments allant de la technique à l'organisation et au travail en équipe. J'ai eu la chance d'avoir une équipe dans laquelle régnait une bonne ambiance, et de travailler avec mon colocataire, Axel Glorvigen, ce qui m'a grandement aidé pour rester motivé tout au long du projet.