

INF2990
Projet de logiciel graphique interactif
Hiver 2017

Rapport de fin de session

Soumis à : Michel Gagnon

Présenté par : Équipe 04
Hermann Kamsu-Tchuate

Félix Agagnier

Khalil Bennani

Joël Poulin

Simon Felteau

Alexandre Clark

19 avril 2017

École Polytechnique de Montréal
Département de Génie Informatique et Logiciel



**POLYTECHNIQUE
MONTREAL**

LE GÉNIE
EN PREMIÈRE CLASSE

1. Introduction

Maintenant que nous avons pratiquement terminé le quatrième sprint, il nous semble pertinent de revenir sur le projet en entier et d'analyser ce qui en est de notre progression face aux méthodes de gestion de celui-ci. Nous parlerons donc principalement de nos méthodes de planification au fil du projet, de la répartition des tâches selon les compétences, du suivi des tâches et besoins, du respect de l'assurance qualité ainsi que de l'approche agile.

2. Stratégie de planification adoptée

À ce qui a trait à la stratégie adoptée pour la planification globale du projet, nous avons suivi la méthode suggérée dans le cours, c'est-à-dire adopter un principe de développement SCRUM, divisé en quatre sprints. Au départ, nous nous sommes consultés dans le but de s'entendre sur des principes de planification généraux. Étant donné qu'il y avait une charge d'apprentissage très imposante au début du projet, nous avons anticipé le tout et établi qu'il serait plus long avant de pouvoir commencer le développement du projet. De plus, étant donné que notre niveau de compétence en développement web était faible au début du projet, nous avons choisi de prioriser les tâches les plus faciles. Le SCRUM Master nous a aidé durant les premières semaines du projet pour déterminer la faisabilité des tâches dans le temps demandé. Au fil des semaines, nous avons pu déterminer, principalement grâce à *Trello*, la quantité de code et de fonctionnalités que nous pouvions réaliser dans un laps de 3 semaines.

Ensuite, un autre principe général que nous avons choisi d'adopter est de ne pas garder les tâches les plus difficiles pour la fin du projet, mais plutôt de séparer chacune des tâches les plus corsées équitablement à travers les sprints. Ainsi, nous nous sommes évités d'avoir un dernier sprint trop chargé.

Au cours du dernier sprint, un changement d'organisation de l'équipe fut avancée par le professeur et acceptée par l'équipe, où trois membres jouaient le rôle de mentor et trois autres devaient développer. C'est pendant cette période que le *Peer Programming* a été appliqué le plus durant le projet. Nous avons pu constater les effets sur la productivité et que

celle-ci n'a pas été gravement affectée. Toutes les tâches planifiées ont pu être accomplies et il restait du temps pour l'optimisation du projet et des changements ergonomiques.

3. Évaluation de l'effort, assignation des tâches et suivi.

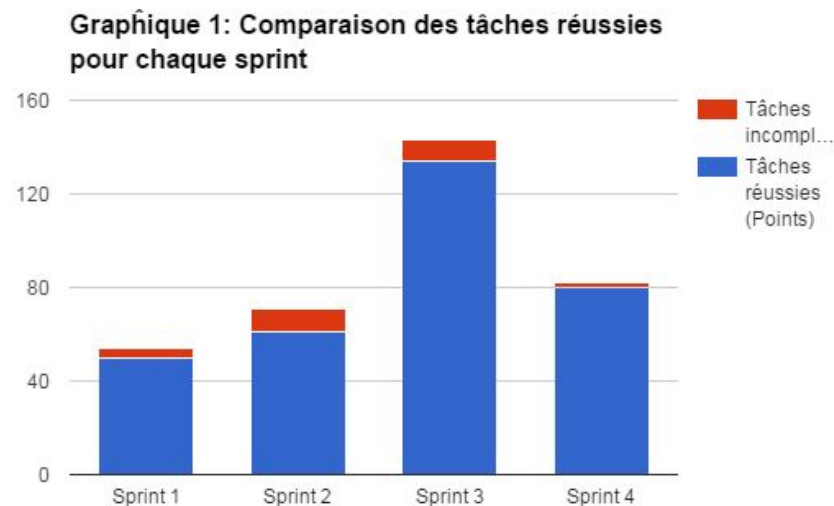
Pour le sprint 1 les membres de notre équipe ont choisi avec l'aide du SCRUM Master les tâches permettant de faire le squelette pour chacun des trois jeux. Parmi toutes les tâches initiales choisies, chaque membre a choisi les tâches qui lui convenaient. Chaque mardi nous nous sommes rencontrés pour discuter des progrès, difficultés et inquiétudes de chacun. Pour les sprints deux et trois, les tâches ont toujours été laissées au choix, mais un accent a été mis sur le fait que chacun devrait travailler sur les trois jeux. Pour le dernier sprint, le suivi a essentiellement été fait lors des différents échanges entre les trois qui codaient et les trois qui assuraient le mentorat.

Notons qu'avec l'outil de communication Slack, on s'est créé un canal pour chaque jeu et un forum général, dans le but de savoir exactement l'avancement de chaque jeu et de chaque personne en dehors des heures de rencontre. L'évaluation de l'effort réside dans: le temps mis pour réaliser une tâche, le nombre de tâche faite et enfin le nombre de sollicitation de la part des autres membres de l'équipe.

Pour ce qui est de la manière d'assurer le suivi, tout se portait autour de notre session *Trello* ainsi que de notre présence lors des rencontres du mardi. Cette présence pouvait être qu'en ligne, notamment à l'aide de *Slack*, mais une vraie rencontre était préférable. Un problème que nous avons fait face est le fait qu'il était presque impossible de trouver un niveau d'entente pour les horaires de chacun. Ainsi, nos rencontres n'étaient pas toujours composées de tous les membres de l'équipe.

Un des problèmes que nous avons fait face au premier sprint est de ne pas mettre à jour les cartes du *Trello* assez régulièrement. Ce pépin a rapidement été réglé après le deuxième sprint. Une bonne subdivision des groupements de cartes a beaucoup aidé à la visualisation des tâches en cours ("À faire" "En cours" "Validation" "Terminé").

4. Progrès de la capacité de planification



Le graphique ci-dessus démontre bien les différents impacts que nos décisions sur la gestion ont eu sur notre projet. Dans un premier temps, il est possible de voir qu'au fil des semaines, notre capacité à déterminer la charge de travail réalisable pour le sprint s'est améliorée et que notre pourcentage de tâches réussies est passé de 86% à 97%. De plus, on peut voir que le nombre réduit de tâches lors des deux premiers sprints nous a permis de bien maîtriser les technologies et de générer une incroyable production au Sprint 3. De cette façon, le dernier sprint a été moins chargé et nous a permis d'apporter les correctifs nécessaires et d'appliquer le *peer programming* en mentorat.

L'effort à travers les différents sprints n'a pas été égal; notre équipe s'est mise d'accord sur le fait que pour le sprint 3, soit à un point dans notre session où nous n'étions pas ensevelis sous les travaux, nous étions capables de travailler plus à l'extérieur du cours et dépasser le 12 heures prescrits. Cependant l'effort n'a pas été fourni par tous les membres et il y a eu un débalancement dans la contribution de chacun.

5. Assurance qualité

D'une part, l'assurance qualité n'a pas été notre point fort lors du projet. En effet, notre couverture pour les tests a été loin d'être complète. Des tests sont bel et bien présents, mais ne sont pas issus de TDD (Test Driven Development) et ne permettaient pas de bien discerner les problèmes lorsqu'ils surviennent. Au fil des semaines, nous avons surtout mis l'accent sur le Tslint et les normes de codages.

D'autre part, la séparation de la logique dans des services et la modularité de nos serveurs auraient d'ailleurs pu être grandement améliorés. Comme mentionné plus tôt, la répartition des tâches et de l'effort n'a pas été égalé dans l'équipe et certains membres ont dû compenser lors des trois premiers sprints pour s'assurer de compléter les tâches, ce qui a diminué le temps consacré à la qualité du code dans les autres tâches.

6. Approche agile

La méthode agile est une démarche efficace pour maintenir un rythme de production constant tout au long de la période de développement. Grâce à cette méthode, nous n'avons pas eu de période de stress intense quelques jours avant la remise tel que le Projet 1, où nous n'avons pas utilisé la méthode agile et avons réalisé le projet en grande partie dans les derniers jours. Nous croyons que plus d'encadrement en début de projet aurait grandement aidé, notamment à ce qui a trait à l'introduction aux nouvelles technologies et au déroulement global du projet. Un plan d'apprentissage avec des objectifs précis et des exercices pratiques aurait pu également aider à faire raccourcir la période d'apprentissage du départ.