



Projet Génie Logiciel

Compilateur Deca

BILAN

Auteurs :

M. Benjamin ARGENSON
M. Jean-Baptiste LEFOUL
M. Thomas
POUGET-ABADIE
M. Robinson
PRADA-MEJIA
M. Alexandre PROY

Encadrants :

Pr. Philippe BODIGLIO
Pr. Catherine ORIAT

Version du 26 janvier 2017

Table des matières

Introduction	1
1 Organisation de l'équipe	2
1.1 Pilotage du projet	2
1.2 Méthodes de travail	2
1.3 Répartition des tâches	3
2 Historique du projet	4
2.1 Les différentes itérations	4
2.2 Planning prévisionnel	5
2.3 Planning réel et rétrospective	6
3 Enseignements tirés	7
3.1 Bilan global	7
3.2 Commentaires personnels	7
3.2.1 Benjamin	7
3.2.2 Jean-Baptiste	8
3.2.3 Thomas	8
3.2.4 Alexandre	8
3.2.5 Robinson	8

Introduction

Ce document a pour ambition de vous expliquer en détail le déroulement du projet. Nous espérons qu'à la fin de la lecture de ce rapport vous aurez un meilleur aperçu de l'environnement de travail que nous avons choisi de mettre en place pour mener à bien ce projet.

Le rapport rend compte, dans un premier chapitre, de la manière dont notre équipe s'est organisée et a réparti les tâches. Il fait l'historique du projet en s'attardant notamment sur l'ordre choisi pour la conception et le développement des différentes étapes. Cette partie détaille également les plannings prévisionnel et réel réalisés avec l'outil *planner*. Une dernière partie sera consacré aux enseignements que nous avons pu tirer de ce premier projet d'une telle envergure. Ce chapitre comprendra un commentaire personnel pour chacun des membres de l'équipe.

Chapitre 1

Organisation de l'équipe

1.1 Pilotage du projet

Au début du projet, et suite à la présentation des méthodes Agiles, nous avons décidé d'expérimenter ces méthodes et de les appliquer à notre projet. Notre principale motivation était d'avoir un compilateur opérationnel et dont le fonctionnement était validé au plus tôt. De cette manière nous avons quelque chose d'opérationnel et sur lequel nous avons pu ajouter des fonctionnalités au fur et à mesure. Cela nous ôtais le risque d'arriver à la fin du projet avec un compilateur, peut-être complet, mais qui ne fonctionne pas.

Nous avons donc mis en application la méthode Scrum. Nous avons découpé le travail en plusieurs itérations et définis le travail à réaliser sur chacune d'elle. Chaque matin nous faisons un *stand-up* durant lequel nous disions ce que nous avons fait la veille et ce que nous comptons réaliser au cours de la journée. Au début nous avons pensé élire un *Scrum Master* à la journée afin que chacun puisse avoir une expérience du poste. Mais il était difficile pour la personne élue de se sentir impliquée et suite aux conseils reçus pendant un suivi nous avons décidé de réformer la pratique. Nous avons ainsi décidé que la personne élue l'était pour une semaine. Cela a en effet permis aux *Scrum Masters* suivants d'être plus impliqués et de donner un réel dynamisme au groupe.

1.2 Méthodes de travail

Afin de former une équipe soudée et motivée nous avons décidé de nous retrouver tous les matins à l'ENSIMAG pour travailler ensemble. La plupart du temps nous nous retrouvions à 8h15 dans une des salles du bâtiment H. Il était important pour nous de nous isoler et de nous concentrer sur notre groupe. Assez rapidement nous nous sommes rendu compte que personne n'utilisait l'amphithéâtre H et qu'il n'y avait aucun cours de programmé. Nous avons donc pris nos quartiers dans cet amphithéâtre. Travailler dans

cet espace qui nous était réservé nous a permis d'avancer dans une ambiance sereine. De plus nous avons pu profiter du tableau à notre guise pour y effectuer nos raisonnements et travailler en coopération.

Pour nous aider à rester organisé nous avons opté pour un outil de gestion de projet en ligne appelé *Trello*. Cet outil nous a permis de créer un tableau par *User Story*. Sur chaque tableau nous avons défini les colonnes suivantes : description (*la fonctionnalité qu'elle doit permettre à l'utilisateur*), à faire, en cours, à tester, testé, fait. Au début d'une *User Story* toutes les tâches se trouvaient dans la colonne à faire, puis quand on choisissait une tâche on la déplaçait dans la colonne en cours. De cette manière tous les membres du groupe savaient qui faisait quoi en temps réel. Cela nous a ainsi évité d'être plusieurs à modifier un même fichier et donc d'avoir des problèmes de fusion sur le dépôt partagé.

1.3 Répartition des tâches

Le compilateur comprend trois grandes étapes. Une étape d'analyse syntaxique, englobant avec elle l'analyse lexicale, une étape d'analyse contextuelle et une étape de génération du code.

Nous avons tous travaillé à un moment donné sur chacune des trois étapes différentes. Cela nous permettait à tous d'avoir une vision globale du projet et d'avoir une bonne compréhension de chacune des étapes. Puis nous avons réalisé que nous avions des affinités plus pour certaines parties et à partir de ce moment nous avons commencé à nous spécialiser un peu. Pour les besoins de l'extension Robinson et Alexandre se sont spécialisés en génération de code. Quant aux autres membres de l'équipe ils ont conservé leur polyvalence afin de pouvoir se concentrer sur l'ajout de la partie avec objet à notre compilateur

Chapitre 2

Historique du projet

2.1 Les différentes itérations

Étant donné que nous avons choisi d'appliquer la méthode *Scrum* nous avons définis plusieurs itérations :

1. Première itération : compilateur Deca sans objet opérationnel sans les options
 - (a) *User Story 1* : pouvoir compiler un fichier vide, un programme vide, un programme affichant des chaînes de caractères
 - (b) *User Story 2* : pouvoir compiler un programme dans lequel on peut réaliser des opérations arithmétiques et déclarer des variables
 - (c) *User Story 3* : pouvoir compiler un programme ayant des structures de contrôles (*if then else*, *while*), des opérations booléennes (*AND*, *OR*) et des inégalités
2. Deuxième itération : qui était une sorte de "*mini sprint*" ayant pour but d'avoir un compilateur Deca sans objet entièrement opérationnel et de nettoyer un peu le code avant le rendu intermédiaire
 - (a) *User Story 1* : pouvoir compiler un programme en utilisant les différentes options possibles
3. Troisième itération : comprenant deux branches travaillant en parallèle : une pour le compilateur Deca avec et sans objet opérationnel et une pour l'extension
 - (a) Partie Objet : compilateur Deca avec objet opérationnel
 - i. *User Story 1* : pouvoir compiler un programme avec des classes vides et des classes avec attributs seulement
 - ii. *User Story 2* : pouvoir compiler un programme avec des classes avec également des méthodes
 - iii. *User Story 3* : pouvoir compiler un programme respectant la hiérarchie des classes et pouvant redéfinir des méthodes

- (b) Partie Extension : compilateur Deca sans objet opérationnel pour la génération de bytecode Java pouvant être exécuté par la Java Virtual Machine
 - i. *User Story 1'* : idem que celle de la première itération mais en générant le bytecode Java
 - ii. *User Story 2'* : idem
 - iii. *User Story 3'* : idem
- 4. Quatrième itération : comprenant elle aussi deux branches en parallèles : une pour la rédaction de la documentation et une pour le nettoyage du code et la correction des derniers bugs
 - (a) Documentation :
 - i. *User Story 1* : rédiger les différentes documentations
 - ii. *User Story 2* : préparer la soutenance et la démonstration
 - (b) Code :
 - i. *User Story 1'* : correction des bugs et nettoyage du code
 - ii. *User Story 2'* : continuer l'extension et travailler sur la partie avec objets

Il est important de noter que chaque *User Story* comprend les étapes d'analyse, de conception, de codage et de validation. En effet les tests étaient réalisés au fur et à mesure afin de pouvoir valider le fonctionnement de notre compilateur à chaque étape de développement. De plus ce découpage correspond au découpage réel, il a été affiné au début de chaque itération.

2.2 Planning prévisionnel

Au début du projet nous n'avions qu'une idée globale de son déroulement. Nous ne savions pas par exemple quelles seraient les *User Stories* qui allaient composer chacune des itérations.

Nous avons donc fait un découpage grossier du projet en différentes itérations, puis comme nous ne savions pas qu'elles seraient les tâches à réaliser nous avons commencé par découper chaque itération selon les étapes suivantes : analyse, conception, codage, validation et documentation. Ce planning prévisionnel prévoyait que lors de la dernière semaine, à part quelques petites corrections mineures, nous n'aurions que très peu de code à écrire. La majeure partie du temps était consacrée à la rédaction des documentations et à la préparation de la soutenance. Notamment dans ce planning prévisionnel nous avons prévu de réaliser l'extension dans sa totalité, pouvoir compiler des fichiers avec ou sans classe.

Finalement ce planning prévisionnel reflétait bien la sérénité qui a toujours régné au sein de notre équipe. Nous avons prévu de nous laisser du temps sur la fin et de ne pas nous laisser prendre au dépourvu. Mais qu'en est-il dans la réalité ?

2.3 Planning réel et rétrospective

Le planning réel de notre projet est le fruit de mises à jour successives. Nous avons modifié ce dernier au début de chaque sprint lorsque nous avions une idée un peu plus précises des *User Stories* et des tâches à réaliser pour chacune d'elles. Le résultat est celui trouvé dans la partie 2.1 Les différentes itérations.

On peut constater que le planning est légèrement différent de celui prévu initialement. Bien que le squelette soit le même, on peut remarquer que la granularité des tâches est bien plus fine. On peut aussi voir que nous avons été un peu trop optimiste quant à la réalisation de l'extension. En effet après négociation avec nos professeurs encadrants nous avons décidé de ne réaliser la génération de bytecode Java que sur la partie sans objet. Nous pouvons aussi voir que la quantité de travail sur la fin du projet est un peu plus importante que celle anticipée. Cependant nous pouvons voir que les différentes échéances ont été respectées et correspondent à celles définies au départ.

En conclusion, bien que nos prévisions aient été un peu trop optimiste nous avons pu correctement terminer ce projet et livrer un compilateur opérationnel. De plus tout cela a été réalisé dans les temps.

Chapitre 3

Enseignements tirés

3.1 Bilan global

Ce projet était le premier projet de grande envergure que nous réalisions à l'ENSIMAG. Ce projet était conséquent de par sa durée de 3 semaines, de par la charge de travail à réaliser, ainsi que de par la taille de l'équipe (*5 membres*). Cela fait beaucoup de paramètres à prendre en compte et pour lesquels nous avons dû adapter notre manière de travailler.

Premièrement cela nous a appris à travailler en groupe et à mettre en place des règles de travail. Ces règles étaient nécessaires au bon fonctionnement du groupe et sont regroupées dans la charte de l'équipe que nous avons écrite au commencement du projet. Nous avons tous des caractères différents ainsi que des manières de travailler différentes. Le fait de clarifier ces règles au début a permis de prendre en compte chaque individualité et de trouver un terrain d'entente propice à une atmosphère de travail saine.

Nous avons également découvert les méthodes Agiles et avons pu les mettre en pratique. Nous pensons que ces méthodes ont été essentiel dans la réussite de ce projet. Travailler par itération nous a permis de toujours avoir un compilateur fonctionnel et de pouvoir en livrer une version à tout moment.

3.2 Commentaires personnels

3.2.1 Benjamin

Ce projet fut une bonne expérience pour moi. En effet, je l'ai trouvé bénéfique pour plusieurs raisons. D'une part, il m'a permis d'améliorer ma capacité à travailler en équipe, d'autre part il m'a apporté une vision plus concrète de la gestion à mettre en oeuvre pour mener à bien un tel projet, notamment avec la découverte des méthodes Agiles, que nous avons utilisées tout au long de ce projet. De plus, j'ai trouvé le sujet pertinent car il porte

sur un outil que l'on utilise quotidiennement. Ce fut donc intéressant de comprendre les mécanismes qui se cachent derrière un compilateur. Enfin, d'une manière plus globale, je pense que ce projet est particulièrement valorisant, puisqu'il permet d'avoir un aperçu du travail d'un ingénieur en informatique.

3.2.2 Jean-Baptiste

Ces trois dernières semaines de projets ont été très enrichissantes, que ce soit sur le plan technique que sur le plan organisationnel. A partir de maintenant, à chaque fois que je compilerai un fichier, je ne pourrai plus m'empêcher d'imaginer un arbre qui se construit, puis qui se décore, pour enfin générer un code en un langage de plus bas niveau. De plus, cette expérience très positive et très plaisante, me rassure sur mon choix de parcours professionnel.

3.2.3 Thomas

Je trouve que le projet GL mérite parfaitement sa réputation du projet le plus formateur de l'ENSIMAG et comme l'un des plus valorisant vis-à-vis des entreprises. En effet, ce projet a non seulement renforcé ma capacité à travailler en équipe mais m'a également beaucoup appris d'un point de vue scientifique. Je trouve cela extrêmement enrichissant de comprendre dans sa globalité ainsi que dans les détails le raisonnement permettant de passer d'une langue plus ou moins compréhensible à un humain à du langage plus basique et mieux conçu pour les machines.

3.2.4 Alexandre

Travailler sur le projet GL était pour moi enrichissant à tous points de vue. Sur le plan technique j'ai pu apprendre comment fonctionnait un compilateur. Sur le plan relationnel j'ai découvert les ressorts d'une bonne dynamique de groupe. Et enfin sur l'aspect gestion de projet j'ai pu expérimenter les méthodes Agiles. C'est d'ailleurs cet aspect du projet qui m'a le plus plu. Je tiens à remercier au passage les encadrants pour leurs conseils ainsi que mes coéquipiers pour leur motivation à toute épreuve.

3.2.5 Robinson

Ce projet a été une expérience complètement nouvelle pour moi, c'était ma première approche réelle d'un projet de travail d'équipe proche de la vie professionnelle. La formalité et l'importance accordées à ce projet nous permettent de mieux comprendre les tâches, échéances et suivis de l'entreprise sur les progrès réalisés. Par ailleurs, les connaissances acquises sont importantes et me donnent un nouveau point de vue différent sur la programmation.

ENSIMAG
681, rue de la Passerelle
38400 Saint-Martin-d'Hères