

Organisation de travail

Mohammad Nauval

14 décembre 2017

Dans cette partie, nous examinons l'organisation de notre projet de compilation. Ce projet développé par une équipe de six personnes, est pour nous l'occasion d'expérimenter le développement en agilité, évoqué en cours de génie logiciel. L'agilité est une approche basée sur 12 principes et 4 valeurs¹. Le développement agile est entre autres basé sur la collaboration entre le client du produit et les développeurs. Il prévoit de satisfaire le besoin client en livrant des versions fonctionnelles du produit, de manière itérative et incrémentale.

L'un des principes d'agilité est de "livrer fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts". Alors le développement du logiciel en agilité progressent par des sprints. Dans chaque sprint, le développement du logiciel fait un cycle. Ce la veut dire que les fonctionnalités sont conçues, codées et testées pendant ce cycle. Le logiciel est livré régulièrement aux clients dans chaque *release*, qui consiste de quelques sprints.

Alors nous avons décidé d'effectuer 2 livraisons et chaque *release* composé de 3 sprints de 2 semaines.

1 Outils

1.1 Trello

Trello² est un outil de gestion de projet en ligne. Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des

1. <http://agilemanifesto.org/iso/fr/principles.html>

2. <https://trello.com/>

tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement³.

Nous avons utilisé Trello pour planifier et gérer l'organisation de notre travail. Nous avons créé des tableaux pour planifier des sprints. Dans ces tableaux, nous avons eu plusieurs planches contenant des tâches que nous avons eu à compléter pour chaque tâche.

1.2 Slack

Slack⁴ est une plate-forme de communication collaborative. Pendant le développement de notre projet, nous avons communiqué en utilisant cet outil. L'intérêt d'utiliser Slack est sa facilité à envoyer des messages et partager des fichiers. Le fait que nous pouvons avoir plusieurs espaces de travail qui nous permet d'avoir différents espaces de travail pour différentes équipes de projet est un point fort de Slack. Dans un espace de travail, nous avons aussi la possibilité d'avoir plusieurs chaînes pour différents types de conversation.

2 Organisation de travail

2.1 Produit final et ses versions

Avant le développement, nous avons défini des fonctionnalités que notre projet doit accomplir. Les exigences sont exprimées en tant que user story et stockées dans le backlog du produit. Comme mentionné ci-dessus, nous livrons le produit en 2 releases : les fonctionnalités sont ensuite découpées pour chaque *release*.

3. <https://fr.wikipedia.org/wiki/Trello>

4. <https://slack.com/>



FIGURE 1 – Backlog produit du projet

Car l'agilité s'appuie sur une équipe auto-organisationnelle, nous avons amené de définir les fonctionnalités fournies pour le release 1. Alors pour release 1 nous avons décidé d'implémenter les fonctionnalités suivantes :

L'utilisateur peut saisir le code source MiniJaja réduite (MiniJaja qui permet avoir des variable de type int et permet d'additionner des nombres ou des variables), vérifier que le code source ne contient pas des erreurs de typages, le compiler vers le JajaCode et l'interpréter. De plus, permettre aux utilisateurs exécuter le JajaCode obtenu. Étant conscient que les clients (les utilisateurs à qui destiné le logiciel), n'ont pas des compétences techniques pour utiliser les fonctionnalités fournies sans interface homme-machine, bien évidemment, qu'il y aura une interface utilisateur minimum ergonomique pour faciliter l'utilisation de notre logiciel.

Nous avons décidé de développer presque toutes les fonctionnalités attendues du produit final (sauf l'exécution pas-à-pas et point d'arrêt) pour une grammaire réduite, car ça nous permet fournir un logiciel fonctionnel pour le premier release et avoir retour de client sur implémentations des exigences.

Pour le release 2 nous avons prévu que le logiciel va contenir les fonctionnalités attendues pour un produit finale.

2.2 Organisation de sprint

Comme ce que nous avons déjà mentionné, avec l'approche d'agilité, nous organisons le travail en plusieurs cycles appelant sprint.

Au début de chaque sprint, nous avons organisé une réunion pour définir les tâches que nous avons eu à compléter à la fin du sprint. Pour cela, nous avons examiné le *backlog* de *release* suivant, à partir des fonctionnalités décrites dans ce dernier, nous avons pu extraire les tâches à compléter. Le but c'était de pouvoir implémenter toutes les fonctionnalités décrites dans le backlog de chaque *release* à la fin de trois sprints.

Ensuite, nous avons créé un tableau pour chaque sprint. Dans le tableau, il y a eu trois 4 listes. La première, c'était la liste de tâches à faire. La deuxième et troisième sont la liste de tâches en cours de réalisation et la liste de tâches finies à vérifier. Et la dernière liste, c'était la liste de tâches complétées.

Dans la réunion au début de chaque sprint, chaque membre de l'équipe a choisit les tâches qu'il voulait faire. La réalisation de tâche peut aussi se faire en binôme si sa taille est assez importante. La personne responsable de chaque tâche devait informer son avancement en modifiant l'état de tâche (en cours, à vérifier, ou finie). Pour chaque tâche, nous avons aussi prévu sa durée de réalisation et à la complétion d'une tâche, la durée réelle de ce dernier est aussi révélée.

Par exemple, voici à ce quoi notre tableau de deuxième sprint de *release 1* ressemble.

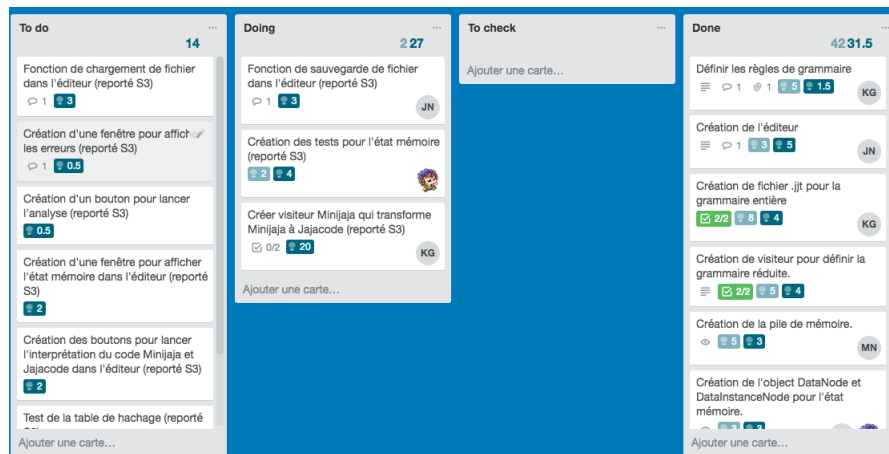


FIGURE 2 – Tableau d'organisation de travail de sprint 2 de *release 1*

Les tâches non-complétées doivent être reportées au sprint suivant. La personne travaillant sur une tâche doit aussi s'occuper de la réalisation des tests pour vérifier ce qui a été fait.

3 Réunions

3.1 Réunions régulières

Nous avons organisé des réunions régulières tout au long du développement de notre projet. Nous avons eu des réunions pendant les travaux pratiques en cours de compilation. Nous avons aussi organisé des réunions avant chaque sprint pour définir les tâches que nous avons eu à compléter pour le sprint suivant. Les réunions sont aussi parfois organisées pendant le sprint pour faire le point sur l'avancement de projet et se discuter entre nous.

3.2 Revue de sprint

Une réunion est aussi organisée à la fin de chaque sprint pour discuter les tâches complétées sur le sprint précédent. S'il y a des tâches non finies, nous les avons reportées au sprint suivant. Nous nous sommes aussi discuté pour répondre des questions posées entre nous.

3.3 Rétrospective

Une rétrospective est organisée à la fin de chaque *release*. Dans cette réunion, nous avons fait le bilan de l'organisation de notre projet pendant les sprints précédents. La rétrospective sera expliquée plus en détails dans la partie Rétrospective.