

Compte-Rendu Réunion n°3

9 Novembre 2016 / 11:00-12:00

Entrevue responsable pédagogique

PARTICIPANTS : TOUTE L'ÉQUIPE DE PFA ET JULIEN ALLALI

SECRÉTAIRE : ELODIE FENG

Sujet de la rencontre :

Mise en oeuvre de la méthode de développement

- M. ALLALI commence par nous demander de **spécifier la demande du client** pour voir si c'est bien clair après rencontre avec ce dernier.
—> On se pose alors la question de l'entrée de notre logiciel :
 - **1 ou plusieurs fichiers SVG ?**
 - **Une ou plusieurs formes (textures)** par fichier ?
- Notre équipe annonce que la méthode de développement choisie sera agile, et ce sera la **méthode Kanban**. Donc les deux premières choses à faire d'après M. ALLALI sont :

Etape 1 : Faire le Product Backlog

- Analyser les **besoins fonctionnels**, du point de vue **utilisateur** (≠ développeur !)
- Faire de **petites** UserStories
- /\ Attention, un Product Backlog n'est pas un cahier des charges. La notion de **priorité** est majeure, et l'**exhaustivité** n'est pas demandée (voire pas bienvenue)
- Formalisation des UserStories grâce au **critère INVEST** :
 - Les US doivent être indépendantes les unes des autres
 - Négociable : une US doit être annulable (plus un *choix* qu'une *étape*)
 - Une US doit avoir de la valeur pour le client
 - ? (point « E » non abordé, cf internet ;))
 - Une US doit être la plus petite possible, ça devrait rentrer dans une itération (SCRUM), c'est à dire ~1-2 semaines
 - Testable : indiquer comment on valide chaque US -> souvent de l'**observation** (point de vue client) mais on peut mettre en place des **tests logiciels** pour vérifier que notre code marche toujours bien, car on peut casser des choses lors de refactorings.

Une US c'est :

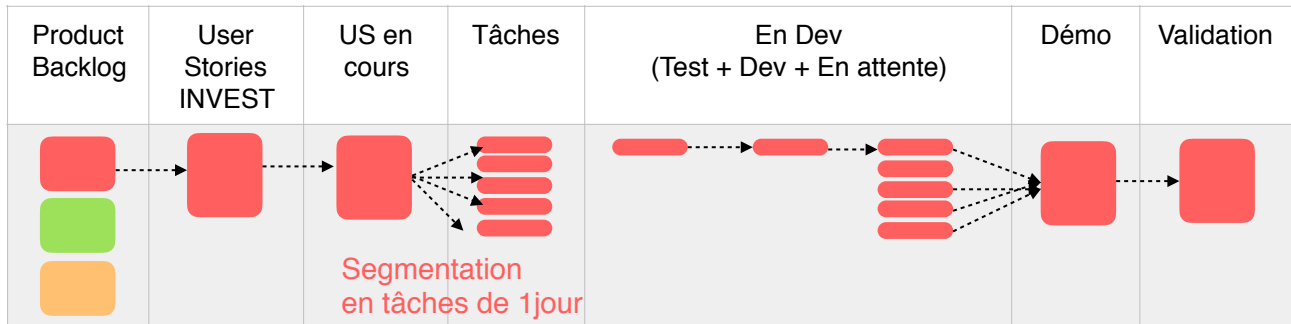
« En tant que, je veux pouvoir dans le but de »

ex : En tant que client je veux pouvoir donner un fichier SVG avec toutes les formes et être brécupérer un sans chevauchement ...

- Ne pas hésiter à aller sur Internet chercher des exemples, des explications pour faire de bonnes US (c'est simple mais avec le temps on devient de plus en plus précis)

Etape 2 : Choisir les colonnes de notre Tableau de Kanban

- On peut utiliser l'outil numérique de Bordeaux INP : kanboard.ipb.fr
- Voici la proposition de M. ALLALI :



- Règle : **On ne développe que ce qu'il y a dans le tableau !** Cela correspond au besoin client et non à l'envie du développeur (eh oui, c'est frustrant, mais c'est sensé nous rendre plus efficaces !)
- Difficulté : pondérer les tâches et accepter de faire les choses dans un ordre moins logique/intuitif, mais dépendre du besoin client (*exemple de l'autoradio dans une voiture, prioritaire sur le moteur pour le client, et c'est ainsi !*)
- Essayer de faire les US en **parallèle** (US vues comme des choix) et non en **linéaire**. Néanmoins c'est inévitable lorsque l'on doit développer un logiciel du début comme là.
- Une Démo, c'est quand on montre la validation des US.

Notre cahier des charges (pour lun-21-novembre)

- Ce ne sera PAS un cahier des charges classique, demande M. ALLALI, en dépit de ce qui a été dit avec le client. Il va falloir trouver le moyen de rassurer le client en lui fournissant l'architecture souhaitée tout en suivant notre méthode.
- Notre document comportera (en outre) :
 - LA DEMANDE : présentation du client, de son métier, de sa motivation à faire ce projet (pourquoi ?), explication du contexte et du besoin.
 - Le Product Backlog

Note finale

L'idée, c'est de **rentrer rapidement dans la phase de développement**, quite à **refactoriser souvent** !! On ne prévoit pas d'architecture très précise à l'avance, on fait sur le tas pour être au plus proche des besoins (changeants) du client et éviter de « sur-architecturer » notre code.