

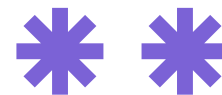
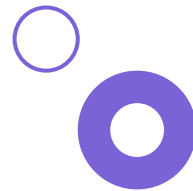


# Méthodes Agiles

Guillaume Bouyer

# Définition

La méthode Agile est une approche de gestion de projet, souvent utilisée dans le développement de logiciels, qui privilégie la flexibilité, la collaboration client, et la réactivité aux changements.


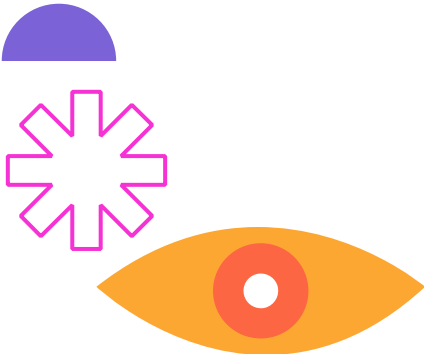




# Les 4 piliers de l'Agilité

Les piliers de l'agilité sont les principes fondamentaux qui forment la base de la méthodologie Agile. Ils sont essentiels pour comprendre et mettre en œuvre efficacement cette approche dans la gestion de projet, en particulier dans le développement de logiciels.

**Ces piliers sont soutenus par les 12 principes du Manifeste Agile**



# Les 4 piliers



**Les individus et les interactions plutôt que les processus et les outils**



**Un logiciel fonctionnel plutôt qu'une documentation complète**



**Collaboration avec les clients plutôt que négociation de contrats**



**Répondre au changement plutôt que suivre un plan**

# Les 4 piliers



## **Les individus et les interactions plutôt que les processus et les outils**

Ce pilier met l'accent sur l'importance des personnes et de leurs interactions plutôt que sur les processus rigides et les outils. Il souligne que la communication et la collaboration au sein de l'équipe sont cruciales pour le succès du projet.



## **Un logiciel fonctionnel plutôt qu'une documentation complète**

Agile privilégie un logiciel fonctionnel plutôt que de la documentation exhaustive. Bien que la documentation soit importante, le principal objectif est de fournir un logiciel opérationnel qui répond aux besoins des clients.

# Les 4 piliers



## **Collaboration avec les clients plutôt que négociation de contrats**

Ce principe valorise la collaboration avec les clients plutôt que de se reposer uniquement sur ce qui a été négocié dans les contrats. L'objectif est de travailler étroitement avec les clients pour comprendre leurs besoins et s'adapter aux changements au fur et à mesure qu'ils surviennent.

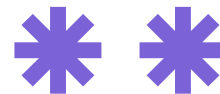
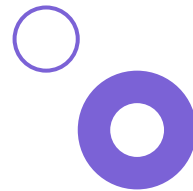


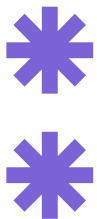
## **Répondre au changement plutôt que suivre un plan**

L'agilité est basée sur la capacité à répondre aux changements plutôt que de suivre rigideement un plan. Cela permet aux projets de s'adapter et d'évoluer en fonction des besoins changeants du marché ou des clients.

# Les 12 principes

Les 12 principes de l'agilité, issus du Manifeste Agile, sont des directives clés pour la mise en œuvre de l'approche Agile dans le développement de logiciels et la gestion de projet. Ils soulignent l'importance de la satisfaction client, la flexibilité face aux changements, la livraison continue, et la collaboration étroite entre les équipes et les clients. Ces principes favorisent également l'auto-organisation et l'amélioration continue, essentiels pour un développement réactif et efficace.





# Les 12 principes

## **Satisfaction du client par livraison rapide**

Prioriser la livraison continue de logiciels utiles pour assurer la satisfaction et le retour d'information du client

## **Livraison fréquente de logiciel fonctionnel**

Produire et livrer régulièrement des versions fonctionnelles du logiciel, de préférence dans un court laps de temps

## **Accueil des changements de besoins**

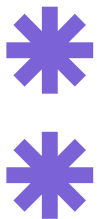
Être flexible et ouvert aux changements de besoins à tout moment du projet, utilisant ces changements pour l'avantage compétitif du client.

## **Collaboration entre les équipes d'affaires et de développement**

Encourager une collaboration quotidienne entre les professionnels des affaires et les développeurs pour une meilleure compréhension des objectifs







# Les 12 principes

## **Construction autour d'individus motivés**

Fournir un environnement de soutien et de confiance aux individus motivés, en les habilitant à mener à bien le projet

## **Mesure du progrès principal : Logiciel fonctionnel**

Considérer un logiciel fonctionnel comme la mesure principale du progrès dans un projet Agile

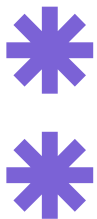
## **Communication en face à face**

Favoriser la communication directe, considérée comme le moyen le plus efficace et efficient de partager des informations au sein d'une équipe

## **Développement à un rythme soutenable**

Maintenir un rythme constant et soutenable de développement pour assurer la qualité et l'efficacité sur le long terme





# Les 12 principes

## **Attention continue à l'excellence technique et à la conception**

Se concentrer constamment sur l'excellence technique et une bonne conception pour améliorer l'agilité

## **Meilleures architectures, exigences et conceptions issues d'équipes auto-organisées**

Encourager les équipes auto-organisées à créer des solutions innovantes et efficaces

## **Simplicité**

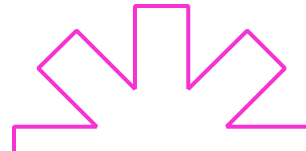
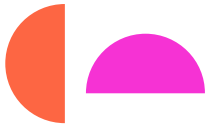
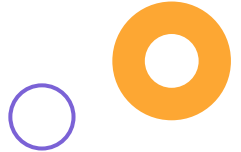
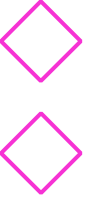
Maximiser la quantité de travail non fait en se concentrant sur la simplicité et l'essentiel

## **Réflexion et ajustements réguliers**

Procéder à des réflexions d'équipe à intervalles réguliers pour s'adapter et devenir plus efficace dans les processus

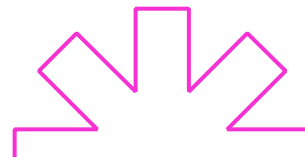


# Différentes méthodes Agiles



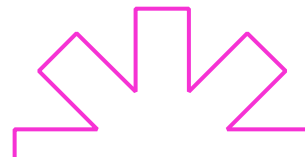
# Les méthodes

1. **Scrum** : Un cadre de travail très populaire qui se concentre sur la gestion de projet itérative, avec des rôles définis, des cérémonies (comme les sprints, les stand-up quotidiens, les revues de sprint) et des artefacts (comme le backlog de produit)
2. **Kanban** : Une méthode axée sur la visualisation du flux de travail, souvent à l'aide d'un tableau Kanban, pour optimiser le processus de production. Elle met l'accent sur la limitation du travail en cours et sur l'amélioration continue



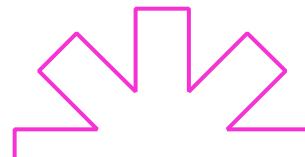
# Les méthodes

3. **Extreme Programming (XP)** : Met l'accent sur la qualité technique du logiciel, avec des pratiques comme la programmation en binôme, le développement piloté par les tests, le déploiement continu, et le design simple
4. **Lean Software Development** : Inspiré par les principes de fabrication Lean, il met l'accent sur l'élimination des gaspillages, l'amélioration de la qualité, et la délivrance rapide de valeur



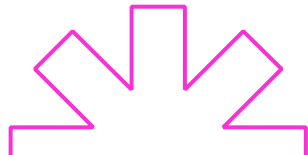
# Les méthodes

5. **Feature-Driven Development (FDD)** : Se concentre sur la conception et la construction de fonctionnalités, avec des pratiques structurées pour le développement de logiciels et une attention particulière à la modélisation et au design
6. **Dynamic Systems Development Method (DSDM)** : Une approche qui intègre des phases de projet et des rôles définis, avec un accent sur la livraison fréquente de produits et la participation active des utilisateurs



# Les méthodes


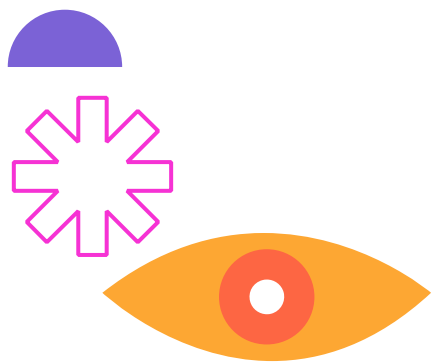
8. **Crystal** : Un ensemble de méthodologies agiles qui sont adaptées à la taille et aux besoins spécifiques de l'équipe, en mettant l'accent sur la communication et les interactions humaines
9. **Scaled Agile Framework (SAFe)** : Un cadre pour appliquer des principes agiles à grande échelle dans des entreprises et organisations de plus grande taille





# Scrum

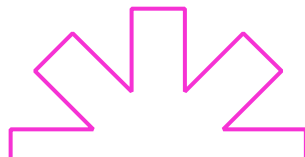
Un cadre de travail très populaire qui se concentre sur la gestion de projet itérative, avec des rôles définis, des cérémonies





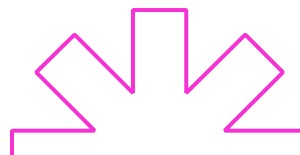
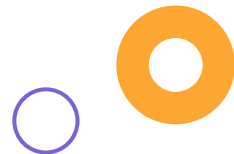
# Scrum

1. Une approche de type agile qui permet de produire la plus grande valeur dans une courte durée
2. Équipe responsable et en auto-organisation
3. Une version produite à la fin de chaque release
4. Un produit partiel testé et livré à la fin de chaque sprint
5. Un esprit de rugby plutôt qu'une course de relais



# 01

## Les rôles





# Product Owner

Représentant du client et “propriétaire” de la vision



# Product Owner

- Pilote le projet d'un point de vue métier
- Communique une vision claire
- Accepte ou rejette le produit
- Responsable des livraisons
- S'assure que l'équipe se concentre sur les items du Backlog de plus forte valeur ajoutée



# L'équipe

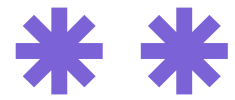
Entre 5/9 personnes - Autogérée





# L'équipe

- Délivre le produit et est responsable de sa qualité
- Travaille avec les utilisateurs finaux, le client, le Product Owner pour comprendre les exigences-métier



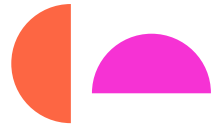
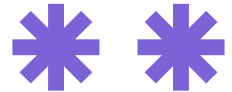
# Scrum Master

Travaille avec le Product Owner et aide l'équipe à avancer

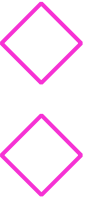
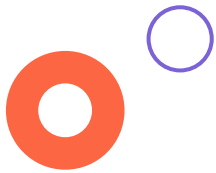


# Scrum Master

- Représente le management de l'équipe
- Élimine les obstacles
- S'assure que l'équipe est complètement fonctionnelle et productive
- Facilite une coopération poussée entre tous les rôles et fonctions







# Rappel



**Product Owner**



**Scrum Master**



**L'équipe**

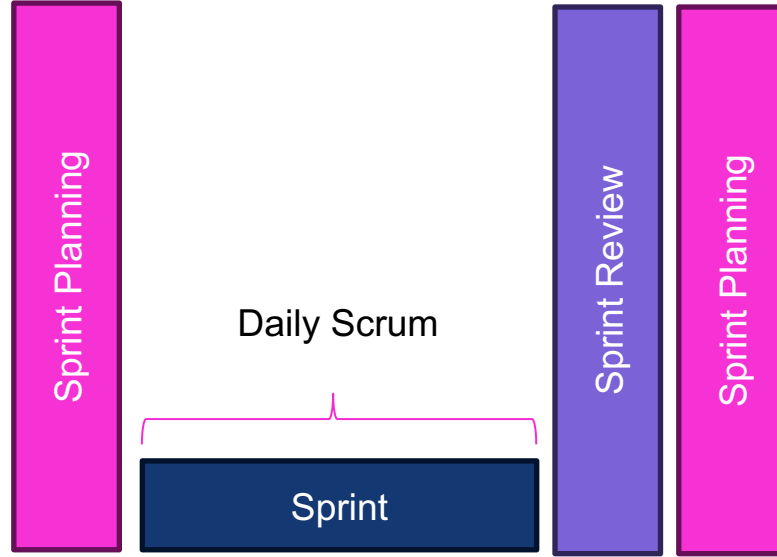


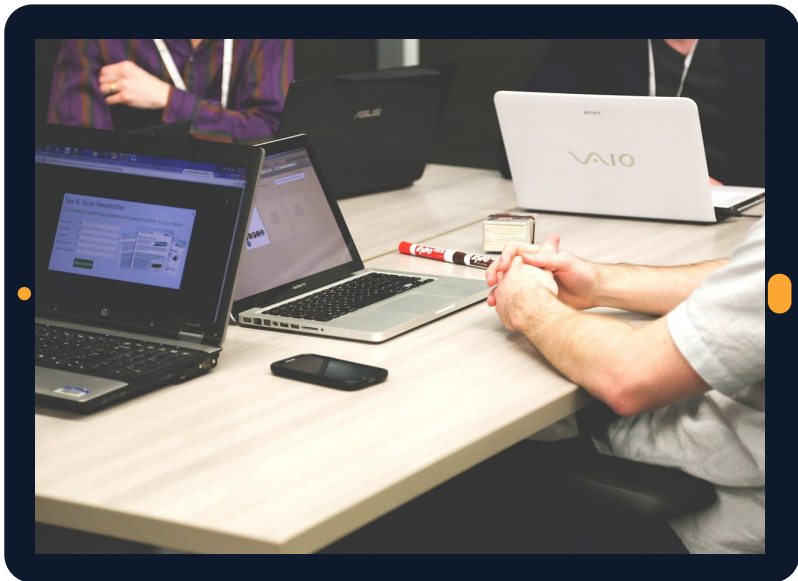


**02**

# **Les cérémonies**

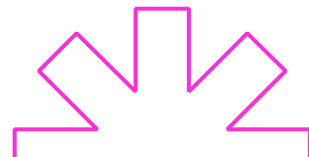






## Sprint planning

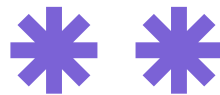
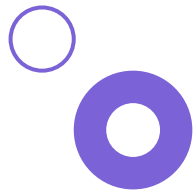
Deux heures chaque semaine dans lesquels les « stories » stocké dans le « Product Backlog » sont découpées en tâches



# Un story

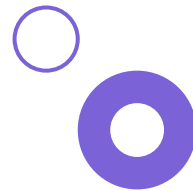
Les User Stories dans Scrum sont des outils concis et centrés sur l'utilisateur pour définir les fonctionnalités d'un produit. Elles sont formulées dans un langage simple, suivant le format 'En tant que [utilisateur], je veux [fonctionnalité] afin de [bénéfice]', ce qui facilite la compréhension et la communication des besoins. Conçues pour être suffisamment petites pour être complétées en un seul sprint, elles permettent une planification flexible et réactive.

Chaque story est accompagnée de critères d'acceptation clairs, garantissant que le travail accompli répond aux attentes. Les stories jouent un rôle clé dans la planification des sprints, aidant les équipes à prioriser le travail et à clarifier les objectifs pour tous les participants, du développeur au client



# Product Backlog

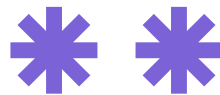
Le Product Backlog est une liste de toutes les fonctionnalités, améliorations, et corrections nécessaires pour le développement d'un produit dans Scrum. Géré par le Product Owner, il évolue constamment en fonction des besoins des clients, des retours d'expérience et des changements du marché. Le Backlog sert de base pour la planification des sprints, où les éléments les plus prioritaires sont sélectionnés pour le prochain cycle de développement. Il assure que l'équipe reste concentrée sur les tâches qui offrent la plus grande valeur au produit

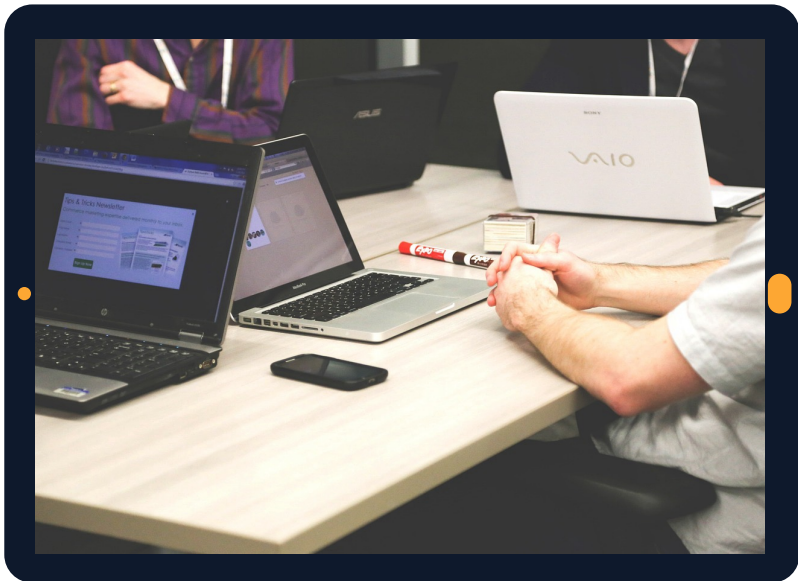




# Contenu du Product Backlog

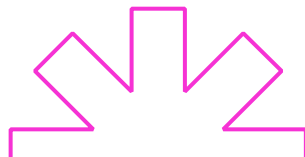
- Elaboré et partagé avec toute l'équipe
- Une liste ordonnée des « stories »
- Un story = une fonctionnalité
- Changements continuels en fonction des retours





## Sprint

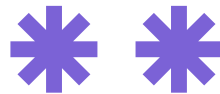
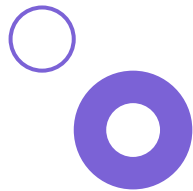
C'est le temps sur lequel les développeurs vont devoir réaliser les tâches choisies. Généralement planifié sur 2 semaines (ou plus)

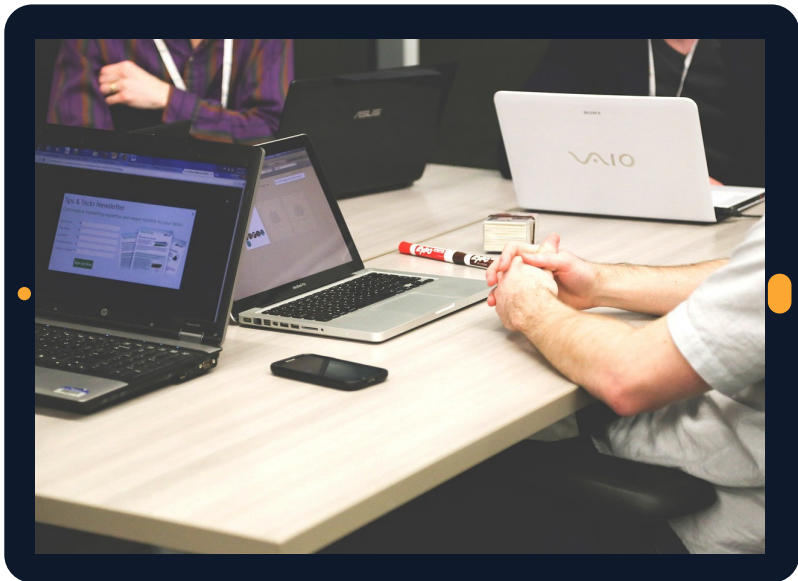




# Un sprint

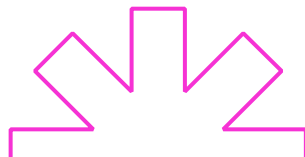
Un sprint dans Scrum est un cycle de développement délimité dans le temps, démarrant immédiatement après le sprint planning. Durant un sprint, qui dure généralement entre deux et quatre semaines, l'équipe s'efforce de compléter les tâches choisies lors du sprint planning. C'est une période durant laquelle l'équipe de développement se concentre intensément sur la réalisation des user stories sélectionnées, transformant les exigences en fonctionnalités concrètes et testables du produit. À la fin du sprint, l'équipe présente les fonctionnalités développées lors de la revue de sprint, suivie d'une rétrospective pour évaluer le processus et s'améliorer pour le prochain sprint. Ainsi, chaque sprint est une étape autonome vers l'accomplissement global du projet





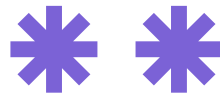
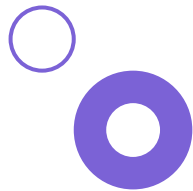
## Daily Scrum

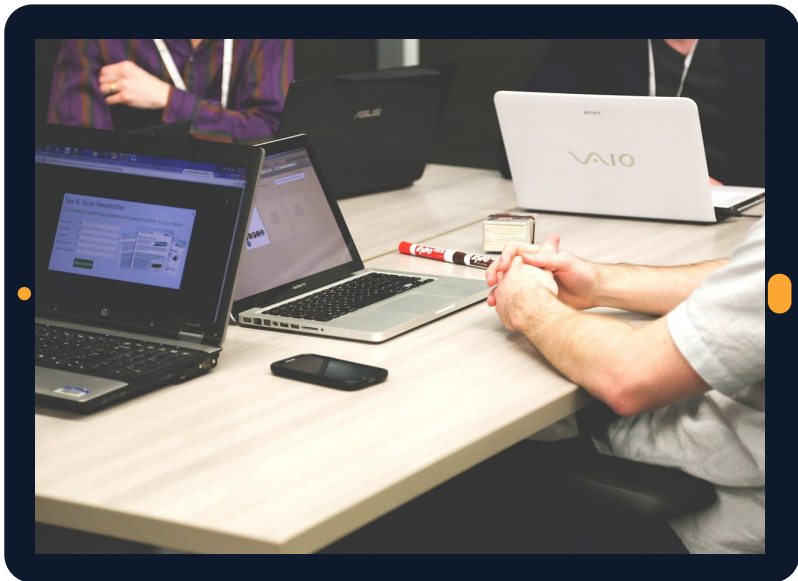
Une réunion tous les matins d'environ 10/15 minutes à chaque développeur parle de son avancement ou des blocages rencontrés.



# Daily Scrum

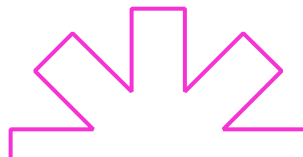
Le Daily Scrum est une réunion brève et quotidienne, généralement limitée à 15 minutes, où l'équipe de développement se synchronise sur l'avancement du sprint. Chaque membre partage ce qu'il a accompli la veille, ce qu'il prévoit de faire aujourd'hui, et évoque les éventuels obstacles rencontrés. Ce rituel favorise la transparence, permet une adaptation rapide et assure que l'équipe reste focalisée sur les objectifs du sprint. C'est une composante essentielle de Scrum, contribuant à maintenir le rythme et l'alignement de l'équipe tout au long du projet.





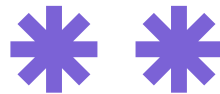
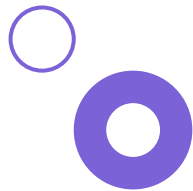
## Sprint Review

A la fin de chaque sprint où l'équipe présente son travail.

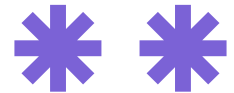
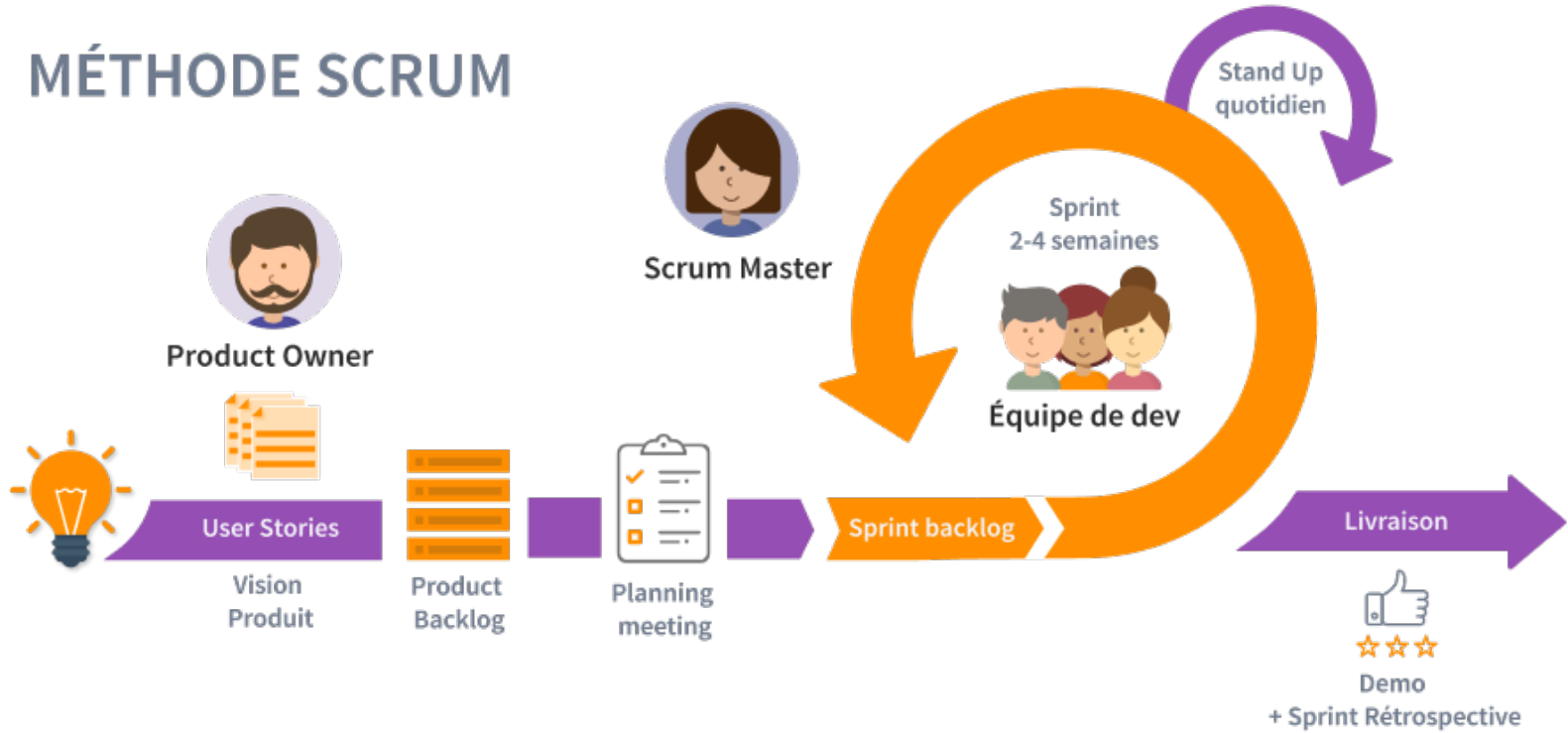


# Sprint Review

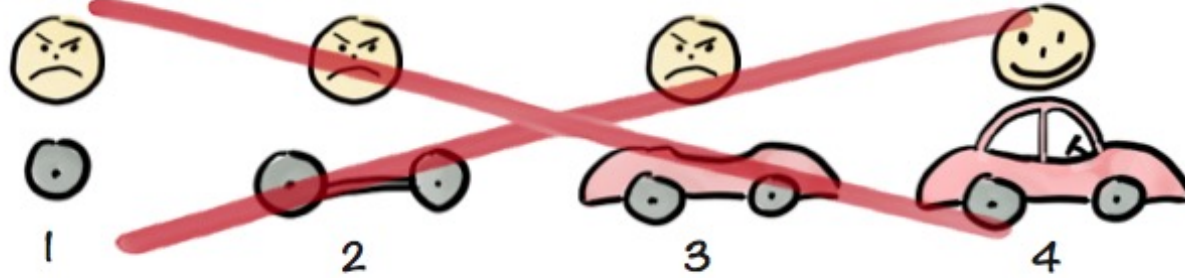
Le Sprint Review est une réunion tenue à la fin de chaque sprint dans Scrum, où l'équipe présente ce qu'elle a accompli pendant le sprint. Cette rencontre, d'une durée d'une heure par semaine de sprint, permet aux parties prenantes de voir les progrès réalisés et de donner leur retour. Le Product Owner révisé le backlog produit en fonction de ce retour et des changements du marché, permettant ainsi une adaptation continue du produit. Le Sprint Review est crucial pour assurer l'alignement entre les besoins des utilisateurs et la direction du produit



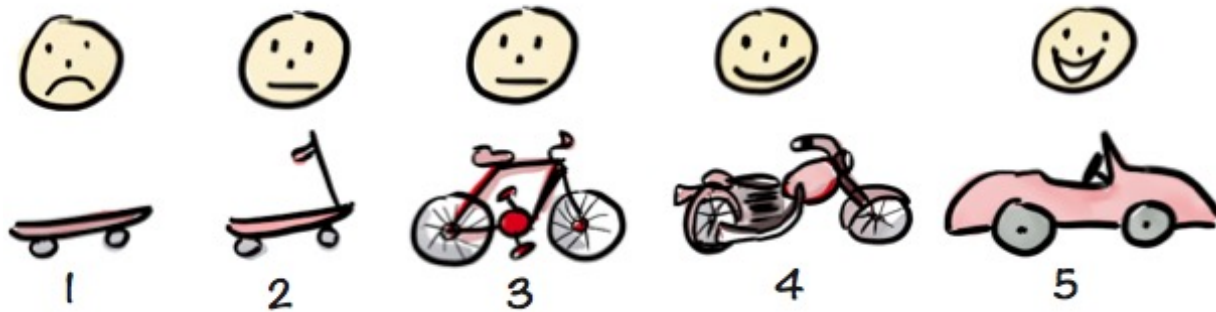
# MÉTHODE SCRUM



Not like this....




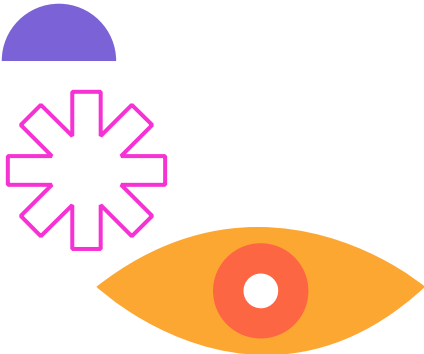
Like this!





# Kanban

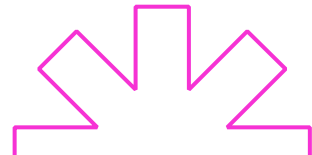
Optimiser le flux de travail et à améliorer la  
continuité et l'efficacité des processus. En  
intégrant les concepts de Work In Progress (WIP)  
et de sa limitation

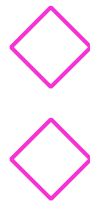




# Kaban

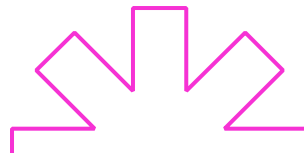
1. Visualisation du travail
2. Limitation du WIP "Work In Progress" (Travail en Cours)
3. Gestion du flux de travail
4. Amélioration continue

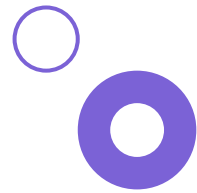




**01**

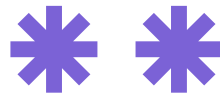
# **Visualisation du travail**





# Visualisation du travail

La visualisation du travail est un aspect fondamental de la méthode Kanban, car elle permet de voir clairement l'avancement des tâches et d'identifier rapidement les problèmes potentiels dans le flux de travail.



# Détails sur la Visualisation du Travail dans Kanban

- **Tableau Kanban :**

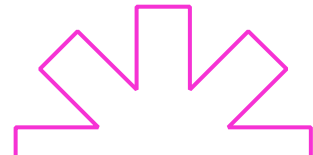
Le tableau Kanban est traditionnellement un tableau physique avec des colonnes et des cartes, mais il peut aussi être numérique.

Chaque colonne représente une étape différente du processus (par exemple, "À faire", "En cours", "En revue", "Terminé").

- **Cartes Kanban :**

Chaque tâche ou élément de travail est représenté par une carte.

Les cartes se déplacent d'une colonne à l'autre au fur et à mesure que le travail progresse.



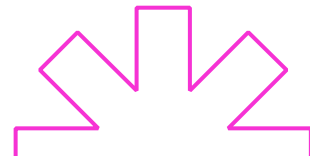
# Détails sur la Visualisation du Travail dans Kanban

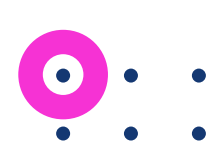
- **Informations sur les Cartes :**

Les cartes peuvent inclure des détails tels que le titre de la tâche, la description, la personne responsable, les délais, les priorités...

- **Mise à jour et Suivi :**

Le tableau est régulièrement mis à jour pour refléter l'état actuel du travail. Cela offre une vue d'ensemble instantanée de l'état d'avancement du projet.





# Outils

## Trello

Populaire et  
personnalisable

## Jira

Gestion de projet et  
gestion de grandes équipes

## Asana

Tâches et  
sous-tâches

## Kanban Tool

Dédié à la méthode  
Kanban

## Monday.com

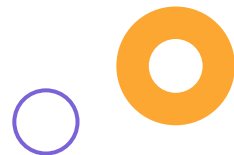
Gestion de projet  
flexible

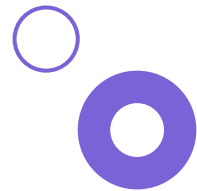




02

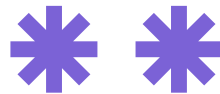
# Limitation du Travail en Cours (WIP)



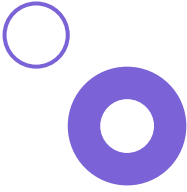


# Limitation du Travail en Cours (WIP)

La limitation du travail en cours (WIP) est un principe essentiel de la méthode Kanban, qui joue un rôle crucial dans la gestion efficace du flux de travail et la prévention des goulots d'étranglement.

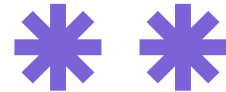






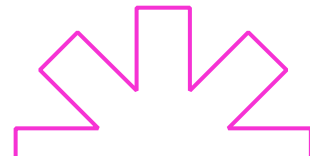
# WIP ?

Le terme "WIP" signifie "Work In Progress" (Travail en Cours). Dans le contexte de la méthode Kanban et d'autres méthodologies de gestion de projet Agile, le WIP fait référence à la quantité de travail qui est actuellement en cours de réalisation à un moment donné.



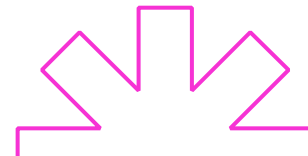
# Détails

- **Objectif de la Limitation du WIP :**
  - Réduire le surmenage et améliorer la concentration sur les tâches en cours. Cela augmente l'efficacité et accélère le temps de réalisation.
- **Définition des Limites de WIP :**
  - Chaque colonne du tableau Kanban se voit attribuer une limite maximale de tâches pouvant y être travaillées simultanément.
  - Les limites doivent être adaptées à la capacité de l'équipe et peuvent être ajustées au fil du temps.



# Détails

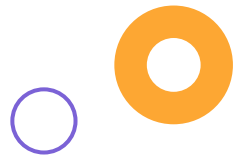
- **Avantages de la Limitation du WIP :**
  - Elle aide à identifier et à résoudre les goulots d'étranglement.
  - Elle favorise un flux de travail plus régulier et prévisible.
  - Elle réduit le temps de cycle global des tâches.
- **Gestion des Dépassements de WIP :**
  - Lorsqu'une limite de WIP est atteinte, l'équipe se concentre sur la finition des tâches en cours avant d'en démarrer de nouvelles.

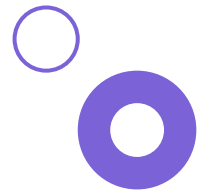




**03**

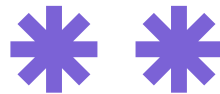
# **Gestion du Flux de Travail**





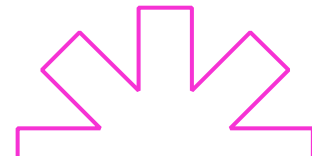
# Gestion du Flux de Travail

La gestion du flux de travail est un aspect crucial de la méthode Kanban, centré sur l'optimisation et le contrôle efficace du déroulement des tâches à travers les différentes étapes du processus.



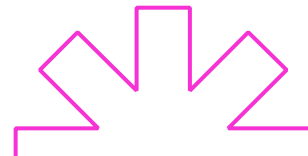
# Détails

- **Surveillance du Flux :**
  - Le suivi constant du flux de travail permet d'identifier où les tâches sont en cours, en attente, ou achevées. Cela aide à détecter les goulots d'étranglement et les retards.
- **Fluidité du Travail :**
  - L'objectif est de maintenir un mouvement régulier des tâches à travers le tableau Kanban, évitant ainsi les accumulations et les retards inutiles.



# Détails

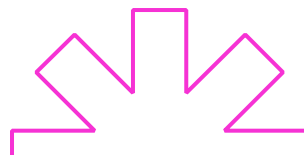
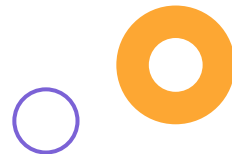
- **Équilibrage de la Charge de Travail :**
  - Répartir de manière équilibrée les tâches entre les membres de l'équipe pour maintenir un flux constant et éviter la surcharge de travail.
- **Adaptation et Flexibilité :**
  - Être prêt à réagir et à s'adapter aux changements, que ce soit en ajustant les limites de WIP, en redistribuant les tâches, ou en modifiant les priorités.
- **Utilisation de Métriques :**
  - Des métriques telles que le temps de cycle et le temps de latence sont utilisées pour mesurer et analyser l'efficacité du flux de travail, permettant des ajustements basés sur des données concrètes.



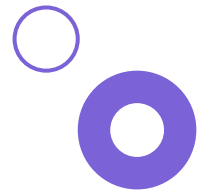


# 03

## Amélioration Continue

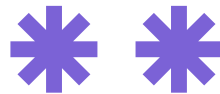






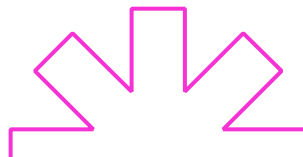
# Amélioration Continue

L'amélioration continue est un principe fondamental de la méthode Kanban, visant à augmenter régulièrement l'efficacité et la qualité du flux de travail. Cette approche repose sur l'évaluation constante et l'optimisation des processus.



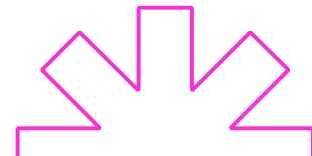
# Détails

- **Évaluation Régulière :**
  - L'équipe évalue régulièrement les performances et l'efficacité du processus de travail. Cela peut se faire lors de réunions régulières ou de rétrospectives.
- **Identification des Goulots d'Étranglement :**
  - Analyser le flux de travail pour identifier les goulots d'étranglement, les retards et les points d'amélioration.
- **Expérimentation et Ajustement :**
  - Tester de nouvelles stratégies ou changements dans le processus pour résoudre les problèmes identifiés. Cela peut inclure l'ajustement des limites de WIP, la modification des étapes du processus ou l'introduction de nouvelles pratiques.



# Détails

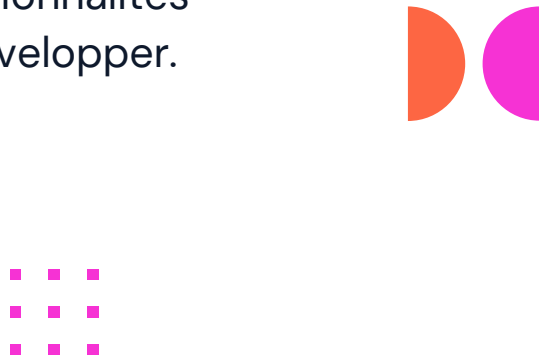
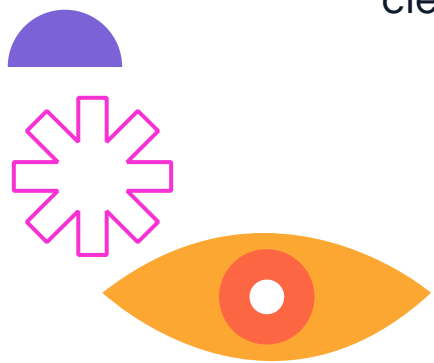
- **Mesure des Changements :**
  - Utiliser des métriques pour mesurer l'impact des changements introduits. Cela aide à déterminer si les modifications apportées améliorent le processus.
- **Culture de la Rétroaction :**
  - Encourager une culture de feedback ouvert au sein de l'équipe, où les suggestions et les idées pour améliorer le processus sont les bienvenues.





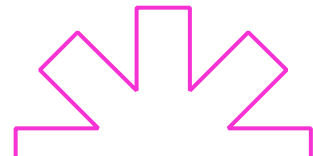
# Exemple

L'équipe travaille sur une application mobile destinée à faciliter la gestion des finances personnelles. Le projet est à un stade intermédiaire, avec plusieurs fonctionnalités clés déjà en place, mais d'autres encore à développer.



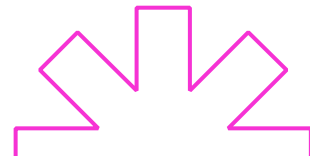
# Utilisation de Kanban

- **Visualisation du Travail :**
  - L'équipe utilise un outil en ligne comme Trello pour créer un tableau Kanban. Le tableau est divisé en colonnes : "À faire", "En cours", "En revue", et "Terminé".
  - Chaque fonctionnalité ou tâche à développer est représentée par une carte sur le tableau.
- **Limitation du WIP :**
  - Des limites de WIP sont établies pour chaque colonne. Par exemple, la colonne "En cours" a une limite de 3 tâches à la fois.
  - Cela permet de s'assurer que l'équipe ne se surcharge pas et se concentre sur la finition des tâches en cours.



# Utilisation de Kanban

- **Gestion du Flux de Travail :**
  - Au quotidien, l'équipe examine le tableau pour suivre le progrès des tâches.
  - Si une colonne atteint sa limite de WIP, l'équipe se concentre sur la complétion des tâches en cours avant d'en commencer de nouvelles.
- **Amélioration Continue :**
  - L'équipe tient des réunions hebdomadaires pour discuter des progrès et identifier les goulots d'étranglement ou les retards.
  - Les membres de l'équipe proposent des améliorations au processus, comme ajuster les limites de WIP ou réorganiser les priorités dans le backlog.



# Soucis rencontrés

À un moment donné, l'équipe remarque que les tâches dans la colonne "En revue" s'accumulent, ce qui indique un goulot d'étranglement. En réponse, l'équipe décide de réduire la limite de WIP pour la colonne "En cours" et d'allouer plus de ressources à la revue et aux tests. Cette adaptation permet de résoudre le problème de congestion et d'améliorer le flux global de travail. En parallèle, l'équipe continue de recueillir des retours d'utilisateurs sur les fonctionnalités déjà lancées, intégrant ces informations dans le backlog pour de futures itérations.

