

BILAN DE GESTION D'ÉQUIPE ET DE PROJET

Vous trouverez dans ce document le bilan de notre projet d'un point de vu gestion d'équipe
et organisation

SOMMAIRE

Organisation du projet	2
Organisation de l'équipe	2
Chefs de projet	2
Répartition du travail dans l'équipe	2
Réunions quotidiennes	3
Télé-travail/Présentiel	3
Méthode Agile-Classique	3
Mise en place	3
Ajustement	4
Environnement de travail	4
Notion	4
Git	4
Discord	5
Bilan	5
Historique du projet	5
Etape A	6
Etape B	7
Etape C	7
Extension	8
Conclusion	9

CONSIGNES : Sur la méthode de développement, en s'appuyant sur Planning.pdf et Realisation.pdf :

— Description critique de l'organisation adoptée dans l'équipe.

— Présentation de l'historique du projet : ordre choisi pour la conception et le développement des étapes B et C, temps passé sur les différentes activités (analyse, conception, codage, validation, documentation).

L'objectif de l'enseignant n'est pas d'évaluer la qualité intrinsèque de la démarche adoptée, mais l'analyse qui en est faite a posteriori. Il sera lu par l'enseignant SHEME et l'enseignant en informatique.

Le bilan doit être au format PDF et être commité dans le dépôt sous le nom docs/Bilan _Projet.pdf.

Organisation du projet

Organisation de l'équipe

Chefs de projet

Nous avons choisi deux chefs de projet dans l'équipe avec deux objectifs distincts.

Il fut d'abord désigné un chef de projet orienté sur une vue d'ensemble du projet. Il avait pour rôle de s'assurer que les livrables seraient prêts dans les temps, d'organiser les réunions et les préparations de suivi et soutenances et de s'assurer du maintien d'une bonne communication et d'une bonne entente au sein de l'équipe.

Ensuite, il fut désigné un second chef de projet dont l'objectif était de coordonner l'avancée des travaux. Il devait s'assurer de la cohérence et de la qualité du code produit sur chaque partie. Il devait vérifier que le compilateur respecterait bien les normes et les formats attendus par le client.

Répartition du travail dans l'équipe

L'ensemble de notre travail a toujours été divisé en trois parties distinctes. Pour chaque version du langage deca à atteindre, nous divisons le travail de la façon suivante : Étape A, Étape B, Étape C. Nous répartissions ensuite les membres de l'équipe sur ces différentes parties selon les besoins du moment et l'expertise de chacun. Cela nous a également permis de nous spécialiser et donc d'être plus efficace sur les différentes étapes.

Vers le début de la deuxième semaine du projet, nous avons été amenés à modifier la subdivision des effectifs selon les parties afin de les faire avancer à un rythme équivalent. Cette modification a été fluide grâce à la communication qui a réduit le temps d'adaptation. Afin d'aider sur les parties où il restait plus de travail, nous avons également essayé de

réorganiser la répartition en fin de projet, cependant à l'aube de la fin du projet cette adaptation a été bien plus complexe.

Réunions quotidiennes

Afin de garder une idée de l'avancement général et de permettre à chacun d'exposer les problèmes rencontrés, nous organisons chaque matin une courte réunion d'équipe. Cela nous permettait aussi de planifier notre avancée pour la journée.

Télé-travail/Présentiel

Nous avons décidé de privilégier le travail en présentiel pour ce projet. Cela nous semblait plus intéressant et efficace de nous retrouver à l'Ensimag chaque jour que ce soit pour pouvoir s'entraider et se motiver entre nous ou communiquer plus facilement. Cependant, les horaires étaient souples et adaptables selon les obligations et volontés de chacun.

Méthode Agile-Classique

Afin de gérer la planification et la gestion du projet, nous avons utilisé une organisation alliant la méthode agile à la méthode classique.

Nous avons divisé le projet en 4 sprints :

- Hello World : atteindre un compilateur supportant le langage Hello World
- Sans Objet : atteindre un compilateur supportant le langage Sans Objet
- Essentiel : atteindre un compilateur supportant le langage Essentiel et la compilation séparée
- Complet : atteindre un compilateur supportant le langage Complet et la compilation séparée

Finalement, nous avons fait le choix de nous concentrer sur les trois premiers sprints sur le temps qui nous était accordé.

Chaque sprint était préalablement planifié. Pour cela, nous décomposons le travail à effectuer en tâches (regroupées par étape A/B/C), nous donnions une estimation du temps passé sur chaque tâche, puis nous réalisons un Gantt du sprint.

Mise en place

Au début, estimer le temps d'une tâche nous paraissait difficile car nous avions un faible aperçu du travail effectué en amont ou de la quantité de travail à fournir pour compléter le sprint. Ainsi, nous avons planifié le sprint 1 rapidement en supposant qu'il faudrait certainement admettre une grosse marge d'erreur sur cette première planification. Nous avons donc décidé de nous lancer sur le sprint 1 afin d'atteindre Hello World avec une

planification des tâches approximative. Ce choix nous a permis de nous plonger pleinement dans le projet, de prendre conscience de la charge de travail à effectuer sur chaque partie et donc dans un second temps de mieux nous organiser en divisant le travail plus clairement sur les sprints.

Ajustement

Au cours de chaque sprint, nous faisons des points réguliers pour ajuster le planning en cas de non-respect de la durée anticipée sur chaque tâche. Nous avons adopté une stratégie d'estimation plutôt large. De ce fait, nous étions rarement en retard et cela nous a permis de garder une vision plutôt confortable du futur proche et de ne pas nous retrouver avec de gros retards juste avant les deadlines.

Cependant, pour la fin du dernier sprint, l'étape C avait été mal découpée, donc le temps à passer dessus avait été mal prévu. De ce fait, on a légèrement dépassé le planning prévu, mais le retard a été contenu grâce à notre tendance à avoir une estimation large sur le temps pris par nos tâches.

Environnement de travail

Notion

Afin de gérer l'avancement de notre projet, nous avons utilisé un espace de travail Notion dédié à notre travail. Notion est un outil puissant permettant de faire de la gestion de projet et de gagner en productivité. Vous pouvez trouver à partir du lien ci-dessous une présentation de notre espace de travail :

<https://intriguing-suit-5e3.notion.site/Pr-sentation-de-notre-espace-de-travail-03fef8bba464410fa818a7518a809548>

Dans Notion, nous prenions des notes, nous planifions nos tâches, nous pouvions observer l'avancement du projet grâce à différents outils comme le Gantt et le tableau d'état (to do/doing/done).

Git

Pour pouvoir travailler sur les mêmes scripts et se partager les modifications, nous avons mis en place un système à plusieurs branches :

- branche master : branche qui sert de rendu, les fichiers ne sont déposés sur cette branche qu'après validation et sont fonctionnels
- branche develop : branche sur laquelle tout le développement est fait
- branche release : branche sur laquelle sont déposés les programmes finis de develop, ils doivent ensuite être validés avant de passer sur la branche master

En plus de cela, nous avons des sous-branches de develop (Etape A, Etape B, Etape C, CLI, Extension, etc) pour que chaque équipe puisse travailler sur chaque tâche sereinement sans craindre un push illicite d'un autre développeur.

Discord

Discord a été un outil extrêmement utile pour notre équipe. Nous l'utilisons comme messagerie pour échanger à distance, mais il a aussi servi à plusieurs reprises à réaliser des réunions à distance lorsque certains des membres de l'équipe étaient indisponibles en présentiel.

Bilan

Nous sommes globalement très satisfaits de notre organisation. Celle-ci nous semble plus que correcte et nous a permis de mener à bien ce projet tout en rencontrant peu de difficultés (il y en a eu mais cela était acceptable). Certains points sont quand même largement perfectibles.

En effet, la gestion des branches Git peut être grandement améliorée. Le fait d'avoir beaucoup de branches diverses nous a causé quelques problèmes au moment de fusionner les différentes parties. Il aurait été intéressant de rassembler notre travail plus régulièrement pour éviter les problèmes de collision et les pertes de temps.

Ensuite, il pourrait être très favorable de revoir la priorisation des tâches de chaque partie. Après coup, nous trouvons qu'il aurait été plus intelligent de mieux définir nos tâches sous forme de user stories et de progresser sur ces user stories dans les trois étapes (A/B/C) en même temps. Plus concrètement, il faudrait choisir un ordre de développement pour les différentes expressions et respecter ce même ordre pour les trois étapes. Par exemple, d'abord développer les expressions binaires complètement et avoir un compilateur qui supporte les expressions binaires de bout en bout. Cela nous aurait permis de pouvoir tester plus tôt nos codes et de s'assurer d'une meilleure cohésion entre les parties.

Historique du projet

Lors de ce projet, les sous-équipes travaillant sur chacune des étapes A, B et C étaient quelque peu indépendantes, et chacun était libre de son choix de méthode. Ainsi, les différentes activités se sont passées différemment sur les diverses étapes, à l'exception de la documentation. En effet, la documentation a été commencée à la fin de la troisième semaine du projet, et est toujours en cours de rédaction.

Les autres activités ont été effectuées en fonction des dates de début et de fin des 3 sprints sur les langages deca "Hello World", "Sans objet" et "Essentiel", et sont spécifiques par étape.

La répartition de notre équipe sur les différentes étapes a évolué sur ces 3 semaines.

La première semaine, nous étions trois sur l'étape A, un sur l'étape B et un sur l'étape C.

Les 3 personnes de l'étape A ont donc chacune pu atteindre une bonne compréhension de cette étape tout en mettant en place l'environnement de travail de groupe et l'organisation que nous allions suivre tout au long du projet. De plus, répartir deux autres personnes sur les étapes B et C nous a permis d'atteindre une compréhension suffisante pour ne pas être pris par le temps et commencer l'analyse et la conception du travail à réaliser sur ces deux étapes. De cette manière, nous avons garanti que l'étape A ai toujours de l'avance sur les autres étapes ce qui permettrait de pouvoir développer et tester progressivement chacune des étapes suivantes.

A la fin de la semaine, nous sommes ensuite passé à deux personnes sur l'étape A, une sur l'étape B et deux sur l'étape C. Autrement dit, une personne à glisser de A à B, et une de B à C. Cela nous a ainsi permis d'avoir deux personnes avec une vision un peu plus globale du projet, tout en gardant des personnes spécialisées dans leur partie. Faire une phase d'analyse poussée en première semaine a également permis d'estimer le temps et le nombre de personnes nécessaire pour chaque partie et nous a aidé à arriver à cet équilibre.

Nous disposions ainsi d'une partie A très bien testé en fin de 2ème semaine, ce qui nous a ensuite permis de dépêcher une personne sur les tests des autres parties lors de la 3eme semaine.

Etape A

Pour l'étape A, la phase d'analyse a duré deux jours, du lundi 3 janvier au mardi 4 janvier, pendant laquelle la sous-équipe assignée a lu les spécifications fournies, et cherché à comprendre ce qu'il y avait à faire. L'analyse de l'étape A a également été faite au cours des phases de développement, car le fait de développer nous a permis de mieux comprendre ce qu'il y avait à faire.

L'étape A avait deux parties: le lexer et le parseur. Au cours de la première semaine, nous avons codé la quasi totalité du lexer. Nous nous sommes lancés sur le parseur de la partie "Hello World". Le lundi 10 janvier la partie A du langage Deca "Hello World" était entièrement implémentée. La conception quant-à-elle était principalement fournie, et n'a donc pas pris de temps supplémentaire.

La validation du lexer a cependant attendu la moitié de la deuxième semaine, pendant laquelle la partie sans objet du parseur fut codée, ainsi que la validation de celui-ci.

Enfin, pendant la troisième semaine, nous avons fait en parallèle le codage et la validation du langage essentiel.

Etape B

Lors de la première semaine, nous nous sommes occupés du sprint 'Hello World' et de la mise en place des outils qui nous permettraient d'être plus efficace par la suite et de concevoir les sprints suivants de manière concise.

Ainsi, nous avons principalement effectué un travail de conception et de validation.

Nous avons donc listé les tâches à réaliser sur chaque sprint afin d'avoir une vision plus claire de cette partie de la conception du compilateur

Pour des raisons de répartition de tâches, nous avons choisi de changer de référent sur cette étape en fin de semaine et nous avons commencé par implémenter l'option -p. Celle-ci a permis de pouvoir tester efficacement l'Étape A et d'obtenir une meilleure compréhension de la structure du Projet pour l'étape B, aidant ainsi à l'analyse.

Lors de la deuxième semaine, nous avons construit et implémenté étape par étape les différentes méthodes permettant la compilation de la partie Sans-Objet, donnant une grosse phase de codage. La conception n'étant pas préparée, elle eût été faite à la volée. Dès que l'étape A a été fonctionnelle, il a été nettement plus simple de tester la partie B et ainsi de garantir un code correct pour la Partie C.

On a ainsi pu fonctionner pas à pas lors de l'implémentation de la Partie B en conceptualisant et codant fonction par fonction ce qu'il était nécessaire de coder pour la partie Sans-Objet d'abord et Essentiel lors de la 3eme semaine. Cela tout en testant au fur et à mesure.

Etape C

L'étape C étant quand-à-elle moins guidée, l'analyse était plus compliquée et a duré jusqu'au début du sprint Sans Objet. Le travail à réaliser ne nous apparaissait pas clairement, rendant l'analyse incomplète avant le début du développement.

Le sprint sur Hello World a donc été principalement constitué d'une phase d'analyse, sachant que le développement requis pour le rendu Hello World de cette étape était déjà fourni.

Au début du sprint sur le langage Sans Objet pour l'étape C, la conception et le codage de l'étape C ont commencé. La distribution des tâches au sein de la sous-équipe a nécessité d'avoir de petites phases de conception sur certaines matinées, afin de choisir et de décrire la manière dont le travail de chacun allait être intégré avec celui de l'autre. Enfant ce temps, le principal du temps passé était sur le codage. Il y avait également un peu de validation au cours de ce temps de codage, afin de s'assurer que ce qui avait été codé fonctionnait. Cela dura toute la semaine. Le weekend, une fois toutes les fonctions implémentées, nous avons eu une dernière grosse phase de validation, pendant laquelle l'étape a été intégrée aux autres.

Nous avons ensuite eu une deuxième phase d'analyse au début du sprint sur le langage Essentiel, alimentée par les cours TD sur l'étape C.

Pendant la première moitié de la semaine du dernier sprint, nous avons repris la même organisation temporelle que pendant le début du sprint précédent. Cependant, nous ne pouvions plus faire de vraie validation car les arbres devenaient trop compliqués pour être faits à la main, et donc sans l'étape B. L'une des deux personnes repassa donc sur l'étape B une fois le codage terminé afin d'accélérer son développement, pendant que l'autre se concentra sur l'écriture de tests, et une fois que l'étape B fut fonctionnelle, le sprint fut terminé par une grosse phase de validation et de debug, du vendredi soir au lundi du rendu final.

Extension

Pour ce qui est de la partie Extension, la manière d'aborder les choses a été nettement différente.

En effet, une longue réflexion a été nécessaire afin de pouvoir ajouter efficacement les imports de fichiers. Nous avons donc d'abord eu une phase d'analyse commune avec les autres membres du groupe afin d'avoir une idée claire de la structure que nous allions adopter, qui a duré tout le sprint Hello World et le sprint Sans Objet.

Ensuite, nous avons eu des phases de conception et codage spécifiques pour chaque étape, dont le principal but était d'adapter l'étape à l'extension. Nous avons également eu une petite étape "D", correspondant au développement de l'outil d'édition de liens.

La phase de conception spécifique à la partie A a été répartie sur une ou deux journées du deuxième sprint. Son codage a quant-à-elle été faite en même temps que le codage de l'étape A pour le langage Essentiel, et y a été bien intégrée.

Lors de la phase de conception spécifique à la partie B, on a pu choisir les structures de données que nous allions utiliser, et l'ordre dans lequel nous allions implémenter les différentes fonctionnalités pour que l'on puisse se baser sur ce qui avait déjà été construit afin de faire les tests.

De cette manière, on a d'abord fini de tester la partie Objet de la partie B afin de s'assurer que tout fonctionnait.

Nous l'avons ensuite généralisé à l'aide d'une nouvelle structure de données, puis retester pour s'assurer que tout fonctionnait au moins comme avant.

Et nous avons pu rajouter à la fin la gestion des imports, cette étape a été rendue très simple grâce à tout le travail fait en amont pour généraliser le compilateur afin d'accueillir les imports.

L'extension pour l'étape C était très petite. La phase de conception de la partie C a donc été très rapide, presque inexistante, et faite à la volée avant le codage. Le codage en question a été fait la veille du rendu final, et sa validation dans la foulée.

Pour l'étape additionnelle, que nous avons nommés étape D, la phase de conception était elle aussi très courte, et la phase de codage a été faite en deux fois, une fois avant la fin de l'étape C, et une fois après. Finalement, la phase de vérification fut rapide. Tout cela a été fait en une soirée.

D'une manière générale pour l'extension, l'analyse fut commune et assez conséquente, et les phases de conception et codage furent plus courtes et rapides. La validation était également rapide.

Conclusion

Dans l'ensemble, une grosse partie du temps a été utilisée pour le codage, et une partie un peu moins imposante a été sur la validation. L'analyse a quant-à-elle eu lieu au début du projet et au début de chaque sprint, et la conception était souvent faite à la volée, pendant le codage ou juste avant. L'exception fut l'extension, où l'analyse fut une partie primordiale, et la conception fut faite en amont du codage et de la validation.

La documentation vint ensuite dans un dernier temps, quand la pression fut moins importante.