

# Bilan de gestion d'équipe et de projet

## I/Description critique de l'organisation adoptée dans l'équipe

### I.1/Planification du projet:

Pendant la première semaine, nous ne nous connaissions pas suffisamment et n'avions pas de vision parfaitement claire sur nos propres compétences, c'est pourquoi nous avons décidé de nous lancer dans le projet sans nous attribuer de rôle en particulier. Dans cette continuité, nous avons organisé notre planning avec l'outil Planner, sans entrer trop dans les détails. Après quelques jours, nous avons décidé d'adopter le modèle Agile et de fonctionner par user stories.

Voici un résumé de nos sprints (prévisionnels, bien que les résultats aient été ajoutés après) :

(Nous n'avions pas réussi à découper le premier sprint en user stories, mais nous nous sommes améliorés par la suite)

<p><b>Sprint 1:</b>  <b>Objectif : Langage HelloWorld</b>  Début : 01/01/2021  Fin : 11/01/2021  total des points : [20] = 200h (5 jours ouvrés)</p> <p>-&gt; lire le cahier des charges + cours [6]  -&gt; Etape A:  - lexer [1]  - parser (+ début sans objet) [2]  - décompilation [1]  - tests et validation [1]  -&gt; Etape B:  - décoration, environnement, verify [3]  - test et validation [1] (création de fichiers scripts pour automatiser les tests [1])  -&gt; Etape C:  - langage assemblage [1]  - test et validation [1]  -&gt; Extension:  - documentations sur les algos existants (présentation pour le suivi) [1]  -&gt; Préparation des rendus pour le 1er suivi (11/01) : Chartes, Gant, sprints [1]</p>	<p><b>Sprint 2:</b>  <b>Objectif : Langage Sans Objet</b>  Début : 11/01/2021  Fin : 18/01/2021  total des points : [20] = 200h (5 jours ouvrés)</p> <p>User et technical stories (pas forcément dans l'ordre chronologique):  -&gt; lire le cahier des charges [3]  -&gt; fonctionnalités [14] :  - déclaration de variable [2]  - expression binaire [6]  - expression unaire [2]  - print [1]  - if [1]  - while [2]  - readInt [1]  -&gt; Extension[1]  -&gt; Préparation des rendus pour le 2ème suivi (18/01) [1]</p> <p>Résultat:  On a réussi à finir le langage sans objet le 17/01/2021. On va donc pouvoir commencer Objet un peu plus tôt que prévu.</p>
---	--

<p>Résultats: Légère difficultés dans la compréhension du sujet. Mais bon progrès.</p>	
<p><b>Sprint 3:</b> <b>Objectif : Langage Objet</b> Début : 16/01/2021 Fin : 20/01/2021 total des points : [20] = 200h (5 jours)</p> <p>User et technical stories (pas forcément dans l'ordre chronologique): -&gt; lire le cahier des charges [2] -&gt; fonctionnalités [11]     - déclaration de classe     - déclaration de champs     - déclaration de méthodes     - new     - appel méthode     - accès champs     - return     - this     - null</p> <p>-&gt; Extension [7]</p> <p>Résultat: Léger retard, mais on y est presque. (se débloquera 1 jour plus tard)</p>	<p><b>Sprint 4:</b> <b>Objectif : Langage Complet</b> Début : 20/01/2021 Fin : 24/01/2021 total des points : [20] = 200h (5 jours)</p> <p>User et technical stories (pas forcément dans l'ordre chronologique): -&gt; fonctionnalité [7]     - cast     - instance of -&gt;Extension[7] -&gt;soutenance [4] -&gt; lire le cahier des charges [2]</p> <p>Résultat: Tout le projet fonctionne, on est dans les temps ! Baudouin a même pu faire des petites optim pour augmenter nos perf pour le classement.</p>

En pratique, les sprints 1 et 2 se sont déroulés sans encombre (avec même un peu d'avance pour le sprint 2). Nous avons pris du retard sur le sprint 3 au niveau de l'étape C des déclarations et accès aux méthodes. Nous avons donc fini ces tâches pendant le sprint 4. D'un autre côté, nous avons pu commencer les étapes A et B des tâches du sprint 4 dès la fin du sprint 3.

Nous n'avons pas découpé l'extension en sprints, sa réalisation s'est faite en continu sur le projet bien que rythmée par des objectifs temporels bien définis.

Enfin, nous avons rapidement convenu d'une charte de mise en production, afin de prévenir des difficultés le jour des rendus.

### Retour critique:

Il est vrai que nous avons pris un peu de temps pour prendre nos marques et nous organiser au mieux, mais cela nous a permis de vraiment comprendre où se trouvaient nos limites individuelles et collectives, et comment les dépasser. De

cette manière, nos décisions en matière d'organisation ont pu s'adapter naturellement à notre manière de fonctionner.

Nous avons mal estimé le temps nécessaire pour les sprint 3 et 4 qui étaient respectivement très dense et assez rapide. Ces deux erreurs se sont heureusement compensées mais nous aurions pu le prévoir en étant plus attentifs aux spécifications demandées.

Malgré nos précautions pour la mise en production, nous avons tout de même été en danger le jour du rendu final (Lundi 25) : nous avons rencontré des problèmes inattendus (la suppression de commentaires a eu pour conséquence malheureuse de rendre les sorties de nos tests différentes des sorties de références), et nous avons presque manqué l'heure de rendu. Nous avons sous-estimé l'impact des lignes de codes qui paraissent sans influence sur le code. Pour éviter cela, il nous paraît aujourd'hui important d'ajouter à la charte des informations sur le "Quand faire la vérification" en plus du "Quels sont les risques?" et du "Comment les gérer?".

## I.2/Réunions

Au début nous n'avions pas d'heures de réunion fixes, puis conformément à la méthode Agile, nous nous sommes habitués à faire une réunion de 30 minutes tous les matins à 10h.

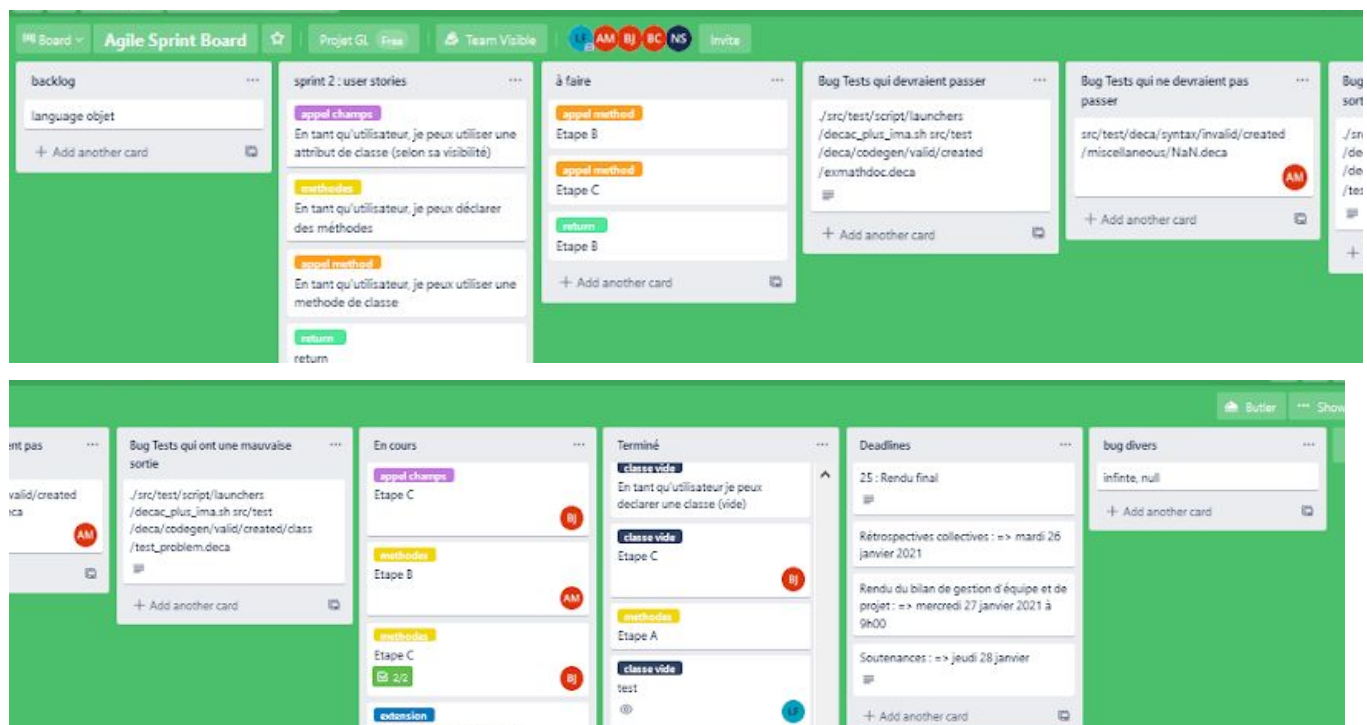
Le but de ces réunions était de répondre à trois questions pour chacun : Qu'avons-nous fait hier ? Qu'allons nous faire aujourd'hui ? Avons nous des questions ou besoin d'aide ?

Parfois, nous avons ressenti le besoin de poursuivre les réunions plus longtemps, pour résoudre des soucis de conception ou de compréhension des spécifications par exemple.

Les outils que nous avons utilisés pour ces réunions sont dans un premier temps Zoom, pour communiquer à l'oral, mais aussi pour partager nos écrans. Le cadre plus officiel de cette plateforme a très bien complété nos échanges très réguliers sur Facebook Messenger.

## I.3/Outils:

Du point de vue de l'organisation pratique, à partir du milieu de la 2e semaine le diagramme de Gantt ne nous a plus suffi pour nous organiser. nous avons besoin d'une vision plus détaillée de nos tâches, et d'une flexibilité que planner ne nous permettait pas. Sur une suggestion de l'enseignant SHEME (Philippe Bodiglio) nous sommes passés à Trello, que nous avons configuré comme ci-dessous.



### Retour critique:

Nous aurions aimé faire un burn down chart mais nous avons manqué l'opportunité.

### I.4/Répartition des rôles:

Cette première organisation “libre”, intuitive, nous a permis de prendre nos marques.

Après avoir tâtonné pour attribuer les tâches, chacun s’est attribué le rôle qui correspondait le mieux aux compétences cruciales qu’il possédait. Ci-dessous les détails de nos rôles respectifs:

Baudouin étant le seul d’entre nous à être à l’aise avec l’assembleur, il a donc pris en charge la partie de génération de code.

Benjamin, avec ses connaissances en programmation shell, a commencé par organiser l’automatisation des tests en développant de nombreux scripts. Il a terminé cette tâche au cours de la 3<sup>e</sup> semaine, ce qui lui a permis d’apporter son soutien pour les autres tâches.

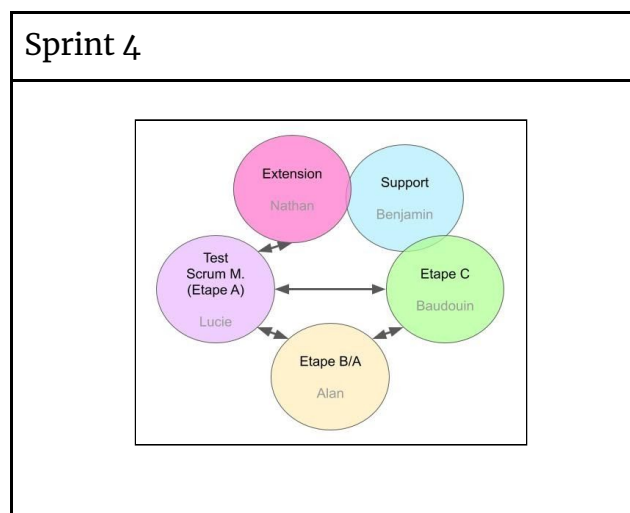
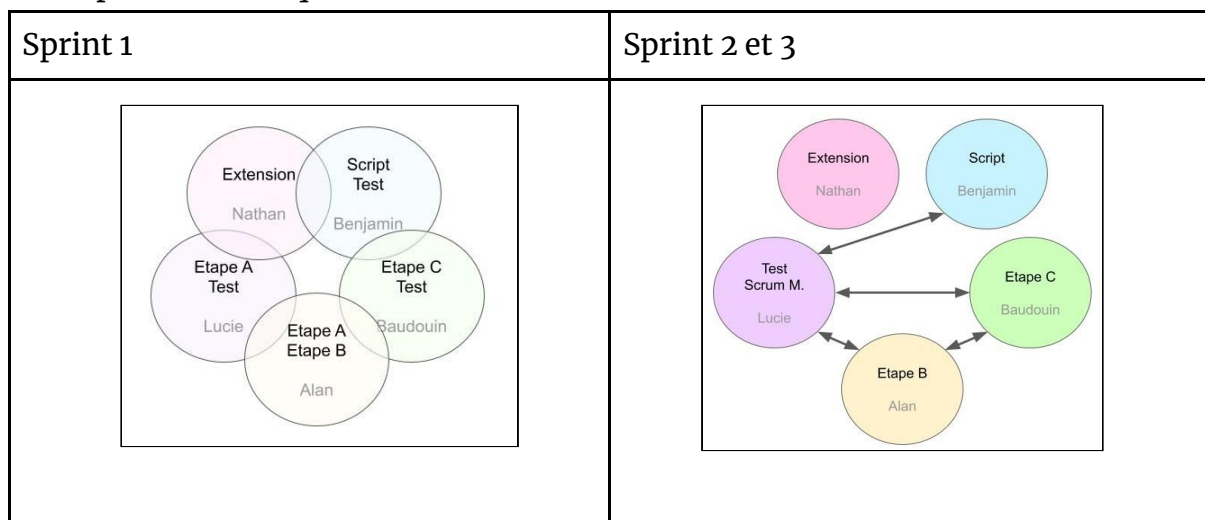
Lucie a participé dans un premier temps au Parser, puis s’est concentrée sur la partie test (fichier .deca). Elle a aussi joué le rôle de Scrum Master sur le reste du projet.

Alan s’est chargé en particulier de l’étape B (analyse contextuelle), mais aussi partiellement de l’étape A (analyse syntaxique), en collaboration avec Lucie.

Nathan s'est occupé de l'extension dès le début du projet, il a commencé par chercher des algorithmes permettant d'approcher les fonctions trigonométriques puis il les a codés en Python. Il a réalisé des tests de précision et de rapidité puis il a traduit les algorithmes Python en Java puis en Deca.

Ensuite, par souci d'efficacité (et par confort), chacun s'est cantonné au fil du projet au même type de tâche, avec relativement peu de perméabilité (quoiqu'il nous arrivait de naviguer parfois entre différents rôles lorsque nous étions oisifs). Par exemple, Benjamin, une fois que les scripts étaient plus au moins finalisés, a aidé Nathan à implémenter l'extension TRIGO en Deca et également fait une partie du Cast en java.

Cette façon de procéder nous a quelque peu isolé parfois mais elle a simplifié la communication : en cas de bug, il suffisait de savoir quelle partie était mise en cause pour savoir à qui s'adresser.



### Retour critique:

Si la division des tâches, en nombre suffisant par rapport à notre effectif, nous a permis d'avoir une progression confortable sur le projet, elle nous a aussi placé dans une situation de haut risque car nous avons fait entièrement confiance aux capacités de chaque maillon à remplir son rôle; pour exemple, la partie de génération de code était aussi conséquente que critique, or elle a été effectuée quasiment exclusivement par Baudouin.

## II/Présentation de l'historique du projet

### II.1/Ordre choisi pour la conception et le développement des étapes B et C

La conception des étapes B et C s'effectuait en parallèle afin de pouvoir disposer de toutes les informations utiles lors de la génération de code. Si une décoration manquait pour la génération de code, un simple message suffisait pour qu'elle soit ajoutée en quelques minutes.

Le développement de l'étape B se faisait en amont de celui de l'étape C pour chaque user story, ce qui permettait de tester et valider l'implémentation des deux parties immédiatement. A chaque incrément fonctionnel, les tests validaient ou non l'implémentation afin de passer au suivant.

Cette méthode bien rodée s'est révélée très efficace.

### II.2/Temps passé sur les différentes activités

#### II.2.1/Analyse

Nous avons passé beaucoup de temps à comprendre le sujet (surtout au début du projet : environ 50 % de notre temps), que ce soit dans la lecture du pdf de 231 pages, mais également le visionnage des vidéos ainsi que les cours en visio.

Par ailleurs la compréhension du sujet s'est poursuivie jusqu'à l'échéance du rendu.

Il nous est arrivé très souvent de lire et relire un point ensemble afin de s'assurer de regarder dans la bonne direction.

Pour l'extension, Nathan a passé beaucoup de temps d'analyse en recherchant sur internet des algorithmes, il a ainsi pu constituer le début d'un dossier pour garder des traces de ses recherches et les transmettre au reste de l'équipe .

### II.2.2/Conception

Pour l'extension Nathan a passé beaucoup de temps sur la conception. Il a commencé par coder les algorithmes en Python, puis en Java avant de les implémenter en Deca. Cette méthode a permis de s'assurer de la compréhension et de la bonne mise en œuvre des algorithmes rapidement. C'était plus évident de réaliser des tests de performance ou de déboguer en Python.

Pour la génération de code plusieurs heures ont été passées à comprendre et s'approprier les algorithmes récursifs pour les opérations arithmétiques et booléennes.

### II.2.3/Codage

Le codage du langage s'est fait tout au long du projet, particulièrement pour les parties B et C, où la manipulation des classes favorisait la compréhension du travail attendu.

L'implémentation de l'extension en deca a été faite en 2 ou 3 jours vers la fin du projet. Cela est dû au fait que la partie objet du langage n'était prête qu'à la fin du projet, et cette dernière est nécessaire à l'implémentation des fonctions trigonométriques. Ce temps a cependant suffi à implémenter les fonctions trigonométriques assez proprement, bien qu'elles ne soient pas parfaitement précises.

### II.2.4/Validation

Les Scripts de tests, les tests et la validation ont occupé 70% du temps de Lucie et de Benjamin sur le Projet. Développer et déboguer les scripts a pris beaucoup de temps, mais sans cet outil, la validation aurait été impossible, nous avons donc pris cela très au sérieux.

Notre objectif était de faire des tests les plus exhaustifs possibles (pour toutes les étapes), mais sans redondance afin de ne pas rendre la vérification interminable.

Ce que nous pourrions améliorer sur un futur projet serait de se pencher davantage sur les tests unitaires que nous avons boudés ici.

### II.2.5/Documentation

Certains documents devaient être discutés pendant les suivis, nous les avons donc rédigés avec sérieux.

La documentation sur l'extension devant être très dense, Nathan s'y est attelé dès la première semaine et l'a complétée tout au long du projet.

Nous avons commencé le reste de la documentation assez tard (sauf pour la documentation énergétique) car nous avons préféré jeter toutes nos forces dans le compilateur et l'extension sur la dernière semaine.

### III/Conclusion

A l'issue de ce projet, nous sommes un peu fatigués mais très fiers de notre travail. Nous le savons perfectible cependant nous n'avons pas honte de nos erreurs car elles nous ont permis de progresser.

Nous ne nous connaissons pas vraiment et pensons aujourd'hui former une équipe efficace et qui se fait confiance.

Groupe 01 Equipe 01