

# Bilan individuel de compétences

Mathis Lavigne - Groupe GL27

Le projet de Génie Logiciel (projet GL) est un projet se déroulant tous les ans pendant les 4 semaines du mois de janvier. Cette année, le déroulement du projet GL a été particulier en raison de la crise sanitaire que nous traversons. Ce document explique qu'est-ce que m'a apporté ce projet, notamment au regard de deux compétences fondamentales pour un ingénieur en informatique. La première s'intitule "Agir en professionnel responsable" et la seconde "Comprendre finement les implications des calculs faits par une machine".

## *// Agir en professionnel responsable*

Cette première compétence est au centre de toute gestion de projet. On nous pousse dès le départ, avec des documents à rendre, à intégrer cette notion à notre organisation. Alors qu'avant, la gestion de projet se résumait à "discuter et faire quelques réunions de temps en temps", j'ai appris qu'il y avait bien d'autres aspects tout aussi importants pour bien faire fonctionner une équipe qui est plus grande que ce qu'on a l'habitude d'avoir.

En effet, j'ai trouvé le fait de faire une charte d'équipe très intéressant. Ne connaissant pas toutes les personnes de l'équipe dès le départ, ainsi que leurs compétences, cela m'a permis de savoir quelles sont les faiblesses et forces de chacun et de pouvoir les prendre en compte par la suite. Cela pose aussi un cadre en cas de problème. Un exemple typique pour lequel la charte d'équipe nous a servi est pendant la répartition des rôles au sein de l'équipe. Nous savions qu'untel avait davantage de facilité avec telle technologie, et des difficultés avec tel logiciel. Il était alors possible qu'une autre personne de l'équipe, plus familière avec ce logiciel, lui apprenne à s'en servir. Cela a aussi permis d'identifier nos points faibles et mieux les combler par la suite. Par exemple, aucun d'entre nous n'était familier avec le "développement piloté par les tests" (Test Driven Development). Nous avons donc pris du temps dès le début pour nous familiariser avec ces notions.

J'ai appris qu'un autre aspect de cette compétence repose sur la prise de décision. Il est en effet très rare qu'un projet se déroule comme prévu. Nous avons nous-même eu des imprévus, notamment dans notre gestion de projet. Nous avons eu, par exemple, une baisse générale de motivation au milieu du projet. Plutôt que de rester chacun dans son coin et de se reposer sur les autres, nous avons collectivement pris la décision de faire des séances de projet en présentiel. Cette décision, qui n'était pas du tout prévue au départ, a été prise assez rapidement et a

permis de redynamiser le groupe et de ne pas prendre de retard pour la suite du projet.

J'ai finalement appris que le temps alloué à la gestion d'équipe de projet n'était pas du temps perdu. Au contraire, en plus de gagner du temps et de la productivité sur le long terme, il permet d'éviter les problèmes techniques, comme des bugs ou des oublis. Nous avons pu nous rendre compte en milieu de projet, par exemple, que personne n'avait implémenté les options du compilateur. Grâce à notre planning et nos réunions presque quotidiennes, nous nous sommes rendus compte de ce soucis à temps, et cela n'a pas posé plus de problème.

### *II/ Comprendre finement les implications des calculs faits par une machine*

J'ai aussi beaucoup appris au niveau de cette deuxième compétence. Dans un premier temps, je connaissais déjà le langage Java et avais déjà pu faire des projets avec ce langage de programmation (notamment le projet POO et le projet BDD). Mais j'ai véritablement découvert la puissance d'un langage objet à travers ce projet. En effet, pour ce type de projet, l'architecture proposée par les encadrants a permis de me rendre compte qu'une structure de classe rendait le code bien plus lisible et bien plus facile à maintenir qu'un code procédural/impératif, comme on aurait pu le faire en C. J'ai véritablement intégré la notion d'héritage et de classes/méthodes abstraites, au point de naturellement savoir quand utiliser le mot-clé `abstract` et quand ne pas l'utiliser. De même pour le polymorphisme qui, jusqu'à maintenant, m'était une notion assez obscure. Notion que j'ai pu mettre en pratique de A à Z lorsque je me suis occupé de rendre le projet modulable à l'ajout d'un langage assembleur pour l'extension : j'ai dû imaginer l'architecture en classes et en packages de toute la partie GBA.

Par ailleurs, j'avais aussi déjà fait de l'analyse lexicale, syntaxique, sémantique et contextuelle (en cours mais aussi pendant des projets personnels). Ce projet m'a permis de mettre en pratique ces notions très abstraites pour en faire quelque chose de concret. J'ai, maintenant, une vision beaucoup plus claire de ce qu'est l'analyse contextuelle et quels sont les outils pour effectuer les analyses lexicales et syntaxiques (ANTLR4 ici, mais aussi - m'étant renseigné - Bison ou Flex).

Enfin, n'ayant pas pris d'extension en rapport avec le calcul flottant je n'ai pas acquis autant de connaissances dans ce domaine que dans ceux précédents. Ce projet a tout de même été pour moi l'occasion de me renseigner sur le calcul flottant, afin d'évaluer la faisabilité, dans le temps imparti, d'un moteur de calcul flottant pour l'extension sur la console GBA. Cela m'a permis de prendre conscience qu'il aurait été très compliqué de stocker les nombres flottants dans des registres par exemple (car on a difficilement accès à la mantisse et à l'exposant séparément).

En conclusion, le projet de génie logiciel m'a permis de renforcer des compétences variées, allant de la gestion d'équipe de projet à des notions d'informatique avancées. Bien que cela n'ait pas été évoqué ici, si j'avais pu refaire ce projet, j'aurais accordé davantage d'importance aux tests et à la validation, ce qui nous a pénalisé à la fin du rendu, où nous nous sommes rendus compte trop tard de certains bugs qu'il aurait été facile de corriger avant. Je suis néanmoins satisfait de notre gestion de l'équipe et en retire toutes les leçons dont j'ai parlé dans ce document.