



Cursus FRONT

Introduction aux méthodes agiles

M2I Formation 2022

Glodie Tshimini



PLAN

Glodie Tshimini : contact@tshimini.fr

PLAN

- I. Les méthodes traditionnelles
- II. Les méthodes agiles
- III. SCRUM

Annexe

[SCRUM GUIDE 2020](#)



Les méthodes traditionnelles



Les méthodes **traditionnelles**

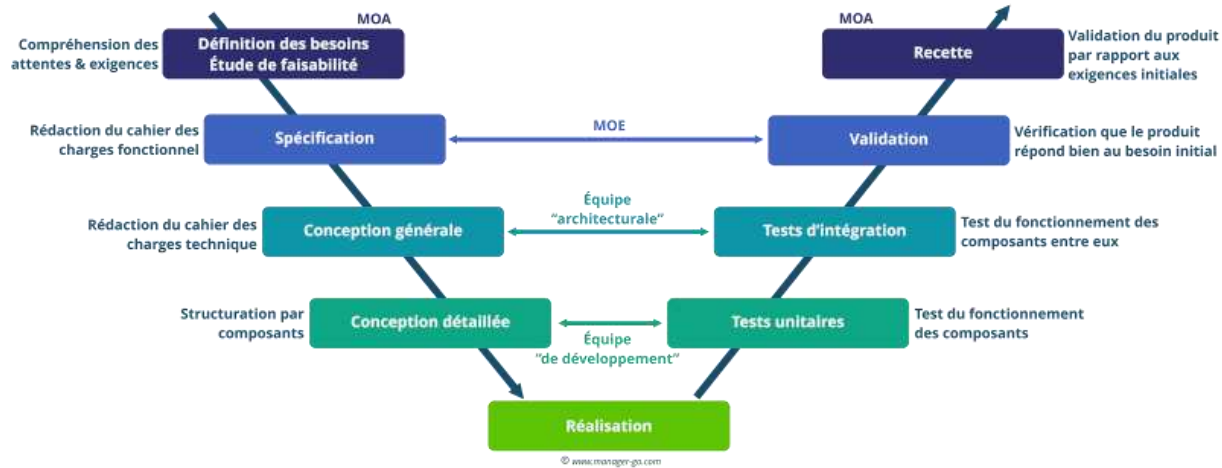
- ▶ Trouve ses origines dans le bâtiment
- ▶ Séquentielle
- ▶ Réaliser les besoins spécifiés dans un cahier des charges
- ▶ Planification rigoureuse dans un diagramme de Gantt



Les méthodes **traditionnelles**

- ▶ Très peu d'interaction avec le client
- ▶ Peu flexible
- ▶ Peu adaptabilité

Le cycle en V





II.

Les méthodes agiles





4 valeurs agiles (Manifeste Agile)

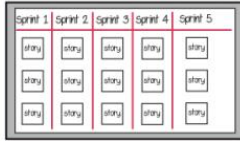
- ▶ Les **individus** et les **interactions** plus que les processus et les outils.
- ▶ Des **logiciels opérationnels** plus qu'une documentation exhaustive.
- ▶ La **collaboration avec les clients** plus que la négociation contractuelle.
- ▶ **L'adaptation au changement** plus que le suivi d'un plan.
- ▶ Nous reconnaissons la valeur des seconds éléments, mais **privilégions les premiers**.


12 principes agiles


Knowledge
TRAIN
Les 12 principes Agile*


1 Satisfaire le client



2 Accueillir les changements


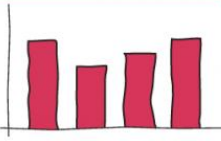
3 Livrer fréquemment



4 Travailler ensemble



5 Confiance et soutien



6 Dialogue en face à face



7 Logiciel opérationnel


8 Développement soutenable


9 Attention continue


10 Privilégier la simplicité


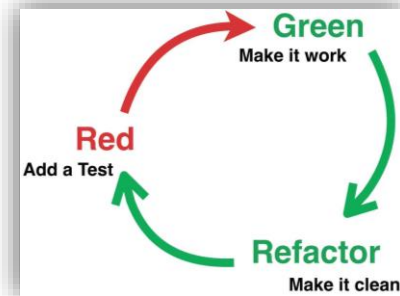
11 Équipes auto-organisées


12 Réfléchir et adapter


Copyright © 2016 Knowledge Train Limited. www.knowledgetrain.co.uk. *Quoted from www.agilemanifesto.org

Autres principes liés au **développement**

- ▶ *BDD* (Behaviour Driven Development)
 - ▶ Développement piloté par les comportements des utilisateurs
 - ▶ Tout le monde participe au recueil et à la définition du besoin
 - ▶ Tout le monde converge vers le vrai besoin à partir des scénarios et des exemples
- ▶ *TDD* (Test Driven Development)
 - ▶ Développement piloté par les tests
 - ▶ Code plus robuste
 - ▶ Meilleur qualité
 - ▶ Efficace (strict minimum)



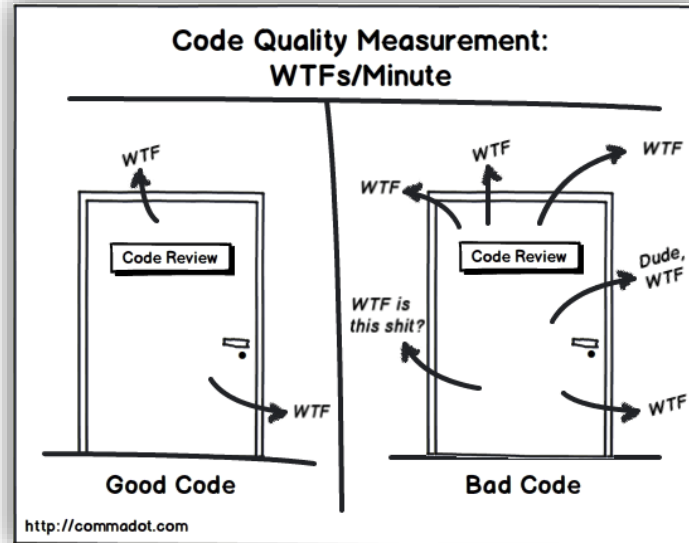
[Source de l'image infoQ](#)



Autres principes liés au **développement**

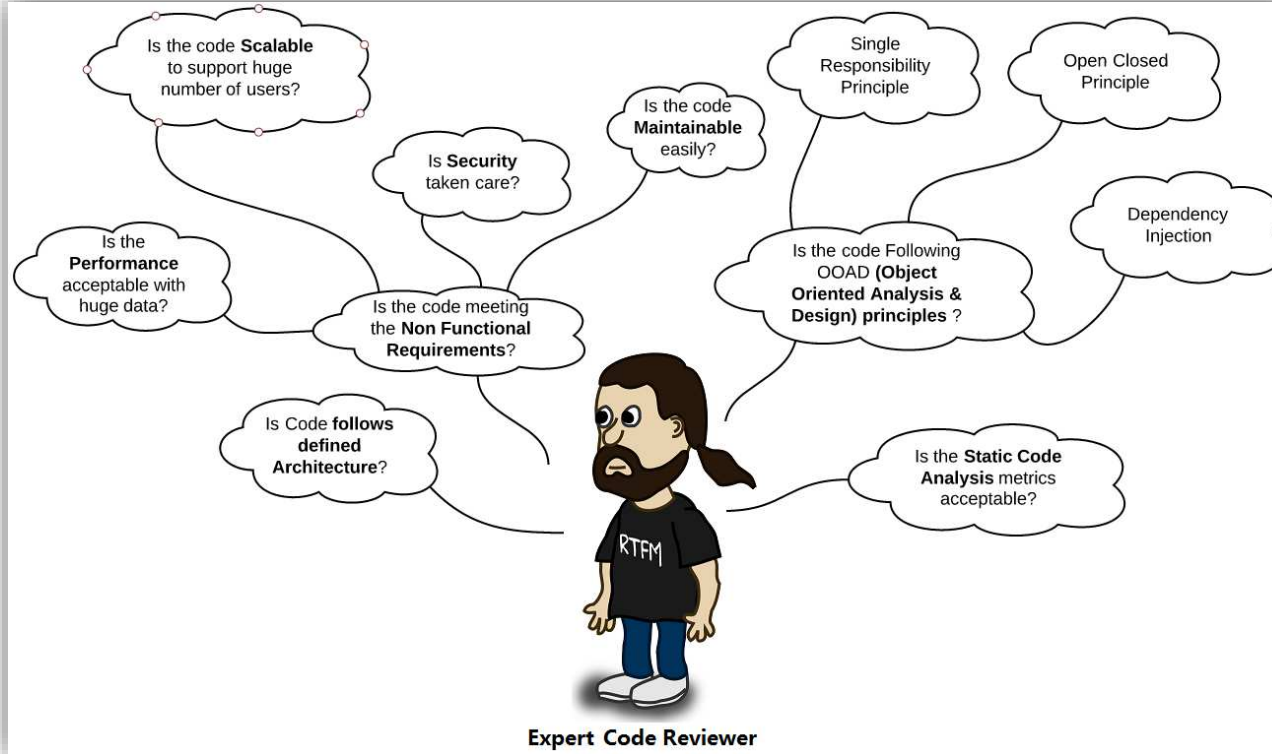
- ▶ *CI/CD* (Continuous Integration / Continuous Delivery)
 - Automatisation des livraisons en production
- ▶ *Pair programming*
 - Développement en binôme
- ▶ *Code review*
 - Revue de code par l'équipe

Grandes lignes du **Code review**



[Source de l'image commadot](http://commadot.com)

Grandes lignes du **Code review**



- Framework
- Travail en flux continu
- Facile à mettre en place



[source image Ionis](#)

Exercice

Glodie Tshimini : contact@tshimini.fr



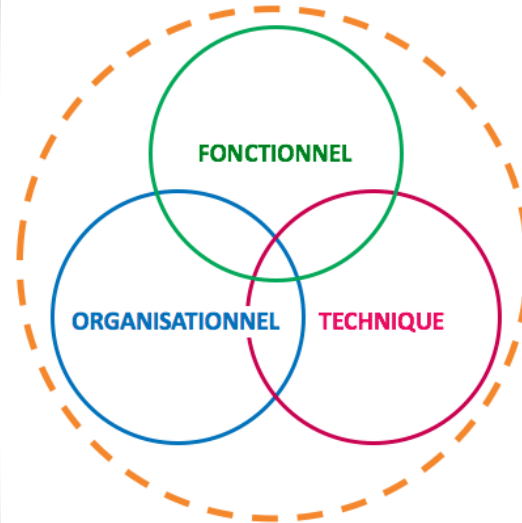
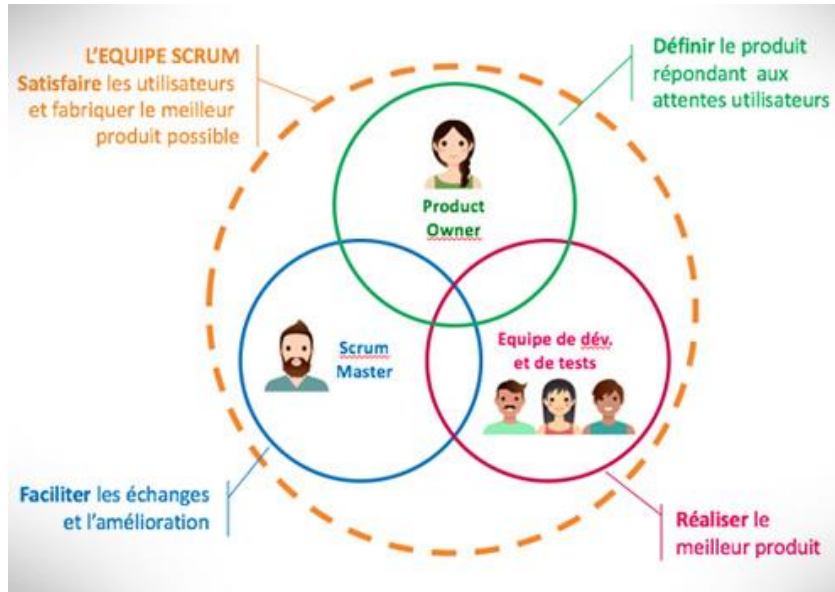
Exercice 1 : avantages & inconvénients méthodes **traditionnelles et agiles**

- Depuis le dépôt local 1-exercices/ex1-methodes.md



SCRUM

Equipe **SCRUM**



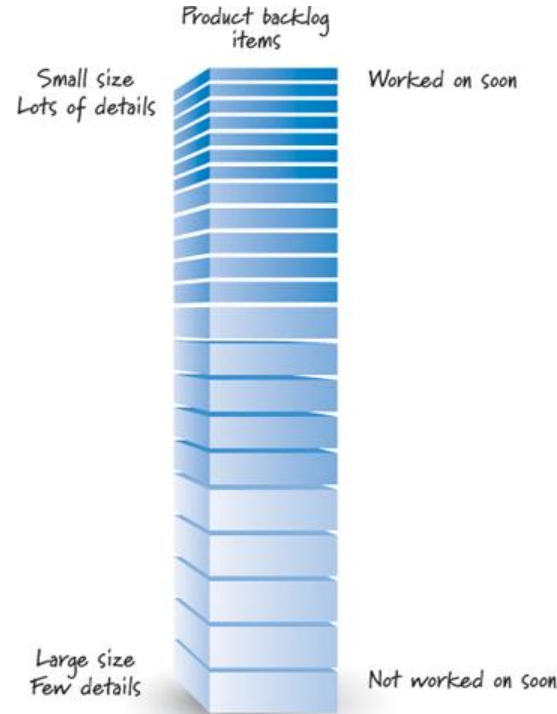


Equipe **SCRUM**

- ▶ **Product Owner**
 - ▷ Représentant du client
 - ▷ Collecte et priorise les besoins
 - ▷ Valide les livrables
 - ▷ Garant du produit et de sa valeur
- ▶ **SCRUM Master**
 - ▷ Au service de l'équipe
 - ▷ Élimine les obstacles
 - ▷ Aide à trouver des solutions
 - ▷ Garant de l'application des pratiques *SCRUM*
- ▶ **Développeurs**
 - ▷ Auto-gestion dans la réalisation des tâches
 - ▷ Livraison régulière des fonctionnalités
 - ▷ Garant de la qualité des livrables

Product Backlog

- ▶ Carnet de produit
- ▶ Ensemble des *User Stories* (éléments) qui permettent de satisfaire le *Product Goal* (objectif de valeur plus large)
 - ▶ Liste des *US* à développer pour améliorer le produit ou le service





User Story

- Découpage du besoin
- Phrase simple pour décrire le besoin d'un utilisateur
 - **Qui** : contexte
 - **Quoi** : fonctionnalité
 - **Pourquoi** : valeur
- **En tant que** *client*, **je souhaite** *contacter un conseiller* **afin d'avoir** *des informations supplémentaires* sur le produit.
- **En tant que** *préparateur de commande*, **je souhaite** *avoir la liste des articles des commandes à traiter triées par leur emplacement dans l'entrepôt* **afin** *d'optimiser les délais de traitement.*

Exercice

Glodie Tshimini : contact@tshimini.fr



Exercice 2 : **backlog**

- Depuis le dépôt local 1-exercices/ex2-backlog.md



Definition Of Ready (**DOR**)

- ▶ En équipe
 - ▷ Définir les critères d'acceptation d'un *US* pour être implémenté par les devs lors d'un *Sprint*.
- ▶ Grille **INVEST**
 - ▷ **Independant** : indépendance technique entre les *US*
 - ▷ **Negociable** : Choix de la solution
 - ▷ **Valuable** : Apporte de la valeur
 - ▷ **Estimable** : Estimable à l'aide d'un outil ou méthode d'estimation
 - ▷ **Small** : Suffisamment petit pour être réalisé lors d'un *Sprint*
 - ▷ **Testable** : Tests techniques et fonctionnels



Definition Of Done (DOD)

- ▶ En équipe
 - ▷ Définir une checklist permettant de vérifier que le développement d'un *US* est terminé
 - ▷ Niveau d'exigence de la qualité du développement

- ▶ *Exemples*
 - ▷ *Tests unitaires*
 - ▷ *Revue de code*
 - ▷ *Critères acceptation*
 - ▷ *Documentation*
 - ▷ *Mise en préproduction*
 - ▷ *Validé par le Product Owner*



Backlog Refinement

- ▶ Avant ou pendant le *Sprint Planning*
- ▶ Objectifs :
 - Affiner les *US* (ajouter des détails)
 - Apporter éventuellement des précisions techniques sans donner la solution
 - L'équipe décidera de la solution à implémenter lors du *Sprint*
- ▶ Formule affinage d'un *US*
 - *En tant que {Utilisateur}*
 - *Je souhaite {besoin/objectif}*
 - *Afin de {résultat}*



Planning **Poker**

- ▶ Estimer la complexité d'un *US*
 - ▶ Consiste à mesurer
 - ▶ *la complexité*
 - ▶ *L'incertitude*
 - ▶ *Risque*
 - ▶ *Périmètre*
 - ▶ *Effort*
- ▶ **Il ne s'agit pas de déterminer le temps de réalisation de l' *US***
- ▶ Utilisation des cartes utilisant la suite de ***Fibonacci***
 - ▶ 1, 2, 3, 5, 8, 13, 21 etc.
- ▶ ***Storypoint*** = carte de complexité attribué à l' *US*

Exercices

Glodie Tshimini : contact@tshimini.fr



Exercice 3 : **backlog refinement**

- Depuis le dépôt local 1-exercices/ex3-backlog-refinement.md



Exercice 4 : **planning poker**

- Depuis le dépôt local `1-exercices/ex4-planning-poker.md`

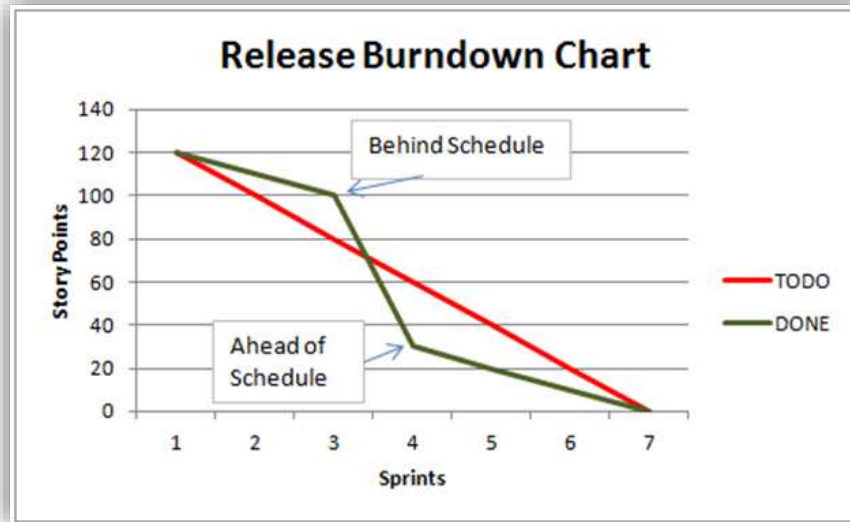


Sprint Planning

- ▶ Définir de l'objectif de *sprint*
- ▶ Sélectionner les *US* à réaliser pour le *sprint* en cours
- ▶ ***Sprint Backlog = Objectif sprint + US sélectionnés + plan d'action***
- ▶ L'équipe peut inviter des personnes en dehors de la Team pour avoir leurs expertises, etc.
- ▶ Autres opérations effectuées durant cette réunion
 - ▶ ***Backlog Refinement***
 - ▶ ***Backlog grooming*** (nettoyage du *Product Backlog*)
 - ▶ ***Planning Poker***

Burndown Chart

- ▶ Destiné à l'équipe
- ▶ Permet de voir le reste à faire
 - ▶ Total des *US* à réaliser durant le *Sprint* – *US* terminés





Sprint

- ▶ Développer (réalisation technique) des *US du **Sprint Backlog***
- ▶ Objectif avoir un incrément opérationnel (livrable qui tend vers le produit final)
- ▶ **On ne livre pas un incrément qui n'est pas fini**
- ▶ **Événement conteneur**
 - ▶ **Inclut tous les autres événements**

Exercice

Glodie Tshimini : contact@tshimini.fr



Exercice 5 : **sprint planning**

- Depuis le dépôt local 1-exercices/ex5-sprint-planning.md



Daily SCRUM

- ▶ **Inspecter** les travaux en cours
- ▶ **Focus** sur *l'Objectif de Sprint*
- ▶ Rendez-vous quotidienne de 15 min
 - ▶ Même lieu et même horaire
- ▶ Obligatoire pour les développeurs
- ▶ Limite la nécessité d'avoir recours à d'autres réunions
- ▶ **S'assurer que tout le monde est en phase avec l'objectif du *sprint* en cours**
- ▶ Répondre aux questions suivantes (n'est plus obligatoire)
 - ▶ Qu'est-ce que j'ai fait hier ?
 - ▶ Qu'est-ce que j'ai prévu de faire aujourd'hui ?
 - ▶ Soulever les éventuels points de blocage ou problèmes ?



Sprint Review

- ▶ Bilan du *Sprint*
- ▶ Tout le monde est le bienvenue
 - ▶ Équipe *SCRUM*
 - ▶ Parties prenantes du projet
- ▶ Présenter l'objectif du *Sprint* et le(s) livrable(s)
- ▶ On peut livrer avant l'événement (*US* validé par le *PO* et l'équipe avec le *DoD*)
- ▶ Avoir le feedback des parties prenantes sur le travail réalisé
 - ▶ Intégrer les feedbacks pour le prochain ou les *sprints* suivants



Sprint Retrospective

- ▶ Bilan humain du *Sprint* précédent
- ▶ Objectif : améliorer les conditions de travail pour le prochain *Sprint*
- ▶ Outils, DoD, interactions, processus, individus etc. passent en revue
- ▶ Echanger sur
 - ▶ Qu'est-ce qui a bien fonctionné ? (continuer à utiliser)
 - ▶ Qu'est-ce qui a moins bien fonctionné ? (arrêté ou revoir l'utilisation)
 - ▶ Qu'est-ce qui peut être amélioré ou être intégré ? (en faire plus ou nouvelles idées)

Exercices

Glodie Tshimini : contact@tshimini.fr



Exercice 6 : **sprint review**

- Depuis le dépôt local 1-exercices/ex6-sprint-review.md

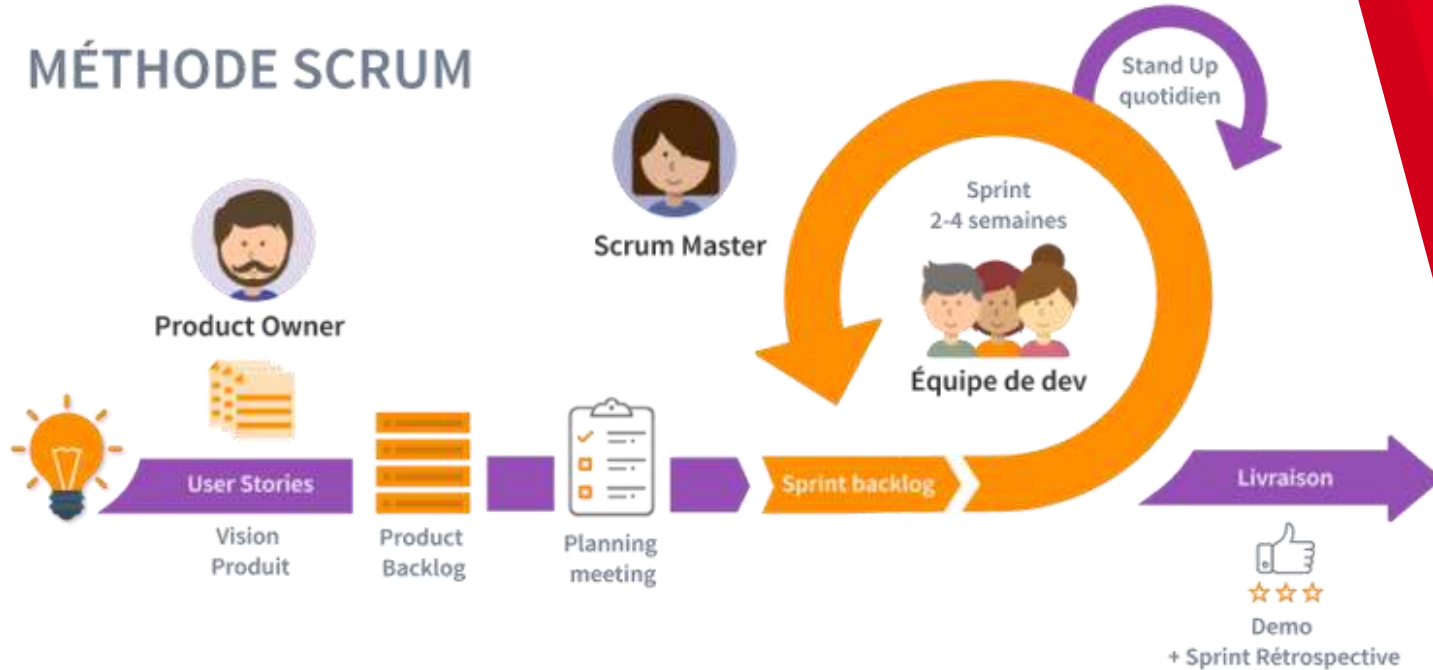


Exercice 7 : **sprint retrospective**

- Depuis le dépôt local 1-exercices/ex7-sprint-retrospective.md

Itération **SCRUM**

MÉTHODE SCRUM



Annexe

Glodie Tshimini: contact@tshimini.fr



SCRUM GUIDE

- [Lien vers le guide SCRUM 2020](#)



FIN.

Glodie Tshimini: contact@tshimini.fr