

# Projet Génie Logiciel

## Documentation : Bilan D'Equipe

Gr2 - G110 - 2023 - 2024

Membres du groupe:

CHRIF M'HAMED Mohamedou

DIAB Dana

KONE Madou

MOHAMED AHMED Mohamed Lemine

# Le Sommaire

|  |          |
|--|----------|
| <b>1. Introduction :</b>                           | <b>2</b> |
| <b>2. Description de l'organisation adoptée :</b>  | <b>2</b> |
| <b>3. Présentation de l'historique du projet :</b> | <b>4</b> |

## 1. Introduction :

Le succès du projet de génie logiciel dépend largement de la gestion efficace de l'équipe et de la mise en place d'une structure organisationnelle robuste. Cette documentation vise à dresser un bilan de notre approche, de nos forces, et des leçons apprises tout au long du développement de notre compilateur pour le langage deca.

## 2. Description de l'organisation adoptée :

Dès le début du projet, nous avons élaboré une charte d'équipe pour identifier et capitaliser sur les forces individuelles de chaque membre. Cette démarche nous a permis de comprendre les compétences oratoires comme une faiblesse collective, mettant en lumière l'importance de la communication dans notre projet. Nous avons travaillé activement pour atténuer ce point faible lors des présentations orales des suivis.

La charte a été essentielle pour distribuer les tâches en fonction des compétences individuelles. La désignation des rôles a été cruciale pour assurer une répartition équilibrée des responsabilités et capitaliser sur les compétences spécifiques de chaque membre.

- **Étape A - Analyse Syntaxique** : Mohamedou était responsable de cette phase.
- **Étape B - Analyse Contextuelle** : Mohamed Lemine était responsable de cette phase.
- **Étape C - Génération de Code** : Dana et Madou étaient responsables de cette phase.

Suite à l'observation positive de l'efficacité de cette répartition des tâches, nous avons maintenu cette allocation tout au long du projet. Ainsi, nous avons opté pour une approche incrémentale classique. Les étapes du développement étaient constituées des sous-langages HelloWorld, SansObjet et Complet de Deca.

Pour garantir la pérennité et l'efficacité de cette organisation, nous avons mis en place plusieurs outils, tels que :

- Trello a été utilisé comme outil principal pour suivre au quotidien l'avancement des tâches.
- Git a été employé pour fusionner nos versions et consolider les progrès de programmation, facilitant ainsi la gestion collaborative du code source.
- Nous avons élaboré un diagramme de Gantt dans le but de comparer les délais effectivement pris avec ceux initialement souhaités, offrant ainsi une visualisation claire de la planification du projet.
- La création de groupes sur des plateformes de réseau social comme Discord et Whatsapp pour la communication.
- Un dossier sur Google Drive a été employé comme moyen de partage de fichiers.

- Un tableau Google Sheets a été utilisé pour consigner les dates clés telles que les réunions de suivi, les échéances de rendu, etc., tout au long du projet.

Ces outils ont joué un rôle crucial dans la gestion d'équipe et de projet en facilitant la communication et en assurant une coordination efficace tout au long du processus de développement. Bien que nous n'ayons pas pu terminer dans le temps prévu, ces plateformes ont néanmoins contribué à maintenir une collaboration organisée et à atténuer les difficultés rencontrées. Ces outils se sont également révélés utiles pendant les périodes de vacances et les jours où nous n'avons pas travaillé en présentiel.

Lors de l'établissement du planning prévisionnel, nous avons surestimé le temps que nous allions consacrer au travail pendant les vacances, en raison de notre situation géographique différente et des horaires de travail variés. Cette surestimation a entraîné un léger retard par rapport au planning initial, impactant notamment la finalisation du sous-langage SansObjet. Ce léger retard s'est prolongé jusqu'à la fin du projet, empêchant l'achèvement de l'étape C pour le langage Complet. Cela souligne l'importance d'une estimation réaliste des ressources et des contraintes pour une planification plus précise dans des conditions de travail non conventionnelles.

D'autre part, les tests mis en place pour les étapes B et C ont été essentiels pour identifier les parties mal implémentées dans les phases précédentes. Cela nous a permis de remonter ces problèmes à la personne concernée, facilitant ainsi la correction proactive des erreurs et contribuant à l'amélioration continue du processus de développement.

En conclusion, la gestion de notre groupe pour le projet de génie logiciel a été une expérience enrichissante. Une estimation plus réaliste du temps que nous allions consacrer au travail aurait probablement facilité la réalisation du projet dans le délai prévu. Malgré cela, ce projet a apporté des leçons importantes sur la gestion d'équipe et la planification, fournissant une base solide pour l'avenir.

### 3. Présentation de l'historique du projet :

Les trois parties du projet ont été développées parallèlement. En effet, la partie B étant plus compliquée que la partie A, son développement laissait suffisamment de temps pour que la partie A soit terminée et utilisée pour tester la partie B. Cela a été le même cas entre la partie B et C.

Nous tenons à signaler que l'idée du développement parallèle était en fait un test au début. Nous l'avons mis en place à priori que pour le Hello World, mais cela a très bien fonctionné donc nous l'avons conservé pour le reste.

Néanmoins, bien qu'on soit parallèle, il est souvent arrivé que la partie C ait besoin d'une implémentation de la partie B, alors cette dernière donnait priorité à cette implémentation.

Pour ce qui est du temps passé dans les différentes activités,

La première semaine était plutôt une découverte du projet en termes d'analyse, conception et codage. Durant cette semaine, on était un peu moins stricte sur les règles du groupe, on laissait tout le monde s'aventurer dans le projet à sa propre guise.

Puis dans la semaine des vacances, c'est là que chacun a bien pris le temps de lire la documentation, d'explorer les packages, de voir les liens entre les fichiers etc...  
Donc, l'analyse et la compréhension profonde ont été effectuées durant les vacances.

Puis au retour des congés, on était maintenant dans le codage des fonctionnalités, l'élaboration des tests et cela a constitué la plus grande partie du temps consacré au projet.

La validation de notre implémentation est effectuée par nos tests, leurs complexités et leurs couvertures. Cependant, l'élaboration des tests s'est faite des fois pendant le codage et d'autres fois après le codage. Mais pour donner une estimation, cela tourne autour de 30-40% du temps de codage.

La documentation a duré 4 jours ( un jour avant le rendu final, ainsi que 3 jours avant la soutenance ).

En conclusion, il aurait été bénéfique d'anticiper la rédaction des documents à rendre. Nous considérons cependant que le temps alloué à chaque partie du projet était approprié. Bien que le temps investi dans l'analyse approfondie et la compréhension préalable du sujet ait retardé notre calendrier prévisionnel, cette approche méticuleuse s'est avérée être un atout précieux tout au long du projet.