

**LOG2990- Projet de logiciel d'application Web**

Groupe 01

Équipe #5

**Rapport final de gestion de projet**

**Présenté à Michel GAGNON**

**David TREMBLAY 1748125**

**Younas EL FOULADI 1750533**

**Maroua DJAROUD 1806028**

**Dominique PICHÉ 1766981**

**Jean TONG 1745133**

Département de génie informatique et génie logiciel

Le 5 décembre 2017

École polytechnique de Montréal

**Table des matières**

[Introduction 3](#_Toc500248217)

[Stratégie de planification adoptée 3](#_Toc500248218)

[Évaluation de l'effort, assignation des tâches et suivi 3](#_Toc500248219)

[Progrès de la capacité de planification 3](#_Toc500248220)

[Assurance qualité 3](#_Toc500248221)

[Approche agile 3](#_Toc500248222)

# Introduction

Durant ce projet, nous avons acquis énormément de notions théoriques qui nous ont permis de créer une application web orientée objet. Nous avons également appris à travailler par pair.

Dans ce qui suit, nous ferons une réflexion globale sur la stratégie de planification adoptée pendant les 4 sprints, nous aborderons notre manière d’assigner les tâches ainsi que nos efforts fournis. Nous analyserons notre progrès au fil du projet, et nous discuterons de l’assurance qualité ainsi que de l’approche agile.

# Stratégie de planification adoptée

# Évaluation de l'effort, assignation des tâches et suivi

# Progrès de la capacité de planification

Nombres d’heures effectué / nombres d’heures planif

73,5/81

--/215 avec plusieurs taches du sprint 1

59/66

168/211

Comme le montre le graphe ci-dessus, nous avons eu du mal à ajuster la quantité de travailler nécessaire avec la quantité de travail réellement fournie. Certaines taches nous ont demandé plus d’efforts que d’autres comme la création de piste (32 au lieu de 21) ou encore le service lexical (46 au lieu de 8). Et c’est en tentant d’ajuster ces déséquilibres des premiers sprints que nous avons essayé de planifier les suivants. Nous avons également subi un changement majeurentre le deuxième et troisième sprint avec le retrait d’un membre de l’équipe et une concentration sur les taches du Racing Game uniquement. C’est pour cela que nous avions une telle différence de quantité de travail entre ces deux sprints. (montrer la difference de nmbre de taches entre les deux sprints) Les sprints 3 et 4 ont été les plus intenses vu le retard pris lors des premiers sprints. Cependant, nous avons fait en sorte de ne pas surcharger le sprint 3 afin de pouvoir fournir un code de qualité et ainsi rattraper les points perdus aux précédents sprints.

Enfin nous avons pu après ce changement fournir plus d’effort afin d’atteindre nos objectifs et ce malgré le peu de temps qui nous sont alloué pour le projet dans la session. Ce qui nous a demandé de se rencontrer en dehors de ces heures.

# Assurance qualité

Dès le début du projet, le professeur a suggéré une approche de développement TDD pour le projet. C'était une méthode de travail qui nous semblait judicieuse d'appliquer et les membres de l'équipe se sont mis d'accord pour au moins l'essayer. Par contre, la réalité est que le TDD n'a jamais été mis en pratique dans l'entièreté du projet. Plusieurs raisons ont causé ceci. Tout d'abord, au début du projet, nous avions moins de temps dédié à l'avancement du projet puisque nous avions 4 heures de cours théorique par semaine. En plus de ce temps en moins, une grande partie du temps restant était consacré à lire sur les différentes technologies et à la planification des différents components, services, modèles, etc.. Nous étions donc plus motivés à commencer à faire du travail concret que de prendre encore plus de temps à lire sur comment faire des tests et planifier le TDD. Nous avons donc fait des tests à la toute fin du sprint 1 finalement. La rétroaction des chargés de cours indiquait que nos tests n'étaient pas adéquats et nous avons donc continué de faire des fonctionnalités au lieu d'appliquer le TDD pour le second sprint puisque cette approche semblait erronée sans une bonne connaissance sur les tests. À partir du sprint 3, nous avons commencé à travailler exclusivement sur le jeu de course et ce qui devait être testé pour cette partie du travail était très flou. Nous avons donc encore une fois continué à développer des fonctionnalités sans nous soucier des tests au départ puisque nous n'étions pas sûrs de ce qui devait être testé. Finalement, pour les mêmes raisons que le sprint 3 et parce que nous n'avions finalement jamais fait du TDD, nous avons continué dans la même direction pour l'étape finale du projet.

L'assurance qualité a été délaissée lors des deux premières remises du projet et une grande incompréhension au sein de l'équipe sur ce qui était attendu par les correcteurs était aussi présente. L'assurance qualité a surtout été ignorée jusqu'à la dernière journée de la remise pour la moitié du projet et nos tentatives pour remédier aux problèmes si tard dans le développement étaient alors plus ardues puisque la taille du projet était plus grande et parce que les erreurs étaient maintenant bien implémentées dans le code (mauvais noms de fichiers, incohérence de code, séparation services/components, etc.).

Nous n'avions pas abordé notre norme de codage lors du premier sprint. Ceci nous a causé de nombreux problèmes lors de la première remise. Des attributs étaient écrits avec des underscores, étaient explicitement déclarés publics ou privés, étaient initialisés, etc. alors que d'autres ne l'étaient pas. Nous avons donc discuté des normes à propos des attributs lors de la planification du sprint 2 pour être tous d'accord sur la marche à suivre. Par contre, le jour de la remise, plusieurs problèmes de normes de codage ont tout de même été décelés. L'incohérence du code était flagrante. Certaines méthodes avaient des types de paramètres et de retour tandis que d'autres non, plusieurs imports de service et attributs étaient inutilisées, etc.

Le résultat est que nous avons eu **25%** pour la partie assurance qualité pour les sprints 1 et 2. Un grand changement devait être fait et nous avons donc passé plusieurs heures avec Dylan pour regarder notre code et comprendre ce qu'il attendait de nous. Nous avons donc appliqué ses conseils et avons fait du refacto de code tout au long des sprints 3 et 4 (pas seulement à la remise). Voici quelques-uns des changements apportés :

* Nous avons fait une librairie de constantes contenant les strings et nombres magiques du projet.
* Nous avons ajouté de nombreux services pour réduire la taille des services et que chacun ait une tâche précise.
* Nous avons coupé les méthodes trop longues pour plusieurs petites méthodes
* Nous avons créé une arborescence pour mieux séparer nos différents services

Le résultat a été une note de **80%** au sprint 3 pour l'assurance qualité. Nous avons été très heureux de cette nouvelle et avons appliqué les mêmes méthodes pour la fin du projet.

# Approche agile