Grenoble INP - Ensimag

Projet Génie Logiciel (GL)

Bilan de gestion d'équipe et de projet

Compilateur Deca – Janvier 2026

ÉQUIPE GL56

Ayoub – Responsable Étape A (Syntaxe)
Hamza – Responsable Étape C (Gencode)
Mouad – Renfort Étape C / Ex-Responsable A
Chaima – Responsable Étape B (Contextuel)
Abdellah – Extension & Support

20 janvier 2026

Table des matières

1 INTRODUCTION

Le projet de réalisation du compilateur Deca constitue une étape charnière dans notre cursus d'ingénieur. Au-delà du défi technique consistant à traduire un langage de haut niveau vers de l'assembleur IMA, ce projet est avant tout une épreuve de gestion de ressources humaines, de temps et de complexité.

Ce document dresse le bilan de l'organisation de l'équipe **GL56**. Il retrace l'évolution de nos méthodes de travail, de la phase *« Sans Objet »* à la phase *« Objet »*, analyse nos choix stratégiques face aux imprévus majeurs qui ont ponctué le développement, et présente le retour d'expérience individuel de chaque membre.

2 ORGANISATION ET MÉTHODOLOGIE

2.1 Philosophie générale : Agilité et Spécialisation

Dès le lancement du projet, nous avons identifié que la complexité du compilateur ne permettrait pas à chaque membre de maîtriser l'intégralité du code dans le temps imparti.

Nous avons donc opté pour une **organisation verticale par tâches (spécialisation)**, tout en conservant une flexibilité (Agilité) pour réallouer les ressources en fonction des goulots d'étranglement.

2.2 Évolution de la structure de l'équipe

L'un des points forts de notre gestion a été la capacité à restructurer l'équipe entre les deux grandes phases du projet.

Phase 1 : Le Compilateur Sans Objet Durant cette première phase, la répartition était classique :

— Un binôme sur le lexique et la syntaxe (Étapes A).

— Un binôme sur la sémantique (Étape B).

— Une personne sur la génération de code (Étape C).

Cette phase nous a permis de découvrir les forces de chacun, mais a aussi révélé un risque majeur : l'isolement du membre travaillant sur l'étape C.

Phase 2 : Le Compilateur Objet (La Stratégie de Rotation) Pour la phase critique *« Objet »*, nous avons opéré une redistribution stratégique des rôles :

1. **Sanctuarisation de l'Étape C (Hamza) :** Nous avons décidé de maintenir Hamza sur l'étape C. Sa maîtrise acquise sur l'architecture bas niveau lors de la première phase était un atout inestimable.
2. **Rotation des autres membres :** Pour favoriser la montée en compétence :

— **Ayoub** a pris le pilotage de l'Étape A (Syntaxe Objet).

— **Chaima** a pris la responsabilité de l'Étape B (Vérification Contextuelle).

— **Mouad** a été basculé en **renfort sur l'Étape C** pour épauler Hamza.

— **Abdellah** a été affecté à l'exploration de l'extension TRIGO.

2.3 Outils de collaboration

GitLab : Nous avons créé une branche « Mouad » pour le développement, qui peut contenir éventuellement des conflits ou des fichiers mal commentés. Nous avons décidé de garder la branche principale pour une version du compilateur fonctionnelle..

Communication : Réunions vocales quotidiennes (*Daily meetings*) pour synchroniser l'avancement.

Planification : Diagramme de Gantt pour le suivi du chemin critique.

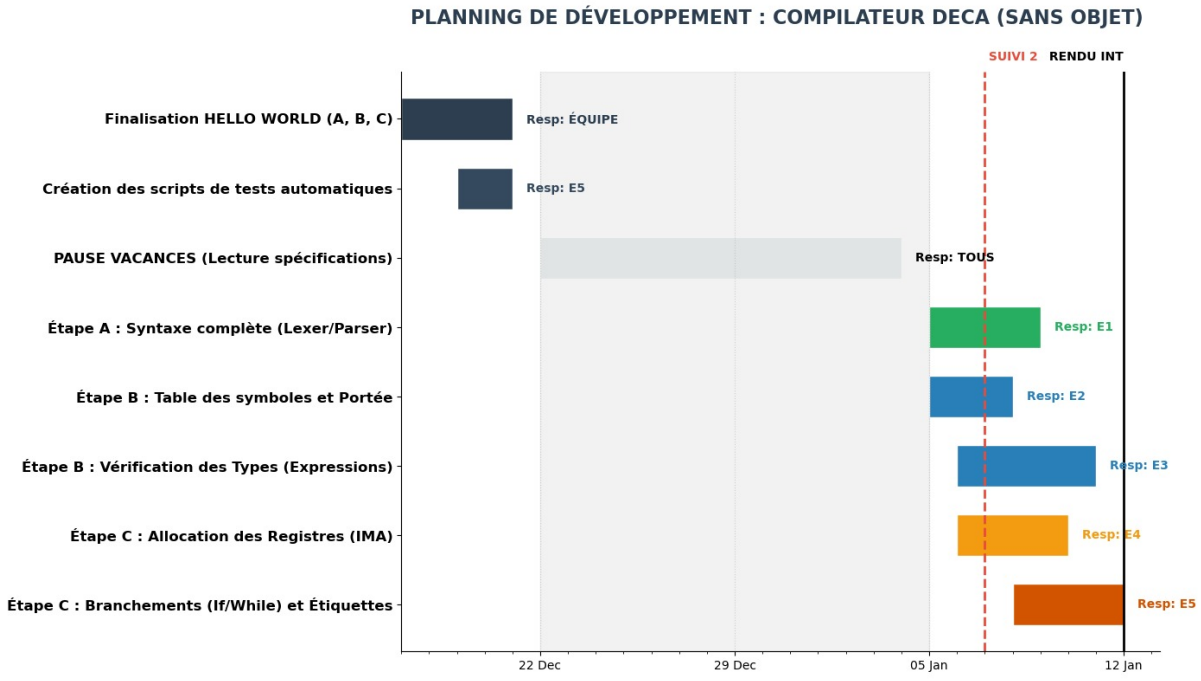

FIGURE 1 – Planning de développement (Phase Sans Objet)

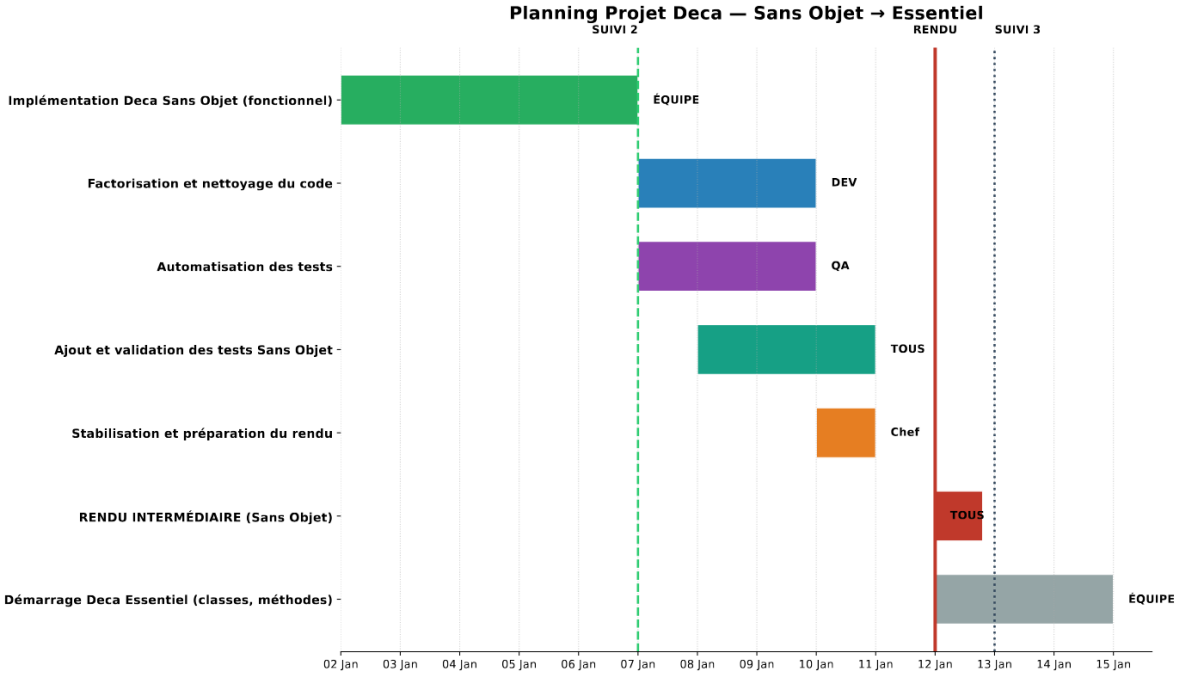

FIGURE 2 – Planning de développement (Phase Objet)

3 GESTION DES IMPRÉVUS ET CRISE DE L'ÉTAPE C

La gestion du projet ne s'est pas déroulée de manière parfaitement linéaire. Nous avons fait face à un défi majeur lors de la transition vers la partie Objet.

3.1 Le "Bug" de Planification

Notre planning initial prévoyait une implémentation rapide de l'étape B (Contextuelle) de la partie Objet en **2 jours** seulement, afin de laisser le maximum de temps à l'étape C (Génération de Code), traditionnellement plus longue.

Cependant, la réalité a contredit cette estimation optimiste. Lors du démarrage de l'implémentation de la génération de code Objet (Étape C), nous avons détecté plusieurs **régressions et bugs critiques** issus de l'étape B précédente (notamment sur la gestion des signatures de méthodes et la validation des héritages).

3.2 L'Effet Domino

Cette découverte tardive a eu un effet paralysant :

— Le binôme responsable de l'étape C s'est retrouvé **freiné** : il ne pouvait pas générer de code correct à partir d'un arbre décoré comportant des erreurs sémantiques.

— Il a fallu corriger l'étape B en urgence tout en essayant d'avancer sur l'étape C, créant une charge mentale importante et une situation de "charnière" difficile à gérer.

3.3 La Réponse : Bascule Stratégique

Face à ce goulot d'étranglement qui menaçait la livraison du projet, nous avons pris une décision radicale de gestion de crise :

1. **Suspension immédiate de l'extension :** Nous avons décidé de geler temporairement le développement de l'extensions (Trigo) qui n'étaient pas vitales pour le rendu final.
2. **Mobilisation générale sur l'Étape C :** Nous avons réalloué les ressources humaines en basculant **3 personnes** (Hamza, Mouad et Abdellah) quasi-exclusivement sur la résolution des problèmes de l'étape C et la correction des dépendances contextuelles.

Cette décision difficile mais nécessaire nous a permis de stabiliser le cœur du compilateur et de rattraper le retard accumulé, garantissant un rendu fonctionnel.

4 ANALYSE CRITIQUE

4.1 Ce qui a fonctionné

— **L'adaptabilité :** Notre capacité à sacrifier temporairement les fonctionnalités annexes (extensions) pour sauver le cœur du projet.

— **La spécialisation de Hamza :** Garder un expert technique sur la génération de code depuis le début a évité de perdre du temps en formation.

— **La robustesse de l'Étape A :** Le travail préparatoire d'Ayoub sur l'arbre abstrait a permis aux équipes B et C de travailler sur une base syntaxique stable.

4.2 Ce qui peut être amélioré

— **Validation inter-étapes :** Le goulot d'étranglement décrit plus haut aurait pu être évité avec une validation plus stricte de l'étape B avant de lancer l'étape C.

— **Tests unitaires Objet :** La couverture de tests automatisés sur l'héritage a été insuffisante au début, ce qui a laissé passer les bugs contextuels détectés trop tardivement.

5 RETOURS D'EXPÉRIENCE INDIVIDUELS

Ayoub (Responsable Étape A – Syntaxe & Architecture)

« En endossant la responsabilité de l'Étape A, particulièrement lors de la transition vers l'Objet, j'ai rapidement pris conscience que je ne codais pas seulement pour moi, mais pour toute l'équipe. La construction de l'Arbre Syntaxique Abstrait (AST) est la pierre angulaire du compilateur : la moindre ambiguïté ou incohérence dans ma structure se répercutait instantanément en complexité exponentielle pour Chaima et Hamza. Ce rôle m'a appris une leçon fondamentale d'ingénierie logicielle : l'anticipation. J'ai dû apprendre à penser "au-delà" de la grammaire, en

me demandant constamment comment une structure de classe serait traitée en mémoire. Avec le recul, je réalise que la communication en amont est tout aussi cruciale que le code lui-même ; définir une interface claire avec les autres membres dès le premier jour nous aurait épargné plusieurs refactorisations coûteuses. »

Hamza (Responsable Étape C – Génération de Code)
« Le développement de la génération de code a été pour moi une véritable épreuve d'endurance technique et mentale. Si la partie procédurale était un défi algorithmique, l'implémentation de la partie Objet en assembleur IMA s'est révélée être un défi architectural majeur. J'ai dû construire mentalement, puis en code, des structures abstraites complexes comme la Table des Méthodes Virtuelles et la gestion dynamique du Tas, le tout avec des outils de débogage limités. Le moment le plus critique a été la découverte tardive des incohérences venant de l'analyse contextuelle, qui m'a forcé à stopper net mon avancement. L'arrivée de Mouad et Abdellah en renfort a été salutaire : cela m'a sorti de l'isolement du "développeur bas niveau" et nous a permis de transformer une situation de crise en une session intense de résolution de problèmes collective. Ce projet a définitivement démystifié pour moi le fonctionnement interne d'un ordinateur. »

Chaima (Responsable Étape B – Vérification Contextuelle)
« Ma transition vers l'analyse contextuelle lors de la phase Objet a marqué un tournant dans ma compréhension des langages de programmation. J'ai découvert que la sémantique est le véritable gardien de la cohérence du code. La difficulté ne résidait pas tant dans l'implémentation des règles de typage, mais dans la gestion fine de l'héritage et des signatures de méthodes. L'épisode où des bugs de ma partie ont bloqué l'étape C a été difficile à vivre, car je me sentais responsable du ralentissement de l'équipe. Cependant, cette épreuve a été formatrice : elle m'a enseigné qu'en développement logiciel, un module n'est jamais "terminé" tant qu'il n'a pas été validé par celui qui l'utilise. J'ai appris à travailler sous pression et à prioriser les corrections critiques pour débloquer mes collègues, une compétence qui me sera précieuse en entreprise. »

Mouad (Renfort Étape C / Ex-Responsable A)
« Mon parcours au sein du projet a été marqué par une mobilité stratégique qui m'a offert une vision holistique du compilateur. Avoir initialement travaillé sur la syntaxe m'a donné une connaissance intime de la structure de l'arbre, ce qui s'est avéré être un atout décisif lorsque j'ai basculé en renfort sur la génération de code. Je n'avais pas besoin de découvrir l'AST, je le connaissais déjà par cœur. Rejoindre Hamza sur l'étape C a été une expérience de collaboration intense : nous avons pratiqué le "Pair Programming" par nécessité, débuggant à deux voix des traces d'exécution assembleur obscures. Cette flexibilité m'a prouvé que la spécialisation ne doit pas être une prison ; au contraire, la polyvalence est la clé pour résoudre les goulots d'étranglement dans un projet complexe. »

Abdellah (Extensions & Support Transverse)

« Occupant une position transverse, j'ai eu l'opportunité d'observer le projet avec un certain recul. Mon travail initial sur l'extension mathématique (CORDIC) était passionnant techniquement, mais j'ai dû apprendre à détacher mon ego de mon code lorsque nous avons décidé de geler cette fonctionnalité pour sauver le cœur du projet. Basculer vers un rôle de support et de test m'a fait réaliser que la qualité logicielle est un combat permanent. En traquant les bugs et en validant les corrections, j'ai compris que la robustesse d'un produit vaut mieux qu'une liste de fonctionnalités inachevées. Ce projet m'a enseigné la rigueur et le sens des priorités : savoir sacrifier le "nice-to-have" pour garantir le "must-have" est sans doute la décision la plus mature que nous ayons prise en tant qu'équipe. »

6 CONCLUSION

Le projet GL56 a été une réussite technique et humaine. Nous avons réussi à livrer un compilateur fonctionnel malgré les imprévus majeurs rencontrés lors de l'intégration de la partie Objet.

La leçon principale est que **l'organisation d'une équipe n'est pas figée**. Le plan initial est une boussole, mais c'est la capacité de l'équipe à se reconfigurer dynamiquement face aux obstacles – comme nous l'avons fait en suspendant les extensions pour sauver l'étape C – qui détermine le succès final.