

Bilan individuel de compétences

2. Travailler en mode projet

Il peut parfois être utile de définir des rôles précis par souci d'efficacité et de clarté, cependant nous ne nous connaissions pas vraiment.

Au début du projet, nous n'avons pas défini de chef et avons préféré nous familiariser au projet et aux autres. Plutôt que de nous enfermer dans un rôle qui ne nous correspond pas forcément, nous avons chacun choisi une tâche qui nous convenait.

Pour ma part, je me suis lancée sur le parser car j'aimais beaucoup la théorie du langage. J'ai trouvé ces premiers jours très importants : le projet a certes démarré plus lentement que si l'équipe existait depuis longtemps, mais cela a permis à une sorte de naturel de s'installer.

Par la suite, nous avons continué sans chef en tant que tel, chacun est le chef de sa partie et tout le monde à le même poids dans l'équipe.

Une fois en confiance avec le projet, il a été plus facile de décider d'une organisation. Étant de nature anxieuse et n'aimant pas les réunions à rallonge où personne ne sait qui doit prendre la parole, je me suis proposée pour être scrum Master la première semaine. J'avais déjà commencé à regarder les deadlines et les documents à rendre, organiser les temps de parole pour les suivis etc.. Au quotidien je ne suis ni un leader, ni très organisée mais je me surprend à aimer le rôle de Scrum Master. Je pense avoir gagné de la confiance dans ces domaines par rapport à ce que j'avais noté dans la charte d'équipe au début du projet.

J'ai choisi d'abandonner le Parser une fois l'analyse syntaxique sans objet terminée, et me suis lancée sur les tests. J'ai trouvé que cette position était très pratique pour le rôle de Scrum Master car j'étais en contact avec tout le monde et pouvait donc avoir une vision globale du projet.

Avec les conseils des professeurs et l'accord de mon équipe, j'ai intégré l'outil Trello dans notre organisation. J'ai aussi proposé un horaire officiel de réunion le matin. Ces manières de procéder étaient en fait une continuité de ce que nous avons déjà établi, donc tout le monde a pu apporter son expertise et ses idées pour améliorer les outils.

Nous avons commencé un diagramme de Gantt sur Planner mais nous l'avons rapidement abandonné car personne n'était très à l'aise avec. J'ai appris à visualiser le projet comme un ensemble de user stories, et ce découpage a été très efficace pour se rendre compte de notre progression sur Trello. J'ai trouvé qu'en plus d'apporter de la clarté, cette méthode est flexible (ce qui nous correspond bien) et très motivante.

L'outil Trello a été très utile pour anticiper les deadlines : il m'a permis de créer une liste où j'ai ajouté les dates au fur et à mesure avec toutes les informations nécessaires pour ne pas être pris de cours. Ces données étaient accessibles à tous en permanence.

Nous avons choisi d'utiliser les applications Messenger Facebook et Zoom pour communiquer respectivement au quotidien et pour les réunions. En accord avec la méthode

agile, j'ai donné la parole à chacun en suivant le schéma : "Qu'ai-je fait hier ?" "Que vais-je faire aujourd'hui?" "est-ce que j'ai envie de partager quelque chose en particulier ?" (question, besoin d'aide ou autre). Le groupe a su rendre cela professionnel et en même temps amusant. Les réunions étaient à la fois un moment agréable et efficace. J'ai appris qu'on n'était pas obligé d'être en contact permanent avec les autres pour bien travailler.

J'ai été très heureuse qu'on me propose d'être le Scrum Master sur la suite du projet. J'avais peur de ne pas être à la hauteur mais comme l'équipe fonctionne très bien, rien n'est vraiment difficile.

Il est compliqué pour moi de ne pas communiquer mon stress aux autres en général mais j'ai essayé de garder une attitude très positive pour ne pas ralentir l'équipe. Cet effort m'a fait voir ce projet sous un angle différent que par le passé. L'atmosphère dans le groupe était très détendue, à ne pas confondre avec relâchée ou désintéressée. Nous rappelions les prochaines deadlines et ce qui nous restait à faire à chaque réunion, mais toujours avec un esprit combatif.

A la fin du 3e sprint nous avions un peu de retard sur la partie C, les méthodes ne fonctionnaient pas, mais celui-ci s'est très vite comblé car nous avons essayé d'aplanir le terrain un maximum pour faciliter le travail qui restait, comme faire des tests, finir le contexte, décider des priorités. Si cela avait été nécessaire, nous avions un plan et savions quelles fonctionnalités abandonner (cast et instance of) pour rendre un livrable honnête.

Lorsqu'on n'intervient pas dans un domaine du projet (toujours dans la même idée de positivité) j'ai compris la grande différence entre être impuissant/spectateur et faire confiance aux autres.

3. Mettre en œuvre des processus de validation

J'ai appris quelques notions de shell grâce à un de mes coéquipiers qui faisait l'automatisation des bugs, mais je me suis surtout concentré sur la création de .deca et sur le fait de faire remonter les informations afin de faciliter au maximum le débogage.

Mon objectif premier est bien sûr d'assurer la validité de notre compilateur mais j'ai pris à cœur d'essayer d'isoler le bug le plus clairement possible pour que la partie débogage ne ralentisse pas l'équipe.

Pour la validation voici ma démarche:

1. Au moment où un de mes coéquipiers finit une étape, je relance la banque de test sur la partie correspondante. Je vérifie si des tests précédemment justes (ou faux) n'ont pas été corrompus ou si les tests nouvellement engagés ont des résultats inattendus.
2. Si c'est le cas, j'isole le problème afin de comprendre d'où il vient. Je mets toutes les informations (nom du test, commande utilisée, type de problème, sortie obtenue et sortie attendue) sur trello.
3. Je lance une alerte sur Messenger avec une analyse rapide du problème.
4. 3 fois par jour en moyenne, je lance toute la banque de test pour vérifier que des parties inattendues n'ont pas été touchées, et j'analyse la sortie JACOCO pour voir si les branches qui doivent être couvertes le sont effectivement.

Pour la création de tests:

J'ai conçu la base de tests en suivant les user Stories : chaque incrément fonctionnel doit être testé le plus exhaustivement possible (tester la génération de code a été le plus dur pour moi car la partie assembleur m'est très obscure).

Il a été difficile de concilier exhaustivité et rapidité de vérification. A ce jour, la base de tests fait sa vérification en moins de 5min. Il est plus facile au moment de la conception d'avoir des tests valides unitaires, mais une fois le code terminé, j'ai préféré fusionner la plupart des vérifications afin de gagner du temps.

Ce temps est précieux d'un point de vue durable mais aussi d'un point de vue pratique, pour les rendus afin de pouvoir faire des vérifications de dernières minutes.

Notre méthode de gestion de rendu repose en grande partie sur cette vérification générale qui permet de s'assurer de la bonne fonctionnalité intégrale du compilateur avant de délivrer le produit.

Ces 5 minutes ont été presque trop longues car nous avons eu un contretemps inattendu pendant le rendu final.

Je pense que la partie test dans ce projet est la plus facile en termes de conception, cependant elle demande du sérieux, de la créativité et de la méthode pour ne pas se disperser. Le découpage m'a bien aidé en cela car il m'est arrivé de me laisser déborder.

Conclusion

Pour conclure, j'ai vraiment beaucoup aimé ce projet et les gens que j'y ai rencontré. Je pense avoir énormément appris sur le travail en groupe, la gestion et l'organisation. Il est vrai que je suis restée dans une certaine zone de confort du point de vue des compétences techniques, mais je suis fière d'avoir appris des aspects humains que je considère tout aussi importants.