



Gestion de projet S8

Scrum

Dans Scrum, on va découper le projet en **sprints**. Un sprint est une période de 30 jours au bout desquels on doit avoir quelque chose de tangible entre les mains !

- Au début du sprint, on fait un **sprint planning** pour l'organiser
- A la fin du sprint, on fait un **sprint review** durant lequel on fait une rétrospective sur le sprint qui vient se terminer.

Sprint planning

Toute l'équipe se réunit, avec le *product owner*. Celui-ci doit présenter toutes les fonctionnalités qu'il voudrait voir implémentées. L'équipe définit le **sprint backlog**, qui contiendra toutes les fonctionnalités que l'équipe pense pouvoir faire en 2 semaines (ou 30 jours selon la durée déterminée du sprint).

Daily Scrum Meeting

Prendre chaque jour 15 minutes pour que chacun dise ce qu'il a fait le jour d'avant, ce qu'il compte faire le jour présent, et parler des problèmes qu'il rencontre. Cela permet de partager les connaissances, d'avoir un coup de main pour les soucis rencontrés.

L'idée, c'est d'éviter d'avoir à programmer des réunions plus tard. Et cela permet aussi de donner aux personnes extérieures un retour sur l'avancement d'équipe.

Sprint Retrospective Meeting

Réunion entre les membres de l'équipe seulement (pas spécialement besoin du *product owner*). On présente ce qui a été fait, et ce qui pourrait être amélioré. On essaye d'en tirer des points à améliorer dans le prochain sprint, histoire d'essayer d'améliorer notre manière de travailler.

Backlog refinement

On peut aussi faire une réunion au milieu du sprint, histoire de préparer en avance le sprint suivant. Permet de revoir les futures fonctionnalités, et de prévoir les changements si c'est nécessaire.

Définition of Done

Il faut aussi se mettre d'accord sur quand on peut décider qu'une fonctionnalité est bel et bien finie. Permet de définir les modalités par lesquelles il faut passer pour pouvoir dire que la fonctionnalité est bel et bien terminée. Cela permet aussi de faire une checklist des choses à faire avant de pouvoir terminer un module.

Burn Down Chart

Représenter toutes les fonctionnalités voulues dans l'application. Représenter une ligne théorique (grâce aux données du premier sprint) du temps nécessaire pour faire toutes les fonctionnalités demandées. Outil très simple dont le but est de ne pas passer trop de temps à estimer le temps nécessaire pour finir le projet. Par la suite, on représente à côté le temps réel pris pour la production de chacune des fonctionnalités. Cela permet de prévoir de plus en plus précisément le temps nécessaire pour tout faire. Et cela permet aussi de donner au *product owner* un pseudo-temps de finitions.

Le but n'est pas de complètement suivre la ligne initiale, mais plutôt de l'utiliser pour s'adapter, et pouvoir adapter les prévisions initiales.

Burn Up Chart

Même principe que la **Burn Down Chart**, mais à l'envers ; on part du temps nécessaire pour faire un sprint, pour essayer de prévoir le temps nécessaire pour faire les modules.

L'idée reste toujours de ne passer trop de temps là-dessus.

Remarques

- Scrum ne marchera pas si les personnes n'ont pas envie de vraiment avancer dans une même direction. Si jamais les gens ne sont pas motivé, ou que les

équipes ont des problèmes en interne, Scrum ne sera pas un remède miracle.
Mais il aura peut être le mérite de mettre ces problèmes au clair !

- Autre remarque : les plannings et Scrum ne sont pas non plus incompatibles, il est bien entendu normal de faire des plannings quand on développe selon Scrum. Il ne faut pas juste voir Scrum en opposition avec Waterfall.
- Il ne faut pas non plus s'imposer de faire quelque chose parce que Scrum l'a dit : Scrum donne des conseils, ou une marche à suivre générale ; mais il faut adapter ces conseils à la nécessité de l'équipe.

Pratiques associées à Scrum

Tableau des tâches

Permet de faire en sorte d'améliorer la transparence, car tout le monde peut voir où en sont les tâches devant être faites, et de pouvoir souligner les endroits/les tâches posant problème.

Planning poker

Essaye de régler le soucis de : Une personne donne un temps assez court, car il s'y connaît bien, alors que les autres prendraient plus de temps. De plus, comme le choix du temps se fait caché des autres, ça permet de ne pas s'aligner sur le premier qui parle, ou de se taire par peur de paraître nul.

En cas de grosses disparités dans les temps donnés, on peut forcer les discussions pour permettre aux gens de donner chacun la raison de son choix !

Extreme Programming

les valeurs :

- Simplicité
- Communication
- Feedback
- Respect
- Courage : la transparence nécessite d'avoir un minimum de courage : afficher où

on en est dans un projet, ça peut être stressant et ça demande de faire preuve de courage.