

Bilan gestion d'équipe et de projet

I] Introduction

Le projet Génie Logiciel est un projet fort unique durant notre cursus d'ingénieur au sein de l'Ensimag dans le sens où il délivre une expérience professionnelle proche de celle trouvée en entreprise. Ainsi, nous nous attendions à gagner non seulement en compétence technique, en traitant un sujet difficile sollicitant un apprentissage autodidacte des livrables exigés, mais surtout en gestion de projet et par extension en compétence interpersonnelle, la durée du travail et les conditions à la composition des équipes rendant propices l'assimilation de cette compétence. Dans ce bilan, nous allons étayer nos acquis de cette première expérience de gestion de projet et de travail en équipe.

II] L'équipe

Notre équipe est constituée de cinq membres :

- ❖ CASSAGNES Thomas : 2A IF
- ❖ FIGUEROA Angelica : 2A ISI
- ❖ FLICHY Antoine : 2A MMIS
- ❖ VALLICIONI Mathieu : 2A ISI
- ❖ ZHU Joël : 2A IF

Étant issus de milieux divers, il était indispensable pour nous d'apprendre à se connaître, non seulement entre membres mais aussi soi-même en admettant nos points forts, nos faiblesses ou nos traits de personnalité. En effet, c'est de cette manière que nous avons pu développer de fortes relations qui nous ont permis une communication ouverte durant tout le long du projet. Ainsi, lors de notre première séance ensemble, nous nous sommes présentés tour à tour en utilisant la fiche d'auto-évaluation afin de mettre à jour nos compétences. Nous avons par la même occasion pris le temps de nous concerter sur les modalités de l'organisation du projet en terme d'horaires, de communication, de gestion de conflits, etc. Après cette première séance, il en a été retenu que l'équipe possède des compétences variées. Antoine par exemple est confiant quant à ses compétences en environnement de développement, nous lui avons donc fait confiance pour tous utiliser Eclipse. De plus, d'après les fiches d'auto-évaluation, nous avons remarqué que nous avons des

compétences complémentaires et donc que chaque membre était en mesure d'apporter à ce projet.

III] Organisation

La répartition des tâches s'est faite en fonction non seulement des compétences des membres de l'équipe mais aussi de leur envie. Ainsi, assez naturellement, Thomas et Joël, étant à l'aise en Théorie des langages, se sont orientés vers l'analyse lexicale et syntaxique, programmant en binôme. Antoine quant à lui, souhaitait s'occuper de l'analyse contextuelle. Mathieu, ayant des bases solides en processeur et système d'exploitation a préféré étendre ses compétences en s'attaquant à la génération de code assembleur. Il fut ensuite joint par Joël qui a utilisé ses connaissances en architecture logiciel pour produire un programme qualitatif. Angelica, possédant de fortes compétences en Java, a traité la commande decac. Une fois leur partie respective terminée, Thomas et Antoine ont attaqué l'extension TRIGO. Avec cette répartition des tâches, les parties pouvaient être réalisées en parallèle ce qui était grandement productif. Chaque partie avait un « responsable » et nous avons donc pu de cette façon appliquer la méthode agile efficacement. La priorité fut d'abord de livrer une partie sans objet solide puis d'attaquer l'objet.

Voici un bref historique du projet :

- Présentation des membres et organisation de l'équipe
- Lecture du sujet
- Partie A sans objet et écriture d'un oracle et de tests pour cette partie
 - Une fois la lecture du sujet faite, nous avons souhaité nous approprier ce dernier en commençant relativement lentement en nous familiarisant avec la structure du programme fourni.
- Partie A et B sans objet, écriture de tests :
 - Une fois familier avec le sujet, Antoine débute la partie B. Mathieu écrit les tests pour la partie A afin de s'assurer d'avoir une base de tests solide.
- Partie B et C sans objet, écriture de la commande decac :
 - Lorsque la partie B fut assez avancé, on a jugé pertinent de commencer à traiter la partie C sans objet car celle-ci a besoin de l'analyse contextuelle pour bien fonctionner.
 - Des tests unitaires étaient écrits en parallèle de la programmation.
- Partie A objet, tests pour la partie B :

- Une fois la partie C sans objet bien entamé, Joël a commencé la partie A objet pour prendre de l'avance et permettre aux parties B et C objet de débiter.
- Partie B et C objet, extension TRIGO :
 - Dernière ligne droite, écriture des parties B et C objets et de l'extension TRIGO.
 - Cependant le principe de segmentation de travail que l'on retrouve dans les méthodes agiles à finalement joué contre nous. En effet, par manque de prévision, la structure du programme permettant la génération de code assembleur pour le sans objet n'était pas adapté pour l'objet, ce qui nous a poussé à revoir la structure entière de la partie C et nous a fait perdre un temps considérable.

Temps passé sur les différentes activités :

- Analyse : 20%
- Conception : 10%
- Codage : 30%
- Tests : 30%
- Documentation : 10%

IV] Notre application de la méthode Scrum

En utilisant l'introduction à la méthode agile que nous avons vu en cours, nous avons pu implémenter cette méthode de gestion de projet. De plus, nous avons fait quelques recherches sur cette méthode, et nous avons utilisé l'essentiel de celle-ci. Comme le sujet était conçu avec l'utilisation de cette méthode en tête, il aurait été dommage de ne pas l'appliquer. Nous avons donc mis en pratique la méthode agile, après avoir associé à chacun sa part du travail.

Nous avons appliqué de plusieurs manières la méthode agile. La première chose que nous avons instauré est le daily Scrum. Chaque matin, nous faisons une réunion de vingt minutes debout pour se tenir au courant de l'avancement de chaque membre depuis le dernier Scrum ainsi que ce qu'on comptait accomplir avant le prochain. Grâce au daily Scrum, nous avons pu rester au courant de l'avancement de chacun, et cela nous a donc permis de déduire l'avancement global du projet.

Ensuite, en sachant l'avancement global, cela nous a permis de corriger le planning, et de situer quand est-ce qu'un sous-langage était compilable par notre travail. Cela a permis de voir la totalité du sujet dès le premier rendu. Cependant

nous avons eu quelques difficultés à assimiler le sujet lors de la découverte de celui-ci, ce qui nous a retardé au début du projet.

V] Les règles que nous nous étions fixés

Lors de la première séance ensemble, nous nous sommes fixés quelques règles pour avoir une productivité optimale. La première décision fut de fixer des horaires et un endroit propice au travail. Nous avons choisi des horaires fixes chaque jour entre 9h et 18h, avec une heure de pause le midi. Ce volume d'heures nous paraissait convenable, surtout en rajoutant des heures le samedi. Nous avons choisi l'Ensimag pour travailler, car les logiciels utiles pour la conception étaient déjà installés sur les ordinateurs de l'école. De plus, nous avons décidé de travailler tous dans la même salle avec des ordinateurs proches, car cela permettait de communiquer et d'aider directement.

Pour les échanges de fichier, nous avons utilisé GitLab en utilisant une branche commune, car c'était la solution proposée par l'énoncé, et la méthode la plus pratique pour ce genre de projet. Ainsi, en regardant l'historique des « commits », nous pouvons voir l'évolution de chacun au fil de la journée, ainsi qu'au fil du projet.

En complément, nous avons utilisé le site web Trello; pour voir la liste des tâches qui restaient à faire et les tâches qui étaient faites. Ces premières permettaient de voir l'avancement, tandis que ces dernières donnaient la motivation pour faire celles restantes.

VI] Le respect de ces règles

Nous avons su respecter ces règles, et ce sans difficulté, car cela permettait une routine, ainsi qu'une inertie de groupe qui était propice à la productivité. Nous avons donc, chaque jour, hormis le samedi, respecté les horaires, et nous avons pu travailler le projet sérieusement et de manière régulière. Cette régularité nous a permis de moins stresser face aux échéances car en suivant l'évolution du projet nous pouvions connaître l'avancement et le retard potentiel, notamment en comparant au planning régulièrement mis à jour. Cela laissait donc moins de place à l'imprévu.

VII] Le respect du planning

Une difficulté rencontrée fut la création d'un planning cohérent. Une autre fut le respect de celui-ci. Lors de la création de notre premier planning, vous nous aviez dit, à raison, qu'il était incohérent et inutilisable. Cela était dû au fait que nous ne connaissions pas encore toutes les tâches qui étaient à réaliser, ainsi que la durée potentielles de celles-ci. Sans un planning cohérent, nous étions sûr de rater ce projet, et il a donc fallu en créer un cohérent rapidement. Nous avons énuméré les tâches à l'aide de Trello, ce qui nous a permis d'obtenir un planning cohérent, et adapté à notre avancement. En effet, le retard que nous avons pris au début était à prendre en compte dans le planning, et il fallait donc réorganiser les tâches. Nous avons réussi à respecter ce deuxième planning, et même à prendre un peu d'avance sur celui-ci sur certaines parties, et il en fallait donc un troisième adapté à notre avancement. Cependant, comme on approchait de la date de rendu, ce dernier était assez court, et avec le retard que nous avons pris sur la partie avec objet, il était irréalisable.

Si ce projet avait été un vrai projet d'entreprise, nous aurions demandé à repousser la date de rendu de projet, car nous n'avons pas réussi à le finir totalement. Nous avons au final préféré le rendre incomplet, mais documenté, plutôt que complet et mal fini.

VIII] Les problèmes rencontrés

Au cours du projet, nous avons fait face à différents problèmes. Le premier problème que nous avons rencontré était la difficulté de compréhension du sujet. En effet, lors du premier suivi, vous avez pu constater que cette difficulté a engendré du retard. Il fallait donc premièrement bien comprendre les enjeux et l'architecture du projet avant de coder, et l'étape de compréhension fut plus longue que prévu, et plus longue que ce que permettait ce projet.

Le problème suivant fut une conséquence directe du premier. Cette difficulté à comprendre le sujet engendra un retard que nous ne pûmes jamais vraiment rattraper.

La partie avec objet du sujet étant moins guidée, nous avons moins de pistes et nous savions moins vers où aller.

De plus, nous sommes de l'avis que nous avons appliqué les méthodes agiles trop à la lettre, c'est la raison pourquoi nous n'avons pas pu finir la partie C avec objet. En effet, afin de produire le livrable sans objet, nous avons pensé à une structure du code spécifique qui n'était ensuite pas ré-applicable à la partie objet. La conséquence de cela est que lorsqu'on a commencé la partie avec objet, toute la structure de la génération de code assembleur était à refaire. Hors, par faute de

temps, cela n'a au final pas été possible. Avec un peu de recul, nous aurions pris plus de temps pour penser à une structure de programme solide qui aurait tenu compte de la partie objet également.

IX] Les solutions envisagées à ces problèmes

Après avoir assimilé le sujet, et s'être réparti les tâches, le retard fut en grande partie rattrapé, grâce à diverses raisons. La première était que en associant à chaque partie un responsable, celui-ci avait une connaissance précise de la partie, et il sut donc l'implémenter en conséquence. La seconde était que la méthode agile nous permettait rapidement de voir vers où nous dirigeait le sujet, et comment l'implémenter. La productivité était donc accrue en utilisant cette méthode.

X] Conclusion

Pour conclure, ce projet nous a permis d'expérimenter la gestion de projet, et de montrer les avantages et difficultés de celle-ci. Sur ce projet, nous pensons avoir su travailler en groupe, et avoir su utiliser la méthode agile. Cette méthode de gestion de projet fut une découverte pour nous, et nous a permis de voir un projet dans sa globalité très rapidement, ce qui est vital dans le domaine de l'informatique.