

Bilan projet GL

Florentin Gonthier Germain Geoffroy Matthieu Bergeron
Adrien Fischman Matthias Beaupere

January 26, 2016

1 Répartition des tâches et organisation de l'équipe

1.1 Réception du projet et mise en place des méthodes agiles

A la réception du projet, il nous semblait essentiel que chacun comprenne le sujet dans son intégralité afin d'avoir une vision globale des tâches à effectuer. Dès lors, après une première lecture complète, nous avons pu mettre en place des méthodes Agiles.

Nous avons tous ensemble, sans répartition préalable, analysé les premiers éléments qui ont constitués notre premier sprint. Nous avons eu recours à Acunote, un site de management de projet agile en ligne, pour mettre en forme un premier backlog accompagné du sprint pour l'implémentation de Hello-World en définissant différentes user-stories.

Nous avons décidé de ne pas désigner de Scrum Master. En effet, nous avons

No.	Description	Owner All	Status All	Est.	Rem.
27	HW En tant qu'utilisateur, je veux que le compilateur decac compile mon Hello World	-		25	0
29	HW, Lexer En tant qu'utilisateur, je veux que mon fichier Hello World soit convertis en tokens	Germain	Completed	5	0
30	HW, Parser En tant qu'utilisateur, je veux que mon fichier Hello World soit analysé afin que les erreurs de syntaxe soient détectés via la commande decac -v (2 comments)	bergerom	Completed	8	0
31	HW, Parser En tant qu'utilisateur, je veux obtenir un arbre syntaxique à partir de mon fichier Hello World via la commande -p (1 comment)	matthias	Completed	0	0
32	HW, Semantic En tant qu'utilisateur, je veux des vérifications de contexte avant compilation via la commande -v	fischmaa	Completed	10	0
33	Assembleur, HW En tant qu'utilisateur, je veux que mon fichier Hello World soit compilé via du code assembleur. (2 comments)	FlorentinG	Completed	2	0
Click here to add a new task					
				Total: 25	0

Figure 1: Sprint Hello-World, Acunote.

compris que la bonne communication au sein du groupe serait suffisante pour l'organisation des méthodes agiles et la répartition des tâches au cours des stand-up.

Ainsi, la répartition des tâches pour le premier sprint s'est fait sans difficulté : chacun a choisi de couvrir un user-story selon ses affinités avec le sujet. Matthieu

et Germain se sont occupé du lexer, Matthias du parser, tandis qu’Adrien commençait l’analyse contextuelle et que Florentin se penchait sur la compilation.

La fin de ce premier sprint a été déterminant dans le fonctionnement de notre équipe. Tous les membres de l’équipe ont fournis un travail pertinent, ce qui a instauré une confiance générale au sein du groupe. La confiance vis-à-vis du travail des autres membres de l’équipe a permis à chacun d’entre nous de se spécialiser sur un aspect en particulier du projet. Notre bonne communication permettait de s’aider mutuellement, et de demander un avis extérieur sur notre propre travail sans remise en cause de sa qualité.

Matthias a rejoint Adrien sur l’analyse contextuelle, Germain débutait la classe Math pendant que Matthieu et Florentin travaillaient sur le langage assembleur.

1.2 Avancement du projet et spécialisation

Suite au rendu du sprint Hello-World, Adrien a rejoint Germain sur la classe Math. Nous avons pris la décision de nous spécialiser chacun sur une partie différentes du projet. Cette décision a été prise par le groupe tout entier, elle avait pour objectif de maximiser les compétences des membres du groupe. Chaque membre de l’équipe travaillait avec un binôme privilégié : Adrien et Germain travaillait ensemble sur la classe Math, et Matthieu assurait la liaison entre l’analyse contextuelle de Matthias et la partie assembleur de Florentin. Ainsi, concrètement des sous équipes se sont distingués au sein du groupe. Ce choix de spécialisation a été permis par la confiance instaurée préalablement et nous a permis à tous de gagner en compétences pointus sur un domaine en particulier.

La spécialisation des membres de l’équipe a impacté notre organisation agile pour le projet. La partie Math et la partie liée au compilateur étaient dorénavant séparées. Dès lors, les sprints ne concernaient plus qu’une partie à la fois. Le travail en binôme permettait une synchronisation permanente des avancées et des problèmes rencontrés, ce qui facilitait grandement la répartition des tâches futurs au sein des sous équipes.

2 Historique du projet

Notre projet s’est articulé autour des méthodes Agiles. Nous avons choisi d’effectuer un total de trois sprints, correspondants à des avancements importants dans le projet :

1. Le Hello World
2. Le rendu sans-objet
3. Le rendu final

La partie maths, qui était indépendante du reste, ne faisait pas partie de ces sprints.

Lors de chaque sprint, les parties B et C étaient développées en parallèle. Notre répartition des tâches nous imposait donc une communication constante entre les responsables des étapes B et C.

Nous avons essayé, tant que possible, d'utiliser des techniques de "Test Driven Development". Les tests étaient donc écrits avant de commencer à coder une fonctionnalité. Nous nous sommes aussi efforcés de créer des binômes testeur/programmeur : sur la partie C par exemple, Matthieu a fait la plupart des tests pour le code de Florentin.

Une grande partie de notre travail s'est donc portée sur la validation des tests. Pour chaque étape, des scripts de validation automatique ont été mis en place, ce qui nous a permis d'éviter d'introduire de nouveaux bugs. Cette démarche a été l'un des points forts de notre projet, puisque cela nous a permis d'avancer par incréments, et donc de gagner en agilité. C'était aussi un point positif pour le moral général de l'équipe, puisqu'on voyait notre compilateur s'améliorer de jour en jour.

Avec du recul, nous sommes d'accord pour dire que nous avons peut-être négligé les étapes d'analyse du projet. Cela nous a posé quelques problèmes, notamment pour la partie C qui exigeait un travail préalable sur la définition des conventions et des structures de données.

3 Conclusion

Florentin :

Pour la gestion du projet, je suis parti trop vite sur la partie C sans prendre assez de recul sur les éléments qui étaient à demander. Cela s'est ressenti lors du debug de la partie C. En effet, toutes les fonctionnalités n'étaient pas implémentées ou complètes.

Si j'avais commencé par faire un cahier des charges précis de ce à quoi devait ressembler le compilateur, je pense que le compilateur aurait été plus complet. Les opérations à faire sur la partie C n'ont pas été détaillées, et si le projet était à refaire, je ferais cela en premier. Sinon ce projet m'a permis d'améliorer ma compréhension de l'assembleur, même si cela restait encore de l'assembleur abstrait (à travers `ima`).

Matthieu :

Le début du projet m'a paru très compliqué, puisque j'ai essayé d'avoir une vision globale du fonctionnement du compilateur. Ce choix s'est avéré payant sur le long terme, puisque j'ai réussi à naviguer entre la partie C et l'écriture de tests.

Si il fallait changer une chose dans notre organisation, je pense qu'on aurait dû effectuer des séances de pair-programming au début des sprints. En effet cela nous aurait permis de mieux comprendre le code des autres, et surtout de se

mettre en accord sur certaines conventions.

A la fin du projet, j'ai compris comment fonctionnait un compilateur dans la globalité, c'était mon objectif et j'en suis très satisfait. Sachant que j'ai travaillé sur la partie C, j'ai aussi acquis une connaissance approfondie de la génération de code assembleur. Enfin, j'ai pu apprendre des techniques de tests avancées pour des projets de grande envergure.

Adrien :

Rétrospectivement, bien que le projet se soit très bien déroulé, je pense qu'une autre organisation, notamment en début de projet aurait été bénéfique. En effet, avant que chacun se spécialise sur une partie en particulier, j'ai cherché à comprendre le sujet dans son intégralité. Je pense que ce fut une erreur car une perte de temps. Il aurait suffi de survoler l'intégralité du sujet et d'attaquer une partie directement. Le sujet est trop vaste pour être compris au premier abord. Je pense que j'aurai gagné en productivité si je m'étais consacré directement à une partie spécifique.

Cependant, si je pense que la spécialisation rapide des membres d'une équipe est une bonne chose, il faut qu'elle soit accompagnée de nombreuses et régulières revues de codes en équipe. Ainsi, chacun serait à même d'expliquer sa partie : l'avancement du projet, les difficultés rencontrés et l'explication du code. Dès lors la spécialisation rapide permet aussi de faciliter la compréhension globale pour le groupe, car un membre spécialiste d'une partie doit pouvoir l'expliquer clairement à un autre membre de l'équipe.

De plus, avec d'avantage de recul, j'imagine qu'un Scrum Master pourrait apporter son aide sur un tel projet. En effet, si quelqu'un avait eu une vision globale et pertinente sur le projet tout entier cela nous aurait permis d'avancer plus sereinement. Toutefois, le travail d'un tel Scrum Master aurait dû être fait en amont du projet afin de guider l'équipe dès le commencement, ce qui n'est pas évident pendant le projet GL.

Germain

Le choix d'implémenter la partie est math est bien d'un côté car comme l'on est pas du tout guidé, on doit vraiment se débrouiller tout seul, chercher des documentations pour avoir des avis sur ce qui a déjà été fait auparavant, et réfléchir à comprendre des méthodes qui sont vaguement expliquées dans les documents, enfin on est obligé de tester plusieurs algorithmes pour voir ce qu'il marche le mieux. Ainsi au début cela m'a paru compliqué et au fur à mesure en testant différents outils j'ai trouvé ça de plus en plus facile. C'est formateur dans un sens pour plus tard si on est amené à résoudre des problèmes techniques et se débrouiller tout seul. Cependant le fait que ce ne soit pas guidé nous oblige à passer du temps à lire des documents, essayer de comprendre comment cela marche, temps que j'aurai préféré à consacrer plus sur la partie compilateur du projet.

Matthias

Le premier enseignement que j'aimerais relevé est celui de la confiance. Pour il s'agit d'un des atouts clé du bon déroulement du projet. Le fait de pouvoir se concentrer totalement sur sa partie tout en sachant que les autres parties du

projets seront correctement réalisées permet de gagner en productivité. Ensuite, il est important à mon sens de dire que le sujet en lui même était intéressant. Quoi de mieux pour bien comprendre le fonctionnement d'un langage qu'en en codant un compilateur. Cela m'a permis de comprendre les différentes passes de vérification contextuelle et de mieux comprendre les erreurs que pouvait envoyer un compilateur. Enfin je pense que la mise en situation du projet GL donne un bon aperçu de ce à quoi peut ressembler un travail d'ingénieur informatique, en tout cas son aspect technique et permet ainsi d'avoir une première idée sur le sujet.