

Rôles

Product Owner (P.O) :

Personne représentant le client et l'utilisateur auprès du *ScrumMaster* et de l'équipe de développement. Il définit le produit et priorise les fonctionnalités voulues.

Scrum Master :

C'est le *facilitateur*, le garant du processus agile. Il n'est PAS un chef de projet mais un levreur d'obstacles qui empêcheraient l'équipe de développement d'avancer. Il protège et guide l'équipe des interférences extérieures pendant le *sprint*.

Team

(équipe de développement) :

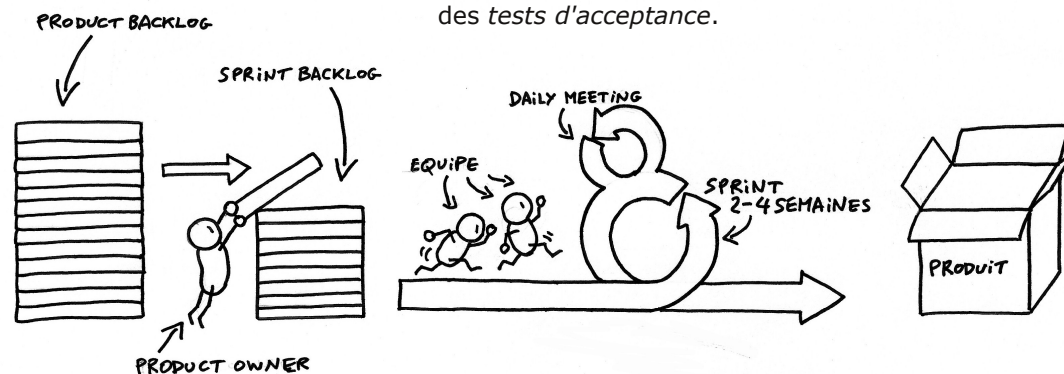
Auto-gérée et multi-disciplinaire (développeurs, testeurs, architectes, etc), les membres travaillent idéalement dans une seule et même pièce. Elle livre un produit *utilisable* à la fin de chaque *sprint*.

Concepts

Story points (points d'histoire) :

Outil d'estimation de l'effort nécessaire pour développer des fonctionnalités. Les points d'histoire permettent de se soustraire du concept de jour/homme. Les points sont attribués à une user story *relativement* à d'autres user stories. Par exemple, une user story estimée à deux points demandera deux fois plus d'effort pour la terminer qu'une user story estimée à un point, ceci sans indication de la durée en jour.

Schéma du cycle Scrum



Velocity :

L'effort, exprimé en nombre de points d'histoire, que l'équipe de développement peut fournir dans un *sprint*. La valorisation en points des user stories permet de déterminer le panier de fonctionnalités absorbable par l'équipe de développement en un *sprint*.

User Story (U.S) :

Description d'une fonctionnalité du point de vue utilisateur. Elle prend le formalisme "*En tant que... Je veux... afin de...*". Une user story peut être divisée en tâches si elle est complexe.

Ci-dessous un exemple de représentation d'une user story sur une carte.



Definition of Done (Fini) :

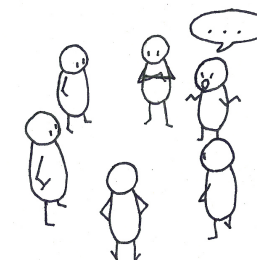
La "définition de fini" est la liste de critères qu'une *user story* doit remplir pour être considérée comme ayant l'état "fini", donc livrable. Cette liste de critères peut inclure, par exemple, une couverture de test minimum, une revue de code d'un autre membre de l'équipe, une javadoc suffisante, etc. Il est important d'avoir une DoD déterminée de façon claire et conjointe entre l'équipe de développement et le *Product Owner*. Ce dernier exprime son acceptation d'une user story via des *tests d'acceptance*.

Rituels

Sprint : Période de 2 à 4 semaines dédiée au développement des user stories *du backlog*, et permettant d'avoir un produit potentiellement livrable à la fin de celle-ci.

Daily stand-up meeting :

Réunion faite debout pour ne pas durer trop longtemps et à heure fixe (généralement le matin), lors de laquelle chaque participant répond aux trois questions : "*Qu'ai-je fait hier ?*", "*Que vais-je faire aujourd'hui ?*" et "*Ai-je un point de blocage ?*"



Sprint review (démonstration de fin de sprint) :

Réunion tenue en fin de sprint durant laquelle l'équipe de développement montre le travail accompli pendant le sprint (i.e. les fonctionnalités, les user stories demandées par le *Product owner*).

Planning poker :

Séance d'estimation menée par l'équipe de développement qui évaluent ensemble l'effort nécessaire pour traiter les user stories du *backlog*.

Pour cela, ils utilisent chacun un jeu de carte sur lesquelles sont inscrit des nombres de points d'histoires dont les valeurs suivent généralement la suite de Fibonacci :

0, 1, 2, 3, 5, 8, 13...

Les estimations sont faites face cachée et dévoilées en même temps pour éviter d'influencer les autres membres de l'équipe.

Retrospective :

Réunion permettant à l'équipe de faire un bilan du sprint qui vient de se terminer.

On y note ce qui fait avancer le projet et ce qui le ralentit. Dans ce dernier cas, l'équipe cherche des actions pour lever les obstacles.

Elle est généralement menée par le *ScrumMaster* et s'organise en 5 étapes :

- set the stage, prendre la température des participants via un vote (de confiance et/ou d'itération);
- gather data, liste le ressenti de l'équipe, les problèmes, les points positifs, les émotions qui l'ont marquée pendant le sprint qui vient de se terminer;
- generate insights, permet une réflexion de groupe sur la perception et les causes des obstacles évoqués précédemment;
- decide what to do, est l'étape qui permet de générer des actions à appliquer lors du sprint suivant pour tenter de lever les obstacles évoqués;
- close the retrospective, marque la fin de cette réunion. On y fait en général un vote nommé ROTI (Return On Time Invested) pour indiquer le degré de satisfaction sur le temps consacré à la rétrospective.

Artefacts

Product backlog :

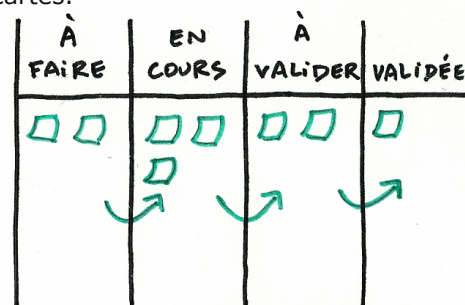
Ensemble des caractéristiques (fonctionnalités ou besoins techniques) qui constituent le produit souhaité. Il doit être priorisé pour permettre de développer les éléments de plus haute importance en premier.

Sprint backlog :

Sous-ensemble des éléments du backlog de produit. Les éléments constituent les user stories à développer au cours du sprint et sont préalablement détaillés pour pouvoir être estimés par l'équipe de développement. Il est également priorisé.

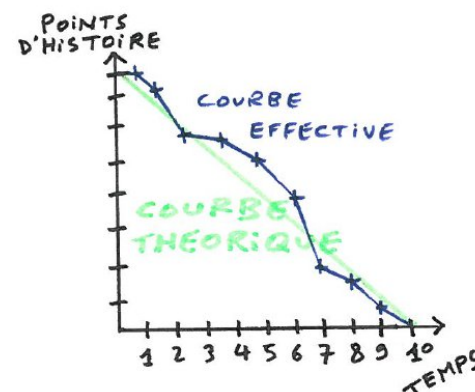
Task board (tableau des tâches) :

Tableau physique ou logiciel reprenant les éléments du *backlog de sprint*. Il possède plusieurs colonnes (ex. à faire, en cours, à valider, validée) permettant de suivre l'avancement des user stories affichées via des post-it ou des cartes.



Burn down chart :

Graphique permettant de suivre le "reste à faire" durant le sprint. Il possède en abscisse le temps et en ordonnée les points d'histoire. La courbe indique le nombre de points d'histoire abattus pendant le sprint. Elles sont mises à jour en continu. Cela permet d'anticiper les dérives et les ruptures de charge. L'idéal étant bien sûr d'arriver à zéro point le dernier jour du sprint.



Hing CHAN

<http://hingchanscrum.blogspot.com>

@HingCChan



Thierry LERICHE

<http://icauda.com>

@thierryleriche



SCRUM

MEMENTO A DESTINATION DE L'EQUIPE

