



Bilan individuel

Antoine Briançon

*Équipe n°31 : Antoine Briançon, Axel Glorvigen, Thibault
Launay, Maxime Martin et Pierre Pocreau*

Au fil de ce bilan personnel, je vais développer deux points essentiels qui ont été les plus importants pour moi tout le long du projet. D'abord le travail en mode projet, ensuite la mise en œuvre de processus de validation efficaces.

Travail en mode projet:

Au tout début du projet, la taille du cahier des charges m'a effrayé. Le fait est que c'était la première fois qu'on devait traiter un problème aussi complet du début à la fin. Fort heureusement, notre équipe a su s'organiser assez vite et, une fois les tâches distribuées au sein du groupe, il était clair que chacun d'entre nous n'aurait pas à lire l'entièreté du cahier des charges.

Cette spontanéité dans l'organisation de mon équipe m'a aidé à avoir à la fois une vision globale du projet grâce aux échanges qu'on avait régulièrement et la possibilité de pouvoir me concentrer sur ma partie afin de la maîtriser le plus possible. Si on avait dû se surveiller entre nous, nous n'aurions jamais pu vraiment pousser nos parties jusqu'au bout et la perte de temps engendrée aurait été fatale pour l'ensemble du projet.

Ainsi, n'étant à la base pas quelqu'un qui apprécie le travail d'équipe puisque je me retrouvais bien souvent à devoir passer derrière mes collègues pour combler leur manque de rigueur ou de motivation, j'ai finalement pris goût à ce mode de fonctionnement où l'équipe devient un tremplin et un moyen efficace de s'entraider.

Régulièrement (presque tous les jours) nous faisons des réunions en vocal pour nous tenir informés de l'avancement de chacun, déjà parce que chaque partie dépendait des autres, ensuite car on ne pouvait se résoudre à laisser un des membres de l'équipe tourner en rond et se frustrer seul dans son coin.

Nous avons mis en place différents outils de planification, notamment un diagramme de Gantt assez complet pour rester conscient du temps qu'il nous restait. L'utilisation de l'outil Issues de GitLab a aussi été très utile pour savoir sur quoi travaillait actuellement chacun des membres. Malgré tout nous sommes restés souples sur la manière d'organiser nos différentes tâches, sachant qu'on ne pouvait pas exactement savoir à l'avance quelle était la durée de développement de chaque étape du projet. La communication est restée l'outil le plus efficace de planification.

Enfin, il était important que chacun ait une idée solide de ce que les autres faisaient. J'en parlerais plus en profondeur dans la deuxième partie mais, étant principalement chargé de produire des tests pour les différentes parties du projet, j'ai éclairci le plus possible les scripts en ajoutant des explications précises sur ce que ces derniers faisaient concrètement.

Processus de validation:

Étant le principal développeur de tests dans le projet, j'ai dû réfléchir à des manières efficaces d'automatiser ces derniers, de les organiser de manière lisible et surtout d'obtenir une couverture satisfaisante pour chacune des parties testées.

Concernant l'automatisation, je me suis largement inspiré des scripts de base fournis en ajoutant des boucles for sur des dossiers précis. Chaque partie a ses restrictions et manières de vérifier si un programme est valide ou non. La rédaction des tests passait donc avant tout par une réflexion sur papier.

J'ai dû réfléchir à la meilleure manière de rendre l'exécution des scripts de tests lisible pour les développeurs des parties concernées. Par exemple, pour tester la génération de code, si un programme de test ne renvoie pas la valeur attendue, le script envoie un message à l'utilisateur lui indiquant la valeur attendue et ce qui est réellement sorti. C'est un détail mais je sais que ça a aidé mes collègues lors du développement du compilateur.

L'organisation des tests a été pensée pour que n'importe qui puisse facilement s'y retrouver et déduire rapidement l'utilité de chaque programme. Pour cela, j'ai veillé à ce que chaque programme de test possède un encart d'introduction avec une description rapide du programme, le résultat attendu lors du lancement du script et une date de création. De plus, chaque fichier de test porte un nom qui donne une première idée de ce qu'il contient. L'arborescence a également été réfléchie pour clarifier l'organisation globale des tests tout en évitant le surplus de dossiers.

Enfin, pour respecter au mieux la couverture des tests, j'ai dû prévoir un maximum de cas possibles. Pour cela, j'ai d'abord lu et relu les règles de syntaxe, de contexte, etc. afin d'établir une liste exhaustive de cas valides et invalides pour chaque partie. Le principe était de comprendre chaque règle et d'imaginer toutes les façons possibles de les respecter et de les briser. De là j'ai écrit une première batterie de tests. Entre-temps, je demandais régulièrement à mon équipe s'ils avaient des idées de tests supplémentaires auxquels je n'aurais pas pensé.

L'outil Jacoco a également été d'une grande aide une fois que les premières batteries de tests étaient validées. Je pouvais directement voir dans le code du compilateur

ce qui n'avait pas été touché par les tests et réagir en conséquence pour tâcher d'étendre la couverture totale. Les multiples retours de la part de nos tuteurs ont aussi été utiles car ils donnaient des idées de tests qui n'avaient pas été implémentés.

Conclusion:

Ce projet m'a bien plus apporté que ce que je pensais avant de l'entamer, que ce soit en termes de travail d'équipe ou de rigueur personnelle. Travailler exclusivement sur un projet pendant presque un mois permet de mieux se rendre compte de la réalité future qui sera sûrement la mienne.

Si c'était à refaire, j'évitais quelques mauvaises manières de faire comme les changements de dernière minute avant un rendu et mettrais encore plus l'accent sur la communication et l'entraide, qui sont deux piliers indispensables pour avancer en équipe.