IUT Sénart-Fontainebleau, UPEC
Méthodo
IUT Info 2A

Florent Madelaine

2018-19

Le projet et son évaluation

Le projet et son évaluation

Il s'agit de travailler sur le jeu de plateau Kahmaté. Je vous ai distribué les règles. Elles sont disponibles en intégralité sur iluvatar. Vous pouvez probablement trouver des videos montrant des parties sur internet.

Je joue le rôle du client. Nous ne ferons pas de cahier des charges à proprement parler car vous avez une assez bonnes expériences à ce sujet dans d'autres cours et avec le projet tuteuré.

En passant, le code produit va jouer sur la note mais les aspects méthodologiques vont jouer pour une très grande part. En particulier, si le produite demandé n'est pas complètement terminé, ce n'est pas un soucis, tant que vous avez joué le jeu de travailler sur le projet avec une méthode agile adaptée de la méthode scrum.

On attend évidemment du code bien écrit, commenté et accompagné d'une documentation raisonnable (par exemple produite avec javadoc). Pour être considéré comme un incrément, tout code correspondant à une *user story* doit être accompagné de tests (nous verrons plus de détails à ce sujet avec JUnit en semaine 2).

Si vous souhaitez faire des diagrammes de classes ou tout autre diagramme UML, c'est votre choix en tant qu'équipe de dev. En tant que client, je n'ai pas besoin de voir ces diagrammes. En tant qu'évaluateur de votre travail non plus. Si cela vous aide pendant un TP que je lise/discute de tels diagrammes avec vous, je le ferais avec plaisir.

Par rapport à l'organisation scrum, je vais vous noter en partie sur ce que j'observe pendant les TPs. Je vais faire une brève soutenance informelle pour chaque groupe en fin de période à la fin des 3 semaines. Je vais avoir besoin d'un petit rapport de 3 ou 4 pages expliquant les aspects méthodologiques de l'équipe, ce qui a marché/ pas marché etc. Il s'agit essentiellement de ce que l'équipe doit produire au moment de la *sprint retrospective*.

Lien avec le client : product Backlog et product owner

Normalement c'est le travail du *product owner* de gérer avec le client le *product backlog*. Vu le temps que nous avons, nous allons tricher et je vous donne cette partie (voir Figure 1 pour les détails).

Le *product owner* sera responsable de travailler un peu plus sur le *sprint backlog* que les autres, et sera le canal dédiée pour que j'explique ce que je veux à l'équipe.

Le Scrum Master

C'est le membre du groupe qui a minima va lire la doc de 20 pages expliquant la méthode scrum et essayer de la digérer pour faciliter sa mise en oeuvre par l'équipe.

Qui fait quoi ?

C'est l'équipe qui s'organise. Le *scrum spring board* (physique pour la plupart des équipes) sert à s'organiser. Dans notre cas, le scrum master et le product owner codent aussi.

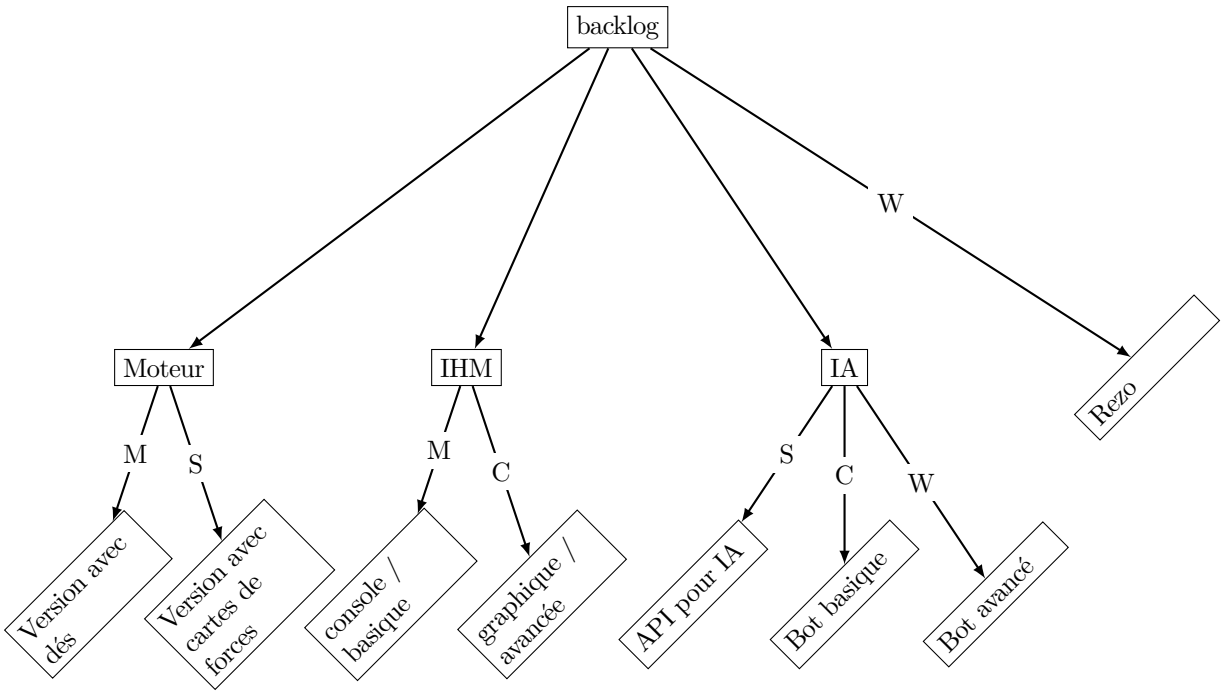

FIGURE 1 – Esquisse du product backlog : les feuilles de l'arbre correspondent à des *use case* auxquels j'ai ajouté une priorité avec la notation MoSCoW.

Les sprints

Nous ne pouvons pas suivre le rythme classique d'un sprint. Comme c'est intéressant de simuler la fin d'un sprint, je propose l'organisation suivante.

- Semaine de cours 1 : sprint 0 et début du sprint 1.
- Semaine de cours 2 : sprint 1.
- Semaine de cours 3 : sprint 2.

Etc

Je recommande fortement d'utiliser le *pair programming*[1].

Le client change d'avis

Si le client change d'avis, c'est plutôt normal. Le produit est pour lui *in fine* et il faut adapter le backlog à ses demandes. Le *product owner* doit toutefois protéger son équipe en donnant des délais raisonnables au client, selon ce que l'équipe peut réaliser. Le *scrum master* est là pour expliquer et trouver des solutions pour remotiver les membres de l'équipe. Par exemple, lorsque un membre d'équipe vient de travailler sur une partie qui devient obsolète.

1. à ne pas confondre avec *peer* ou *pear*.