

Bilan Individuel des compétences

I. Réaliser une solution efficiente en réponse à un besoin ou une demande

Au cours du projet, j'ai, à deux reprises principalement, dû exploiter le cahier des charges pour fournir une réponse adéquate au travail qu'il m'était demandé de faire.

La première fois, j'ai travaillé sur la mise en place du parser pour le langage objet. En effet, j'ai dû créer, à partir de rien, une hiérarchie de classes en prenant exemple sur celle déjà présente. Il était nécessaire de faire une hiérarchie identique surtout pour les autres membres de l'équipe qui allaient travailler avec ces classes. En généralisant la hiérarchie de classes de sans objet, cela permettait de s'assurer qu'une personne extérieure ait plus de facilité à intégrer notre code.

La seconde fois, la plus probante, a été lors de la mise en place de la génération de code du langage objet, à laquelle j'ai participé. Il a été nécessaire pendant ce travail de faire des choix de conception. Nous avons dû faire parfois des choix qui n'étaient pas optimaux mais la contrainte de rendre un produit fini et viable dans le temps imparti a beaucoup influencé nos choix. Il était impératif pour nous de respecter le cahier de charges de base qui était de faire un compilateur fonctionnel. Étant consciente des choix et de leur impact sur le reste du projet, j'ai toujours gardé une note écrite des décisions prises au cours de la génération de code du langage objet afin de pouvoir, dans un premier temps, ne pas les oublier et, dans un second temps, les expliquer et les justifier pour les futures personnes extérieures qui travailleront sur notre code. De plus, lors de la mise en place de

cette partie il a été nécessaire de s'adapter aux outils dont nous disposions, notamment en apprenant à utiliser le débogueur d'im

De ce point de vue, ce projet m'a permis de comprendre que, dans certains cas, il était impossible de fournir un travail complètement parfait. J'ai pu ainsi comprendre les perspectives futures d'un projet car même après le rendu, nous avons réglé les problèmes que nous avons pu observer, et si des utilisateurs commençaient à utiliser notre compilateur il y aurait sans doute d'autres modifications à effectuer.

J'ai aussi appris que l'important dans un projet était de rendre un outil fonctionnel et surtout assez documenté pour que des personnes extérieures puissent facilement reprendre le code et le faire évoluer.

II. Mettre en oeuvre des processus de validation

Au cours de ce projet, j'ai été très impliquée dans la mise en place des tests de nombreuses étapes et dans leur automatisation. En effet, j'ai commencé le projet en travaillant directement par les tests: j'ai implémenté la décompilation pour le sans objet en écrivant les tests puis le code pour que les tests en question passent et réalisent le travail que je voulais. Cela m'a rendu à l'aise avec l'écriture de tests. Cependant, cette création fréquente de tests a vite nécessité que je les automatise.

Je ne connaissais pas beaucoup, voire pas du tout, les scripts en bash pour automatiser les tests, j'ai donc appris directement "sur le tas". J'ai pris exemple sur les scripts déjà présents, écrits par les professeurs et par un de mes coéquipiers, et j'ai directement expérimenté l'écriture de script. Cela m'a permis d'automatiser rapidement mes tests et j'ai pu ainsi aider d'autres camarades sur leur propre automatisation.

Dans la suite du projet, j'ai souvent été amené à faire des tests et j'ai finalement, dans la dernière ligne droite du projet, décidé de gérer l'entièreté de ces derniers. J'ai donc dû les ordonnées, les organisés, assurer leur bon fonctionnement, assurer leur en-tête mais aussi reprendre tous les scripts pour avoir la même idée générale (plus de facilité à les modifier et les comprendre par la

suite), écrire de nouveaux scripts qui avaient été oubliés, ajouter des tests valides et invalides même pour des parties que je n'avais pas étudiées en premier lieu...

Toutes cette organisation et mise en place de tests m'ont permis d'avoir une vision globale du projet, notamment en parcourant toutes les parties (parser, lexer, décompilation, contexte, génération de code et extension).

Cependant, lors du rendu de notre projet nous avons trouvé des failles dans notre compilateur, failles qui auraient pu être évitées par plus de tests précis. En effet, j'ai manqué de rigueur lors de la mise en place des tests car en général nous avons tous utilisé plus notre "sens logique" pour les créer qu'une réelle étude précise du cahier des charges. Par exemple, lors du rendu nous avons compris que notre compilateur ne gérait pas les *if*($(x = x+1) > 4$) alors qu'une étude de la grammaire nous aurait évité cet oubli. Je pense donc qu'à l'avenir je ferai preuve de plus de rigueur envers l'établissement de tests (dans ce cas précis il aurait fallu prendre la grammaire et s'assurer que chaque possibilité était testée).

Cette expérience m'a permis de découvrir l'automatisation de tests, que je n'avais encore jamais mise en place, et j'ai pu comprendre l'importance des tests et de leur automatisation comme les intervenants de conférence métier le répètent souvent.