**Méthode Agile**

**Une User Story**, ou récit utilisateur, en français, correspond à la description d'un besoin ou d'une attente d’un utilisateur/client.

Exemple :

Vous rejoignez donc Anastasia, Jonathan, Samy et Garance : l’ensemble de l'équipe doit être présent. Garance introduit la réunion en présentant la **User Story** n° 1 517 du Product Backlog.

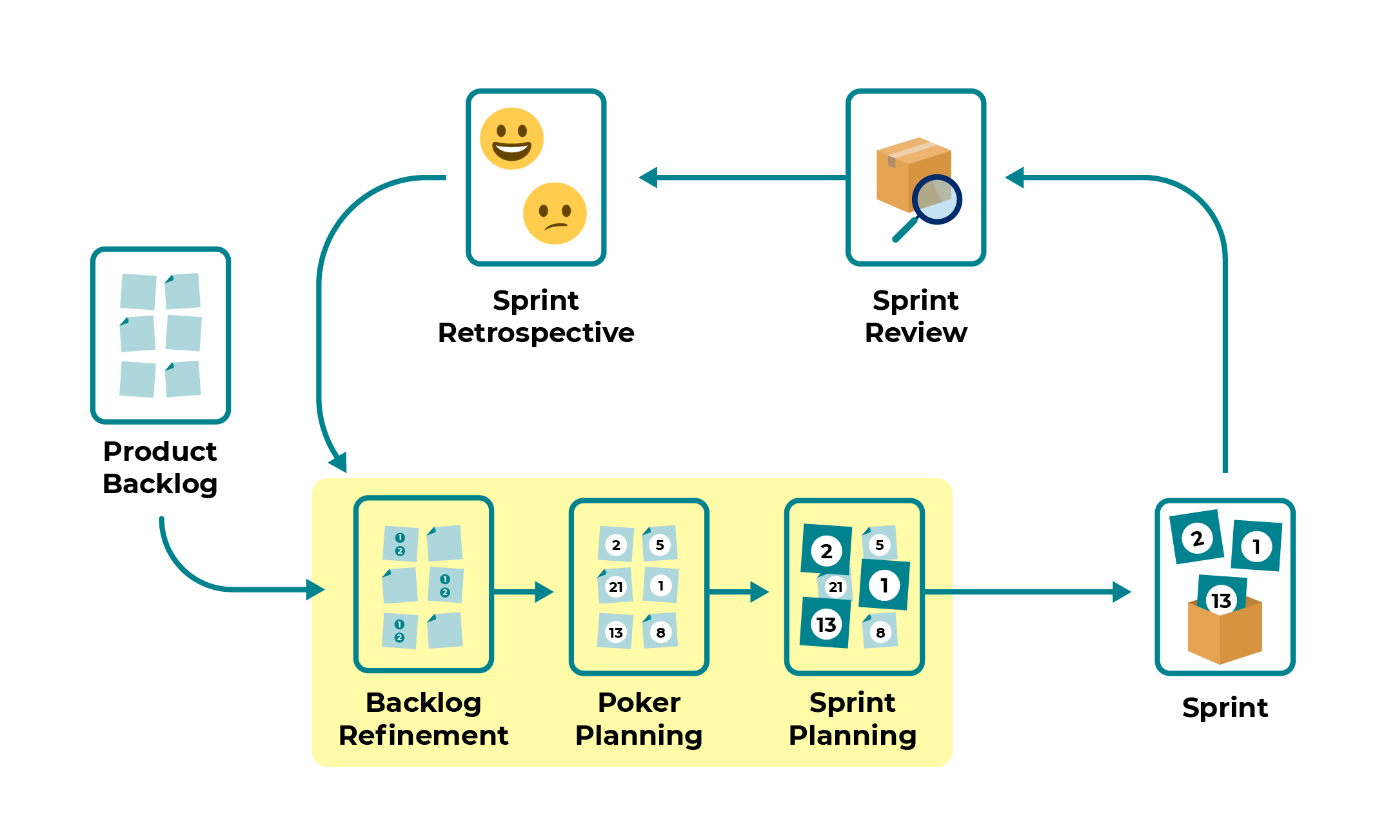
Lors de la réunion de rétrospective , on va discuter des détails et instruction de notre backlog ( product Backlog ) .

Cette réunion est composé de 3 principales étapes : Backlog Refinement , Poker Planning et Sprint Planning .

* **Backlog Refinement**: permet de discuter et de voir les détailles sur les exigences / user storie contenu dans notre product backlog , on va les découper , regrouper en étape technique.
* **Poker Planning :** permet de définir le délai de réalisation d’une tâche en tenant compte de sa complexité par la tenue de carte numéroté ( chaque concepteur va coter la tâche ) et d’un vote pour donner son opinion.

**Nb** coter : c est estimer le temps nécessaire à la réalisation de la tache du fait de sa complexité.

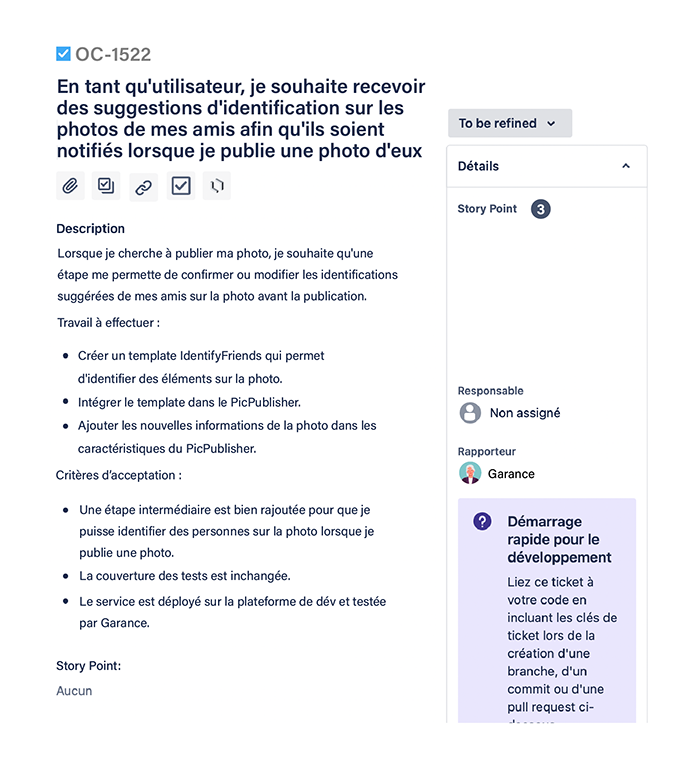
* **Poker Planning :** Product Owner présente tous les ticket seront partis du Sprint et ceux qui vont être traité en priorité pour le déroulement de notre Sprint.



* identifier et découper les tâches à traiter pendant le Sprint ;
* quantifier l’effort alloué pour le traitement de chaque ticket ;
* démarrer votre Sprint.

Pour cerner la portée du ticket, l’équipe a également besoin de définir clairement le travail à réaliser, et les **critères d’acceptation** associés. Ces derniers correspondent aux conditions que doit remplir le logiciel pour être accepté par l’utilisateur final ou le client.

**Exemple de Ticket**



La préparation d’un Sprint repose sur différentes étapes réalisées en équipe :

découper techniquement les tâches du Product Backlog lors du Backlog Refinement ;

estimer la complexité technique ou le temps nécessaire pour traiter chaque ticket grâce au Poker Planning ;

prioriser et s’engager sur le traitement d’un nombre donné de tickets pendant le Sprint Planning ;

communiquer ces objectifs aux parties prenantes.

Anastasia est d’accord avec vous, cette partie du code porte à confusion. Vous décidez de la réécrire ensemble. C’est ce qu’on appelle le refactoring (ou refactorisation, en français).

Le refactoring permet d'améliorer en continu le code source d’un logiciel. Ainsi, quand un morceau de code est gourmand en ressources, peu lisible voire incompréhensible, les membres de l’équipe s’assignent une tâche de refactoring pour produire un code optimisé.

Ces techniques de développement que sont le TDD, le KISS ou le refactoring sont parmi les plus répandues dans le développement agile. Vous en découvrirez d’autres au fur et à mesure de vos expériences ! le KISS. KISS signifie “keep it simple and stupid” (“Garde ça simple et stupide”, en français). C’est un principe en développement qui préconise la simplicité dans la conception.

Le Daily est le cœur d’une bonne communication dans l'équipe. Elle se déroule au même endroit, à la même heure, tous les jours, et dure 15 minutes. Cette réunion est obligatoire pour les développeurs et facultative pour les autres membres de l'équipe.

L’objectif du Daily est de recueillir des informations sur les avancées et la qualité de la collaboration au sein de l’équipe. Chaque membre prend rapidement la parole à tour de rôle. Avec votre équipe, vous utilisez un **tableau Kanban**.

**tableau Kanban** : tableau qui est composé de 5 colonnes .À faire ; En cours ;Bloqué ; À valider ; Terminé.

Le développement agile repose sur la programmation par pairs, qui consiste à développer en binôme.

Plusieurs méthodes de développement agile vous permettront d’obtenir un code lisible et maintenable :

le TDD consiste à développer un programme en commençant par les tests ;

le KISS consiste à développer un code simple et lisible ;

le refactoring consiste à améliorer le code source d’un logiciel en continu.

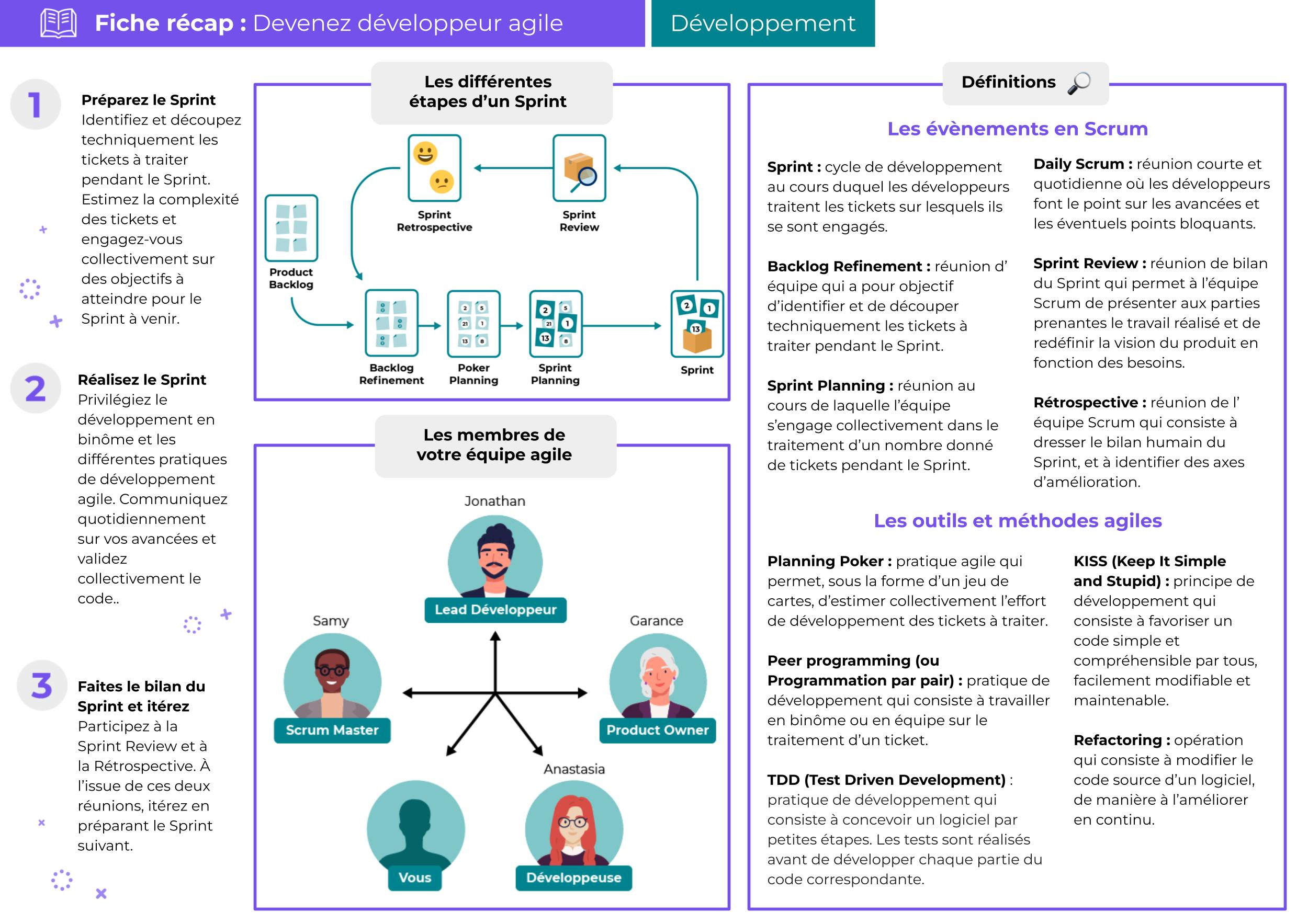
Le développement agile repose également sur une communication quotidienne grâce au Daily, une réunion d’équipe où chacun évoque ses avancées et les éventuels points bloquants.

Tout développement implémenté doit faire l’objet d’une validation collective :

une validation technique par l'équipe de développement lors de Code Reviews ;

une validation fonctionnelle par le Product Owner.

On est ici au cœur du développement agile : identifier les problématiques, intégrer les retours et itérer. Ces concepts s'appliquent au processus de développement en lui-même. Toutefois, la Rétrospective permet aux équipes d'appliquer ces pratiques à une échelle plus humaine.



**Burndown chart :** graphe de l’évolution des tâches en fonction des jours .

