



BILAN SUR LA GESTION DU PROJET

Projet génie logiciel

Equipe : gl07

Naima AMALOU

Yidi ZHU

Jiayun Po

Zaineb Tiour

An Xian

28 Janvier 2020

Table des matières

1	Description de l'organisation adoptée par l'équipe :	2
1.1	La charte d'équipe	2
1.2	Rôles et responsabilités dans l'équipe	3
2	Présentation de l'historique du projet	5
3	Retour d'expérience	6

1 Description de l'organisation adoptée par l'équipe :

1.1 La charte d'équipe

Avant de commencer à travailler, nous avons décidé de mettre en place une charte de travail comme celà nous a été indiqué pour bien se mettre d'accord sur la façon de travail qu'on compte adopter. On a différencier entre deux types principeaux de règles :

Règles comportementales :

- Accepter les contraintes qui nous sont imposées ensemble.
- Respecter la confidentialité des idées et des décisions prises au sein du groupe.
- Participation régulière aux réunions dont on doit décider collectivement de leurs ordre du jour.
- Nous sommes tous co-responsables des décisions qui sont prises vis à vis de ce projet vu qu'aucune décision ne devient effective qu'après qu'elle soit validé par tous les membres de l'équipe.
- Respecter les besoins des autres que ca soit besoin de calme/silence, une pause, de l'aide ...
- Etre ponctuel pour les réunion de l'équipe.
- Chaque responsable s'engage à assurer que l'équipe évolue dans la bonne voie en ce qui concerne son axe de travail.

Et règles techniques :

- Chacun commente très clairement la partie de code qu'il a faite.
- Ecrire des messages de commit bien détaillés.
- Respecter le coding style java.
- S'assurer que le code peut-être compilé avant de pusher sur Git, pour bien suivre les erreurs qu'on a faites à chaque commit ou bien il faut bien noter les erreurs dans le message commit.
- La branche master sur GIT doit être propre et ne contient que les bouts de codes corrects.
- L'objectif de chaque test doit être définie dans le fichier de test.
- Chaque fichier de test est bien placée dans le bon répertoire.
- Avant de prendre une décision de conception, il faut au moins avoir discuter avec

son binôme de tâche, ou avec les autres membres de l'équipe.

- La personne qui teste une partie de code doit être différente de la personne qui l'a développé.
- S'assurer que son code à était lu par une autre personne de l'équipe.

1.2 Rôles et responsabilités dans l'équipe

Pendant la première semaine, nous avons décidé de désigner des responsables pour chaque axe principal de travail sans définir un chef de projet. On ne voulait pas nécessairement établir une hiérarchie dans notre équipe, et nous nous sommes contentés d'avoir une responsable de documentation, une responsable du 'code reviewing', une responsable d'organisation et deux responsables de validation. Le but de désigner différents responsables n'était pas de laissé le responsable tout faire, mais plutôt de définir les tâches à faire qui concerne son axe de travail et les rappeler pendant la réunion du jour. Après deux suivis avec nos encadrants, nous nous sommes rendues compte de l'importance d'avoir un interlocuteur principale et éventuellement un chef de projet pour que quelqu'un ait une vue d'ensemble sur le projet et essaye de le diriger vers la bonne voie. La responsabilité de chaque personne est listée ci-dessous :

- *Chef de projet* : Encadrement général du projet, vigilance aux aspects relationnels du travail commun : communication aisée , gestion et résolution des conflits.
- *Responsable de validation* : Décide et organise la répartition des tests entre les membres de l'équipes.
- *Responsable de documentation* : Prendre en charge le rendu de documentation, rappeler les deadlines, et diviser les tâches de la rédaction entre les membres.
- *Responsable du 'code reviewing'* : S'assurer que chaque partie a été revue par une personne après l'étape de développement et validation. La personne qui a été choisi est normalement une de celles qui ne développe pas la partie et ne fait pas ses tests de validation non plus.
- *Responsable d'organisation* : Organiser des réunions journalières, mettre à jour les tâches sur l'application Trello et le planning.

Pendant l'implémentation du langage "hello world", nous n'avions pas rencontré beaucoup de difficultés vue que la première partie est assez simple. Cette organisation allait bien. Les tâches étaient bien divisées et claires. Malheureusement, le moment où nous avons commencé à implémenter le langage sans objet, nous avons retrouvé quelques soucis de communication entre les membres qui développaient des étapes différentes. Être physiquement ensemble à l'école nous a bien aidé à communiquer, mais nous avons ressenti le besoin d'avoir quelqu'un qui a une vue d'embrée sur le projet et qui connaît bien l'avancement de chaque partie pour nous guider. Pour cela, nous avons donc décidé de changer une responsable de validation en chef de projet et nous avons ainsi gardé cette division de rôle jusqu'à la fin.

Pour ce qui est récapitulation de travail, nous avons décidé de faire une petite réunion chaque jour pour que chacune de nous puisse parler de ce qu'elle a fait et des difficultés qu'elle a rencontré. Ceci a prouvé être une bonne manière de récapitulation journalière.

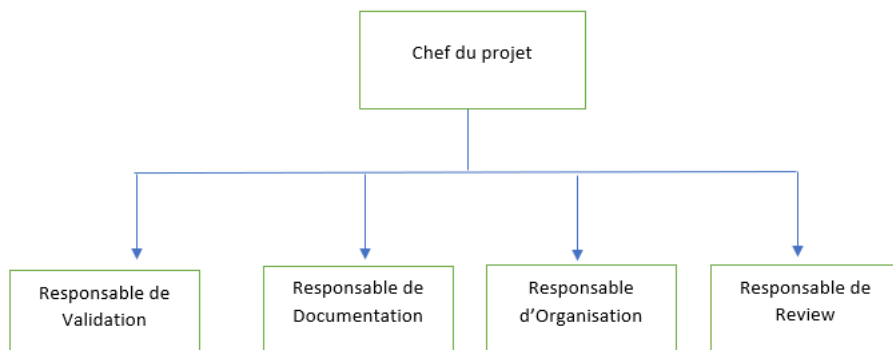
D'après le planning, nous avons pensé à diviser chaque langage en trois étapes : l'étape A est principalement prise par Jiayun Po, l'étape B est faite par Naima Amalou et An Xian, et Yidi Zhu ainsi que Zaineb Tiour sont responsables de l'étape C.

Vue que l'étape A ne prend pas beaucoup de temps, Jiayun Po est responsable de extension aussi.

Nous avons décidé de commencer l'extension en parallèle avec langage sans objet. Et nous avons décider de finir un langage proprement avant de commencer le prochain langage.

Cependant, pendant la réalisation, nous avons trouvé qu'il y a plusieurs taches qu'on peut faire séparément et les regrouper ensemble après, et en plus étape C s'est averée plus compliquée que ce que nous avons pensé. Pour cela, nous avons donné plus de temps à l'implementation de l'étape C qui nous avions prévu et pour ne pas retarder l'avancement, nous avons demandé aux deux personnes qui faisaient l'étape B de continuer leurs travail sur le langage prochain avant les autres finissent l'étape C.

Nous pensons que ce si nous avions à refaire la répartition des taches, nous aurions mis plus de personnes sur l'étape C pour pouvoir bien avancer, ou bien au moins exiger que tout le monde sache ce qu'il faut faire dans cette étape là pour pouvoir la déboguer rapidement lors de l'étape de validation.



2 Présentation de l'historique du projet

Pendant la première semaine, la compréhension du sujet et la lecture du polycopié nous ont pris la plupart du temps. Nous avons commencé l'étape A assez tôt et à la fin de la première semaine nous avons fini notre première langage - le "Hello World" -. Avec les séances de cours que nous avions en parallèle, nous avons réussi à avoir une vue globale sur le projet même si on a pas forcément compris comment construire le compilateur à ce stade là.

Après le premier suivi, nous avons décidé de commencer l'étape B et l'étape C en même temps parce que nous nous sommes rendu compte que les deux parties peuvent prendre un temps de développement important. Mais en réalité, l'étape C a besoin de ce qui est fait dans l'étape B et nous avons donc passé plus de temps à comprendre l'étape C que l'étape B. En effet, on a eu des soucis avec cette parallélisation de tâches, ceux qui faisaient l'étape C ne savaient pas forcément ce qui se passait dans l'étape B alors que par exemple il y'avait bien des fonctions/méthodes implémentées qui pouvait simplifier les algorithmes de l'étape C. Ceci se voit clairement dans la déclaration de méthodes et l'implémentation de la classe DeclField.

Pour l'étape B, on n'a bien compris au départ le fonctionnement des environnements, et on a vite réussi à les implémenter. Une fois nous avons compris le principe avec le langage sans objet, on a réussi à implémenter l'étape B correctement dans 2 jours pour chaque langage et la valider avec plusieurs tests pendant un jour. Au moment du codage de l'étape B nous avons aussi implémenté la décompilation de l'arbre syntaxique pendant une demi-journée pour le langage sans objet, et au fur et à mesure de l'étape B pour le langage Objet.

Pour l'étape C de chaque langage, nous avons passé à peu près une journée à comprendre notre mission et une journée pour dessiner correctement les arbres d'appels et se mettre d'accord sur la conception. Et après nous avons passé environ 3 à 4 journées pour tout coder. Le codage de l'étape C est bien plus difficile que ce que nous avons pensé, et nous avons par conséquent passé plus de temps à vérifier l'arbre que nous avons fait et à réfléchir aux cas spéciaux auxquels nous avons pas pensé précédemment. La partie étant la moins guider par le polycopié, on remettait toujours notre avancement en question et ceci a causé qu'à la fin nous étions pas trop confiantes que ce qu'on a développer couvre tout ce qu'un compilateur correct devrait couvrir.

Pendant le codage, nous avons ajouté les commentaires nécessaires pour gérer la java doc en parallèle. Ce qui nous a aidé à faire la documentation à la fin aussi.

Au tout début, nous avons choisi de faire les tests après avoir fini l'implémentation des codes. Nous nous sommes retrouvées à la fin avec une batterie de tests très faible, et nous étions limitées à tester les fonctionnalités qu'on avait implémenté et non celles

que notre compilateur devrait avoir. Après le premier suivi et surtout après le rendu intermédiaire, nous avons trouvé que le manque de tests a gravement influencé la qualité de notre programme. C'est la raison pour laquelle on a décidé qu'avant de commencer le langage Objet d'écrire les tests en premier et de coder après et de surtout ne pas hésiter à ajouter des tests au fur et à mesure si on en a besoin.

Notre adaptation du développement test-driven nous a permis de ce rendre compte des failles dans notre compilateur au fur et à mesure de notre codage.

Nous avons choisi de demander à une personne de chaque étape de prendre à peu près 30 minutes à 1 heure pour compléter la documentation de conception chaque jour. Et l'autre peut soit continuer sur le développement, soit relire encore une fois le code et le nettoyer/vérifier en lançant les tests et en notant les remarques.

3 Retour d'expérience

C'était une des premières expériences de travail en groupe de cinq personnes et sur un projet qui est bien plus long que les projets auxquels nous nous sommes habitués durant notre cursus scolaire.

Au début du projet nous n'avions pas vite réussi à savoir comment se lancer dans le développement du compilateur en soi, mais nous avons pu nous retrouver juste au bon moment pour réussir à faire un bon départ. Chacune de nous a su retrouver sa place selon ses compétences et ses préférences grâce aux nombreuses réunions qu'on a fait au début où chaque personne était libre de s'exprimer sur le projet en général. On s'est donc répartie les responsabilités, et on a réussi à participer de manière égale aux décisions vis à vis du projet pour finalement aboutir à une équipe auto-organisée.

Nous avons gardé une bonne ambiance et solidarité entre les membres de l'équipe tout au long du projet, et ceci même pendant les moments de fatigue et surtout de doute.

Le retour intermédiaire était un point marquant dans le développement de notre compilateur, il nous a mis face à nos fautes et nous a permis de s'auto-corriger.

Pendant la dernière semaine du projet, la quantité du travail était assez importante (développement de l'étape C, L'extension et jeux de tests) et nous étions très limitées par le peu de temps qui nous restait. Ceci a légèrement baissé le moral de l'équipe, mais nous avons pu tenir et travailler jusqu'aux derniers moments pour essayer d'avancer au maximum à fin de pouvoir rendre un produit fonctionnel.