

Bilan individuel

Ce projet m'a permis d'acquérir différentes compétences. En effet en tant que Scrum Master, j'ai pu gérer les *ScrumTalks* et le fait de travailler de manière agile et ce, chaque matin. Cette compétence nécessitait des capacités d'écoute, et d'entente au sein de l'équipe.

Travailler en mode projet

La première des compétences à acquérir est de travailler avec des personnes que je ne connaissais pas. Nous nous sommes donc présenté tour à tour afin de mieux connaître les forces et faiblesses de chacun. Le fait que les membres se connaissent est important pour la réussite du projet, et il était aussi vital pour celui ci d'avoir une bonne entente dans le groupe. Grâce à cela nous avons une bonne inertie de groupe ce qui permettait une bonne productivité.

Cette responsabilité nécessitait premièrement une organisation efficace. Ainsi nous avons des horaires fixes pour le *ScrumTalk* pour éviter d'attendre chaque membre du groupe. De plus j'ai beaucoup appris en terme de management, et j'ai pu remarquer que certains étaient plus motivés que d'autres, et que nous n'avions pas les mêmes compétences. Ces inégalités avaient des conséquences lors de la répartition du travail, et il fallait gérer cette répartition. Toujours au niveau de la gestion d'équipe, j'ai donc appris à utiliser Trello, et cela nous a beaucoup servi durant ce projet. En revanche, je n'ai pas été responsable de la gestion du planning, même si, avec ce projet ainsi qu'avec les cours de gestion de projet, j'ai appris à faire un planning cohérent, et cela me servira durant mes prochains projet . J'ai de plus appris à adapter la communication à un auditoire spécialisé, car j'avais les compétences techniques nécessaires à force d'étudier ce projet, et à un auditoire non spécialisé, car la vulgarisation est nécessaire lors d'un projet , il est évident que pas tout le monde n'a les compétences pour comprendre toute la complexité technique d'un projet.

Il a fallu de plus considérer le retard que nous avons pris en début de projet. Celui ci était du à la difficulté de compréhension du sujet, de son objectif ainsi que de son architecture logicielle. Je considère personnellement que ce premier obstacle était du à la caractéristique éducative du projet, et que dans le cas de la création d'un projet professionnel, je n'aurais pas à avoir cet obstacle. En revanche s'il s'agit de l'amélioration d'un logiciel, je considérerais cette difficulté dans le planning.

En revanche le projet n'a pas été totalement fini. Il y a donc des points de gestion de projet à améliorer, et je pense notamment à la répartition des tâches. Dans le cas d'un vrai projet, j'aurais demandé à rendre ce projet en retard plutôt qu'incomplet. J'aurais de plus fait un *burndown chart*, et conservé le Trello qui me semble maintenant indispensable.

Mettre en œuvre des processus de validation

Grâce à ce projet, j'ai pu améliorer mes compétences en Java, que ce soit les mécanismes de l'orienté objet, ou de la maîtrise de l'environnement de développement Eclipse qui nous a permis de voir clairement l'arborescence des classes. Le point où j'ai le plus progressé, (du au fait que je n'en avait jamais implémenté) fut les tests. Durant ce projet j'ai pu découvrir ce qu'était un oracle, et j'ai pu en implémenter quelques uns. Ainsi, en effectuant un squelette de script shell, et en écrivant une base de test qui permet de voir si ce que l'on a implémenter fonctionne, et de découvrir les différents bugs. Avec ce squelette, j'ai pu créer différents scripts shell permettant le test du lexer, du parser, et de la génération de code. Antoine s'est chargé du test de la partie contextuelle.

Pour avoir une base de test, il fallait avant tout connaître et savoir exploiter les fonctionnalités du programme testé. Ainsi en m'étant occupé de la partie de génération de code, je pense avoir su tester correctement cette partie avec une base d'environ 300 tests, qui on permis de déceler différents bugs. Pour la conception d'un programme de test, il était nécessaire que ces programmes soient quasi-unitaires, et, dans la plupart des cas, renvoyait une sortie, comparée avec une sortie attendue. On faisait ensuite la différence entre les deux sorties. Cette base de test me semble pertinente car elle effectue tous les cas ou les instructions sont lancées, et seulement ces instructions sont lancées (mis à part le *print*).

Dans le cas où le test d'un fichier ne passait pas, on affiche la sortie attendue avec la sortie renvoyée, en utilisant celles ci, nous arrivions à déceler le bug. Vu que la plupart des tests étaient unitaires, cela permettait de réduire le nombre de fichier où se trouvait l'erreur potentielle.