



ATELIER : qu'avez-vous retenu de l'agilité ?



THE AGILE MANIFESTO

Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.

Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.

Livrez fréquemment un logiciel fonctionnel, dans des cycles de quelques semaines à quelques mois, avec une préférence pour les plus courts.

Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.

Réalisez les projets avec des personnes motivées.

Fournissez-leur l'environnement et le soutien dont elles ont besoin et faites-leur confiance pour atteindre les objectifs fixés.

La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est **le dialogue en face à face**.

Un **logiciel fonctionnel** est la principale mesure de progression d'un projet.

Les processus agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment **un rythme constant**.

Une attention continue à l'excellence technique et à un bon design.

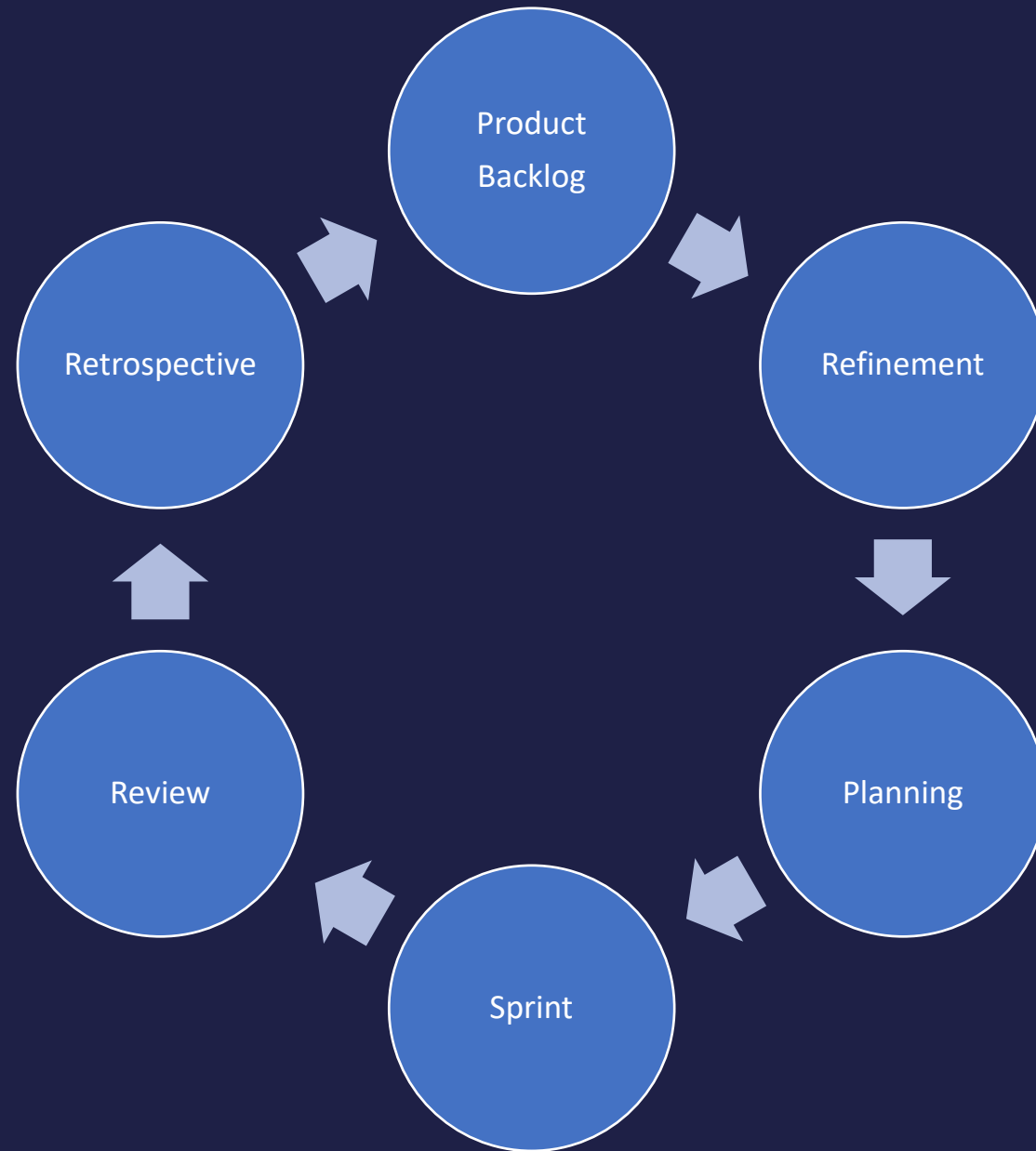
La simplicité – c'est-à-dire l'art de **minimiser la quantité de travail inutile** – est essentielle.

Les meilleures architectures, spécifications et conceptions émergent **d'équipes auto-organisées**.

À intervalles réguliers, **l'équipe réfléchit aux moyens possibles de devenir plus efficace**. Puis elle s'adapte et modifie son fonctionnement en conséquence.

SCRUM

Scrum est un framework ou cadre de développement de produits complexes. Il est défini comme un « cadre de travail holistique itératif »



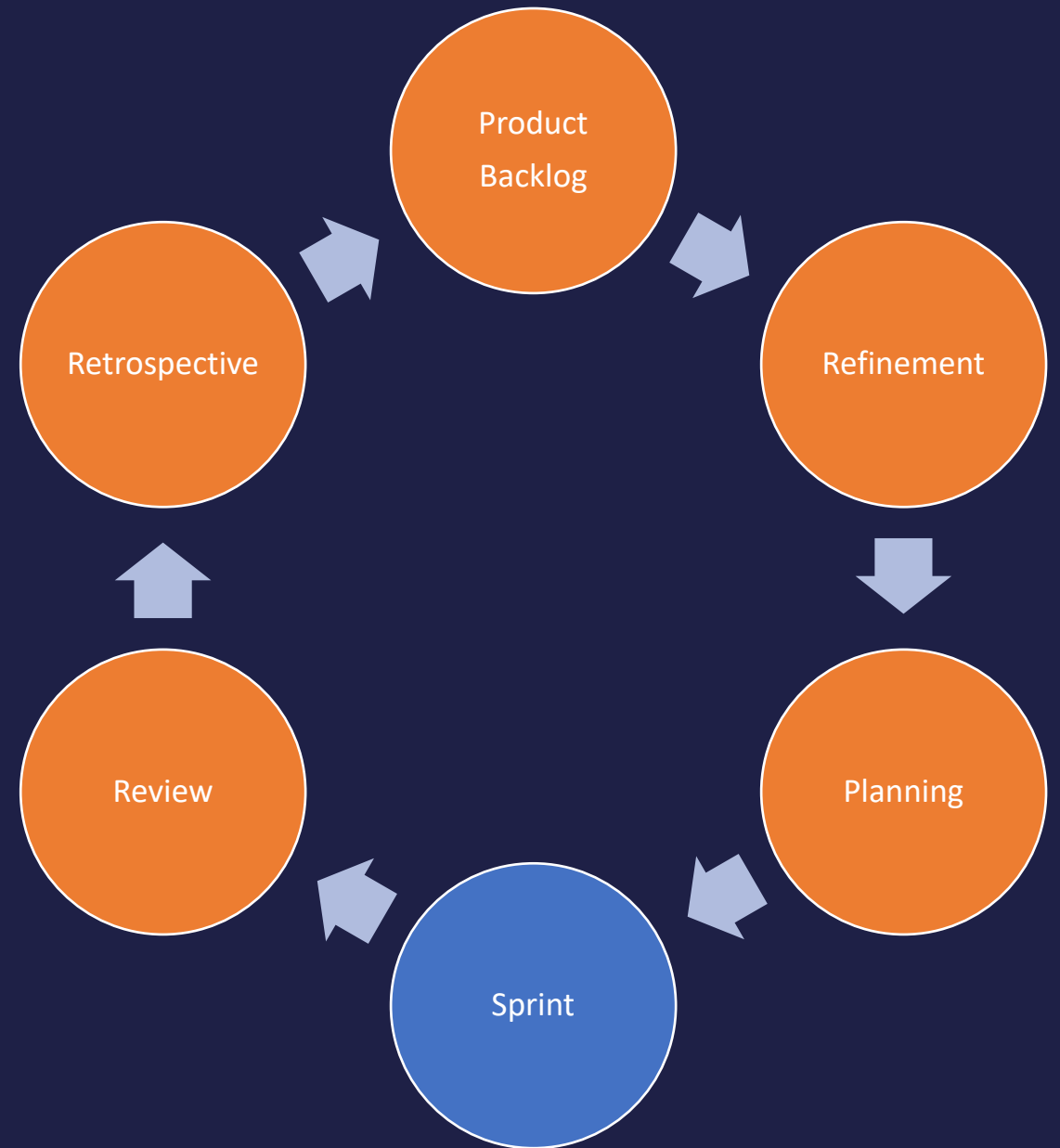
En plus de l'avancé dans la consommation (burn down) du product backlog, tous les jours un standup a lieu pour tenir ses collaborateurs à jour

Dans une équipe Scrum, les rôles associés sont les suivants :

1. Product owner
2. Scrum master
3. Developpeur(euse)s
4. Stakeholders

Le product owner est la source des exigences et de l'expression des besoins.

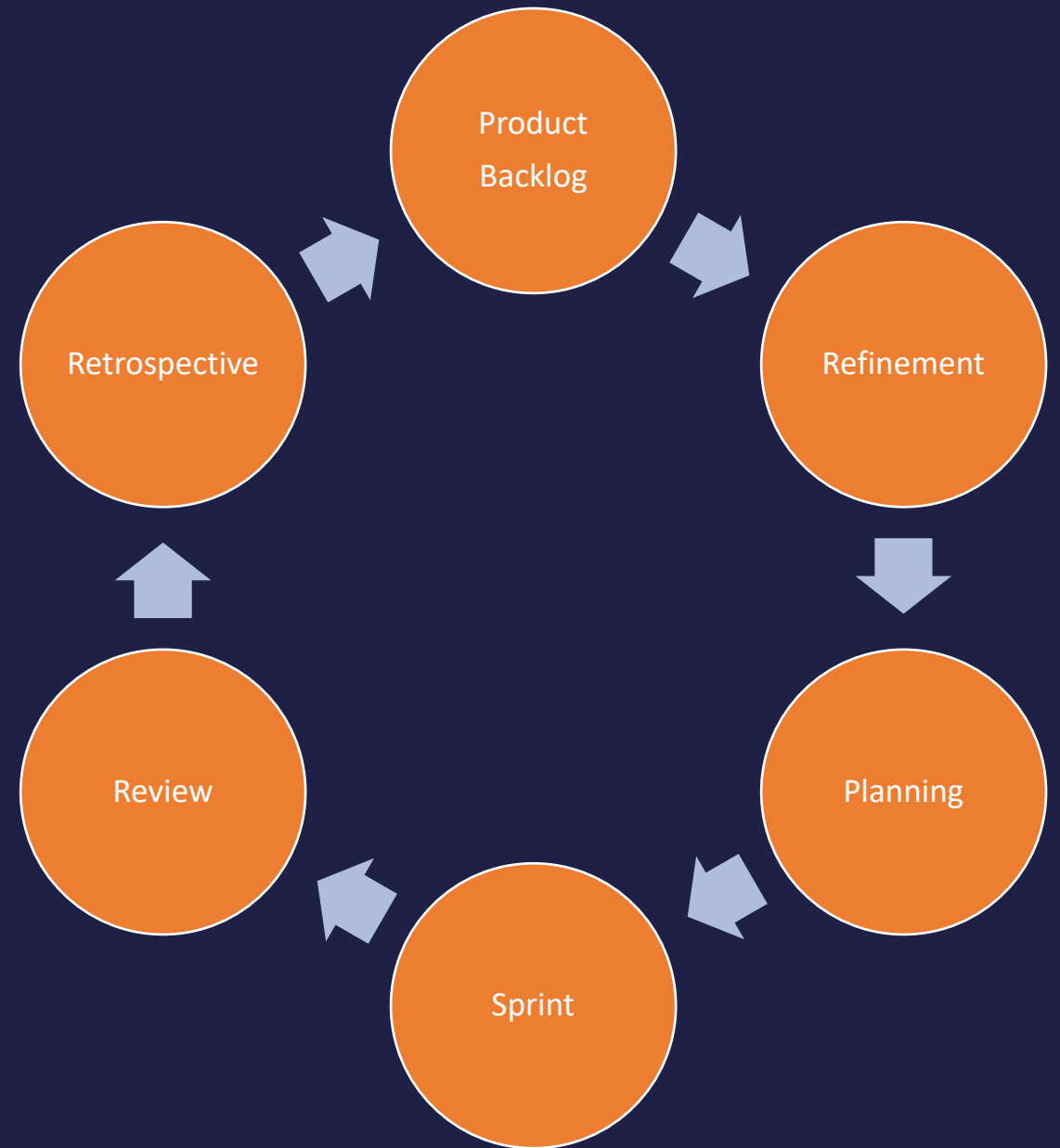
Il remplit le Product backlog avec le scrum master.



Un bon product owner est une personne qui a les idées claires et qui sait ce qu'elle veut et ce qu'elle veut en priorité

Le scrum master est un facilitateur, il s'assure que :

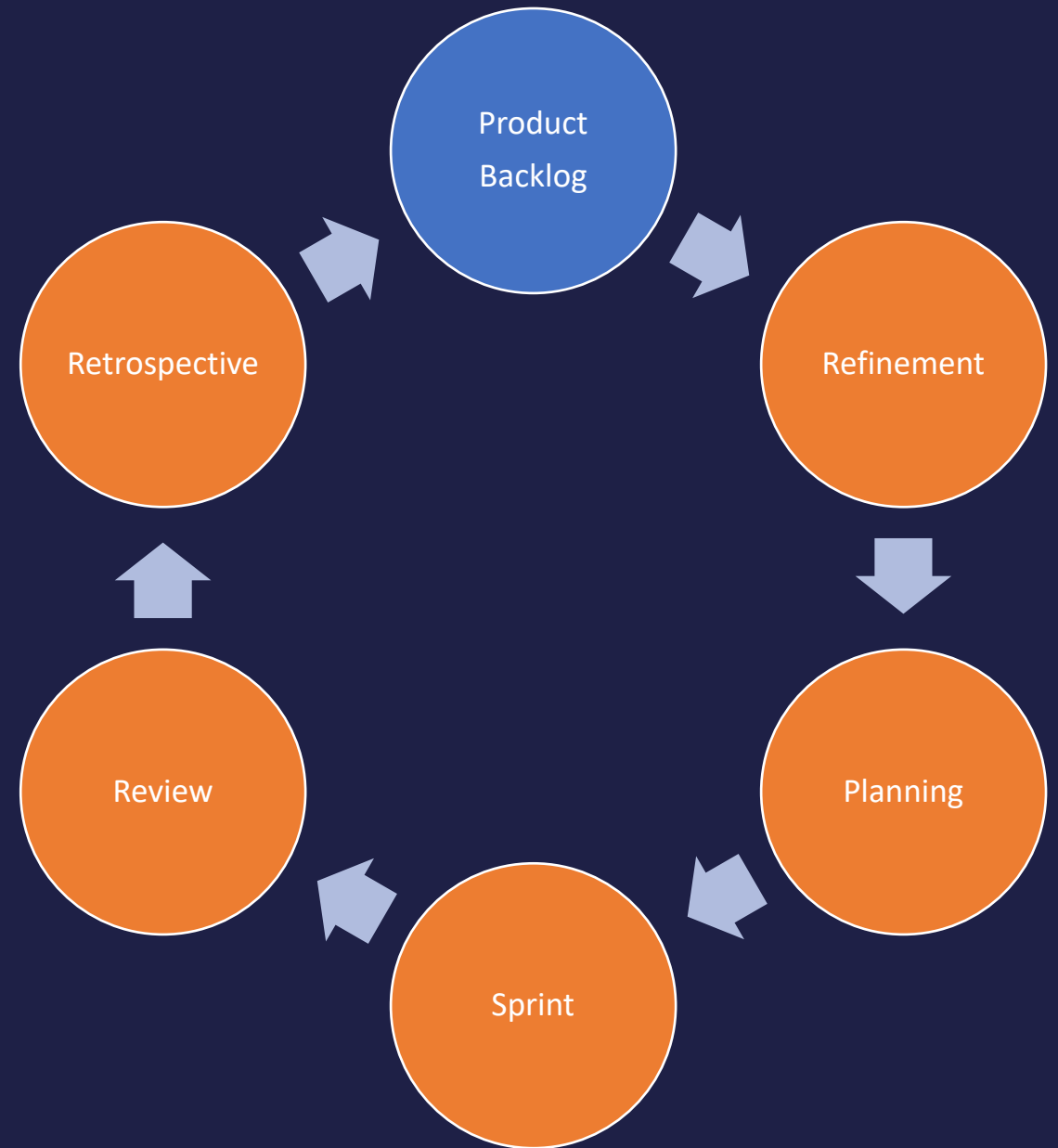
1. le framework soit bien suivi
2. la planification soit précise
3. les développeurs sont protégés des distractions extérieures



Un bon scrum master est une personne qui connaît les capacités de production des artéfacts et qui sait protéger les membres des distractions (qui prend les distractions sur lui)

Les développeur(euse)s :

1. fournissent leur expertise pour construire les incréments
2. sont auto-organisés



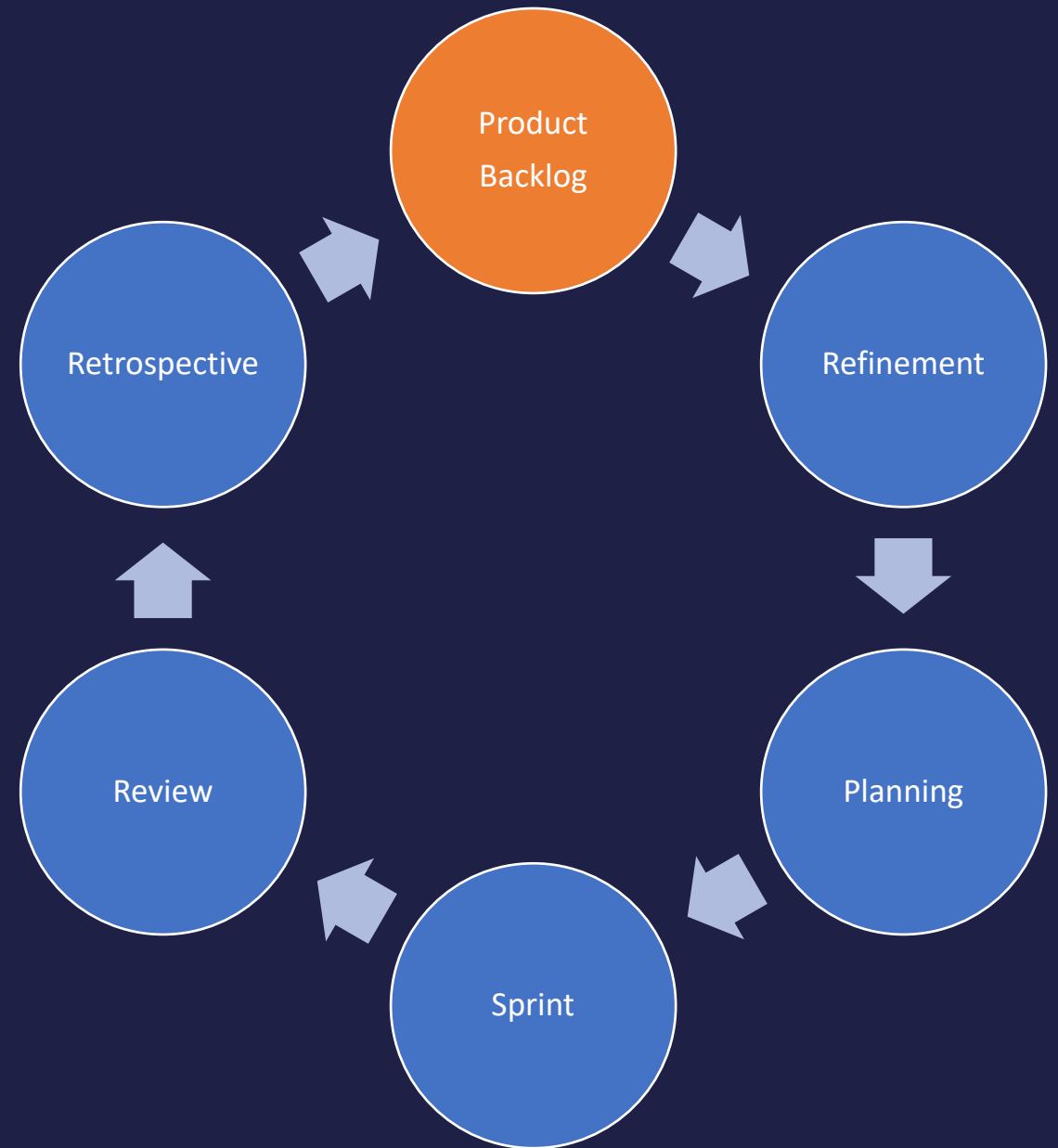
Un(e) bon(ne) développeur(euse) sait estimer le degré de complexité des features de manière cohérente et répétable

DETAILS DU CYCLE

En tant que Product Owner, définissez vos personnas, les plus-values à capturer pour chaque persona et comment il ou elle y parvient

En tant que Scrum Master, assurez-vous que le Product Owner garde une structure persona/valeur/comment

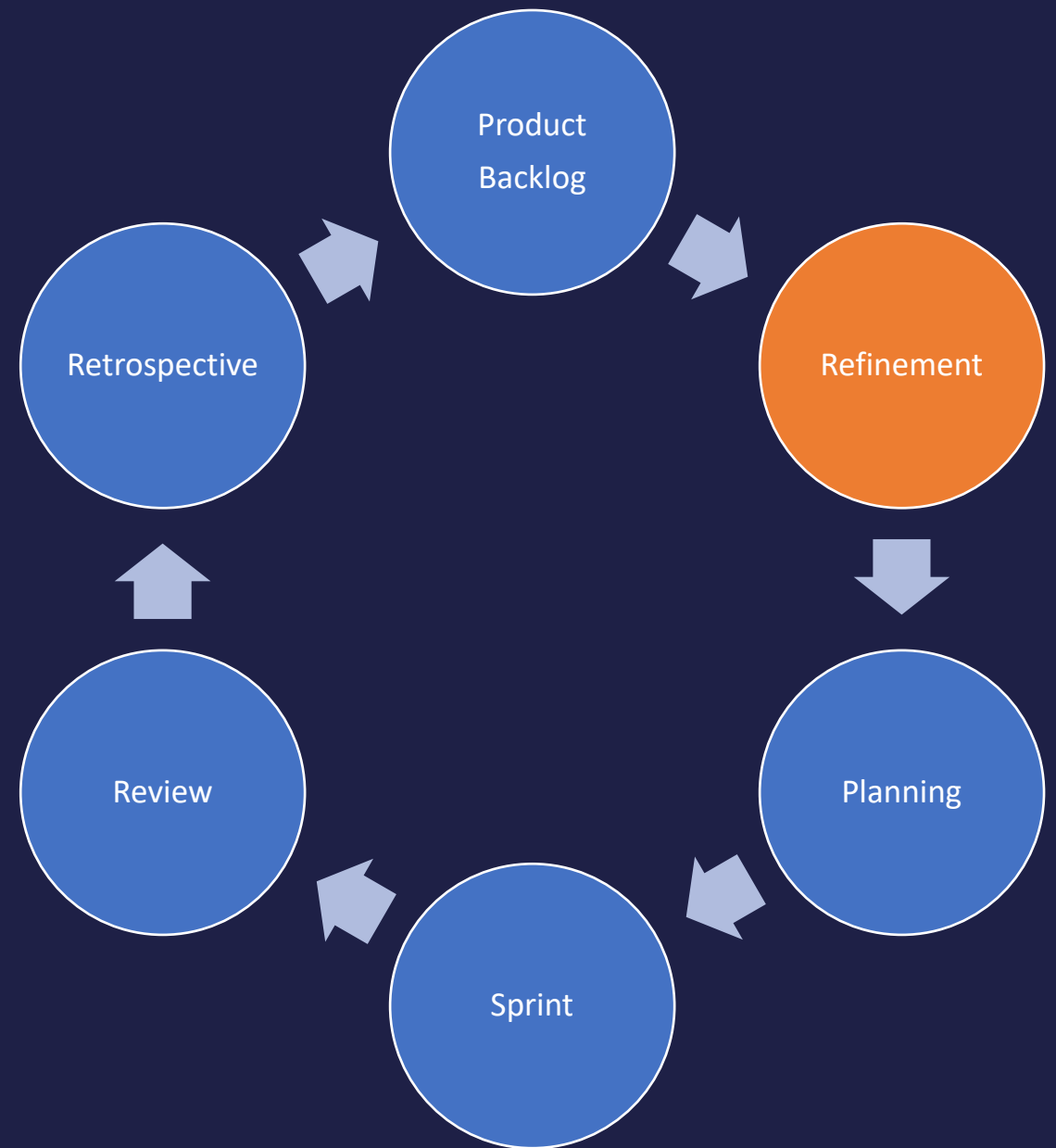
Temps : on-going



En tant que Scrum Master et développeur(euse)s :

1. Votez avec des story points (1 à 5)
2. Découpez les initiatives, les epics -> produisez des user stories

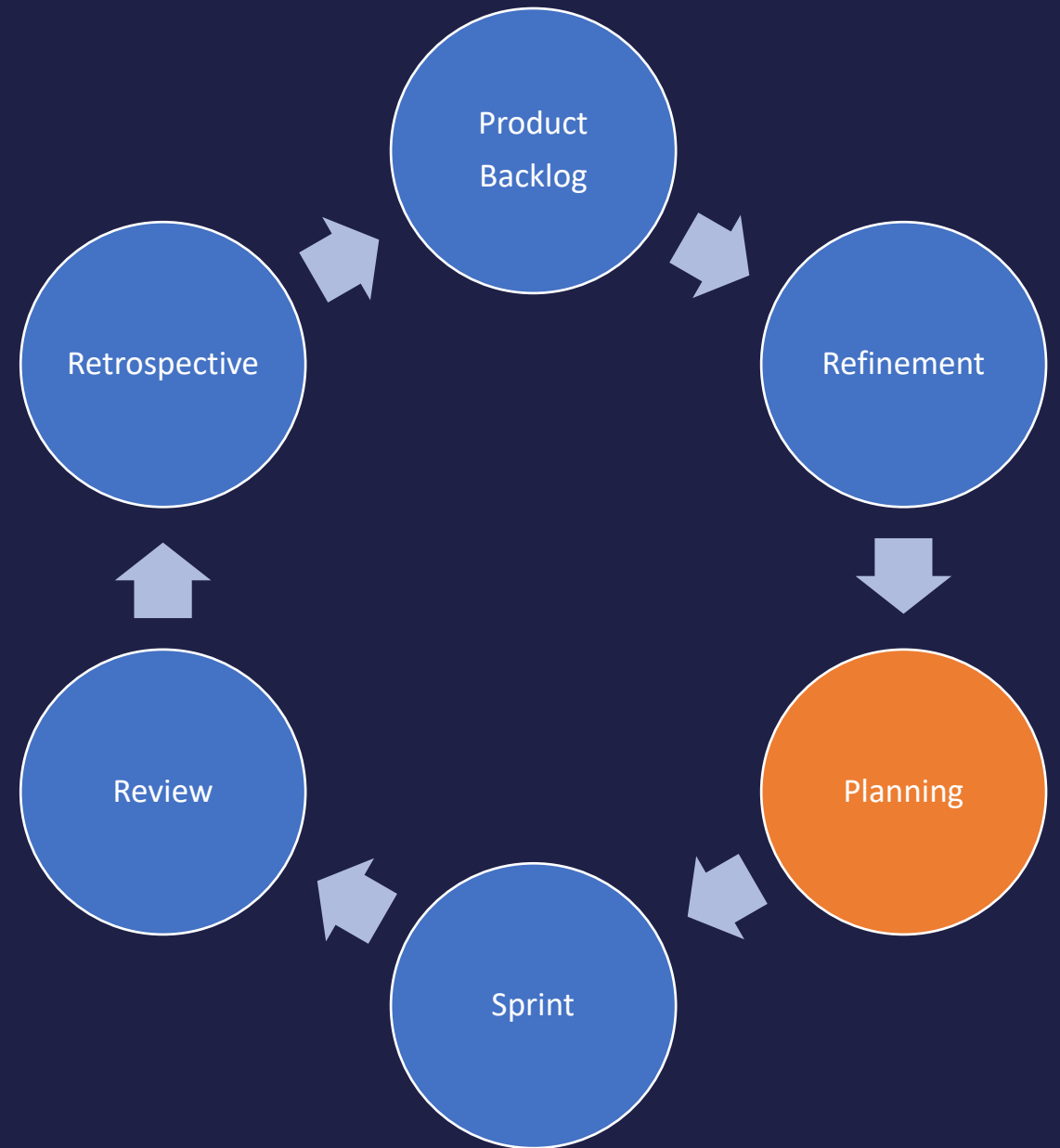
Temps : aucune contrainte



En tant que Scrum Master et développeur(euse)s :

1. Vous pouvez définir les user stories en subtasks si vous avez besoin
2. Plannifiez le sprint

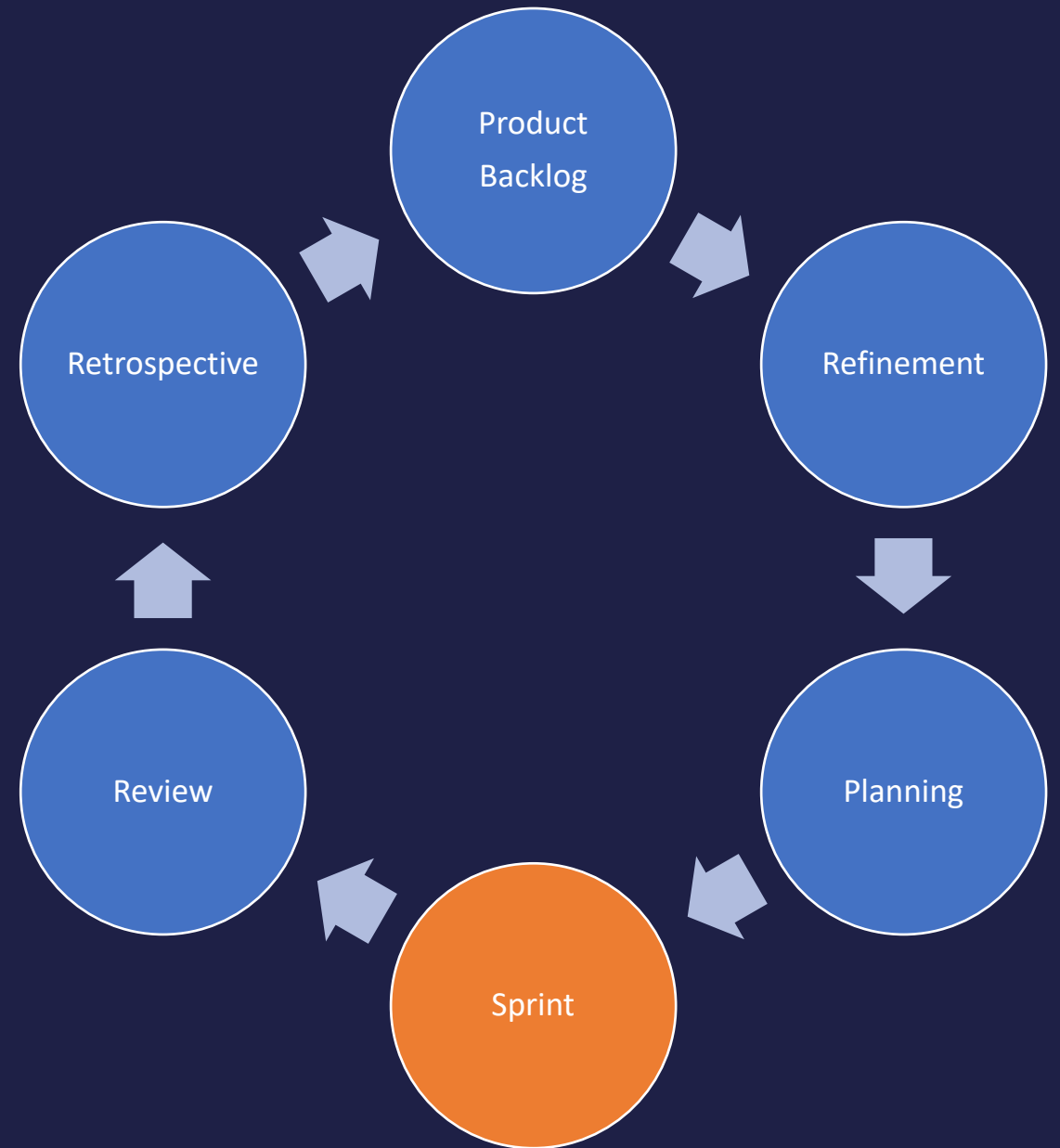
Temps : 10% du sprint



En tant que Scrum Master :
Protégez les
développeur(euse)s de toutes
distractions externes

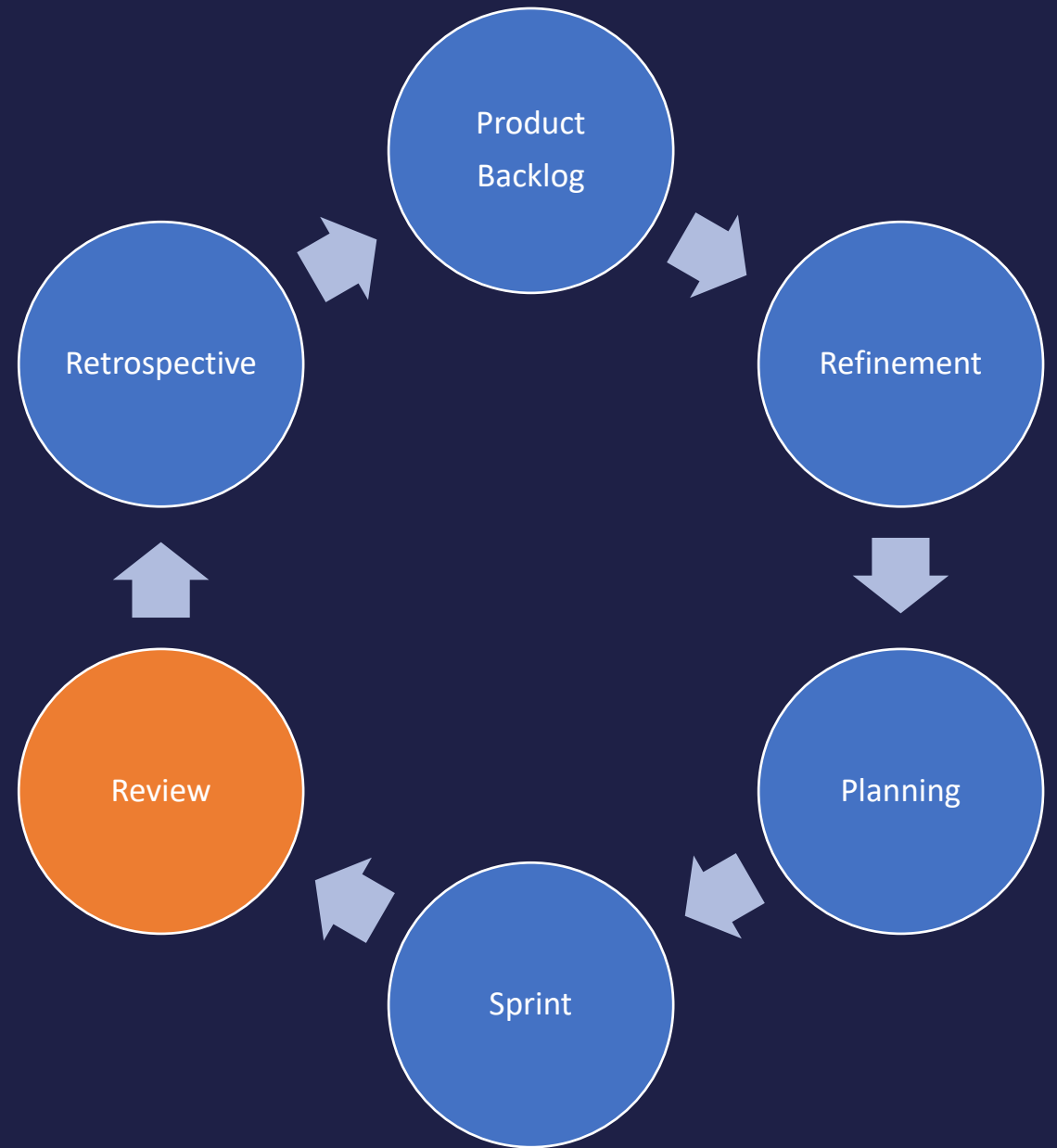
En tant que
développeur(euse)s :
Faites des standups quotidien
pour informer vos collègues

Temps : environ 1 mois



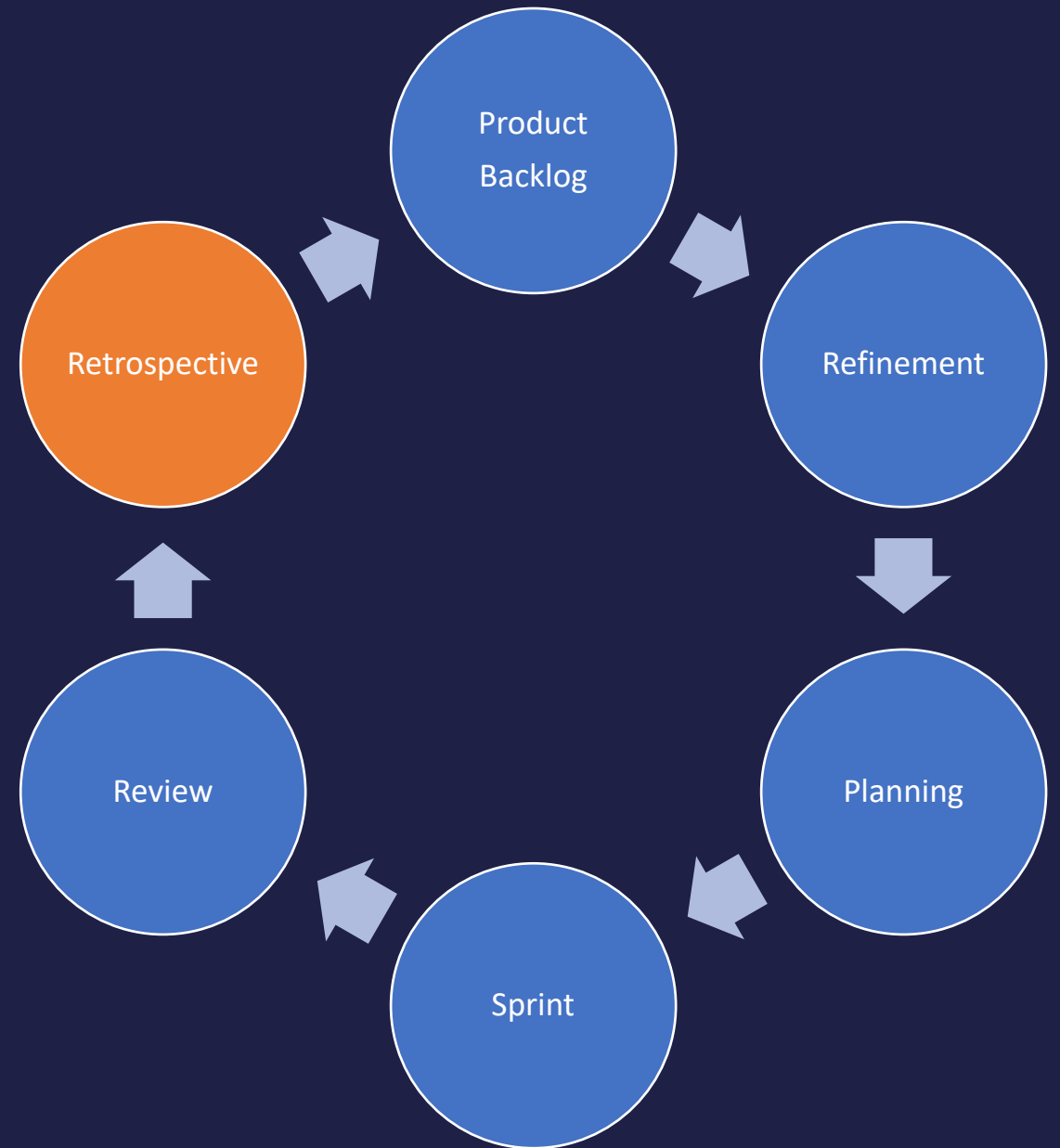
En tant que Scrum Master et développeur(euse)s :

1. Vous faites la demo de votre incrément fonctionnel
2. Vous récoltez les retours à intégrer au backlog
3. Déplacez les user stories à completed



En tant qu'équipe, vous vous améliorer :

1. Que s'est-il bien passé ?
2. Que s'est-il mal passé ?
3. Que faire différemment pour le prochain sprint ?



ATELIER : En groupe, écrivez autant de US que de participants :

Pour la création d'une application pour réserver ses plats à la cantine



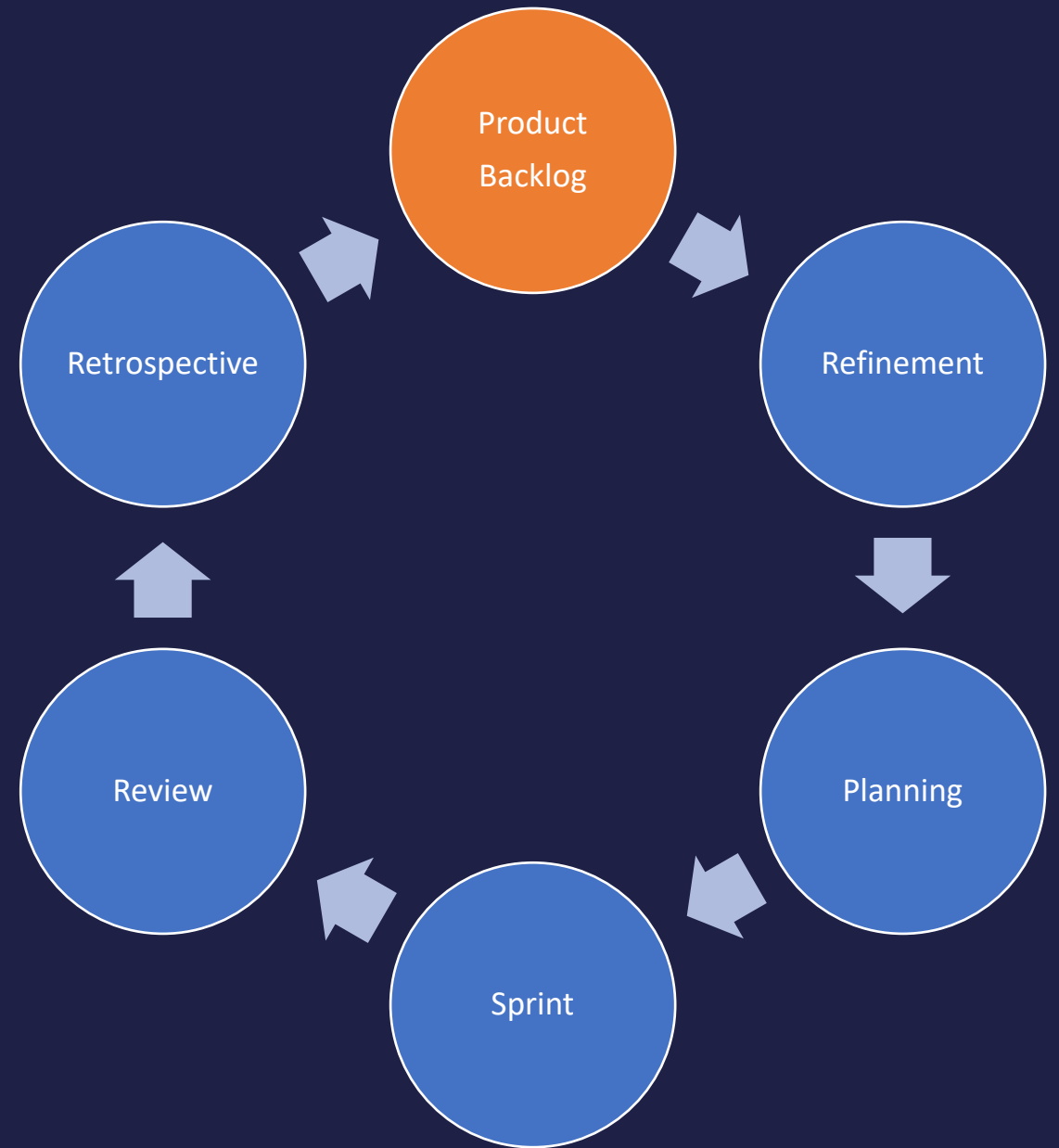
ATELIER : En groupe, construisez votre équipe :

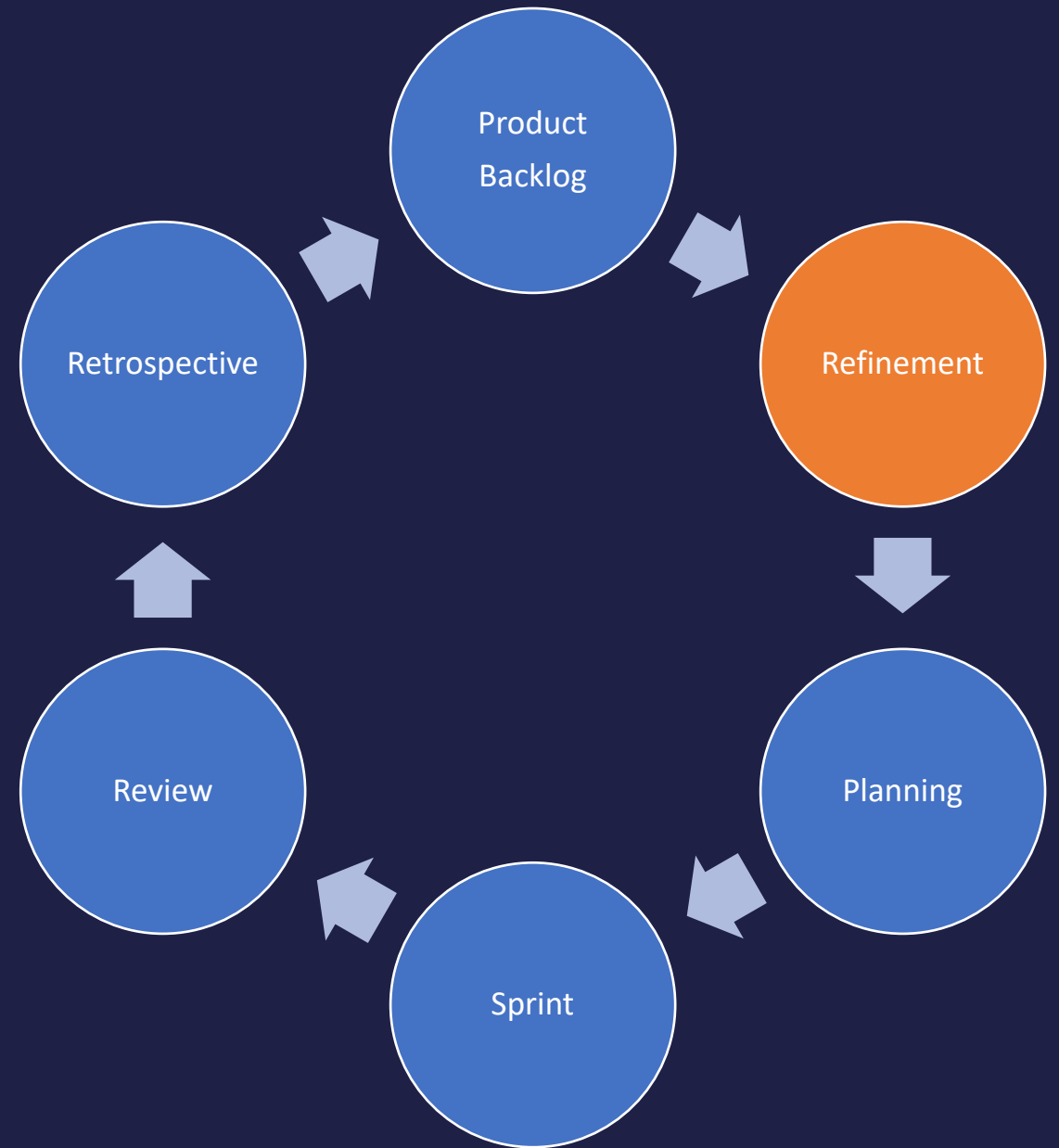
- 1) Choisir 1 PO
- 2) Choisir 1 Scrum master
- 3) Choisir 3 utilisateurs finaux
- 4) Choisir 1 graphiste
- 5) Choisir 1 responsable de la communication
- 6) Le reste seront des développeurs et des développeuses

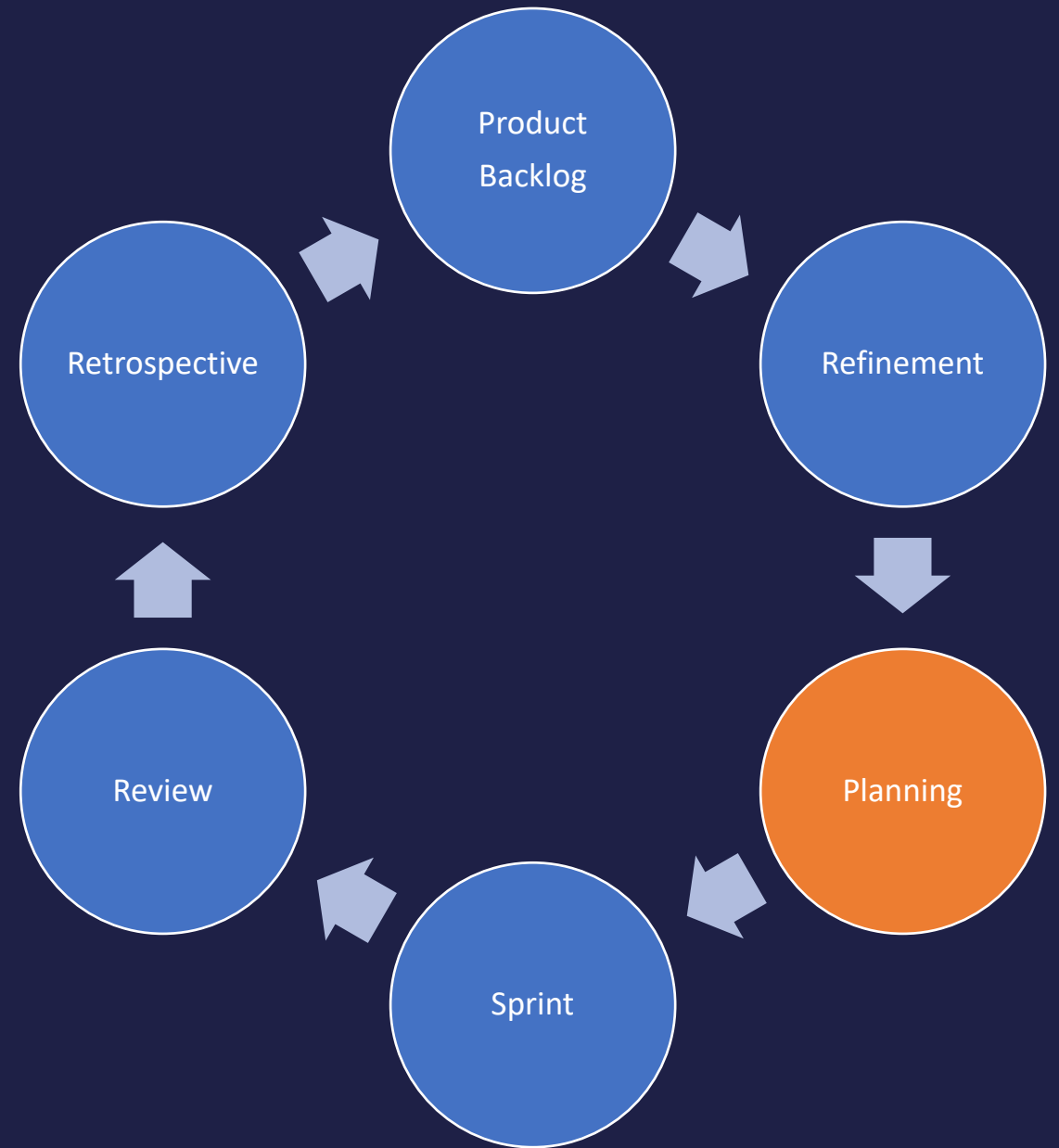
ATELIER : En groupe, simulez des itérations Scrum :

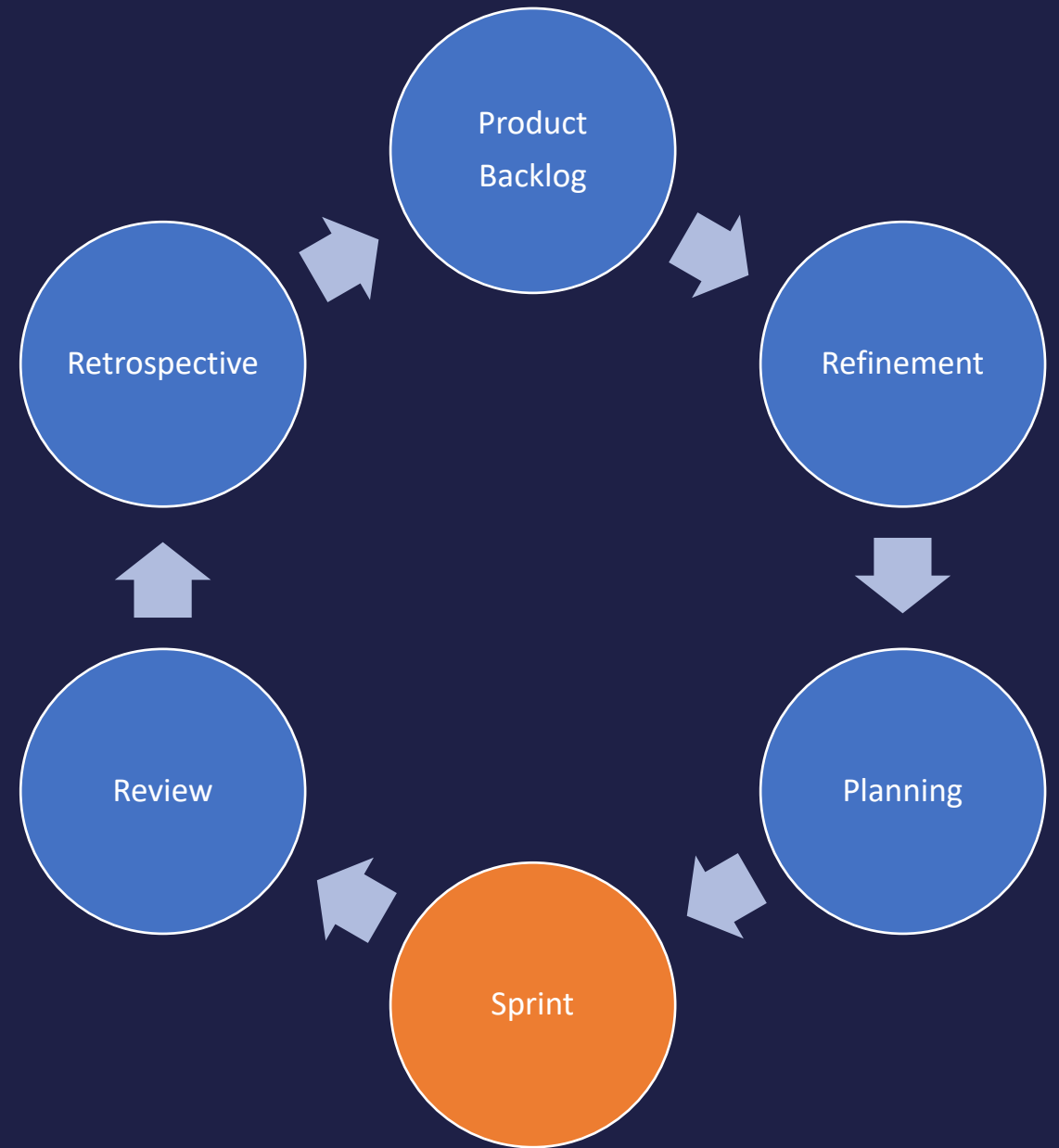
La vélocité de chaque sprint sera fixé par votre professeur

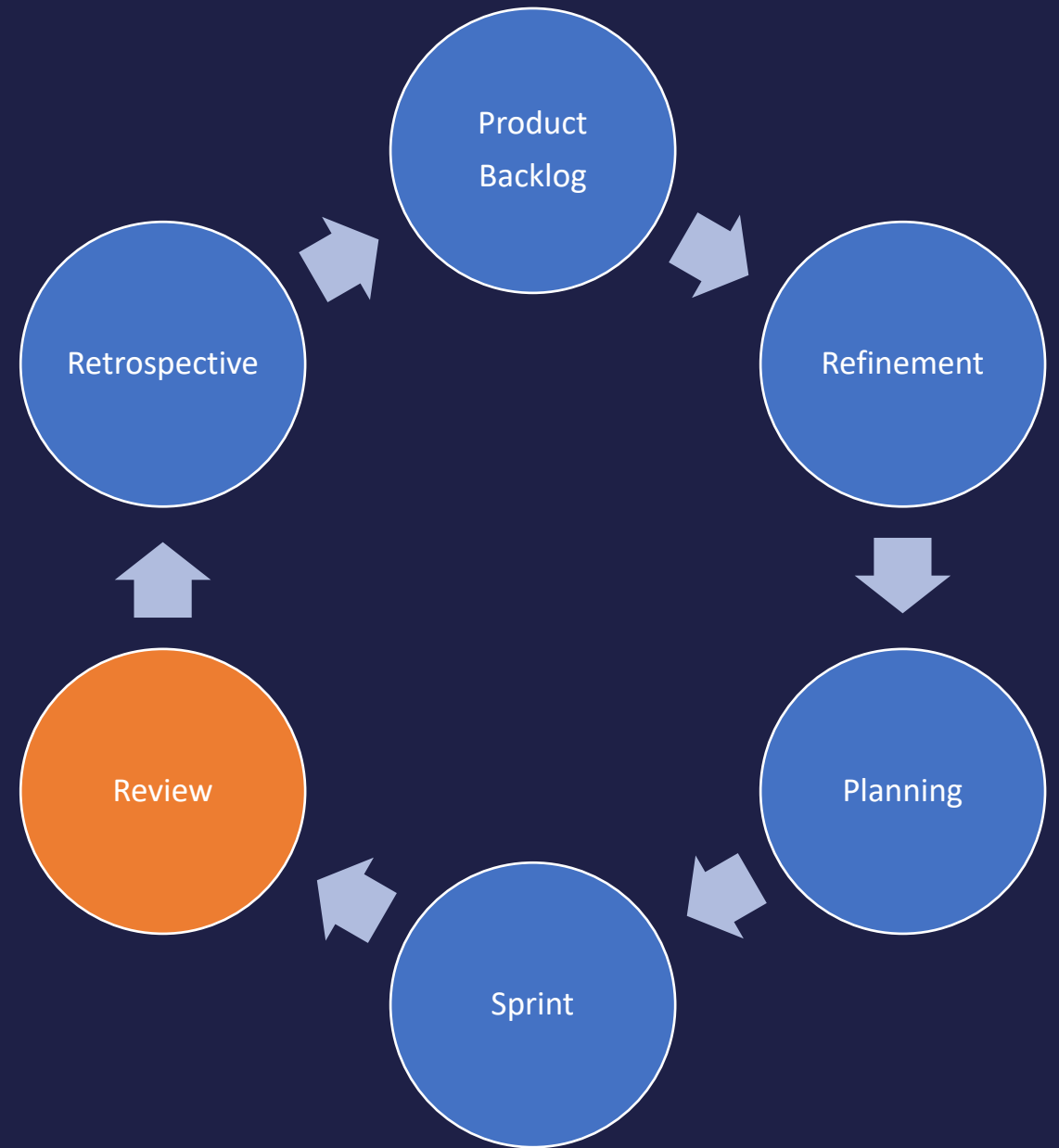


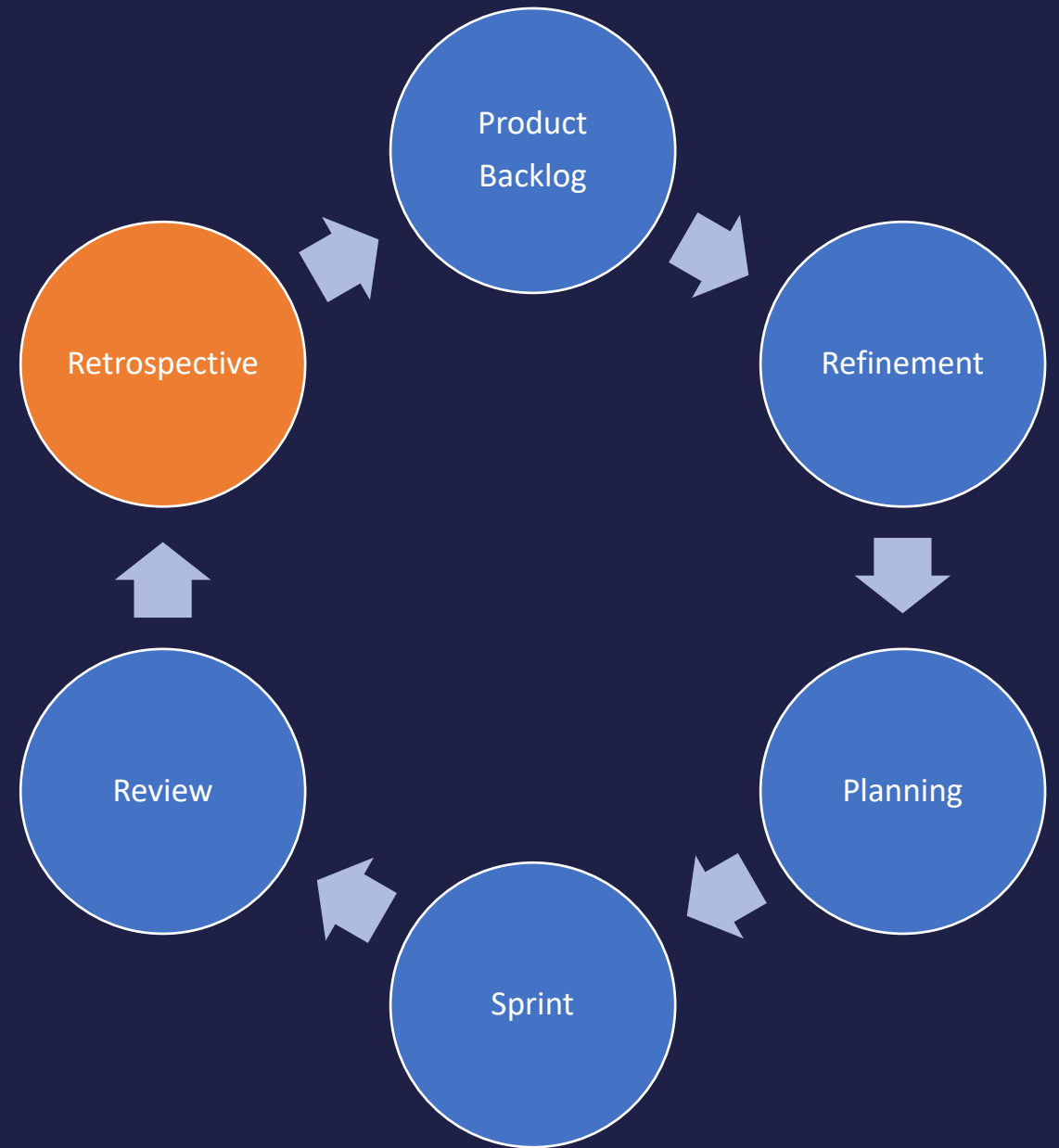












KPI

Lead time : La mesure du temps entre l'émission d'un besoin et sa livraison dans les mains de l'utilisateur (« From Concept To Cash »). La réduction du Lead Time est un autre enjeu fort des programmes de transformation agile.

$\text{lead time} = \text{date de livraison} - \text{date d'émission}$

Prédictibilité (dans le sprint) : Mesure qui permet d'appréhender la précision de l'estimation dans un sprint (Nombre de points réalisés sur le nombre de points prévus initialement)

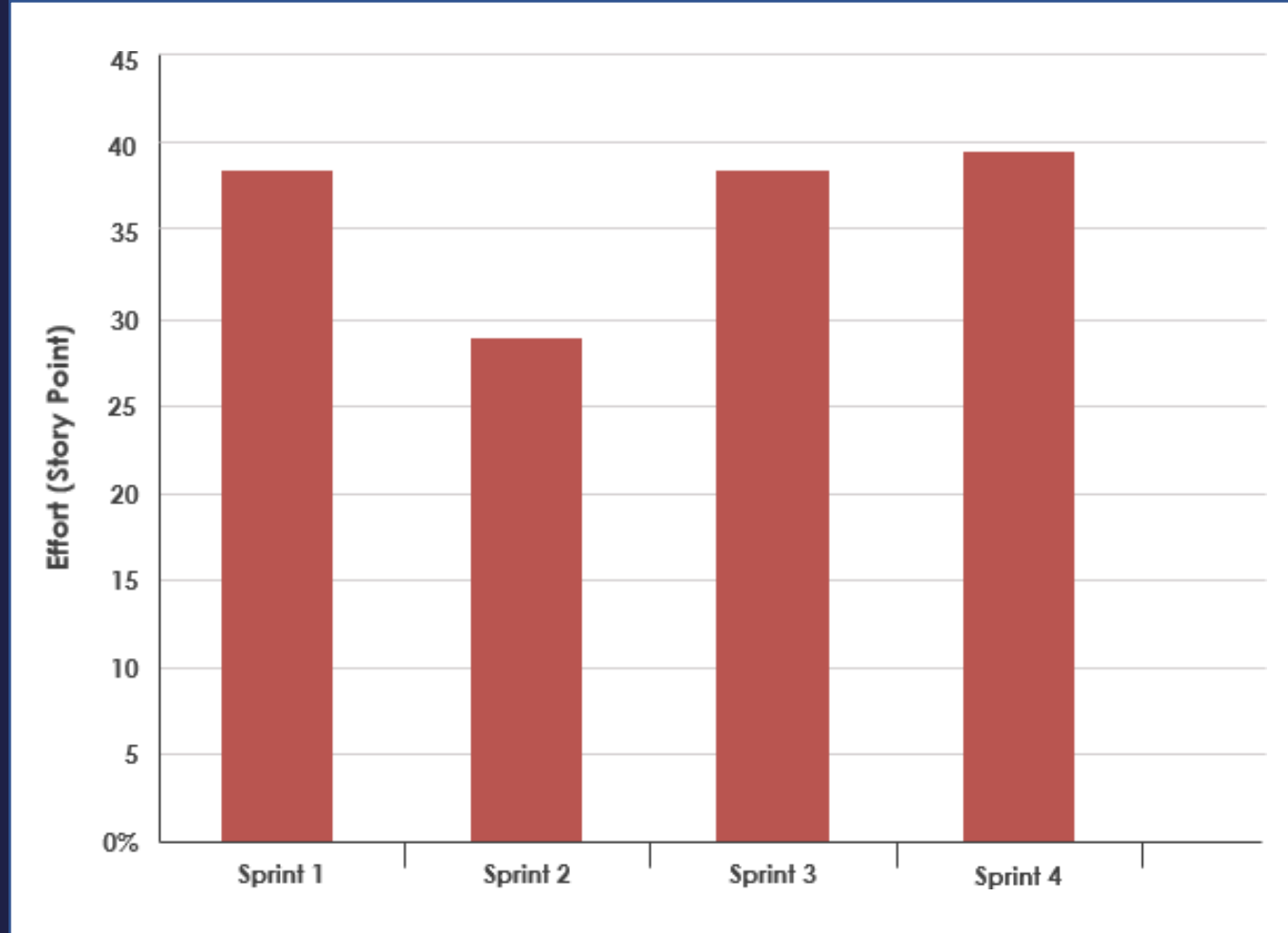
$$\text{predictibilité} = \text{\#points réalisé} / \text{\#points prévus}$$

Le **NPS (Net Promoter Score)** est un indicateur utilisé pour connaître la propension de clients prêt à recommander une marque, un produit ou encore un service

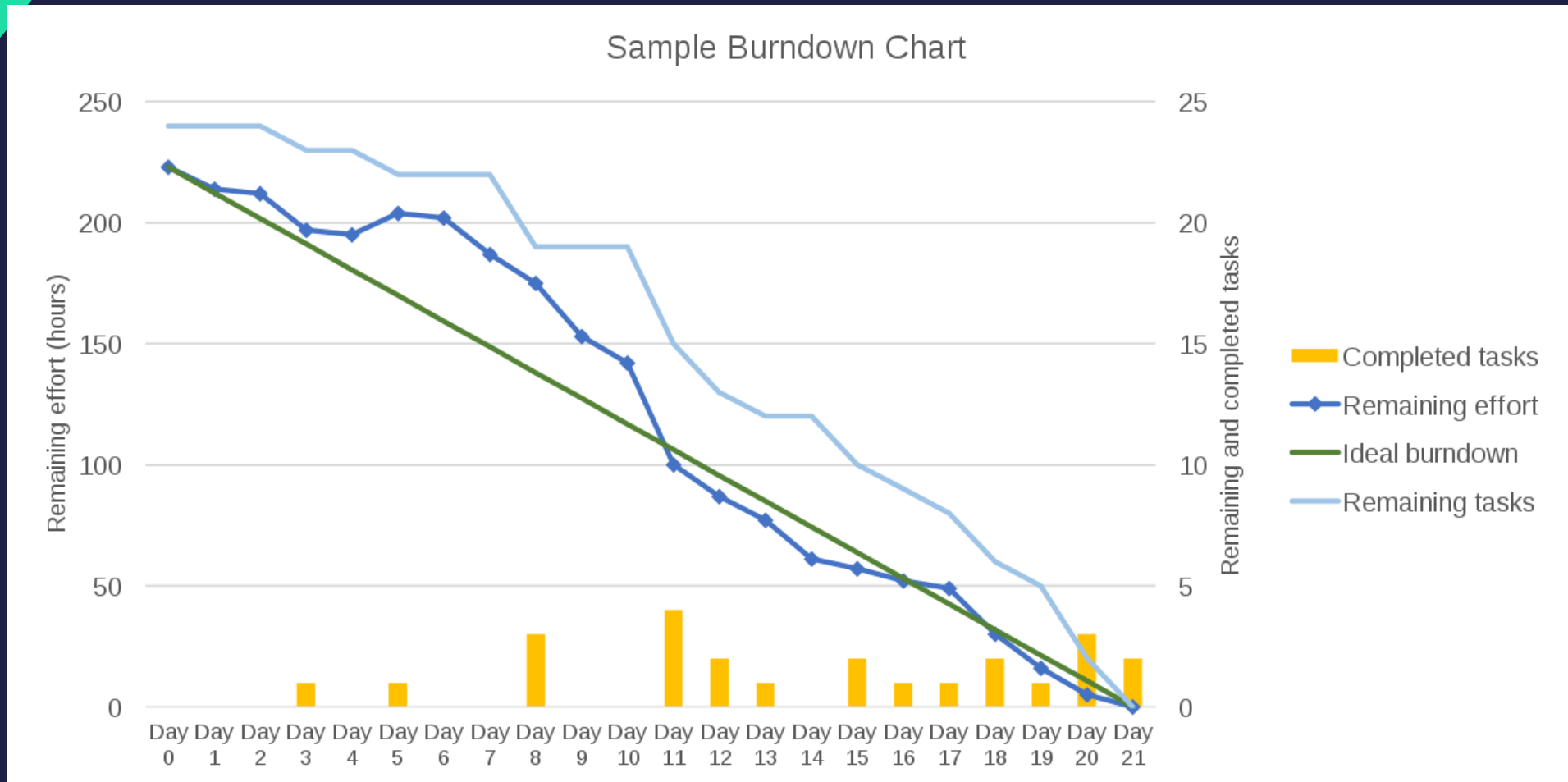
$$\text{NPS} = \% \text{promotor} - \% \text{detractor}$$

La **vélocité** permet de déterminer l'effort qu'est capable de fournir une équipe de développement pour la réalisation des tâches programmées dans un sprint. Elle est exprimée en nombre de points / sprint

Vélocité = Somme points



Le **Burndown chart** est une représentation graphique de l'évolution de quantité de travail restante par rapport au temps sur une période de temps donnée



Pour aller plus loin

<https://www.atlassian.com/blog/agile>