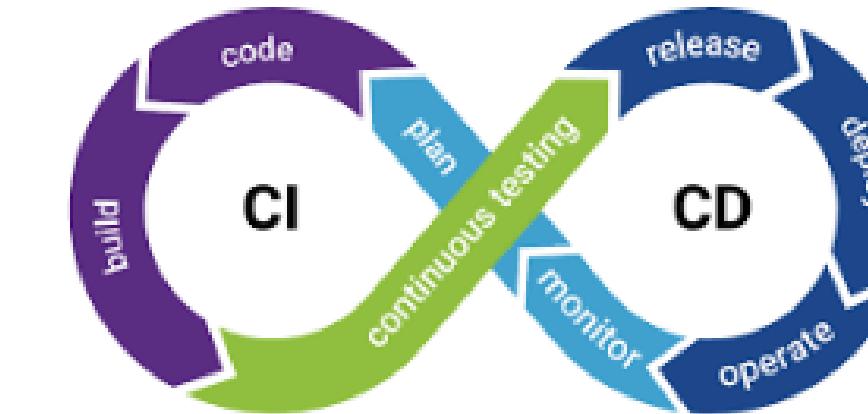


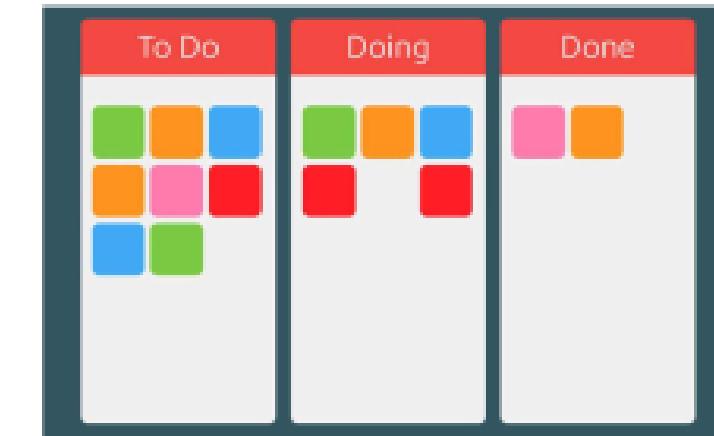
Aujourd'hui: ven 20/10

# La Méthode AGILE

# XP

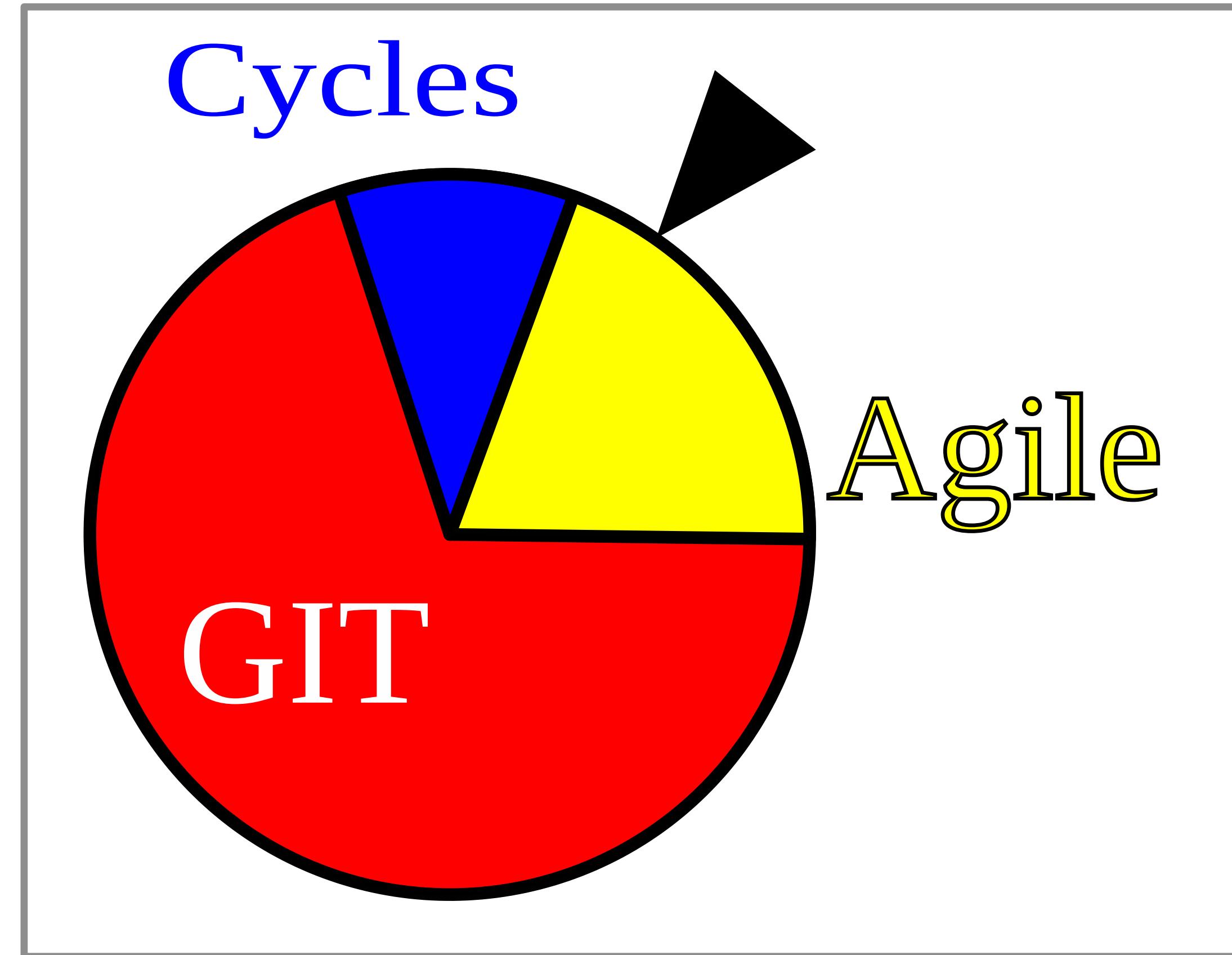


# KANBAN

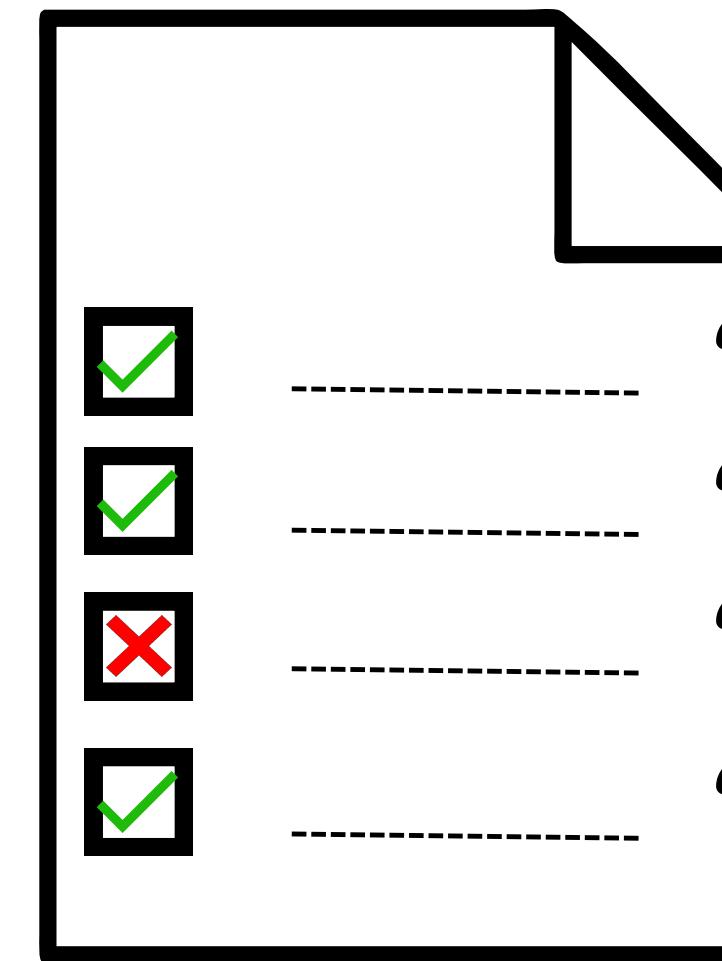
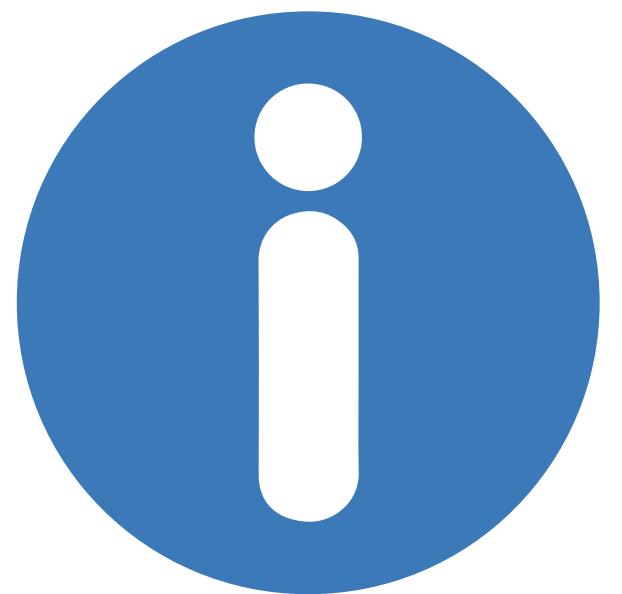
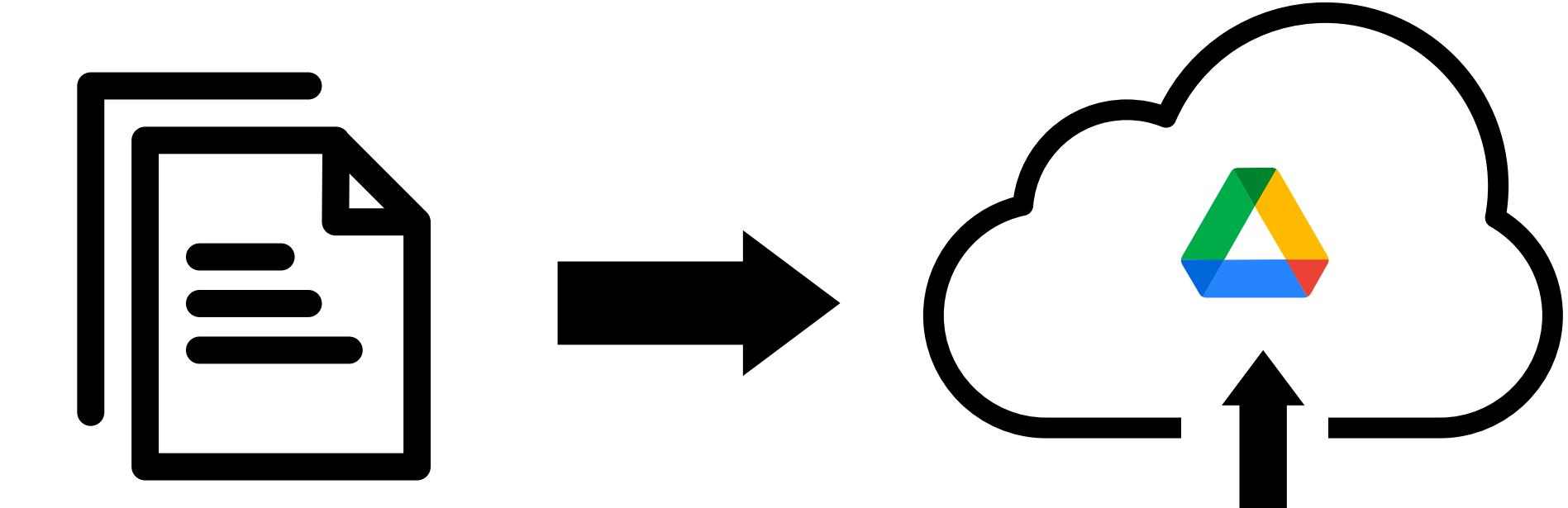


# EPICS/STORY

# Où en sommes nous du cours ?



# Supports

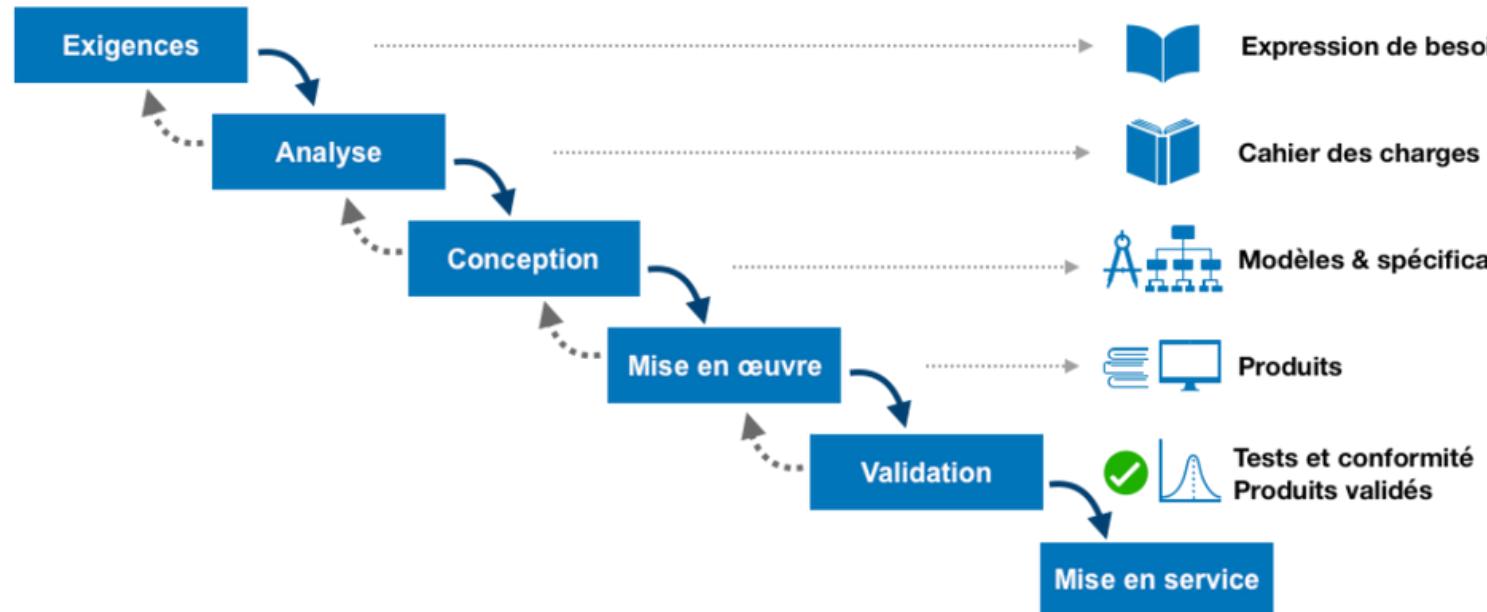


# Évals

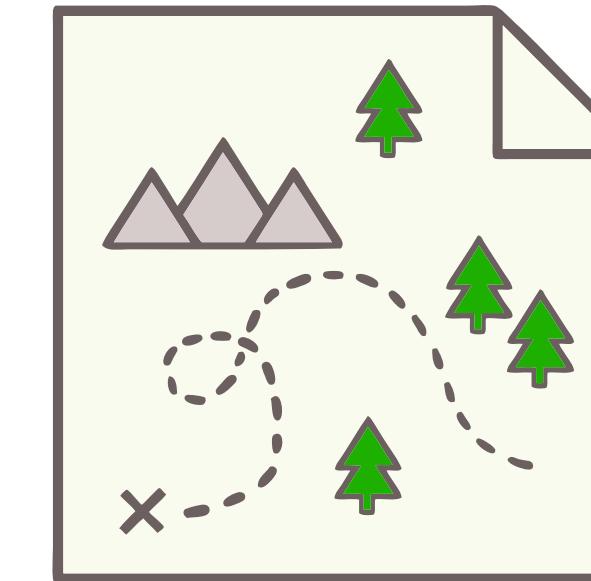
# Rappel des épisodes précédents

“à quoi sert la méthode agile”

Inadapté

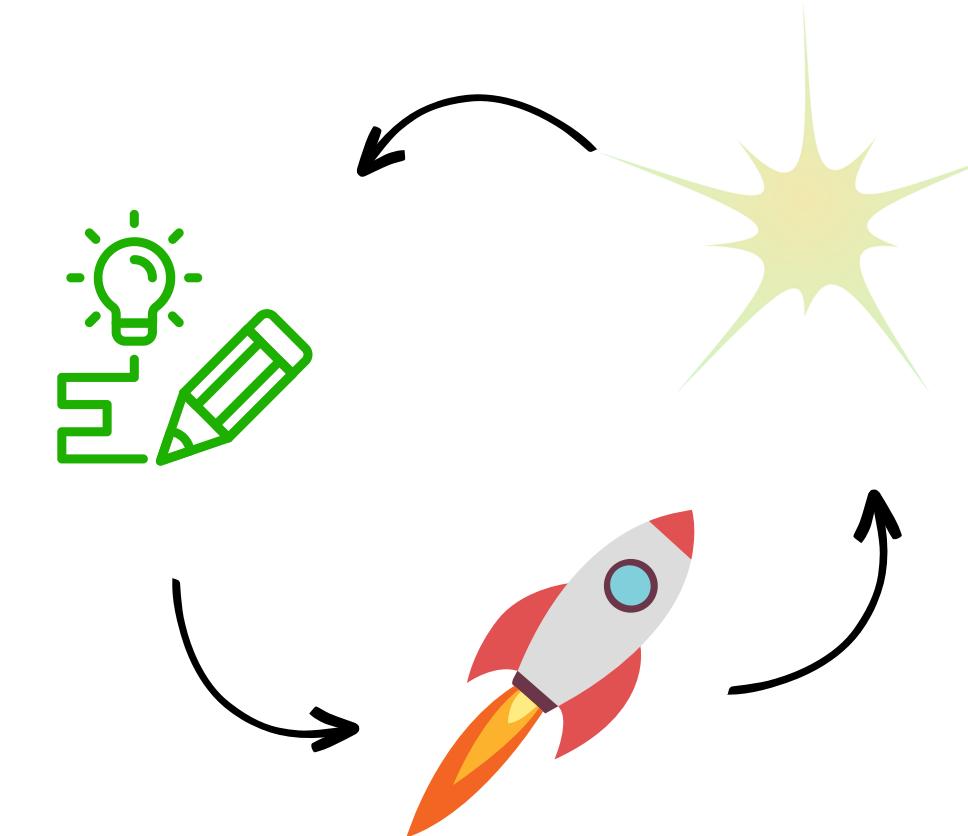


Risqué



Se repérer

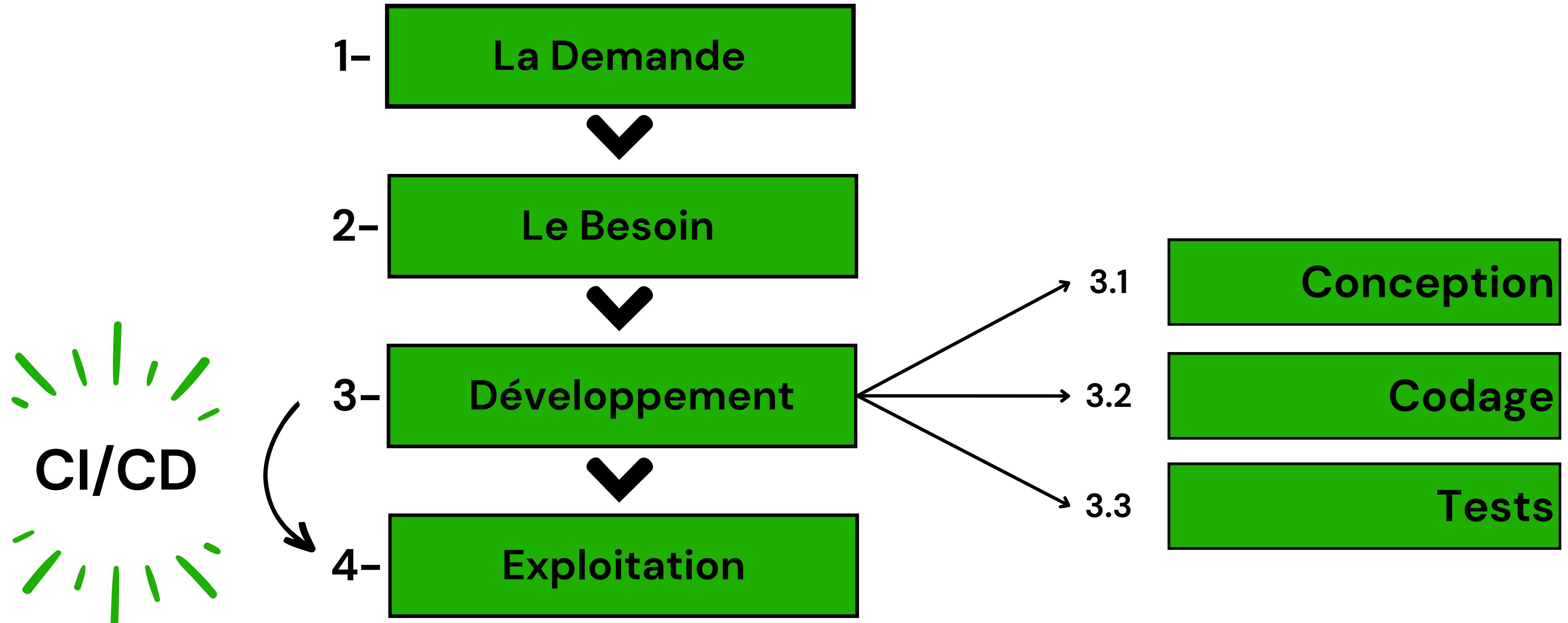
S'adapter



Être efficace

# Rappel des épisodes précédents

“les étapes simplifiées: pour se repérer”



# Rappel des épisodes précédents

**“objectif: satisfaction maximale”**



# Menu du Jour

**Extrem Programming  
Le manifeste Agile**

**La méthode SCRUM**

**Le BACKLOG  
Ateliers User Stories**

# Extrem Programming: 1999

## Objectifs

- une tentative de réconcilier l'humain avec la productivité,
- un mécanisme pour faciliter le changement social,
- une voie d'amélioration,
- un style de développement,
- une discipline de développement d'applications informatiques.



WIKIPEDIA

## Techniques

- la revue de code sera faite en permanence (par un binôme) ;
- les tests seront faits systématiquement avant chaque mise en œuvre ;
- le code sera retravaillé tout au long du projet (refactoring ou remaniement du code) ;
- la solution la plus simple sera toujours celle qui sera retenue ;
- des métaphores seront définies et évolueront en concomitance ;
- les modifications seront faites plusieurs fois par jour ;
- des cycles de développement très rapides faciliteront l'adaptation au changement.

# Manifeste AGILE

<https://www.atlassian.com/fr/agile manifesto>

- Les **individus et leurs interactions** plutôt que les processus et les outils
- Des **logiciels opérationnels** plutôt qu'une documentation exhaustive
- La **collaboration avec les clients** plutôt que la négociation contractuelle
- L'**adaptation au changement** plutôt que le suivi d'un plan

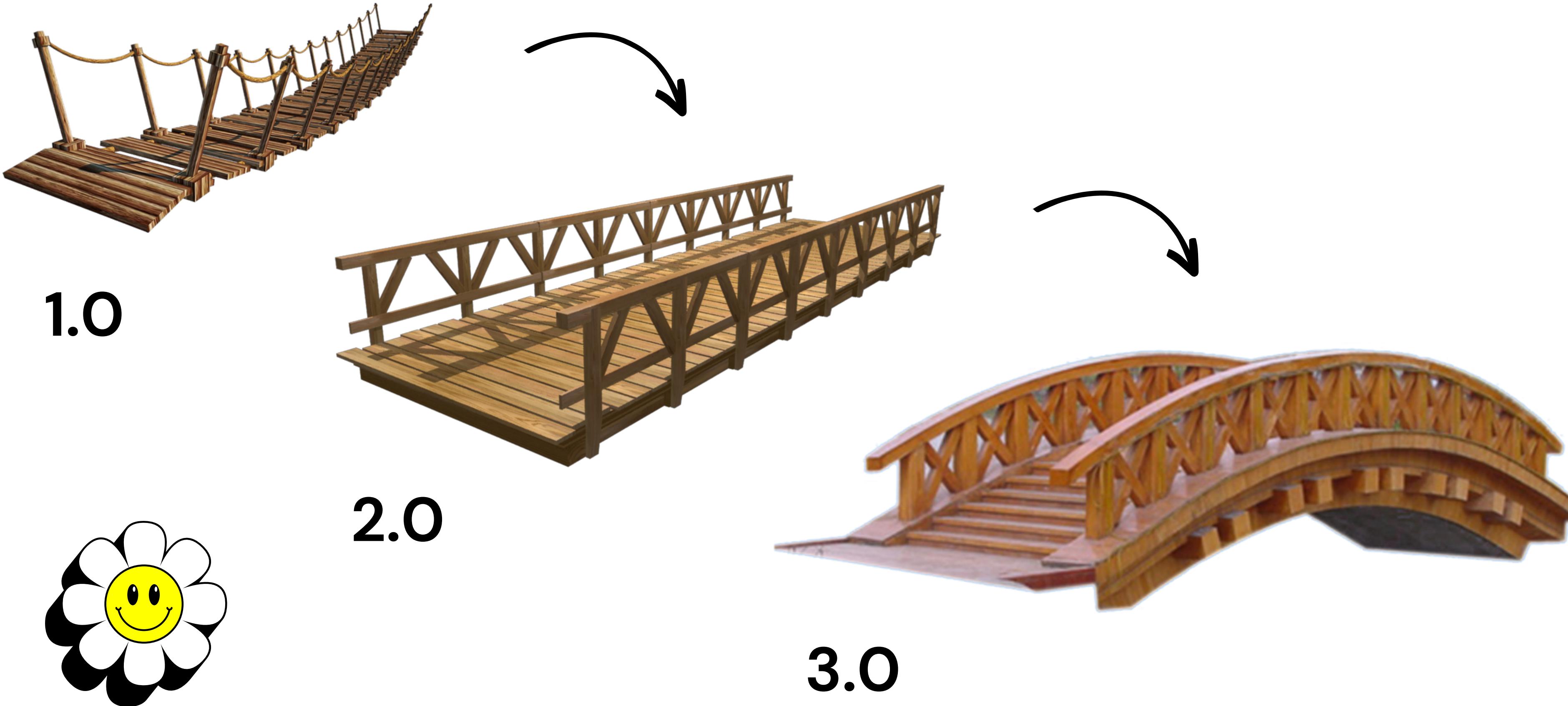
Du 11 au 13 Février 2001, dans l'hôtel The Lodge de la station de ski Snowbird

Kent Beck  
James Grenning  
Robert C. Martin  
Mike Beedle  
Jim Highsmith  
Steve Mellor

Arie van Bennekum  
Andrew Hunt  
Ken Schwaber  
Alistair Cockburn  
Ron Jeffries  
Jeff Sutherland

Kent Beck  
Ward Cunningham  
Jon Kern  
Dave Thomas  
Martin Fowler  
Brian Marick

# Une conséquence des principe Agiles: la progression itérative par prototypes



le premier proto s'appelle le

**“Minimum Viable Product”**



0.1

# “MVP”: c'est quoi ?

*“Version d'un produit qui permet d'obtenir un maximum de retours client avec un minimum d'effort.”*

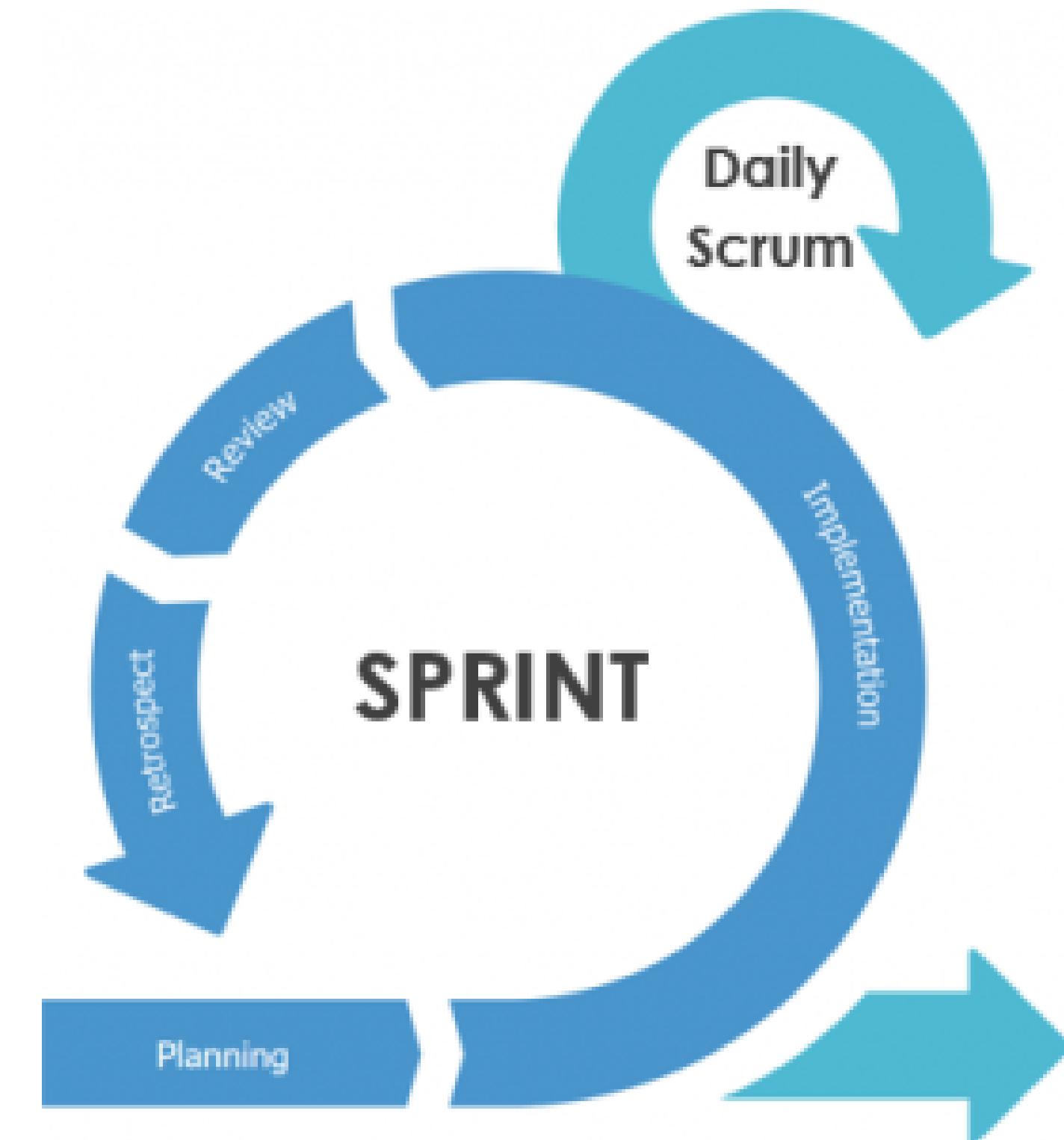
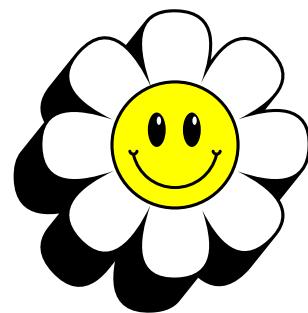
Eric Ries, auteur de “The Learn Startup”

# “MVP”: pourquoi?

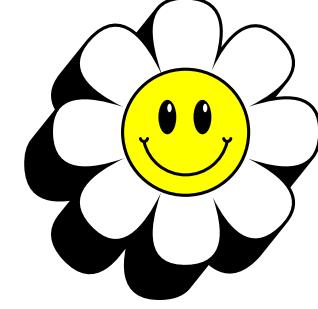
Démarche marketing permettant de développer très rapidement un idée de produit à faire tester par un nombre réduit de personnes (bêta testeurs) afin de recueillir un feed back et ainsi améliorer l'offre par itérations successives du process.

- 1- Économiser son capital
- 2- Limiter les risques
- 3- Accélérer son process
- 4- Valider son concept
- 5- Optimiser les messages marketing

# SCRUM



# SCRUM: un framework de gestion du travail



**SCRUM = MÊLÉE**



**des points de vue  
qui changent**



**des réunions régulières**

**un processus  
itératif**

**de l'auto organisation**

**de l'ouverture**

**de l'amélioration en  
continu**

# SCRUM: des valeurs



## COURAGE:

Les membres de l'équipe ont le courage de faire la chose juste et de travailler sur des problèmes difficiles.

## CONCENTRATION:

Chacun se concentre sur le Sprint et les objectifs de l'équipe

## ENGAGEMENT:

les gens s'engagent personnellement à l'atteinte des objectifs de l'équipe

## RESPECT:

Chaque membre de l'équipe respecte les autres comme étant des personnes capables et indépendantes.

## OUVERTURE:

L'équipe et toutes les parties prenantes sont ouverts à la collaboration et aux critiques constructives qui permettent l'amélioration continue.

# SCRUM: une structure d'équipe avec des rôles



SCRUM MASTER

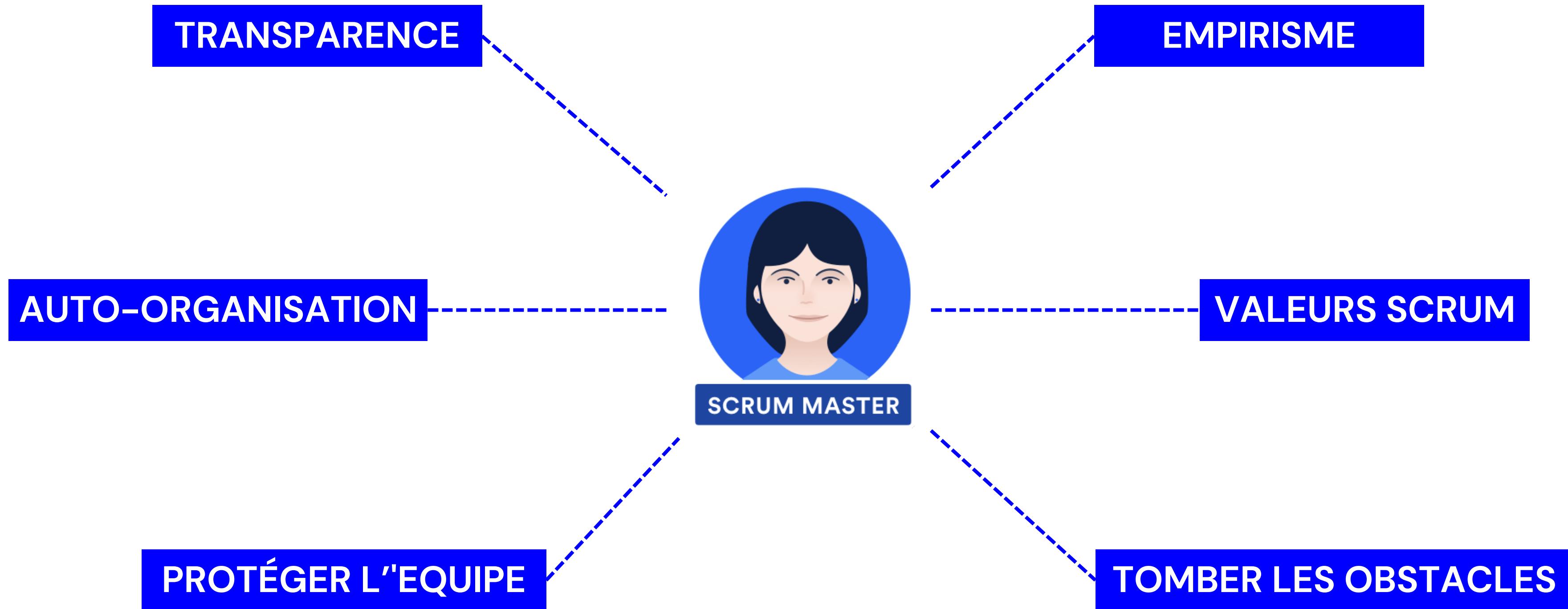


PRODUCT OWNER



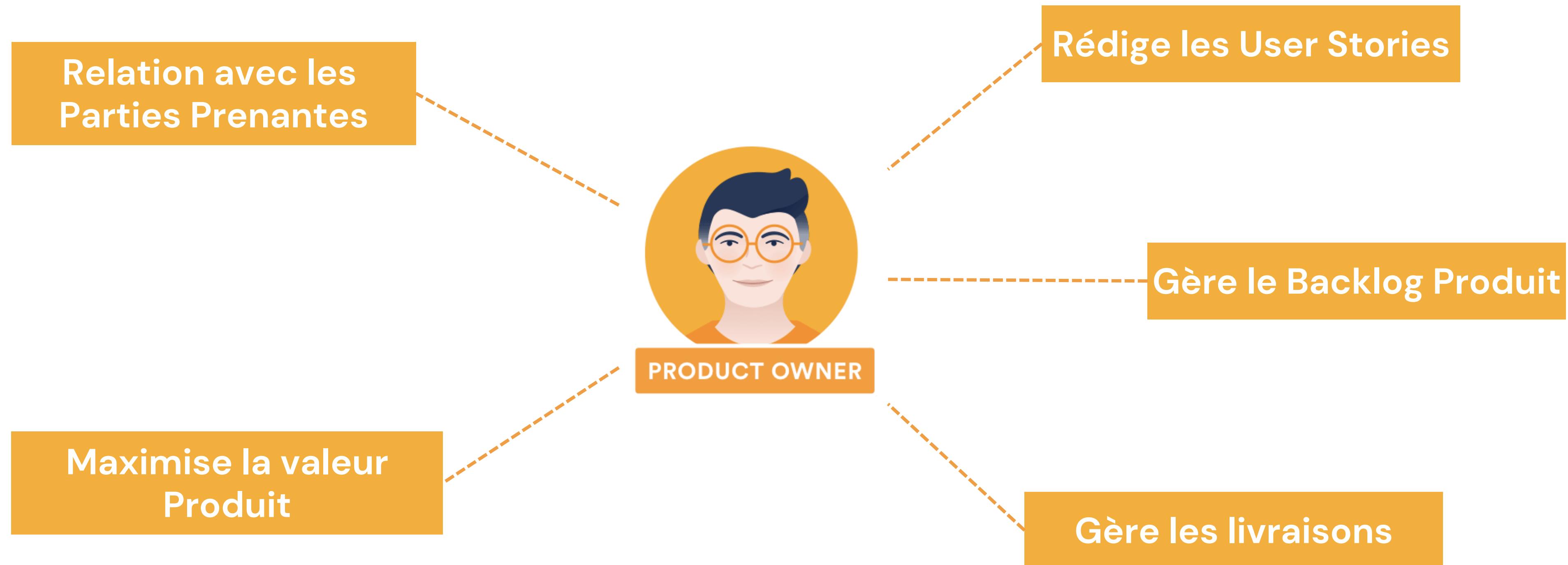
ÉQUIPE DE DÉVELOPPEMENT

# Le SCRUM-MASTER: veille à la méthode

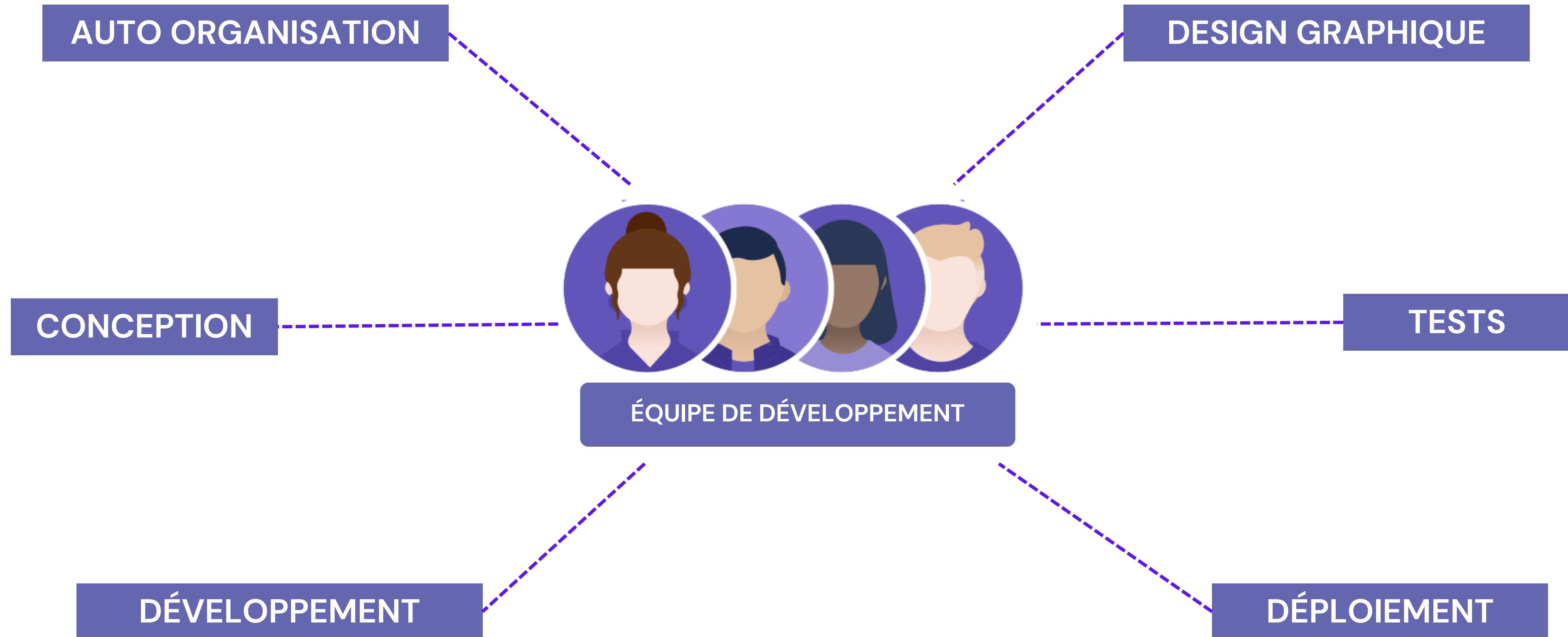


# le PRODUCT-OWNER:

## gère la conformité du produit, la relation client



# L'ÉQUIPE DE DEV: réalise le produit



# SCRUM: un processus structuré

epics

user stories

sprint

backlog produit

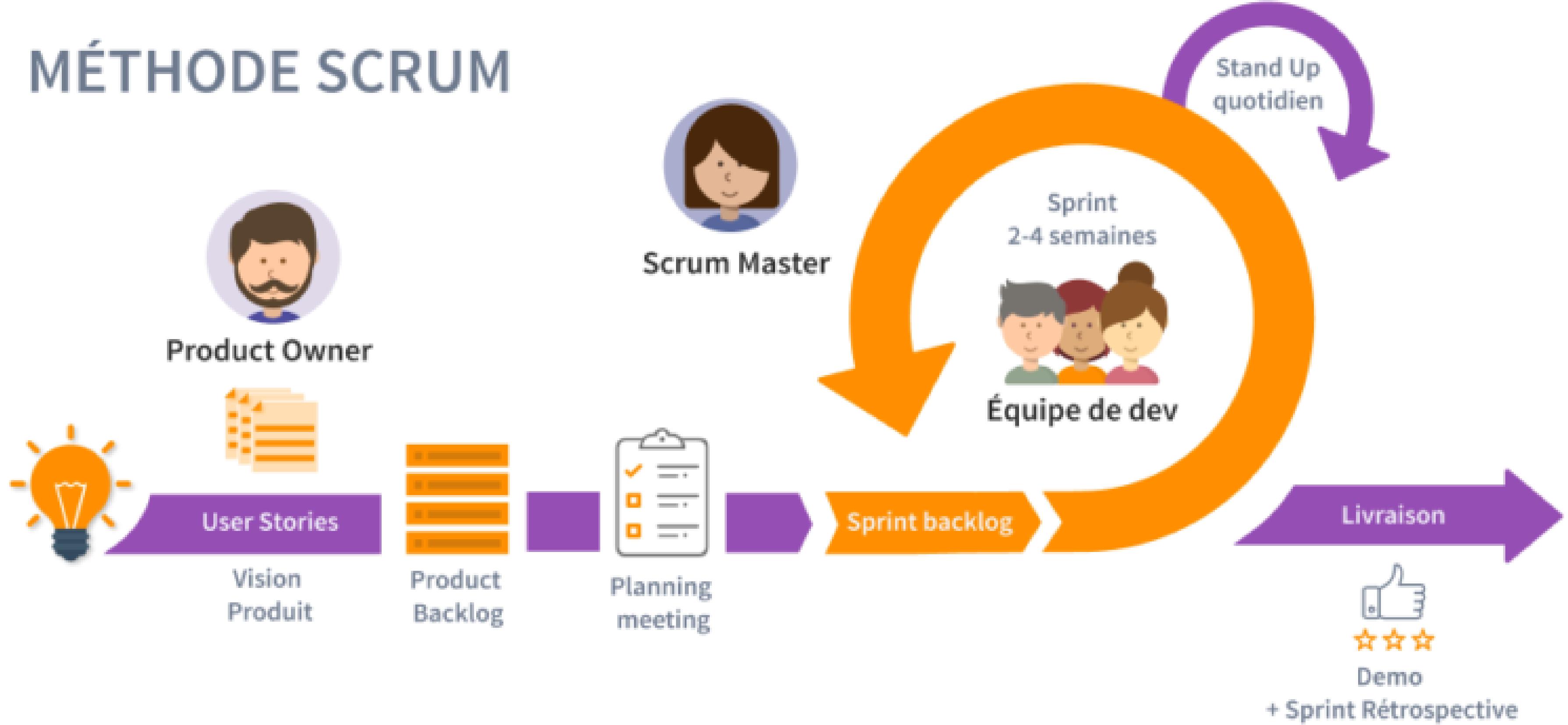
backlog sprint

daily meeting

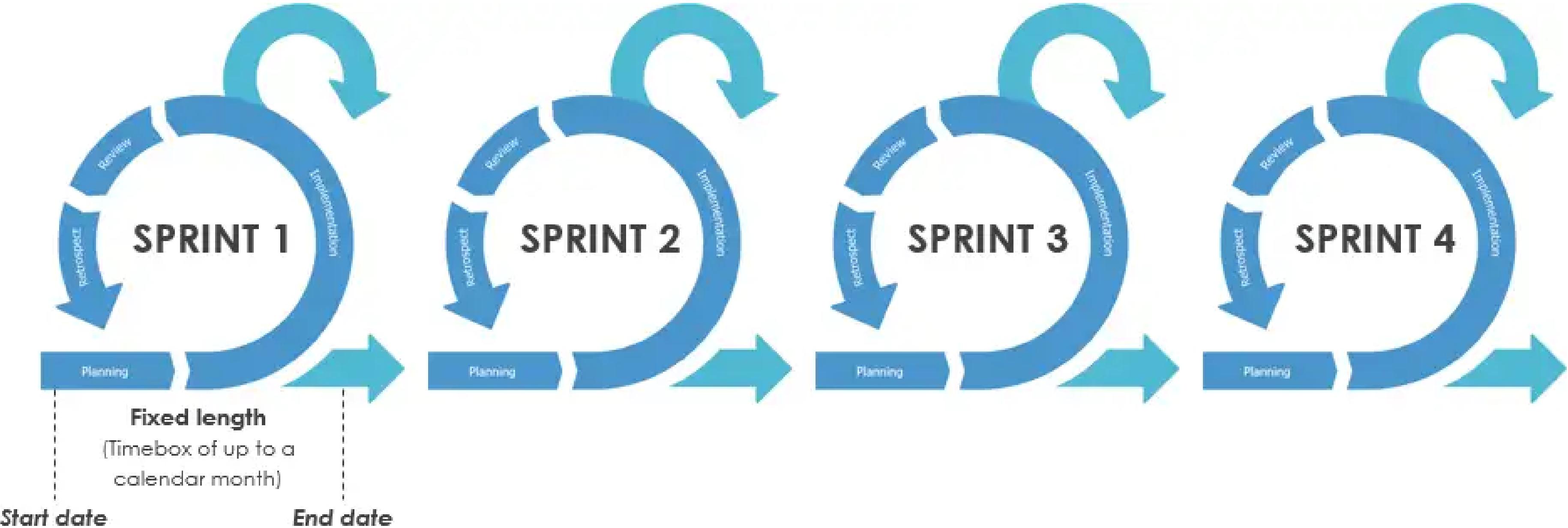
revue de sprint

# SCRUM: un processus structuré

## MÉTHODE SCRUM



# SCRUM: un processus itératif



0.1

1.0

2.0

2.0

**Ne pas confondre**

**BackLog ...**

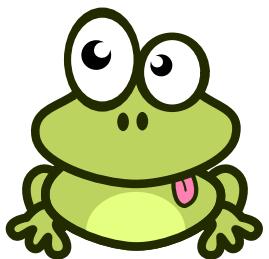
Méthodes Agiles:  
20231020

... BlackFrog

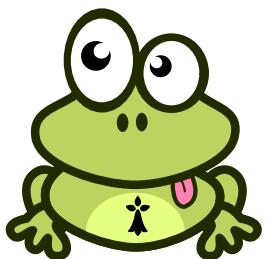


# Ateliers:

## Où en étions nous ?



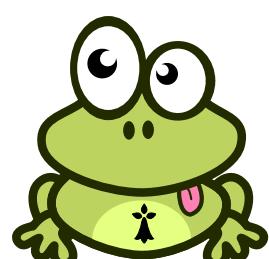
1- Recueil de la demande



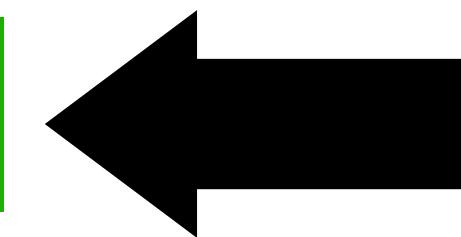
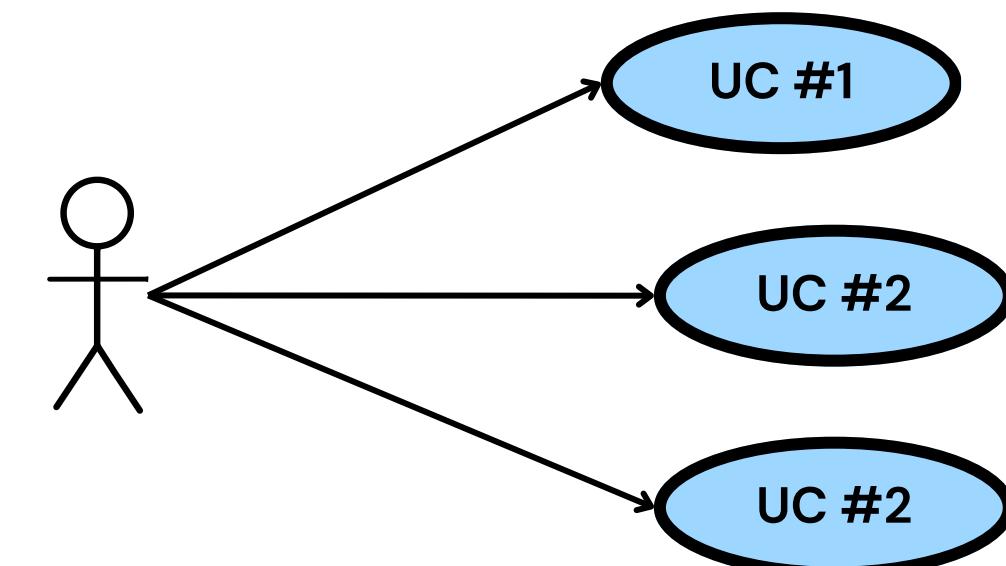
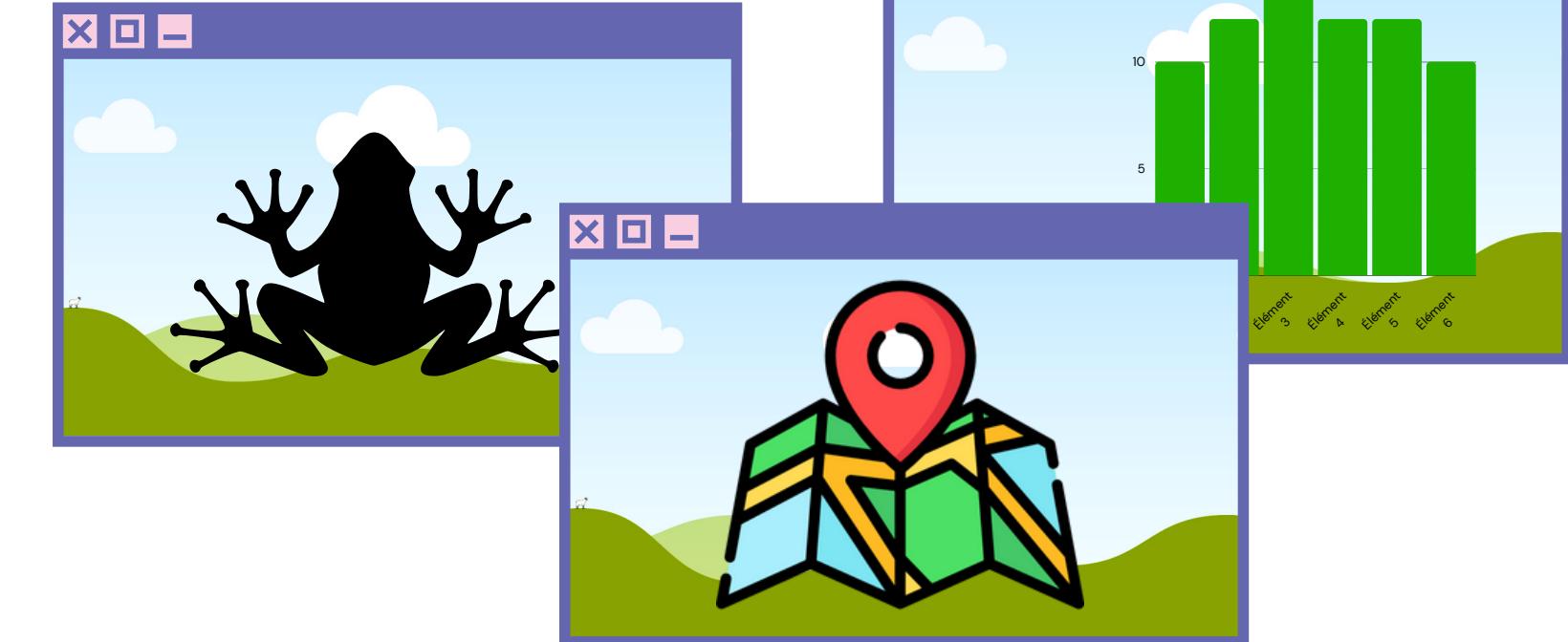
2- Analyse du Besoin



3- Liste des fonctionnalités

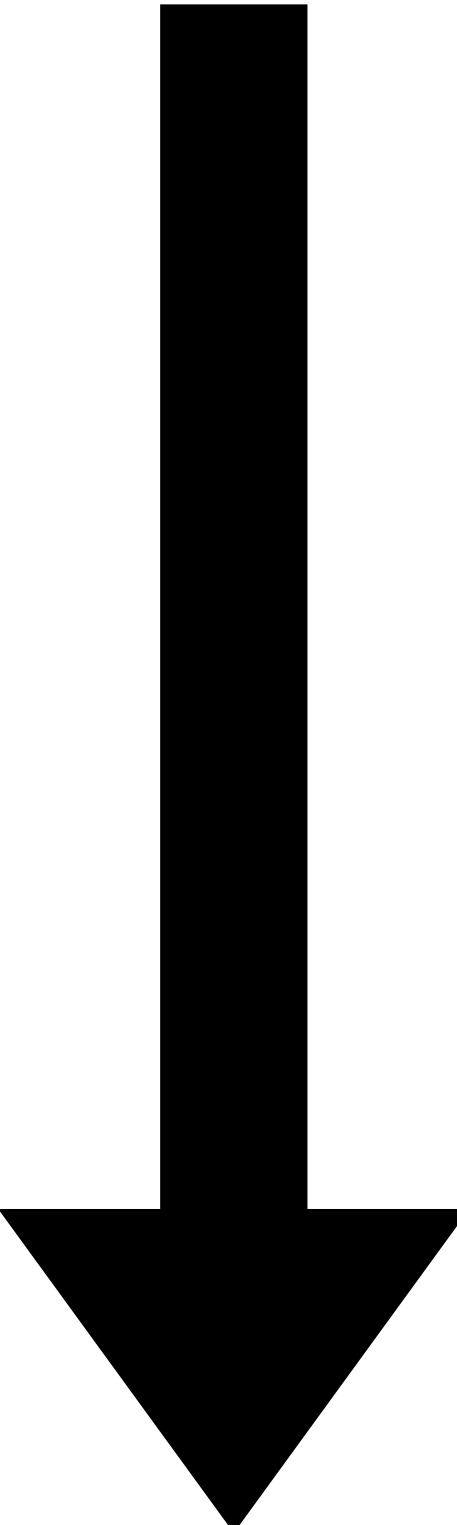


4- Architecture



BackLog ...

# PLANNIFICATION



**EPICS**

**user story VS use case**

**BackLog**

**Planning Poker**

# Raffiner la granularité d'analyse du besoin

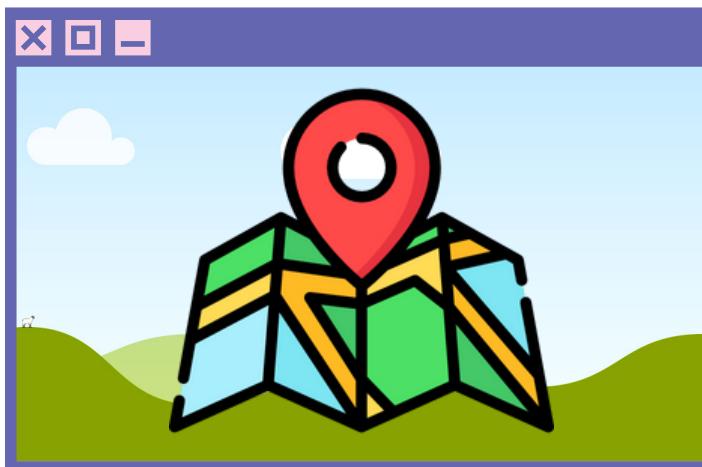


# EPICS

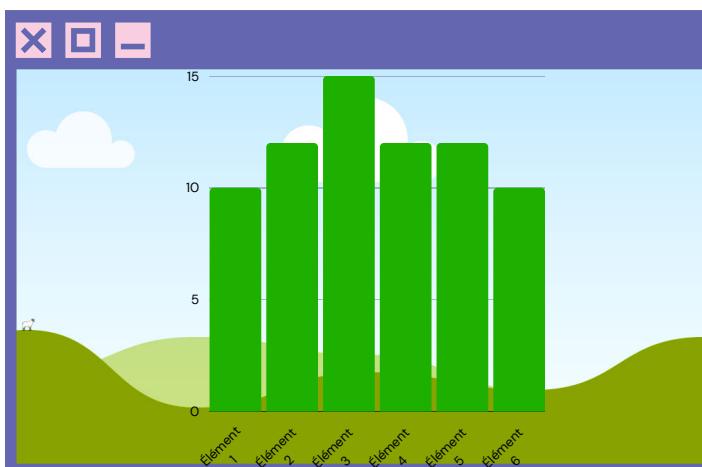


ensemble de  
tâches  
divisibles en  
“stories”

“Accéder à l’application”

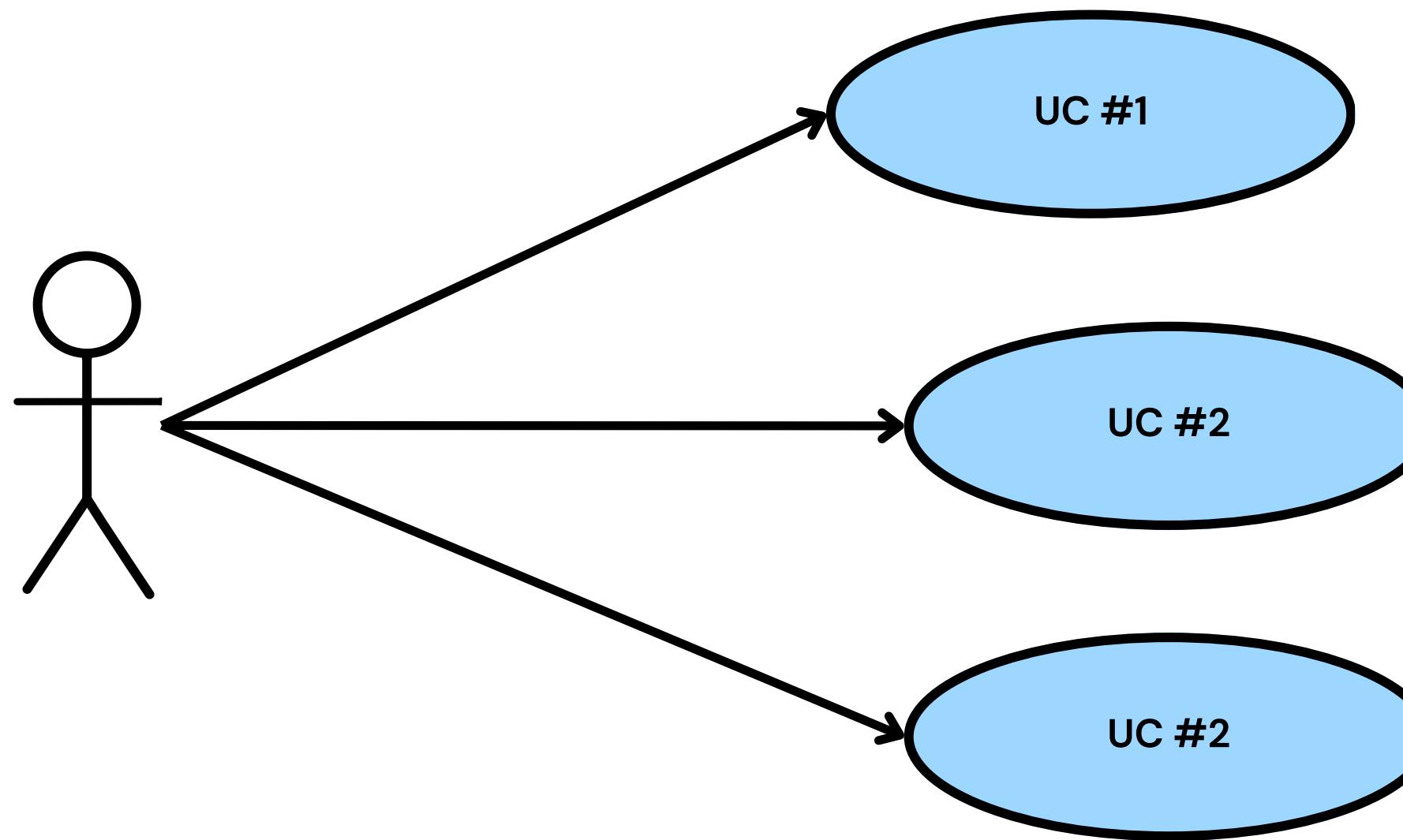


“Voir la répartition géographique”



“Récupérer les données”

# USER STORIES



Une user story est une explication d'une fonctionnalité logicielle écrite du point de vue de l'utilisateur final.

Son but est d'expliquer comment une fonctionnalité logicielle apportera de la valeur au client.

# USER STORIES comment les rédiger

“En tant que \_\_\_\_\_  
je souhaite \_\_\_\_\_  
afin de \_\_\_\_\_”

# USER STORIES comment les rédiger

***“En tant que Gabriel”:***

**qui est la personne qui agit ?**

***“Souhaite que”:***

**quelle est l'intention de Gabriel ?**

***“Afin de”:***

**Quel avantage global Gabriel essaie d'obtenir ?**

# USER STORIES comment les rédiger

**“En tant qu’administrateur  
je veux rajouter une espéce  
afin de l’offrir aux visiteurs”**

# BACKLOG PRODUIT

la liste les User Stories pour l'ensemble du projet

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

# BACKLOG SPRINT

la liste les User Stories choisie pour le SPRINT

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

"En tant qu'\_\_\_ je veux \_\_\_ afin de l'\_\_\_"

# BACKLOG

## CAHIER DES CHARGES VS BACKLOG

### CAHIER DES CHARGES

- Elaboré en début de projet
- Périmètre figé
- Difficilement maintenable
- Budget défini sur un ensemble de fonctionnalités

### BACKLOG

- Elaboré au fil du temps
- Périmètre évolutif
- Affiné régulièrement
- Budget défini sur un ensemble de ressources mobilisées