



Bilan de gestion d'équipe et de projet

Projet GL

Table des matières

| | |
|--|----------|
| Descriptif de l'organisation d'équipe | 3 |
| Historique du projet | 4 |

Description de l'organisation d'équipe

Au niveau de l'organisation d'équipe nous avons choisis d'utiliser une méthode agile, en répartissant le projet sur 3 sprints, un pour le langage "Hello world", un pour le langage "sans objet" et un dernier pour le langage avec objet. Une exception a été faite pour le lexer qui a été complètement fait lors du premier sprint car il représente une plus petite part du travail et ne demande pas de tâches accomplies au préalable contrairement au parser qui dépend du lexer pour la génération des tokens, l'analyse contextuelle et la génération de code qui demandent d'avoir la bonne architecture en sortie du parser pour pouvoir être réalisées. Le planning était revu au début de chacun des sprints et donnait une estimation grossière du temps associé à chaque tâche et la répartition des membres sur chacune des tâches

Chacun de ces trois sprints comportait donc une étape de Parser, d'analyse contextuelle, de codeGen ainsi que des tâches de tests à réaliser sur chacune de ces étapes. Ces étapes étaient elle même divisées en sous-étapes correspondant aux sous-langages de la partie du langage associé au sprint. L'équipe comportant quatre membres actifs, nous avons choisis de répartir le travail de façon à ce qu'un membre travaille majoritairement sur une étape (A, B ou C et les tests) avec les autres membres de l'équipe qui doivent review le code lors des merges requests sur le gitlab afin que chacun puisse comprendre l'intégralité du projet tout en étant plus efficace en travaillant sur une branche avec laquelle il est familier. Certaines étapes demandant moins de travail que les autres (par exemple le parser par rapport à la génération de code) une équipe ayant fini ses tâches dans le sprint rejoignait une autre équipe pour aider à finir ses tâches et ainsi de suite. Nous avons essayé au maximum d'appliquer un travail en parallèle bien qu'avec les dépendances du projet, cela était parfois impossible et nous avons ainsi dû adapter en laissant deux personnes travailler sur une étape puis l'autre.

Au niveau de l'organisation de travail nous ne nous sommes pas fixé une méthode de travail d'équipe. Ainsi certains membres ont passé des journées dans les locaux à coder côtes à côtes tandis qu'un autre membre travaillait sur place le matin et en distanciel l'après-midi. Certains commençais la journée à 8h, d'autres la finissaient après 23h. Nous avons également réalisé des stand up meetings le matin lorsque tous les membres de l'équipe étaient réunis mais ces derniers ne se sont pas montrés très concluants car ceux qui travaillaient côte à côte savaient ce que les autres avaient fait, rendant obsolètes la plupart des informations partagées. Cette organisation hétérogène a également été cause de certains problèmes de communication au niveau de l'équipe, certains membres ne pensant pas à demander de l'aide aux autres ne se trouvant pas à côté d'eux, ou bien quelques tâches qui ont été réalisées en doublon. Cela aurait dû être corrigé en adaptant la même organisation de travail pour tout le monde.

Les problèmes de communication ont d'ailleurs été l'un des gros problèmes rencontrés par l'équipe car certains membres ne demandant pas de l'aide aux autres membres du groupe se sont retrouvés parfois à travailler tout le week end jusque le lundi matin 5h pour réaliser certaines tâches qui auraient pû être faites en parallèle par le reste de l'équipe qui ne savait pas qu'ils avaient besoin d'aide.

Historique du projet

Comme énoncé précédemment, la première étape réalisée fût le lexer après quelques heures de travail. Pour chaque étape du projet, au moins un membre de l'équipe travaillait en permanence sur la validation afin de pouvoir faire remonter et gérer les bugs le plus tôt possible ce qui permettait entre autre d'avoir fraîchement en mémoire le codage de la partie problématique lors du débogage et pouvoir corriger plus facilement le bug que dans le cas où la partie avait été faite plusieurs jours auparavant.

Au niveau des étapes A,B et C, l'analyse syntaxicale était toujours la première tâche finis, généralement pendant l'étape de conception de l'analyse contextuelle et de la génération de code, puis le codage de l'analyse contextuelle en s'appuyant sur l'architecture fournie par le parser, qui permettait de conclure la conception de la génération de code avant la réalisation du codage de cette dernière. Au niveau de la répartition du temps pour les équipe associée, environ 15% du temps était consacré à l'étape A, 30% du temps était associé à l'étape B et 55% du temps à l'étape C de par sa complexité et les corrections nombreuses que l'on devait réaliser sur l'architecture complète du code pour pouvoir gérer l'état de la pile, des registres, la gestion de l'extension ARM...

Pour les documentations nous avons des fichiers google docs enregistrés sur un drive accessible et modifiable par chaque membre du groupe afin de simplifier leur complétion. Les documentations d'extension, de conception et de validation ont été complétées par les équipes travaillant sur chacune de ces étapes lorsque celle-ci le pouvaient (à la fin d'une tâche du sprint par exemple).

La documentation utilisateur a quant à elle était majoritairement réalisée par les personnes s'occupant de l'étape B en faisant remonter les erreurs au fur et à mesure que celle-ci s'ajoutaient. La partie documentant le mode opératoire de l'extension a été remplie par la personne s'occupant de ARM et les limites de notre compilateur et de notre extension ont été remplies par l'équipe entre le rendu final et le rendu du manuel utilisateur.