

Organisation de travail

Mohammad Nauval

14 décembre 2017

Dans cette partie, nous allons examiner l'organisation de notre projet. Comme la taille du projet est assez grande et il y a six personnes dans notre équipe, le projet est découpé en plusieurs modules. Les modules sont :

1. Analyseur
2. Compilateur
3. Interpréteur Minijaja
4. Interpréteur Jajacode
5. L'état mémoire
6. Contrôle de type
7. Interface homme-machine

Nous avons implémenté l'approche d'agilité dans l'organisation du travail de notre projet. Nous organisons le développement en cycles de quelques semaines.

Nous avons eu 12 semaines pour finir le projet. Au début du développement, nous avons planifié deux *releases* pour livrer le projet. C'est-à-dire que nous avons eu 6 semaines avant chaque *release*.

Pour *release* 1, nous avons décidé de réaliser toutes les fonctionnalités du projet. Cependant, celle-ci n'était logiquement pas possible. En effet, le projet livré pour *release* 1 n'a pas couvert la totalité de grammaire Minijaja, il fonctionne pour une grammaire réduite. Ensuite, Le projet était élargi pour couvrir la grammaire complète pour *release* 2.

Avant le développement, nous avons défini des fonctionnalités que notre projet doit accomplir. Les fonctionnalités sont ensuite découpées pour chaque *release*. En effet, nous avons les fonctionnalités à accomplir pour *release* 1

ainsi pour *release 2*. Ces fonctionnalités sont définies dans le *backlog* du produit. À partir du backlog produit, nous avons pu définir les *backlog* de *release 1* et 2.



FIGURE 1 – Backlog produit du projet

Comme ce que nous avons déjà mentionné, avec l'approche d'agilité, nous organisons le travail en plusieurs cycles. Les cycles sont appelés des sprints et chaque sprint dure deux semaines. Nous avons donc eu 3 sprint avant chaque *release*.

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 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 choisi 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.

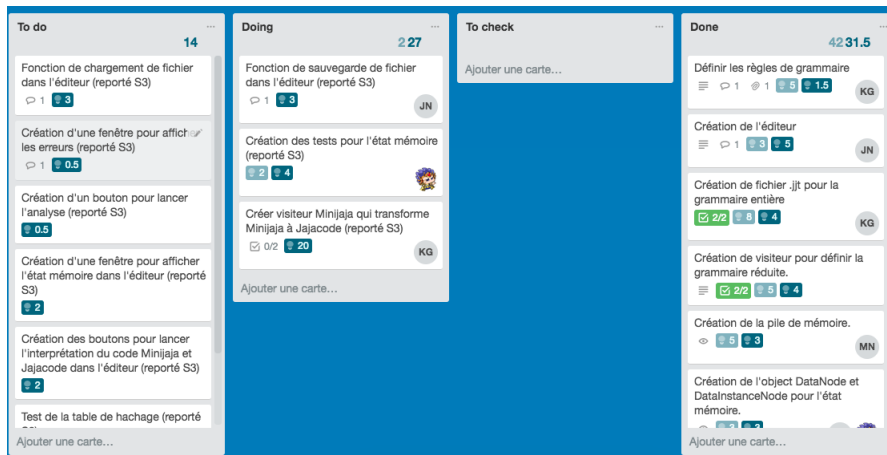


FIGURE 2 – Tableau d'organisation de travail de sprint 2 de *release 1*

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.