

Bilan de gestion d'équipe et de projet

Sommaire :

Organisation adoptée dans l'équipe.

Outils d'organisation et de communication.

Présentation de l'historique du projet.

Difficultés rencontrées et solution apportées.

Rôle de chaque membre de l'équipe.

Bilan de qualité.

Conclusion générale.

Organisation adoptée dans l'équipe

- Nous avons choisi de commencer par l'analyse des besoins, suivie de la conception, puis du codage et enfin de la validation et de la documentation. À noter que nous avons fait l'architecture des tests en parallèle du code, et que nous avons commencé la documentation.
- Nous avons utilisé Google Drive pour stocker les documents importants et Trello pour suivre les tâches. Ces deux outils sont accessibles et modifiables par tous les membres. Ils facilitent donc la transmission d'informations. Chaque personne de l'équipe se connectait régulièrement pour se tenir informée.
- La répartition des tâches s'est faite en fonction des compétences de chacun, malgré les différences de niveaux au sein de l'équipe. Nous avons su partager les tâches et s'entraider quand il le fallait. Personne n'est resté sans rien faire ou n'a bloqué trop longtemps sur une partie.
- Nous avons maintenu une bonne motivation tout au long du projet grâce à une communication efficace et une répartition équitable des tâches. Chacun s'est senti utile à l'avancement du projet et à pris son travail à cœur. Nous avons réussi à garder une bonne ambiance et un bon environnement de travail, qui nous donnaient envie de venir le matin et de travailler ensemble.

Outils d'organisation et de communication

- Nous avons choisi de ne pas utiliser Jira pour gérer les tâches et les bugs, car nous avons trouvé qu'il était plus adapté pour des projets de grande envergure. Au lieu de cela, nous avons mis en place Trello pour gérer les tâches et suivre l'avancée du projet.
- Nous avons également mis en place un système de Daily checkup pour assurer une coordination efficace au quotidien. Tous les matins en arrivant nous faisons une réunion. Chaque membre rappelait ce qu'il avait fait la veille, son avancée, les problèmes rencontrés et ce qui restait à fixer. Nous pouvions alors nous adapter et nous organiser pour la journée. Nous avons rapidement compris qu'il est essentiel de bien communiquer et de savoir ce que les autres font à côté de nous. Cela permet d'éviter de travailler sur les mêmes choses ou de prendre des décisions seul qui pourraient bloquer le travail d'autres membres.
- En ce qui concerne la communication, nous avons choisi de travailler dans le même espace physique et de disposer d'un serveur Discord pour les discussions en temps réel. Nous avons organisé les channels sur le serveur Discord pour faciliter la communication et la coordination au sein de l'équipe, en utilisant des channels dédiés pour les échéances, les documents à rendre, les demandes de corrections, la répartition des tâches quotidiennes, le découpage du projet, les liens de documentation importants, et les messages

généraux. Cela montre que l'équipe a utilisé des outils de communication efficaces pour gérer le projet de manière organisée.

Nous avons ainsi pu faciliter la communication et la coordination au sein de l'équipe. Le channel "deadline" contenait les informations sur les échéances, ce qui nous a permis de suivre les étapes clés du projet et de nous assurer que nous étions en bonne voie pour les rendre dans les délais impartis. Le channel "a-rendre" contenait des informations détaillées sur les documents à rendre, y compris les dates de remise, les nombres de pages et les contenus requis, ce qui nous a aidé à maintenir un bon niveau de transparence et de collaboration. Le channel "fix" permettait aux testeurs de faire des requêtes aux développeurs pour les correctifs, ce qui nous a aidé à résoudre les problèmes rapidement et efficacement. Le channel "day-stuff" contenait la répartition des tâches quotidiennes entre les membres de l'équipe, ce qui nous a permis de nous organiser efficacement et de nous assurer que tout le monde était impliqué et motivé. Le channel "découpage-projet" contenait les informations sur le découpage des projet (user stories) et nous a permis de visualiser les différentes étapes à venir. Le channel "documentation" contenait les liens importants des documents de référence (site du projet, google drive, gitlab, trello, etc.) qui étaient utiles pour la réalisation du projet. Cela a permis de maintenir une bonne gestion de la documentation et de faciliter l'accès aux informations. Enfin, le channel "général" était utilisé pour les messages généraux, tels que les heures d'arrivée le lendemain matin, ce qui a permis une bonne coordination entre les membres de l'équipe. En somme, l'utilisation de ces channels sur le serveur Discord a été un élément clé de notre organisation et de notre coordination au sein de l'équipe, et a contribué à la réussite du projet. Cela montre que l'équipe a utilisé des outils de communication efficaces pour gérer le projet de manière organisée et efficace.

Document *Issues*

L'utilisation d'un document "Issues" sur Google Drive a été une autre stratégie efficace pour gérer les problèmes et les demandes de correction au cours du projet. Il a été utilisé en complémentarité avec le channel #fix du serveur Discord pour avoir une vue d'ensemble des problèmes à résoudre et des demandes de correction. Ce document a permis de maintenir une trace des problèmes rencontrés, de leur statut de résolution et de l'historique des modifications. Il a également permis d'assigner des tâches aux membres de l'équipe pour résoudre les problèmes, et de suivre les progrès de ces tâches. En utilisant ces outils de manière complémentaire, l'équipe a pu gérer efficacement les problèmes et les demandes de correction tout au long du projet, contribuant ainsi à la qualité et à la livraison du projet dans les délais impartis.

No	Issued by	On branch	Day	At time	Description	Fixed by	On branch	At time	Status
4	Lucas	develop	22/01	15:30	Fix: Décoration de la taille des tableaux (on va vérifier si l'expression est entre crochets alors	Felix	develop	16:20	done
5	Lucas	develop	22/01	16:40	Changer le type (int) de 'scalar' de multScalar en float (ArrayFunction.decah)	Camille	develop	16:45	done
6	Lucas	develop	22/01	17:00	Dans ArrayFunction.decah remplacer les 'assert' par des if				in work
7	Felix	all	22/01	17:26	cast les float tres grand vers des entiers (erreur ?)				in work
8	Felix		22/01	19:45	Modifier 'search' pour qu'il renvoie la position du 1er élément				open

Présentation de l'historique du projet

Nous avons choisi de réaliser les étapes B et C en parallèle afin de pouvoir avancer plus efficacement ce qui nous a permis de pouvoir réaliser de premier test et fix sur la partie C pendant que la partie B avançait. Le temps sur chaque activités à été réparti de la manière suivante :

- analyse : cette partie a été assez rapide pour le langage complet car toutes la grammaire et les spécificités du code étaient indiquées dans le polis, elle a duré environ trois jours.
- conception : pour cette activité nous avons passé pas mal de temps sur la façon de créer de nouvelles classes afin de respecter l'architecture déjà fournie, elle a duré environ trois jours.
- codage : ce fut la partie la plus longue, elle a duré six jours
- validation : cette partie a été aussi longue car il a fallu créer une architecture de test puis réfléchir sur la couverture des tests afin de pouvoir tester la majorité du compilateur créé. Elle a duré six jours
- documentation : la documentation fut assez courte à écrire puisque nous étions cinq dessus, cela avançait assez vite. Elle a duré quatre jours.

Difficultés rencontrées et solution apportées

Lors de ce projet, nous avons pu rencontrer des moments difficiles causés par les deadlines serrées et la charge importante de travail. Même si nous avons su réagir rapidement et bien gérer chaque situation, le rendu final s'est avéré compliqué avec git. Nous nous y sommes pris tardivement et la connexion n'a jamais été aussi mauvaise, nous n'arrivions plus à push et pull. Une tension s'est installée, nous devons gérer les derniers conflits rapidement. Malgré la situation compliquée, nous sommes restés calmes et avons persévéré, chacun cherchait une solution et c'est pourquoi nous avons rapidement résolu le problème.

Rôles

- Félix a pris le rôle de Scrum Master au début, mais il a principalement travaillé sur le développement de l'étape B de vérification contextuelle et l'étape C de génération de code. Il a également travaillé sur l'extension (tableaux) et les options de compilation du compilateur.
- Yoann était surtout responsable de l'étape A d'analyse lexicale et syntaxique du compilateur. Il a également travaillé sur la phase C de génération de code en aidant Félix, sur l'extension (tableaux et matrices), les tests pour cette extension, et a fait presque toutes les options de compilation.
- Julien a été responsable des tests. Il a travaillé sur la création de scripts pour l'automatisation de la procédure d'intégration pour le développement, les tests unitaires JUnit, les tests pour la partie sans objet et avec objet, et les scripts pour les tests de l'extension.
- Camille a travaillé sur une grosse partie de l'étape B de vérification contextuelle de la partie du langage sans et avec objet. Elle a également travaillé sur l'extension, les fonctions associées aux tableaux et aux matrices, la méthode de Gauss Jordan. Elle a créé des tests pour l'extension (les tableaux basiques) et fait beaucoup de review de code et de tests.
- Lucas a rédigé une grande partie de la documentation du logiciel : documentation des tests, documentation de conception et documentation utilisateur. Il a également travaillé sur la couverture de tests (Jacoco) et créé des tests sur la partie du langage avec et sans objet, surtout sur la vérification contextuelle. Au début du projet, il a aussi participé à l'implémentation du lexer et du parser.

Bilan de qualité

Les tests unitaires JUnit ont été effectués par Julien, ils ont permis de valider les fonctionnalités du compilateur. Les scripts d'automatisation de la procédure d'intégration pour le développement ont été créés par Julien. La couverture de tests (Jacoco) a été mise en place par Lucas. La documentation utilisateur, la documentation de conception et la documentation des tests ont été rédigées par Lucas.

Conclusion

Notre équipe a mené à bien ce projet grâce à une organisation efficace, une bonne communication et une répartition des tâches adaptée. Chacun des membres de l'équipe a joué un rôle clé dans la réalisation du projet. Il y a eu des difficultés rencontrées tout au long du projet, mais nous avons réussi à les surmonter grâce à une bonne communication et une répartition des tâches adaptée. Pour les projets futurs, nous pourrions envisager d'utiliser un outil de gestion de projet différent, ou d'améliorer notre système de Daily checkup pour encore plus de coordination et continuer à améliorer notre système de tests pour assurer une qualité du logiciel toujours plus élevée.