

Découverte de Scrum

Et de quelques principes de l'agilité



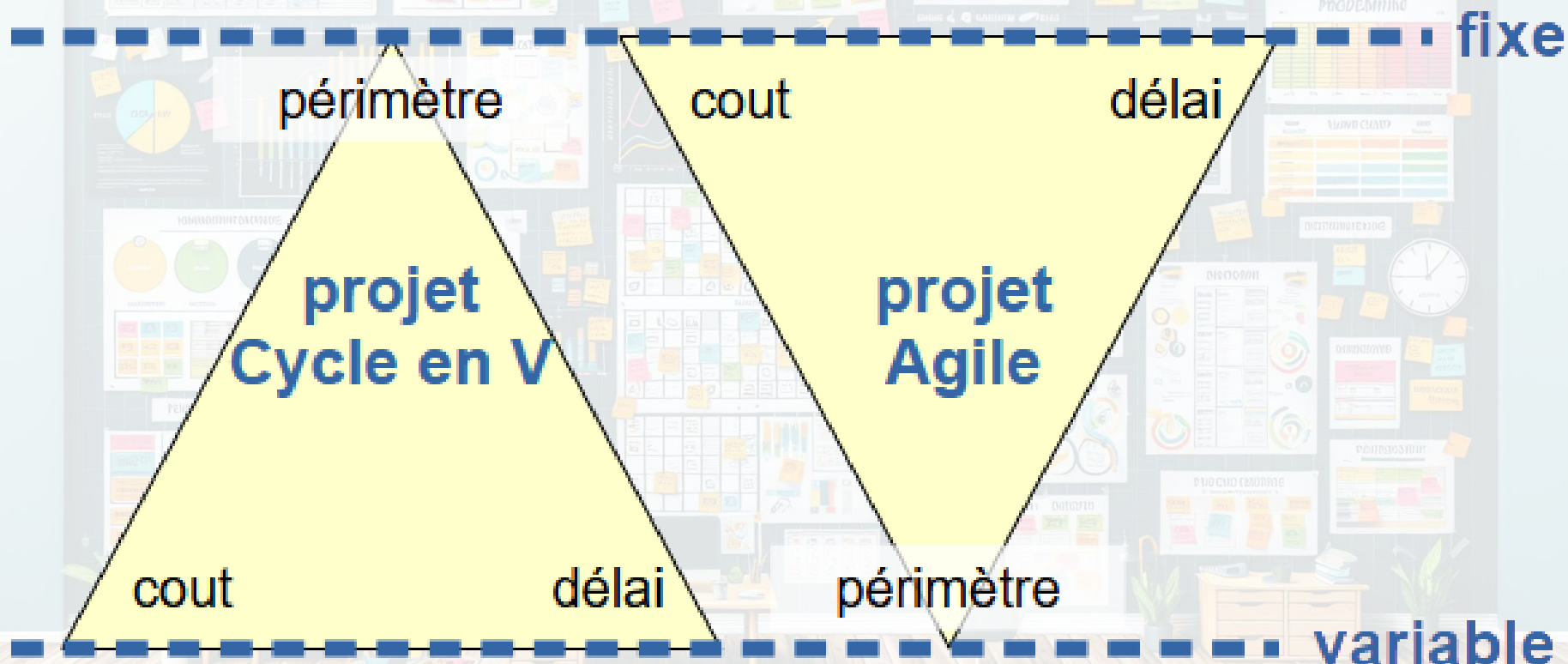
Jeu créé par Alban Catry
alban.catry@gmail.com

L'approche agile met en avant...

- La collaboration avec le client
- Une construction itérative et incrémentale
- L'excellence technique
- L'adaptation au changement

4 valeurs, 12 principes → agilemanifesto.org

Le « triangle de fer »



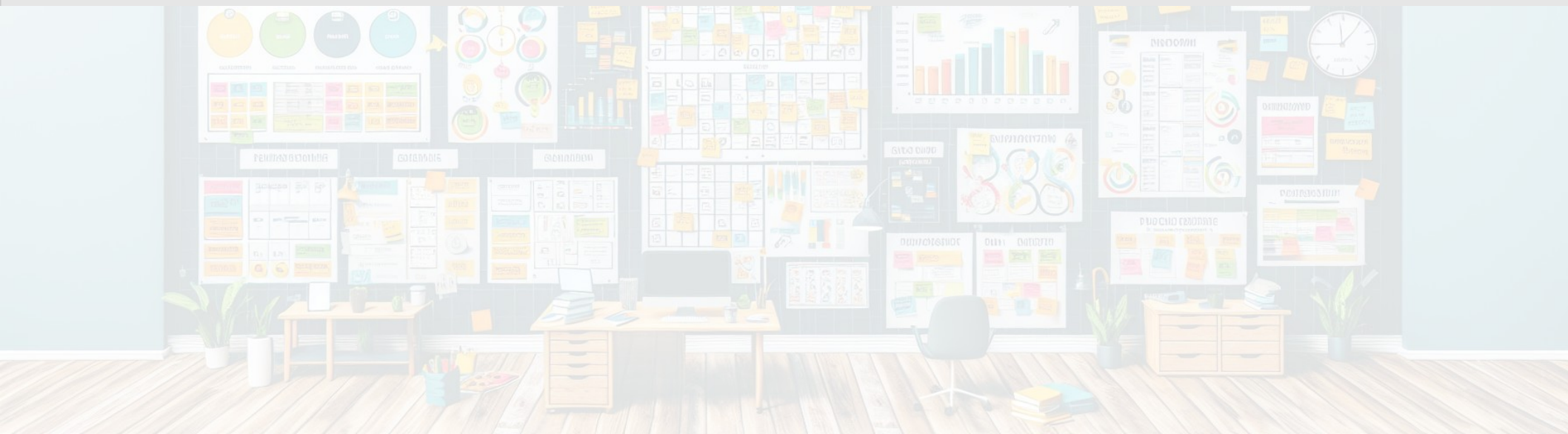
Les règles du jeu

Vous êtes une petite équipe Scrum dans une société de développement informatique...

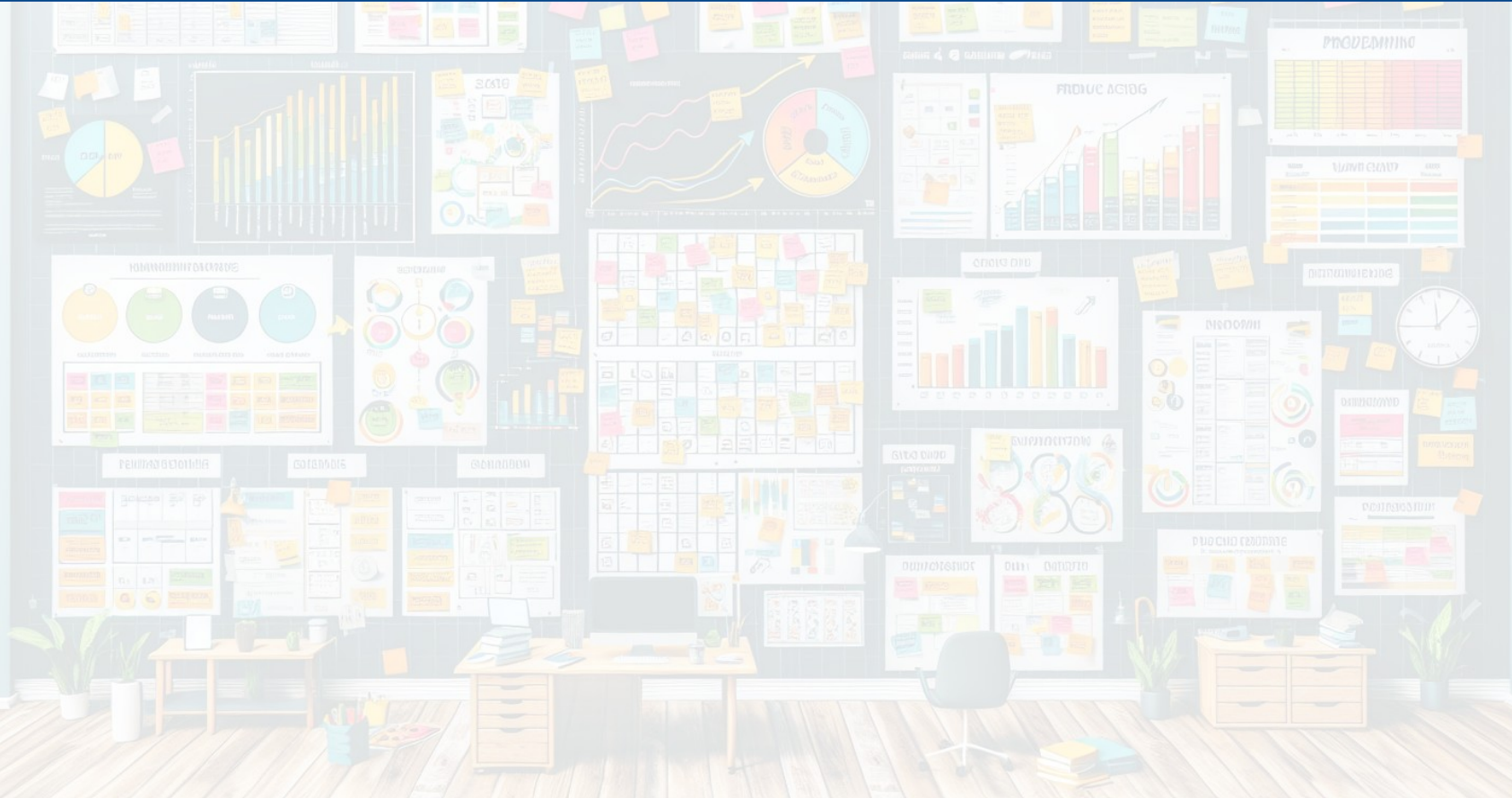
Le site Permalivre va bientôt sortir !

Faire en sorte que chacun puisse produire sa propre alimentation, de façon écologique, voilà l'ambition folle du site de vente en ligne Permalivre qui, selon son directeur Marketing, *« permettra à tous d'acheter pour un prix dérisoire toute la littérature autour de la permaculture et de l'agriculture accessible à tous, locale, solidaire et respectueuse de l'environnement. »*

Promis d'ici moins de 3 mois, ce site sera développé par une petite équipe informatique suivant les principes de l'agilité et une méthode agile, dénommée Scrum. Vous ne connaissez pas l'agilité ? Pas de panique, on vous explique tout, et on vous présente l'équipe !



Les fonctionnalités attendues



Les fonctionnalités attendues ont été regroupées en 9 **Épic**

Epic

Livre

Epic

Recherche

Module

Suggestions

Epic

Catégories

Epic

Panier

Epic

Achat

Epic

Filtres

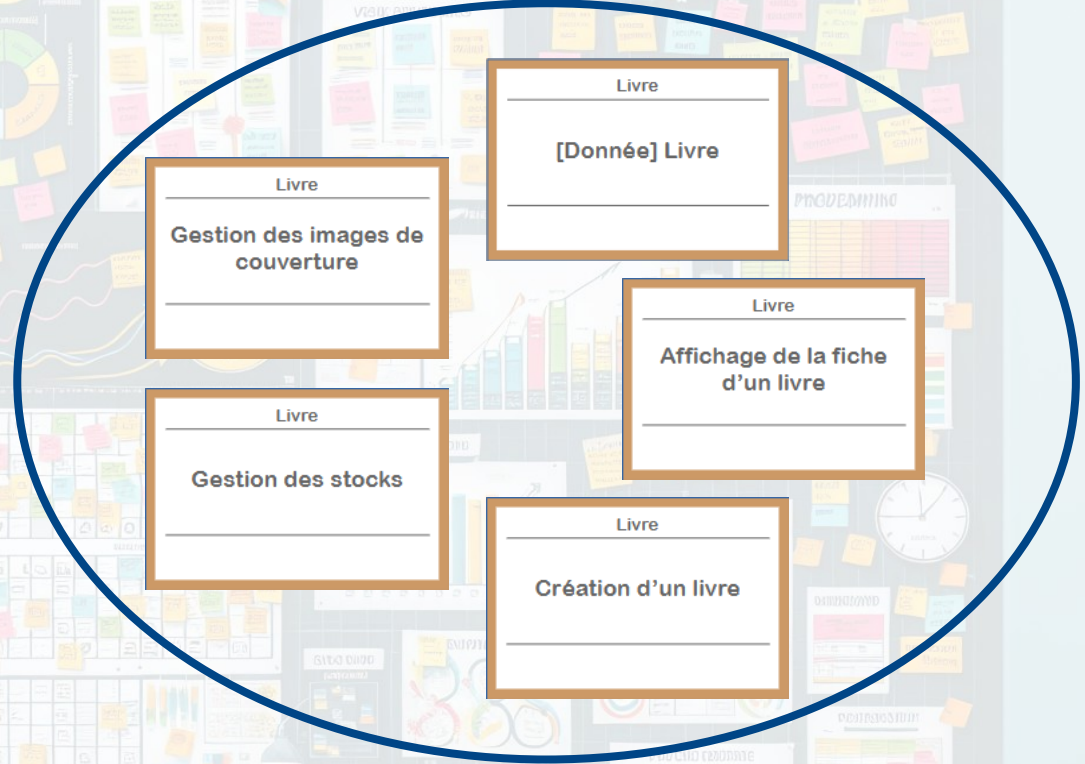
Epic

Compte client

Epic

Expédition

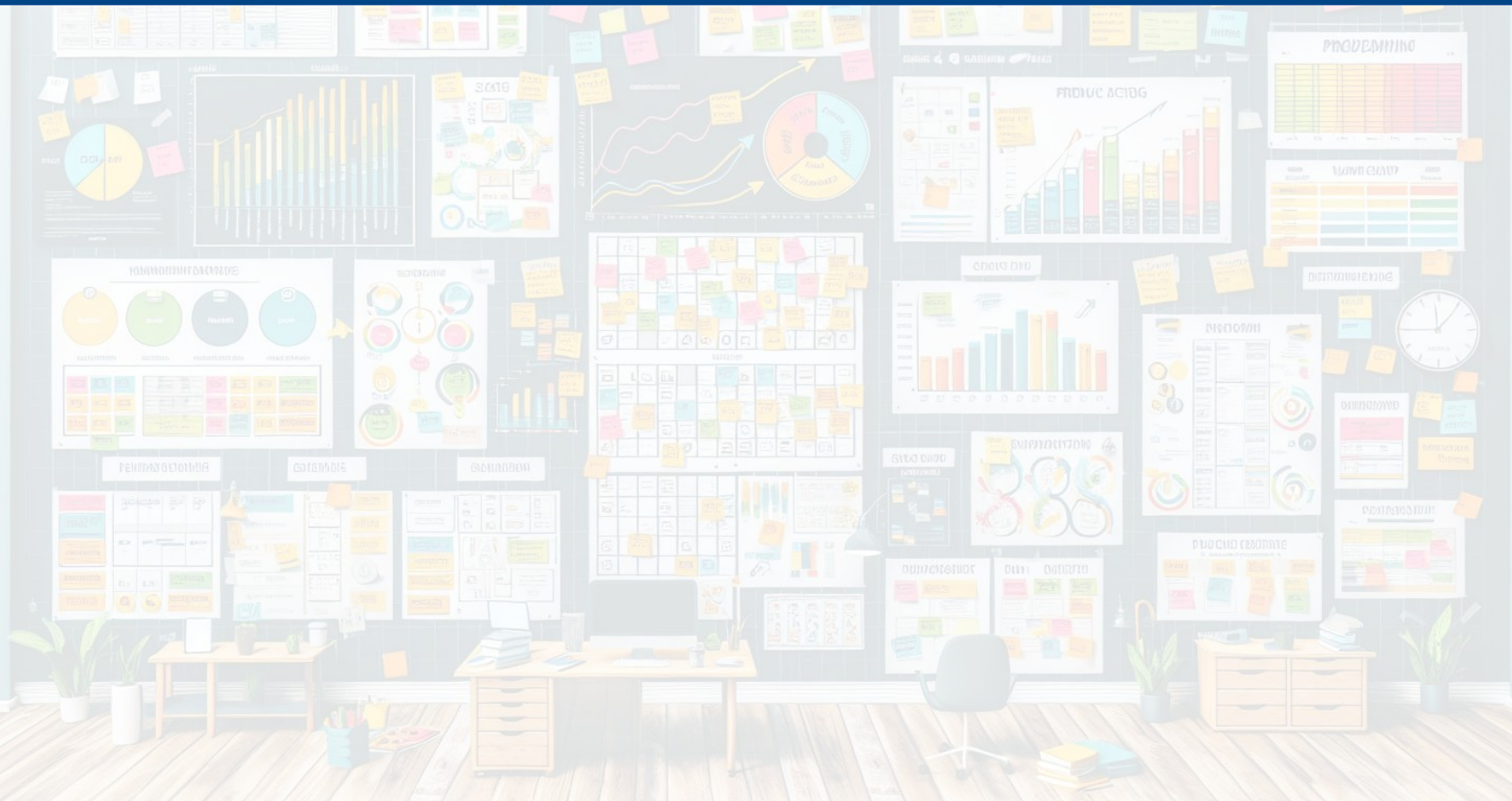
En agilité, on décompose les applications à développer en **User Stories** = « histoires utilisateurs »



Ceci est une **Epic**.

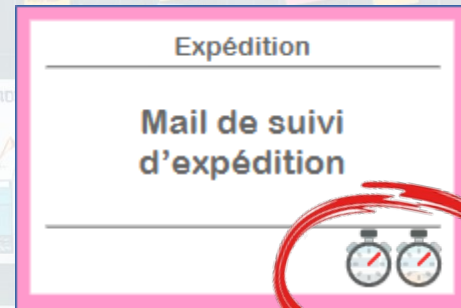
C'est une unité de travail trop grosse pour être réalisée dans un sprint par un développeur et qui doit donc être décomposée en User Stories plus petites.

Déroulement de la partie

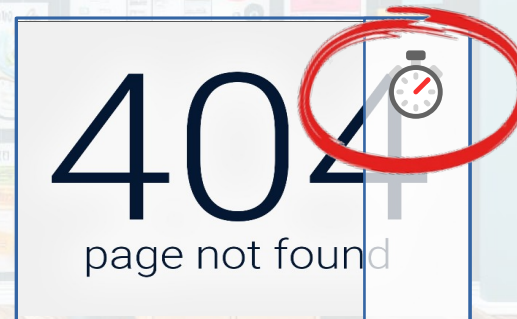


Votre activité principale : développer...

Développer des User Stories



et/ou corriger des anomalies



...mais avec un risque de créer de la **dette technique**



Développeur

Capacité par sprint

Actions

Développer / Corriger une anomalie

Three orange dice showing 6, 5, and 6, followed by a right-pointing arrow and a crossed wrench and screwdriver icon. This entire section is circled in red.

Pour chaque User Story développée et chaque anomalie corrigée, le développeur jette le dé.

S'il obtient 4, 5, 6, on ajoute une dette technique.

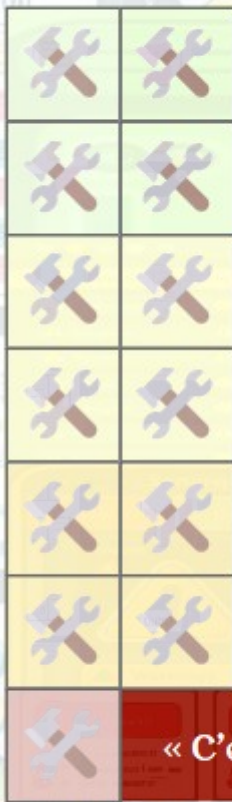
Le développeur débutant a tendance à créer plus de dette technique.

Le développeur confirmé, moins !

La **dette technique** est tout ce qui contribue à rendre le code plus difficile à maintenir et à faire évoluer :

- absence de commentaires,
- code mal indenté / mal écrit,
- conventions de nommage non-respectées,
- duplication du code,
- mauvaises modélisations,
- mauvais choix d'architecture, de librairies,
- ...

Dette technique



Les développeurs ont leur capacité normale
(*effet immédiat*)

Les développeurs débutant et confirmé ont une capacité diminuée de 1.
(*effet immédiat*)

Tous les développeurs ont une capacité diminuée de 1.
(*effet immédiat*)

« C'est vraiment de la merde, vous êtes virés ! »

Masquez la capacité perdue à l'aide de ces tokens



Pour réduire la dette technique, il faut
refactoriser le code



Le projet sera mené en 8 sprints

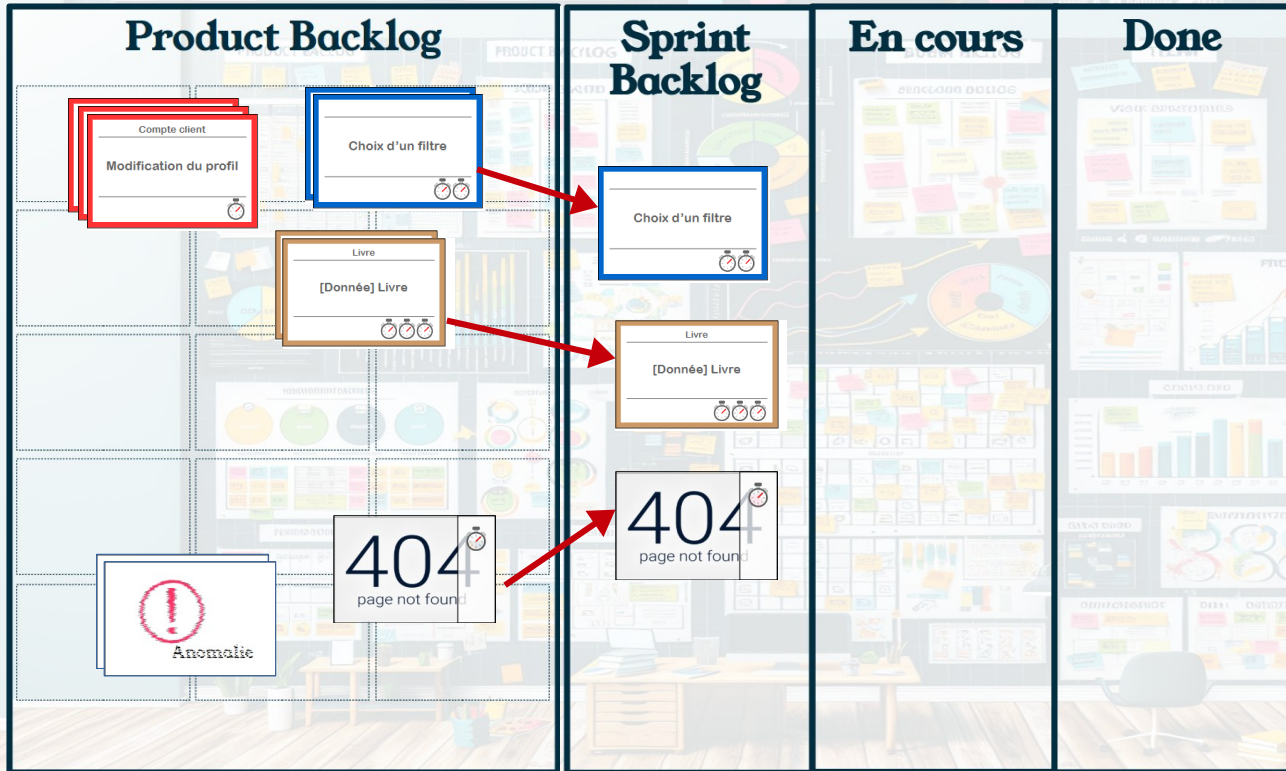


SPRINT

PLANIFICATION
DÉVELOPPEMENT
DÉMONSTRATION
RÉTROSPECTIVE

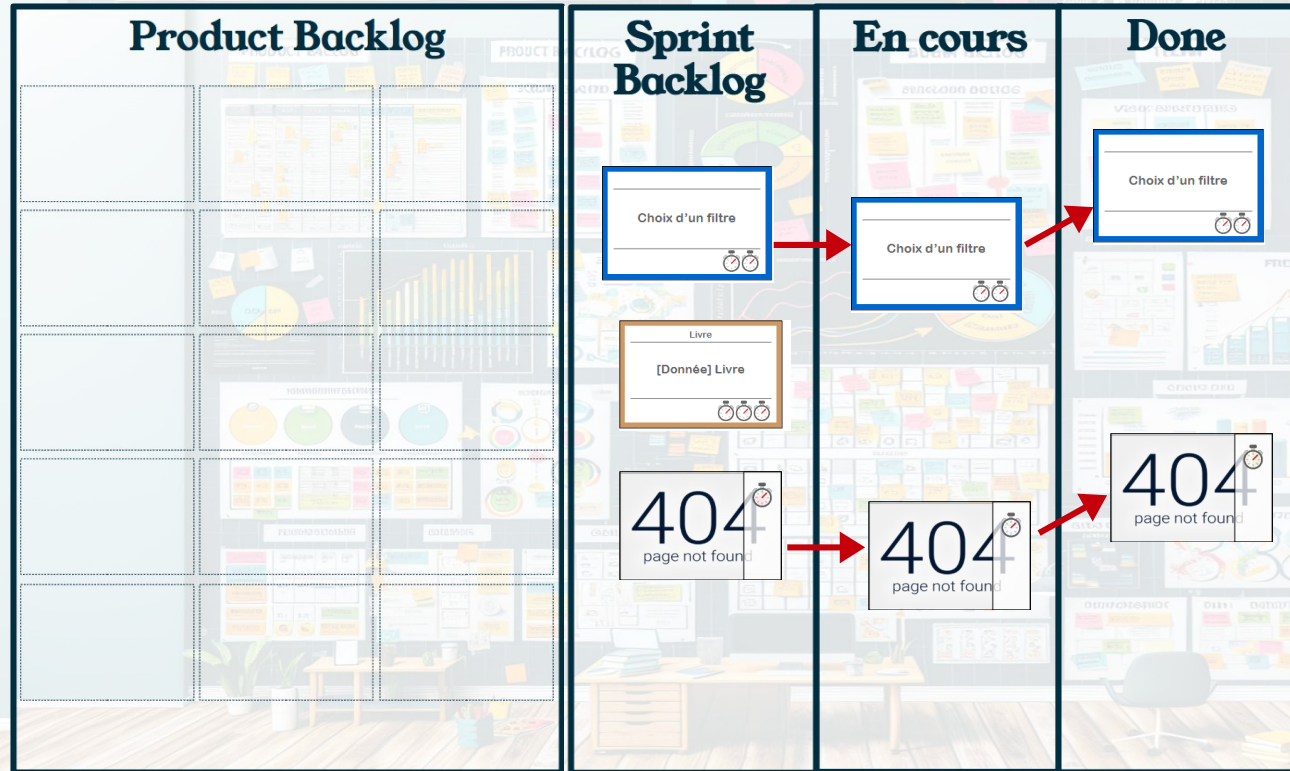
Phase de planification

L'équipe choisit collectivement dans le **Product Backlog** ce qu'elle va réaliser à ce sprint et le met dans le **Sprint Backlog**.



Phase de développement

Chacun son tour, chaque développeur utilise sa capacité pour développer ou refactoriser.



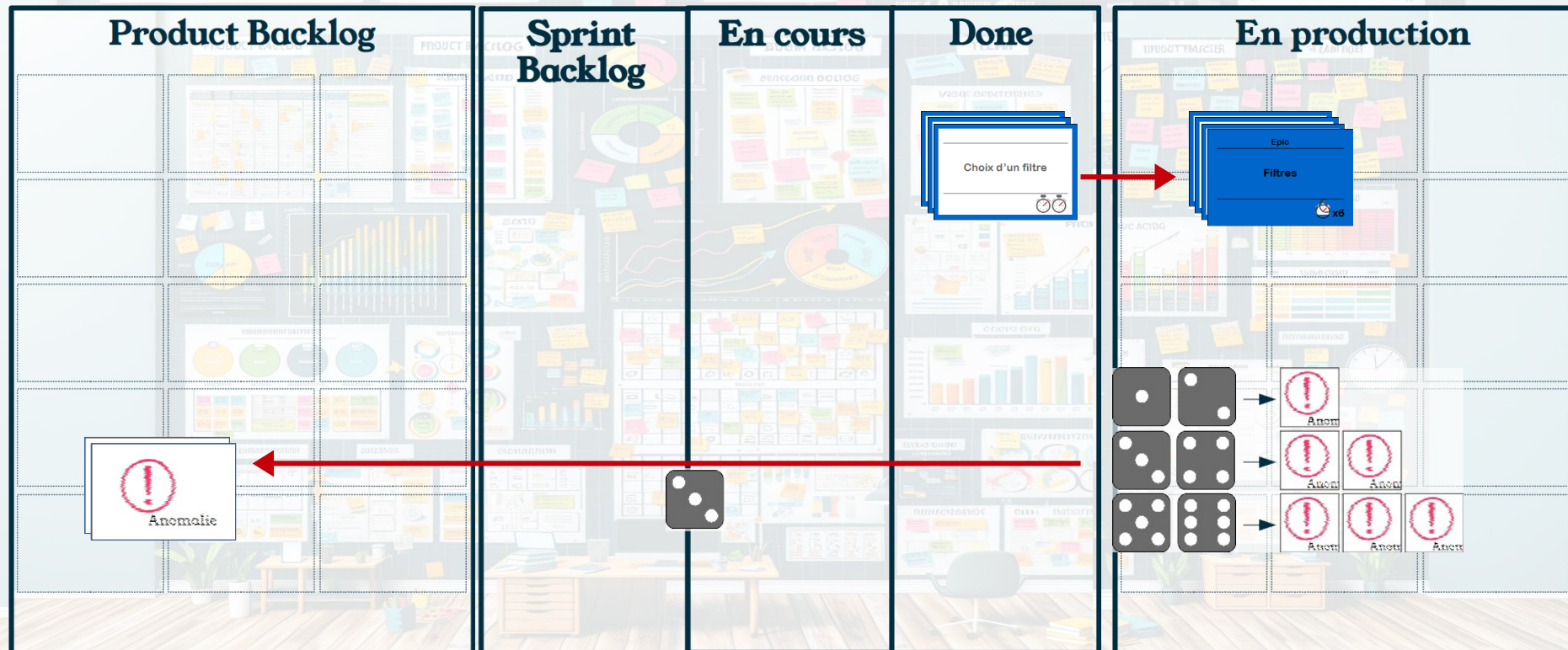
Ne pas oublier de lancer le dé de dette technique à chaque fois !



Phase de démonstration (1/3)

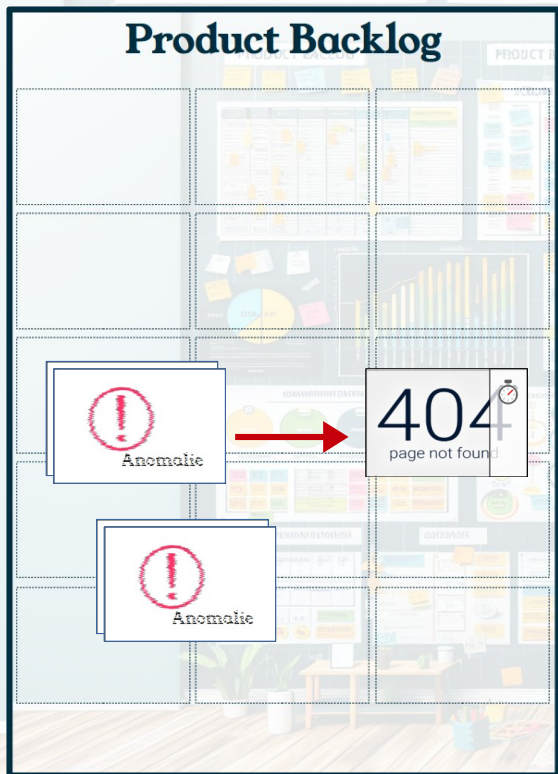
Pour chaque Épic intégralement terminée, on la passe « **En production** » et détermine son nombre d'**anomalies**, qui vont dans le Product Backlog.

Attention, à ce stade, les anomalies sont cachées, car elles n'ont pas encore été trouvées !



Phase de démonstration (2/3)

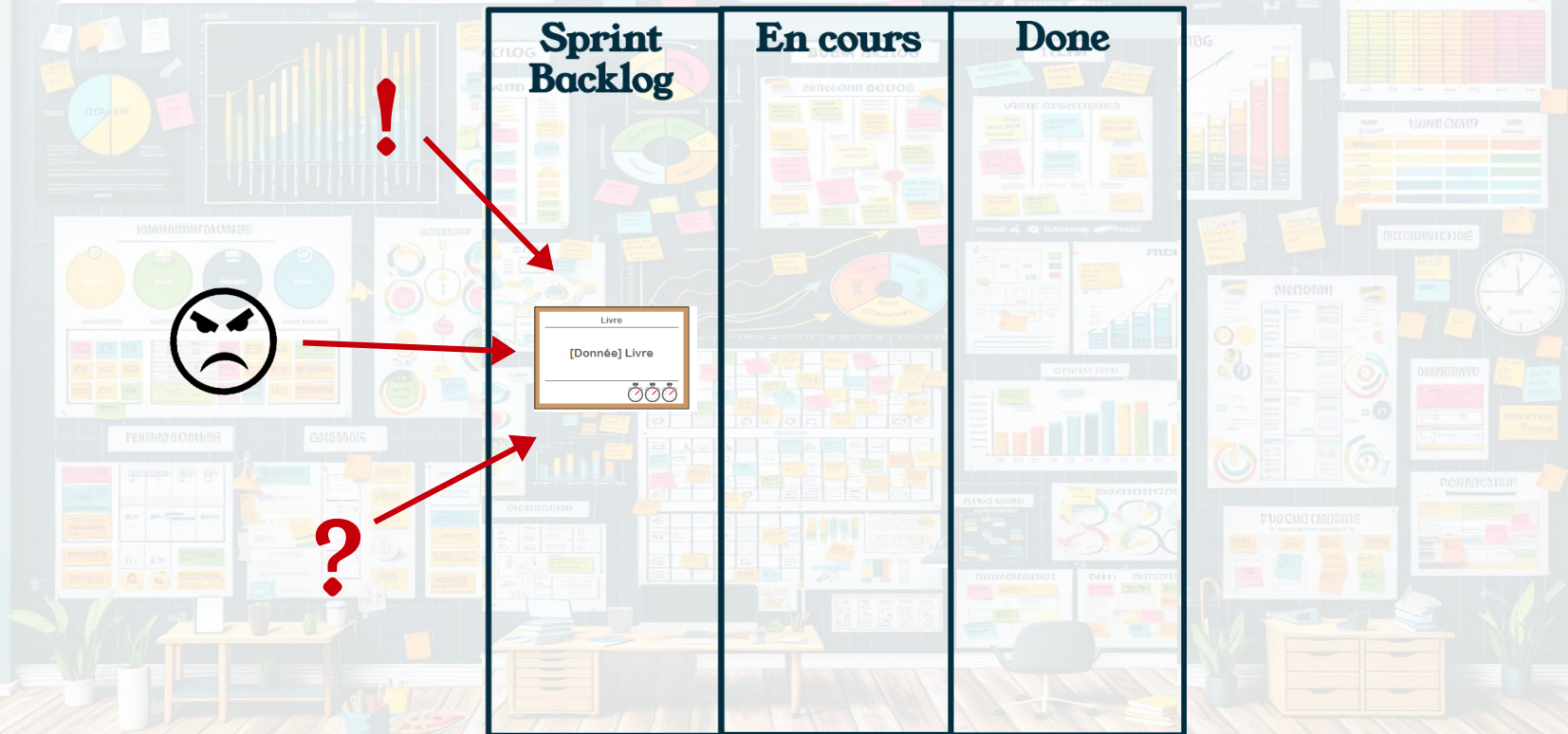
Pour chaque tas d'anomalies qui ne vient pas d'être créé à ce tour, on en révèle une et on la place à part sur le Product Backlog.



Phase de démonstration (3/3)

S'il reste encore quelque chose sur le Sprint Backlog, le client est mécontent (après tout, on lui avait promis qu'on le ferait !).

Les cartes restent sur le Sprint Backlog pour le prochain sprint.



La satisfaction du client

Le client a une satisfaction neutre en début de partie.

Certaines situations augmentent ou diminuent la satisfaction du client. Vous apprendrez vite à les reconnaître !

Satisfaction du client



Phase de **rétrospective**

C'est au moment de la rétrospective que l'équipe trouve des façons de s'améliorer.

- L'équipe pioche trois Idées et en choisit une.
- On applique son effet à partir du prochain tour.
- Les autres sont remises dans la pioche et sont remélangées.



Machine à
café plus
performante

Team
building aux
Baléares

Augmentation
de salaire de
40 %

Le Scrum Master

Le **Scrum Master** a parfaitement assimilé les règles du jeu et sera garant de l'animation à partir du deuxième sprint.

Il doit s'assurer que chaque sprint dure 5 minutes !



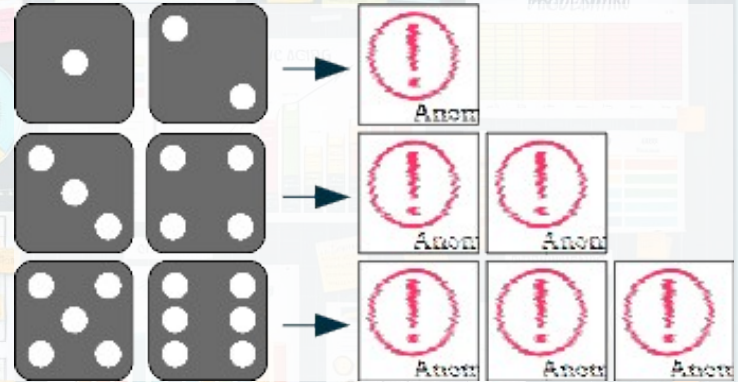
5 min

Le Product Owner

Le **Product Owner** priorise le Product Backlog.

Il organise le site web.

Il jette le dé gris pour la création d'anomalies.



C'est parti !

Faites comme vous feriez pour un vrai projet !

Ah, j'oubliais, il y aura un petit événement à piocher au début de chaque sprint.

Bon courage !

