GRENOBLE-INP ENSIMAG

INGÉNIERIE DES SYSTÈMES D'INFORMATION

2A

GRENOBLE-INP ENSIMAG

INFORMATION SYSTEMS ENGINEERING

2A

ÉCOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET DE MATHÉMATIQUES APPLIQUÉES DE GRENOBLE

PROJET GL

BILAN GESTION DE PROJET ET D'ÉQUIPE

Réalisé par :

DIAKITE Alpha OUSMANE
DREUILLE NATHAN
FLISS YASSINE
KOSSI DAVID
OUDRHIRI IDRISSI SAFWANE

CLIENT :

FREIRE MARCO

Année académique : 2025–2026

Table des matières

1 Introduction

Le projet de Génie Logiciel (GL), réalisé en deuxième année à l'Ensimag, est particulièrement exigeant, tant d'un point de vue technique que par la charge de travail importante qu'il représente. Il consiste à concevoir et implémenter un compilateur pour le langage Deca, en respectant des spécifications précises et des contraintes de temps fortes.

L'objectif de ce document est de proposer un bilan de la gestion du projet et de l'organisation de l'équipe. Cette analyse est menée a posteriori, avec une approche critique, afin de mieux comprendre les difficultés rencontrées au cours du projet, mais également de mettre en évidence les choix et pratiques qui se sont révélés efficaces.

2 Organisation de l'équipe et méthode de développement

2.1 Organisation et répartition des taches

L'équipe était composée de cinq personnes. Lors de la première semaine du projet, un temps important a volontairement été consacré à la compréhension globale du sujet. Cette phase initiale a consisté à lire attentivement la documentation fournie et à visionner les vidéos réalisées par les enseignants spécifiquement pour le projet GL. Cette étape a permis à l'ensemble de l'équipe d'acquérir une vision d'ensemble du projet, de ses objectifs et des différentes étapes de compilation à implémenter.

Afin de mener le projet efficacement, une répartition des tâches par spécialisation a été mise en place. Dès le début du projet, deux membres de l'équipe (David et Nathan) se sont concentrés sur l'étape A (analyse lexicale et syntaxique), un membre (Safwane) sur l'étape B (vérifications contextuelles), un autre (Ousmane) sur l'étape C (génération de code), et un dernier (Yassine) sur la partie dédiée aux tests. Cette organisation visait à permettre un avancement parallèle des différentes composantes du compilateur.

Durant la première semaine, l'équipe a également travaillé sur la réalisation d'un compilateur minimal capable de gérer un programme de type « Hello World ». En parallèle, certaines parties spécifiques de la documentation ont été réparties entre les membres de l'équipe, notamment celles concernant les outils de test, l'utilisation de Jacoco ou encore les tests avec l'interpréteur `ima`. Chaque membre a ensuite synthétisé et expliqué aux autres les points essentiels de la documentation étudiée. Cette démarche a permis de gagner du temps en évitant une lecture exhaustive immédiate de l'ensemble du polycopié, tout en assurant une compréhension suffisante des éléments clés nécessaires au démarrage du projet (avec bien entendu une possibilité de consulter plus tard l'integralité de la

documentation si les explications etaient insuffisantes).

2.2 Analyse critique de l'organisation adoptée

L'organisation mise en place au début du projet a permis une bonne diffusion des connaissances au sein de l'équipe. Lors des premières semaines, un effort particulier a été fait pour expliquer aux autres membres les parties de la documentation étudiées et pour documenter les éléments essentiels, afin que chacun puisse comprendre le fonctionnement global du compilateur et des outils associés.

Cependant, au fur et à mesure de l'avancement du projet, cette dynamique s'est progressivement atténuée. La spécialisation par domaine, initialement choisie pour gagner en efficacité, a conduit chaque membre à se concentrer principalement sur sa propre partie. Les échanges détaillés sur le contenu du travail effectué sont devenus moins fréquents, ce qui a réduit la compréhension globale du projet par l'ensemble de l'équipe.

Cette évolution a permis un gain de productivité sur certaines tâches, mais elle a également renforcé la dépendance entre les membres de l'équipe, notamment lors des phases d'intégration. Avec le recul, un maintien plus systématique des échanges et de la documentation interne aurait sans doute facilité une meilleure compréhension du projet sur certaines parties plus techniques (notamment gencode) et réduit certains blocages en fin de développement.

3 Planification du projet

La planification du projet s'est appuyée en grande partie sur l'organisation imposée par le cadre du projet GL. Bien que le travail ait été réalisé en autonomie, les échéances fixées par les enseignants, ainsi que les séances de suivi régulières, ont constitué une forme de planning que l'équipe s'est efforcée de respecter.

Le projet a ainsi été découpé selon les grandes étapes attendues. La première semaine a été consacrée à la réalisation d'un compilateur minimal capable de gérer un programme de type « Hello World », en vue du premier suivi. Cette étape a permis de valider la prise en main de l'environnement, des outils et de la chaîne de compilation. Dans un second temps, l'équipe s'est concentrée sur l'implémentation du langage Deca sans objet, correspondant au périmètre du rendu intermédiaire. Enfin, la dernière phase du projet a porté sur l'implémentation du langage Deca complet en vue du rendu final.

Cette organisation a permis de respecter les différentes échéances sans prendre de retard, en maintenant un rythme de travail constant tout au long du projet. L'équipe a veillé à anticiper les deadlines afin d'être prête dans le temps imparti, plutôt que de concentrer le travail à l'approche des rendus.

Certaines parties du projet, notamment l'étape A, se sont révélées plus rapides à implémenter que prévu. Les membres de l'équipe initialement chargés de cette étape ont alors pu apporter un soutien aux autres, en particulier en contribuant à la mise en place de tests pour les étapes B et C. Cette entraide a permis de sécuriser l'avancement global du projet et de limiter les risques de blocage lors des phases ultérieures.

4 Déroulement réel du projet

Comme indiqué précédemment, le projet a débuté par une phase de compréhension générale, durant laquelle l'équipe a pris le temps d'assimiler la documentation et les objectifs globaux du compilateur à développer. Cette étape a permis de poser des bases communes avant de commencer l'implémentation.

Au cours de la première semaine, chaque membre a ensuite travaillé sur sa partie respective (étapes A, B, C et tests) afin de parvenir à une première version fonctionnelle du compilateur capable de gérer un programme de type « Hello World ». Cette phase a permis de valider le bon enchaînement des différentes étapes de compilation et de vérifier le fonctionnement de l'environnement de développement.

La période des vacances de Noël a constitué une contrainte particulière dans le déroulement du projet. Tous les membres de l'équipe n'ont pas pu y consacrer le même temps, ce qui a entraîné un investissement variable selon les personnes. Néanmoins, le travail poursuivi durant cette période a permis de faire avancer significativement le projet.

Le fait d'avoir continué à travailler pendant les vacances a permis à l'équipe de prendre de l'avance sur les échéances. En particulier, l'implémentation de Deca sans objet a été finalisée quelques jours avant le rendu intermédiaire. Ce délai a ensuite été mis à profit pour renforcer les tests et démarrer l'implémentation de l'extension TRIGO.

Les étapes B et C s'étant révélées plus longues et plus complexes, Ousmane et Safwane ont continué à travailler sur ces parties jusqu'à la fin du projet. En parallèle, Yassine et David se sont concentrés sur l'implémentation de l'extension TRIGO. La phase de recherche et d'implémentation s'est révélée fastidieuse pour cette extension, mais ils ont néanmoins réussi à identifier et implémenter un algorithme offrant une très bonne approximation des fonctions trigonométriques. Nathan s'est pendant ce temps focalisé sur la mise en place des tests unitaires avec JUnit. Ce travail sur les tests a permis d'améliorer progressivement la couverture de code mesurée avec Jacoco, tout en avançant en parallèle sur la rédaction de la documentation, afin d'éviter une accumulation trop importante de documents à produire en fin de projet.

Bilan La répartition des tâches s'est globalement révélée efficace, même si certaines parties du projet se sont avérées plus longues et plus techniques que d'autres, en particulier la génération de code. Ces différences s'expliquent en partie par des disparités de connaissances initiales au sein du groupe, notamment sur l'assembleur. Malgré cela, l'organisation adoptée a permis à chaque membre de contribuer activement à l'avancement du projet, ce qui a permis à l'équipe de respecter l'ensemble des échéances.

5 Gestion des difficultés

L'une des premières difficultés rencontrées par l'ensemble du groupe a été la prise en main d'une architecture de projet particulièrement volumineuse, comprenant un grand nombre de fichiers. Certains membres de l'équipe n'avaient jamais utilisé des outils tels que Maven ou Jacoco, ce qui a nécessité un important travail initial de documentation et de compréhension du polycopié du projet, dense et très technique au premier abord. Avec le temps et la pratique, l'équipe a progressivement maîtrisé ces outils, ce qui a permis un gain significatif en efficacité.

Un autre problème récurrent au cours du projet a concerné l'utilisation de Git. Son usage n'a pas été optimal dans les premières phases du projet, ce qui a entraîné de nombreux conflits et une perte de temps importante. Ces difficultés provenaient notamment d'une utilisation systématique des *rebases*, peu adaptée à un projet de grande taille avec de nombreuses versions locales divergentes. Une fois cette source de problème identifiée et corrigée, l'équipe a pu gagner un temps considérable et travailler de manière plus fluide.

Un incident notable est également survenu lors du rendu intermédiaire, potentiellement lié à des divergences de versions. Une virgule manquante dans un test a empêché l'exécution correcte de nombreux tests, entraînant une baisse significative de la couverture mesurée par Jacoco. À la suite de cet événement, l'équipe a pris la décision de ne plus effectuer de *push* à l'approche immédiate des échéances, afin de limiter les risques liés à des modifications de dernière minute.

Enfin, lors du rendu intermédiaire, des problèmes techniques liés au vidéoprojecteur ont été rencontrés durant la présentation. Cette expérience a permis à l'équipe d'anticiper ce type de situation et de garder à l'esprit la nécessité de savoir s'adapter rapidement, notamment en vue de la soutenance finale.

6 Bilan critique et enseignements

Le projet de Génie Logiciel, au-delà des apprentissages purement techniques et théoriques, a permis à l'ensemble du groupe de progresser dans la conduite d'un projet de

grande ampleur et de développer des compétences humaines (et oui meme à l'ENSIMAG c'est possible !).

Ce projet a notamment permis de travailler au sein d'un groupe plus important que ceux auxquels nous sommes habituellement confrontés, la majorité des projets étant réalisés en binôme. Travailler à cinq sur un projet de cette taille a constitué une expérience formatrice, en nous apprenant à collaborer efficacement, à nous entraider et à gérer collectivement des problématiques parfois complexes, tant sur le plan technique qu'organisationnel.

Tout au long du projet, l'équipe est parvenue à respecter les échéances fixées, grâce à l'investissement de chacun dans les différentes parties du compilateur. La diversité des parcours et des compétences au sein du groupe a parfois entraîné des différences dans la vitesse d'appropriation de certaines notions, mais elle a également constitué une richesse, favorisant l'entraide et le partage de connaissances.

Enfin, une communication efficace au sein de l'équipe a joué un rôle central dans la résolution des difficultés rencontrées. En particulier, les problèmes liés à l'utilisation de Git, évoqués précédemment, ont pu être identifiés et corrigés collectivement. Sans cette communication, ces difficultés auraient pu perdurer et entraîner une perte de temps significative, chaque membre tentant de gérer les conflits de manière isolée. Cette expérience a ainsi mis en évidence l'importance d'un échange régulier et transparent dans la réussite d'un projet collectif.

7 Conclusion

Ce projet de Génie Logiciel a permis de prendre conscience de l'importance de l'organisation et de la communication dans un projet technique de grande ampleur. La complexité du compilateur Deca a mis en évidence que les difficultés ne sont pas uniquement liées au code, mais aussi à la coordination du travail et à la gestion du temps.

Les choix d'organisation, les ajustements effectués en cours de projet et les difficultés rencontrées ont constitué des enseignements concrets pour la suite de la formation.