

Méthodes agiles

Chapitre I : L'agilité

Qu'est-ce que l'agilité ?

Pourquoi l'agilité ?

L'AGILITÉ

L'agilité est une approche de gestion de projet incluant une méthode de travail, un paradigme, une philosophie, ...

Son but est de repenser la gestion de projet "traditionnelle" de manière empirique pour en réduire les problèmes.



L'agilité est donc plus un concept qu'une méthode de travail stricte. Chaque équipe pourra donc redéfinir sa méthode de travail en utilisant des objectifs, des règles de bonnes pratiques et des outils fournis par ce cadre de travail.

LE MANIFESTE AGILE

Le manifeste agile agilemanifesto.org décrit les préceptes de l'agilité :



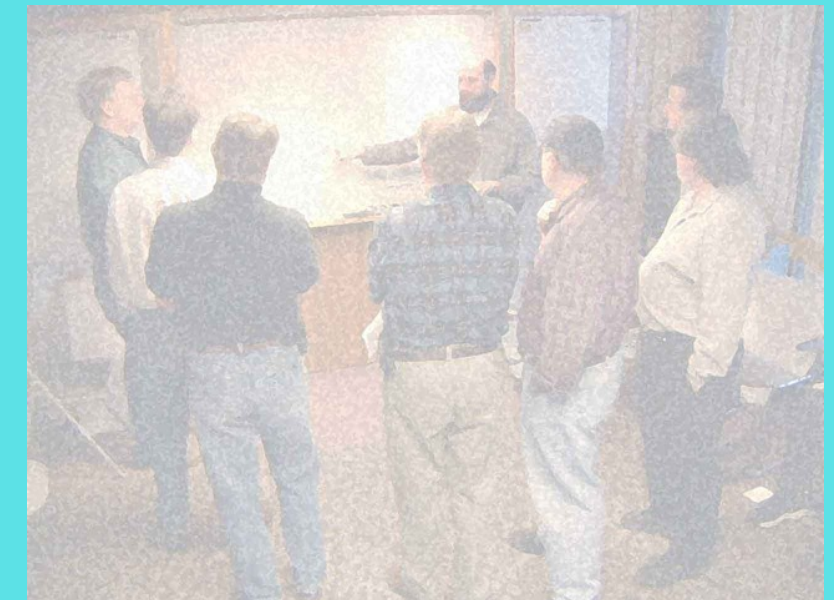
"Les individus et leurs interactions plus que les processus et les outils

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."



Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

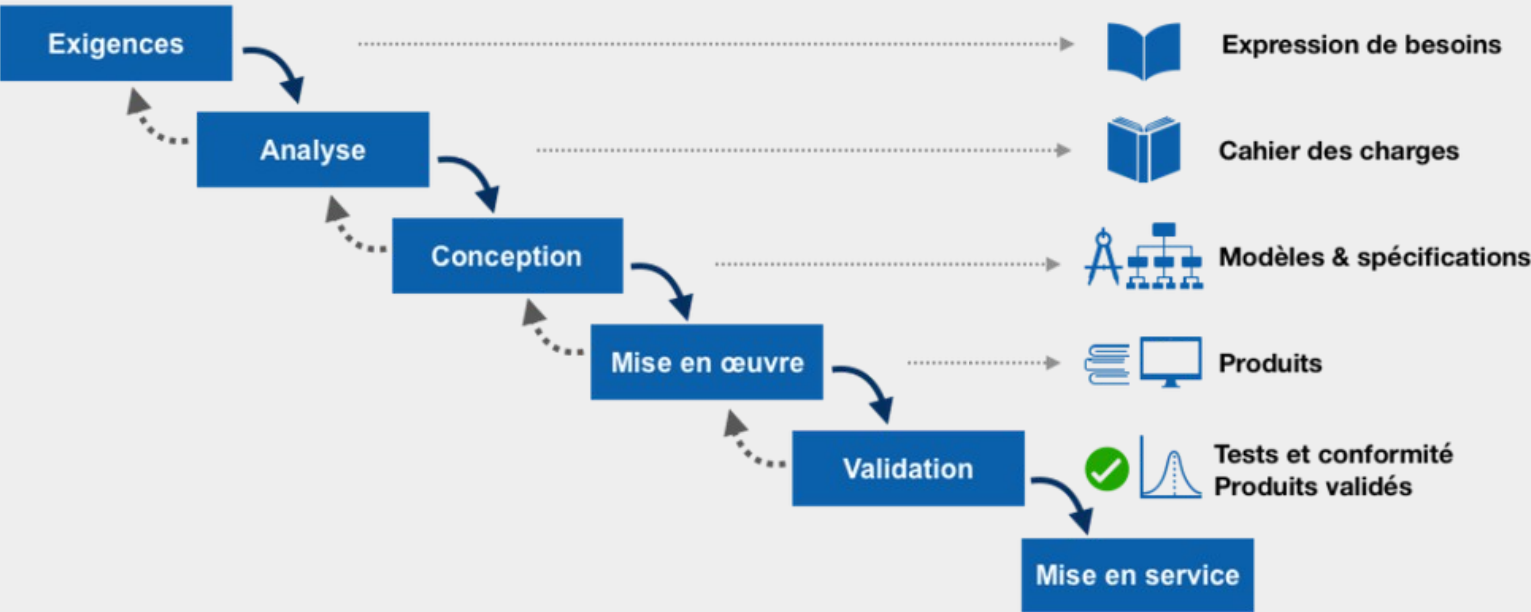
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

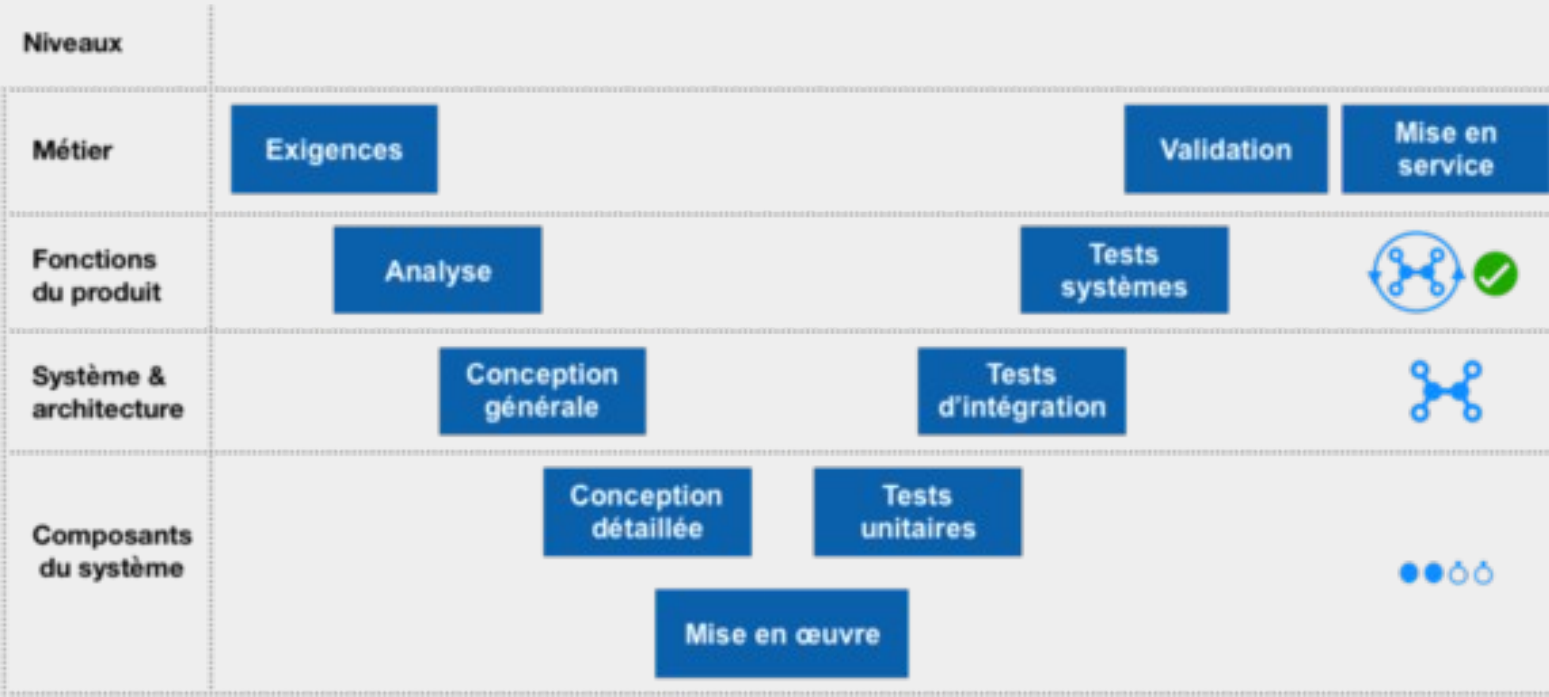
© 2001, the above authors
this declaration may be freely copied in any form,
but only in its entirety through this notice.

LES MÉTHODOLOGIES TRADITIONNELLES : MODÈLE EN CASCADE, CYCLE EN V

Dans l'usage courant des méthodes de gestion de projet traditionnelles (modèle en cascade, cycle en V), le projet est entièrement spécifié au commencement et son développement effectué en une unique itération



Modèle en cascade (Wikipedia)



Cycle en V (Wikipedia)

APPROCHE EMPIRIQUE

Au contraires des méthodes traditionnelles, les méthodes agiles partent du principe que spécifier et planifier dans les détails l'intégralité d'un produit avant de le développer (approche prédictive) est contre productif.



L'agilité est une approche empirique : l'idée consiste à se fixer un premier objectif à court terme pour se lancer sans tarder. Une fois ce premier objectif atteint, on marque une courte pause et on adapte la suite du projet en fonction de nouveaux éléments. Et ainsi de suite jusqu'à atteindre la destination finale.

MÉTHODOLOGIE AGILE

Dans le cadre d'un projet de développement logiciel, le client élabore sa vision du produit à réaliser et liste les fonctionnalités ou exigences de ce dernier.

Il soumet cette liste à l'équipe de développement, communique directement avec elle (plutôt que par papier) qui estime le coût de chaque élément de la liste.

On peut ainsi se faire une idée approximative du budget global.



Les fonctionnalités du produit sont souvent décrites sous forme de "User Stories". Nous verrons ce concept dans la méthodologie Scrum.

LES ITÉRATIONS

L'équipe sélectionne ensuite une portion des exigences à réaliser dans une portion de temps courte appelée itération. Chaque itération inclut des travaux de conception, de spécification fonctionnelle et technique quand c'est nécessaire, de développement et de test.

A la fin de chacune de ces itérations, le produit partiel mais utilisable est montré au client. Ce dernier peut alors se rendre compte par lui même très tôt du travail réalisé, de l'alignement sur le besoin.



L'utilisateur final quant à lui peut se projeter dans l'usage du produit et émettre des feedbacks précieux pour les futures itérations. La visibilité ainsi offerte est clef. Cette transparence peut également apporter davantage de confiance et de collaboration dans la relation client/fournisseur. Les risques quant à eux sont levés très tôt.

AVANTAGES POUR LE CLIENT



Si le client a priorisé avec soin son besoin, il peut saisir l'opportunité d'accélérer le "time to market" si il estime que le produit en l'état (partiel) peut aller en production. Économisant ainsi son budget et récoltant un premier retour sur investissement.



Il a aussi la possibilité de changer en cours de route la priorité des fonctionnalités qui n'ont pas encore été développées (prévues pour les itérations futures). Afin de retarder une fonctionnalité dont le besoin n'est pas mûr, ajouter une nouvelle fonctionnalité cruciale en échange du retrait d'une autre (respectant ainsi budget et délais), etc.

Cette souplesse ainsi offerte est donc un véritable atout pour le client.

Chapitre II :

SCRUM

LA MÉTHODOLOGIE SCRUM

Scrum est (de loin) la méthode agile la plus utilisée.
C'est un modèle complet pour gérer des projets à la fois imprévisibles et complexes.

La méthode Scrum repose sur trois piliers :

- Transparence
 - L'ensemble des acteurs (client, développeur, ...) doit avoir accès à toute l'information nécessaire
- Inspection
 - Le contenu du projet est validé périodiquement par l'équipe et le client
- Adaptation
 - Le projet évolue continuellement pour se rapprocher des demandes du client

LA MÉTHODOLOGIE SCRUM

La méthode Scrum est un schéma d'organisation utilisant :

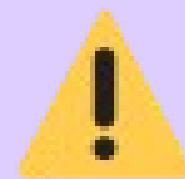
- Une approche empirique : amélioration continue du produit par une inspection quotidienne de l'état du projet pour orienter les décisions.
- Un cadre de travail global : Le produit n'a de valeur que dans son ensemble
- Une méthode itérative : Le développement du projet est réalisé en cycles de durée identique (itérations)
- Un développement incrémental : Chaque itération doit être utilisable
- Une pratique agile : Le client et les utilisateurs finaux participent à la gestion de projet



LES RÔLES SCRUM

 Pour mener à bien le projet, Scrum définit 3 rôles :

- Le Product Owner qui porte la vision du produit à réaliser
- Le Scrum Master garant de l'application de la méthodologie Scrum
- L'équipe de réalisation qui réalise le produit.



Ces rôles ne sont pas forcément attribués à des personnes différentes !

En pratique, le Scrum Master est généralement un membre de l'équipe de réalisation

LE PRODUCT OWNER



Les méthodes agiles placent le client au centre des négociations, or celui-ci n'est pas toujours disponible ou qualifié pour travailler en agilité.

Le Product Owner (PO) est donc le représentant des intérêts du client dans le projet.

Sa mission consiste en :

- L'expression des besoins
- La priorisation des besoins
- La validation des résultats



Le Product Owner peut être :

- Le client (directement)
- Un membre de l'équipe projet représentant les intérêts du client (jamais un membre de l'équipe de réalisation pour éviter les conflits d'intérêts)
- L'initiateur du projet s'il n'y a pas de client (projet interne, développement de produit sans client)

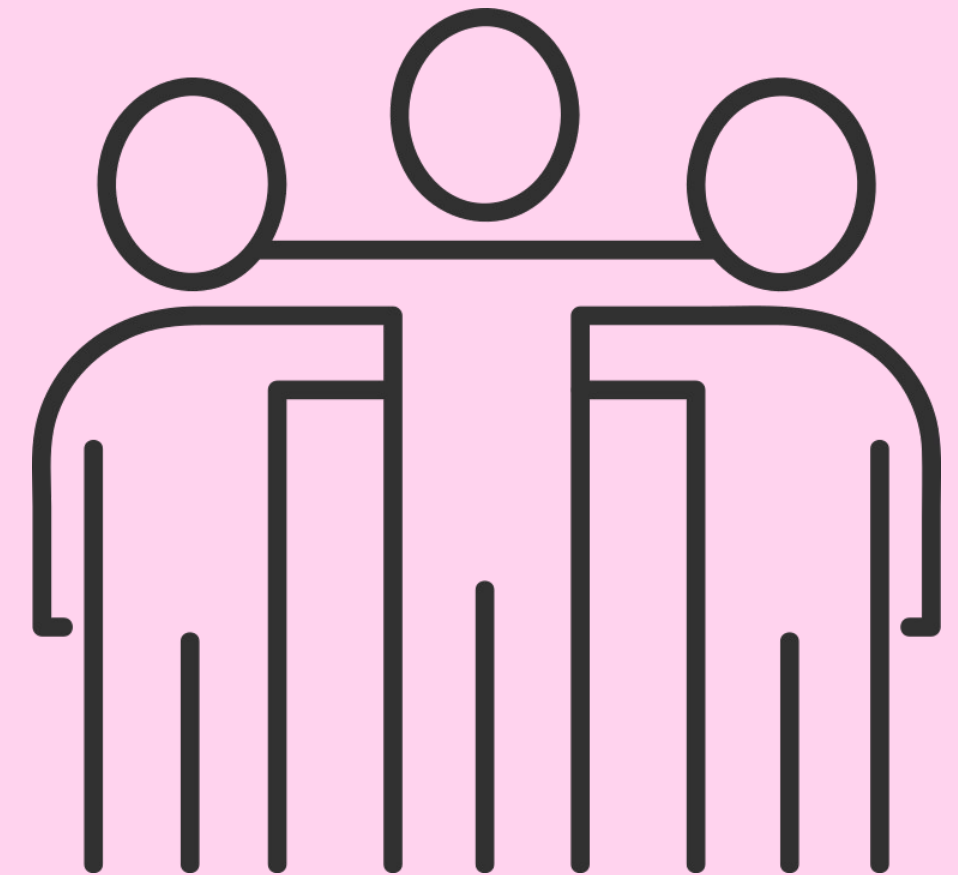
L'ÉQUIPE DE RÉALISATION



Les équipes agiles sont de petites équipes (moins de 10 personnes) possédant un bon niveau d'autonomie. Les responsabilités sont distribuées et les compétences partagées au maximum, afin de limiter la criticité de chacun.

La mission d'une équipe agile est la suivante :

- L'équipe estime et organise sa charge de travail et est seule décisionnaire des choix techniques
- Tous les membres sont considérés au même niveau
- L'équipe est responsable de la réussite du projet et de l'atteinte des objectifs. Elle participe avec le product owner à toutes les cérémonies Scrum.



L'équipe de réalisation s'adresse directement au Product Owner, sans intermédiaire.

LE SCRUM MASTER



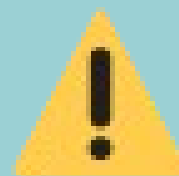
Le Scrum Master est le garant de la méthodologie. C'est avant tout un facilitateur, gérant les interactions au sein et à l'extérieur de l'équipe projet.

Sa mission est la suivante :

- Améliorer la communication à l'intérieur et en dehors de l'équipe.
- Expliquer et faire respecter les règles et les usages de la méthodologie projet, et encourager son amélioration.
- Suivre et communiquer tous les indicateurs pendant la gestion du projet.
- Maintenir la motivation au plus près des enjeux du client, et limiter les difficultés qui se présentent à l'équipe pour fluidifier la production.



Le rôle de Scrum Master est souvent cumulé avec un ou plusieurs autres rôles, mais il ne doit jamais être également le Product Owner.



Le Scrum Master est le garant de la méthodologie - ce n'est ni un manager, ni un supérieur hiérarchique !

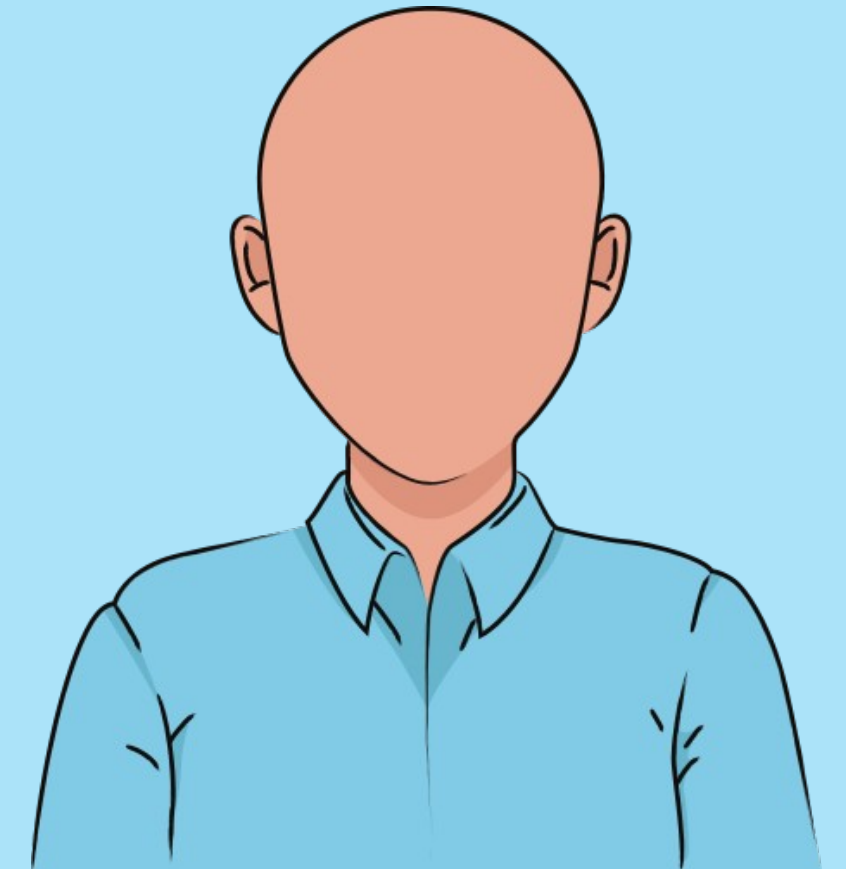
LES RÔLES ANNEXES

Au-delà des trois rôles principaux que nous venons de voir (Product Owner, Equipe de développement, Scrum Master), il est possible selon les besoins du projet d'ajouter des rôles supplémentaires :

- Le coach agile : présent occasionnellement pour former l'équipe à l'usage des méthodes agiles et proposer des axes d'amélioration.
- Le facilitateur : sa mission est d'aider le groupe à atteindre des objectifs communs en l'accompagnant mais en restant neutre.
- Les Stakeholders (parties prenantes) : ce sont des acteurs externes intéressés par le projet.

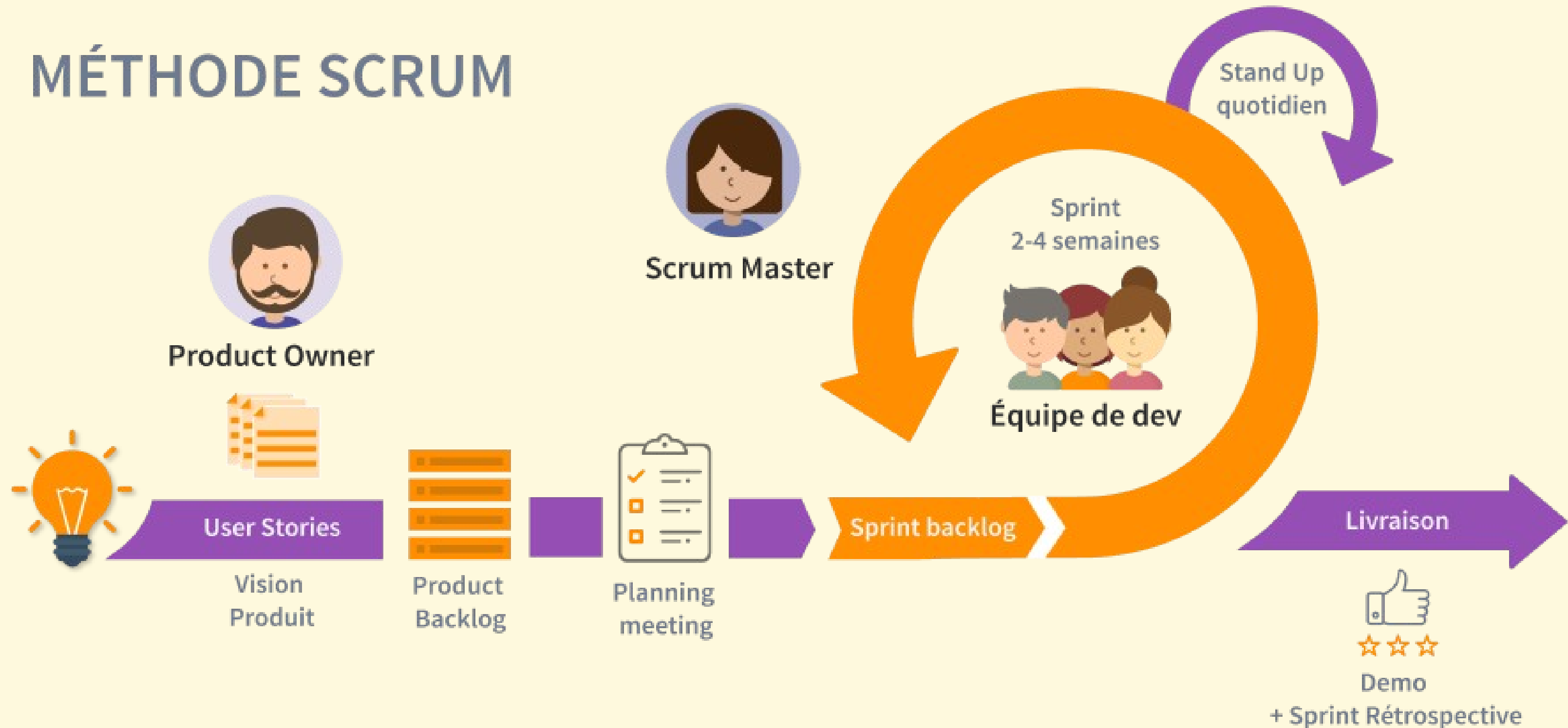


Aucun de ces rôles n'intersecte avec le rôle de product owner : ces rôles annexes ne prennent pas part au développement du produit mais proposent des interactions avec l'équipe projet.



SCRUM EN PRATIQUE

MÉTHODE SCRUM



SCRUM EN PRATIQUE : USER STORIES

La vision du produit est décrite sous forme de User Stories.

Ces histoires courtes décrivent les fonctionnalités attendues par le client pour le produit : ce ne sont donc pas des spécifications techniques !



Les User Stories restant à implémenter sont regroupées dans un panier accessibles à tous (Product Owner et développeurs) appelé le Product Backlog.

SCRUM EN PRATIQUE : USER STORIES

Le coût de développement des User Stories est estimé par l'ensemble de l'équipe (en utilisant le plus souvent une unité fictive : les **story points**). Cette estimation a lieu lors de séances de **Poker Planning** : chaque participant fournit une estimation en utilisant les valeurs de la suite de Fibonacci (1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144), et on recommence jusqu'à l'unanimité.

D'autres méthodes d'estimation existent, comme RICE = (Reach x Impact x Confidence) / Effort

- Reach : créer un référentiel de calcul (ex: 100 clients x 30% taux conversion x 3mois)
- Impact : 1 (faible), 2 (moyen), 3 (fort)
- Confiance : 0.1 à 1
- Effort : JoursHomme (JH), ...

Qu'est-ce qu'une bonne User Story ?

INVEST

- **Indépendante** des autres User Story
- **Négociable** : peut être discutée avec l'équipe de réalisation.
 - Bien : En tant que client, je peux connaître le montant total des factures impayées
 - Mauvais : En tant que client je veux, lorsque je clique sur le bouton "Calculer", qu'une ligne s'ajoute et qu'on affiche sur cette ligne le montant total des factures impayées
- **Vertical** : décrit une fonctionnalité complète (éviter : "créer la base de données", "créer l'IHM", ...)
- **Estimable** par l'équipe de réalisation (avec un risque d'erreur faible).
- **Suffisamment petite** ("sized-appropriately") : pour faciliter son estimation (idéal : conçue, développée et testée au sein d'une itération).
- **Testable** : par un test d'acceptation (tester la fonctionnalité d'un point de vue utilisateur)
- Également :
 - Source de valeur : porteuse de valeur pour le client ou l'utilisateur.
 - Courte : 1 à 3 phrases

SCRUM EN PRATIQUE : SPRINT

Le développement est incrémental, réalisé sous forme d'itérations (en général de 2 semaines) appelées Sprint.

Chaque Sprint démarre par une réunion de planification.

C'est l'une des étapes les plus importantes d'un projet Scrum : l'équipe de développement sélectionne les éléments prioritaires du Product Backlog qu'elle pense pouvoir réaliser au cours du sprint.



Cette planification aboutit à la création d'un plan de Sprint.
C'est un travail collaboratif de toute l'équipe Scrum.

SCRUM EN PRATIQUE : SPRINT



Durant le Sprint, l'équipe de développement, en collaboration avec le Scrum Master, développe les User Stories sélectionnées pendant la réunion de planification.



Une User Story doit normalement être réalisée en une unique itération. Si cela n'est pas possible, c'est souvent le signe que le découpage du projet est trop grossier.





Chaque Sprint se terminera par un livrable déployable en production. Une démonstration du travail réalisé sera faite au client, ce qui permettra de valider l'adéquation des changements à la vision du produit par le client.

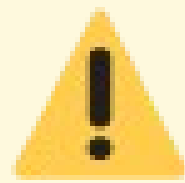


SCRUM EN PRATIQUE : STAND-UP

Chaque jour (en général avant de démarrer la journée de travail), le Scrum Master réunit l'équipe de développement quelques minutes pour faire le point sur l'avancée du projet : c'est le stand-up meeting.

 Cette réunion est primordiale pour suivre la bonne avancée du Sprint. C'est également un espace de discussion peu formel entre développeurs, pour aborder les problématiques rencontrées, les choix d'implémentation, les soucis de dépendances, etc.

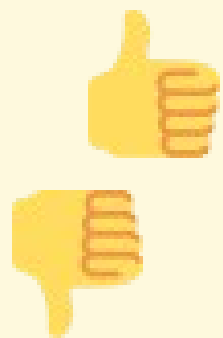
 C'est généralement durant le stand-up meeting que les tâches de la journée sont réparties, sur la base du volontariat.

 Le stand-up meeting doit être une réunion très efficace, pour garder l'attention de tous les membres de l'équipe : le Scrum Master veillera donc au respect du temps de parole, et à la bienveillance des échanges.



SCRUM EN PRATIQUE : LIVRABLE ET DÉMO

A la fin du Sprint, l'équipe de développement crée un nouveau livrable intégrant l'ensemble des modifications effectuées pendant le Sprint. Ce livrable est ensuite utilisé pour faire une démonstration au client, qui valide ces changements.



Suite à la démonstration et aux retours du client, le Product Owner valide (ou invalide) le Sprint. En cas d'échec, le livrable n'est pas déployé et les User Stories retournent dans le backlog.

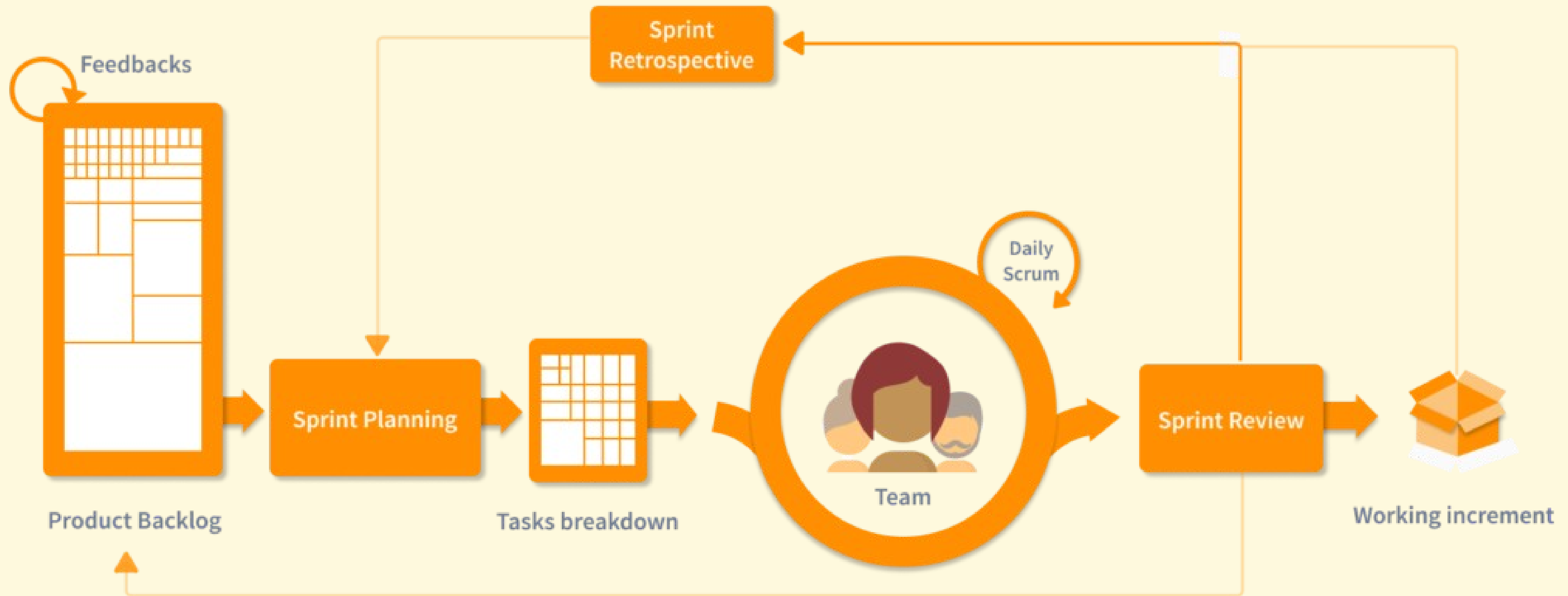
SCRUM EN PRATIQUE : RÉTROSPECTIVE

Une fois le sprint terminé (et la démonstration), le Scrum Master réunit l'ensemble de l'équipe de développement pour effectuer une rétrospective.

En analysant les graphiques Scrum (Burndown chart, Vélocité), en discutant librement en prenant du recul avec le sprint écoulé, l'équipe cherche à améliorer les interactions entre individus, à gagner en bien-être et en motivation, en qualité produit, et de façon globale, à améliorer sa productivité.



RÉSUMÉ DES CÉRÉMONIES SCRUM



Tuleap

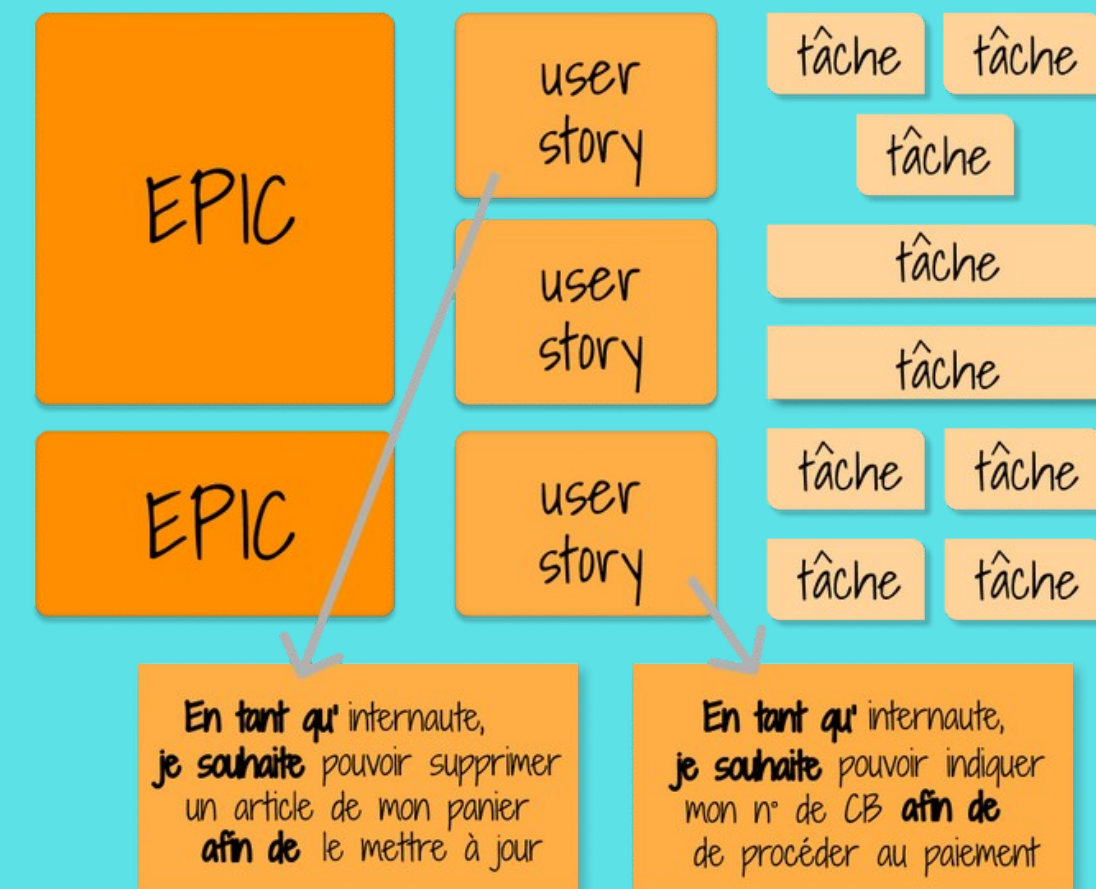
EPIC, USER STORIES, TÂCHES

Une User Story est l'énonciation d'une attente utilisateur (ce n'est ni une tâche, ni une spécification). Pour cette raison, on les décrit dans le format suivant :

As a [persona], In order to [but à atteindre], I want to [expression du souhait]

Une User Stories étant liée au cycle de vie d'un Sprint, il peut être intéressant :

- de découper cette User Story en un ensemble de tâches simples (~1 jour)
- de regrouper des User Stories autour d'Epic, par exemple pour décrire des fonctionnalités unifiées



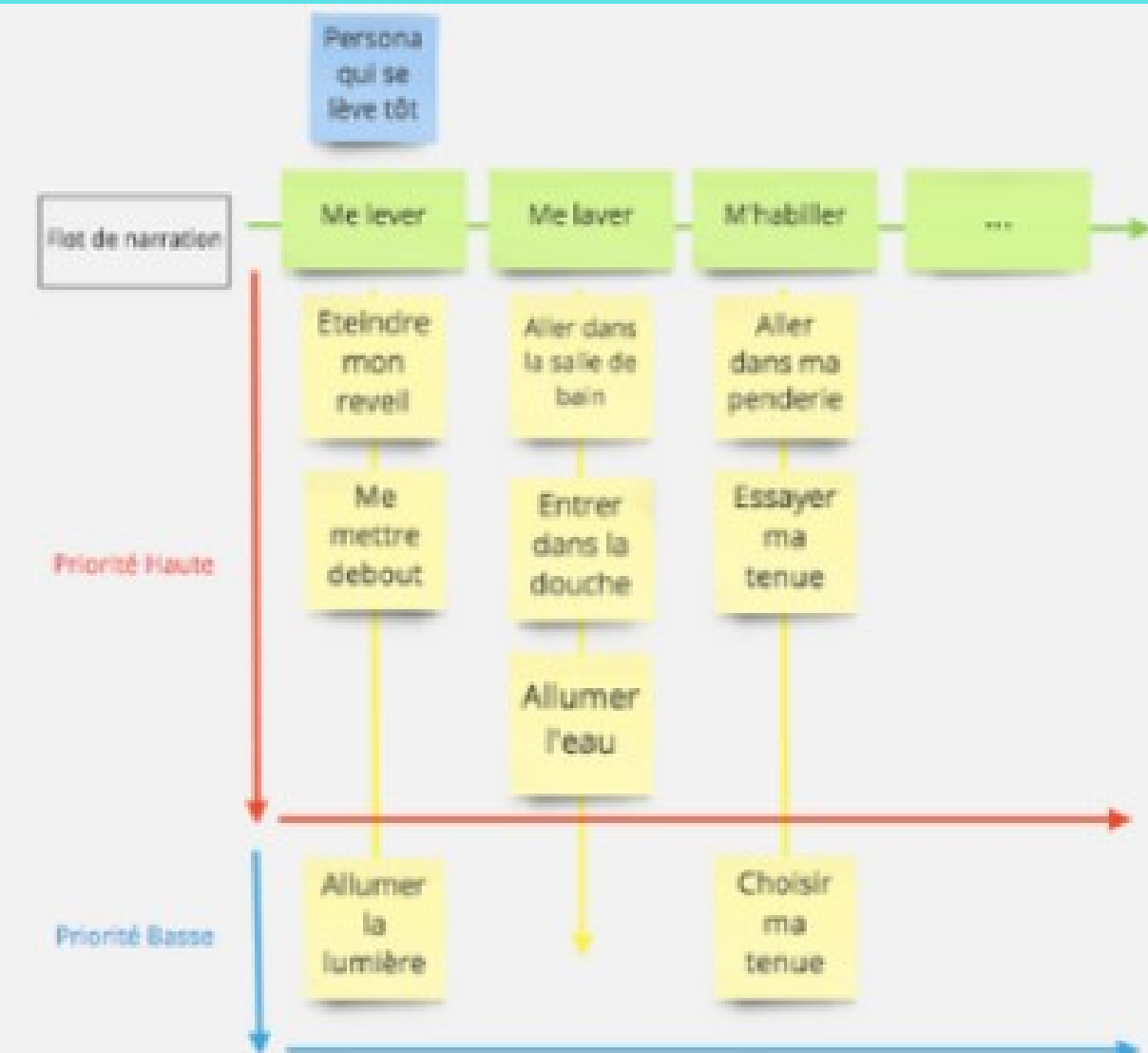
STORY MAPPING

Pour décrire le backlog du produit, on peut utiliser le Story Mapping. Il s'agit d'un plan en 2 dimensions :

- l'axe horizontal représente des flots de narration : un utilisateur type du produit effectue un ensemble de grandes actions l'une après l'autre (ce sont des Epics)
- l'axe vertical représente les User Stories associées aux Epics de la narration. Ces User Stories sont triées par priorité.

Les User Stories étant triées par priorité, il est possible de décrire des groupes de User Stories permettant de générer un produit viable pour le client.

On parle de Minimum Viable Product (MVP).



Sources : <https://blog-gestion-de-projet.com>

Product Vision Board

Très utile, le Product Vision Board donne une vision d'ensemble du produit :

- Vision : BUT du produit, quel changement positif introduit-il ?
- Groupe cible : segment de marché, groupe cible
- Besoins : problème à résoudre ? bénéfice(s) ?
- Produit(s) : qu'est-ce que le produit, en quoi est-il spécifique, est-il réalisable ?
- Objectifs business : quel(s) bénéfice(s) pour la compagnie ? Quels objectifs business ?

Spike

Que faire lorsqu'une User Story est difficile à estimer ?

Le "Spike" permet d'ajouter une activité d'exploration (souvent technique) afin de lever de l'incertitude. Un spike :

- N'est **pas une User Story**
- Est **court** : 2 à 3 jours en principe
- N'est pas estimé : pas d'impact en vélocité
- Crée un livrable (prototype, doc, ...) examiné lors de la **revue** du Sprint
- Permet d'estimer / modifier la User Story dans le Sprint suivant.

Un Spike est souvent représenté par un post-it de couleur différente.

KANBAN

La méthode Kanban permet de représenter visuellement l'état du Sprint en cours.

Il s'agit d'un tableau blanc sur lequel sont affichées les user story (et parfois les tâches) sous forme de post-it, dans différentes colonnes représentant l'état courant de la tâche.

Chaque user story ou tâche possède son propre post-it, et on utilise au moins 3 états : à faire, en cours et terminé.



Il est possible d'ajouter d'autres colonnes pour identifier d'autres états d'une tâche : par exemple, une colonne à tester est très utile avant de passer en état terminé.



Pour améliorer le management visuel, on pensera à ajouter de l'information sur les post-it : estimation de la story (story point), qualification de l'état "terminé" (test nécessaire, livrable, ... souvent renseignés au dos du post-it), type de story (couleurs de post-it), personne à qui la tâche est assignée, et tout autre indicateur qui fait sens dans le projet..

Kanban Board



Visual signal



Columns



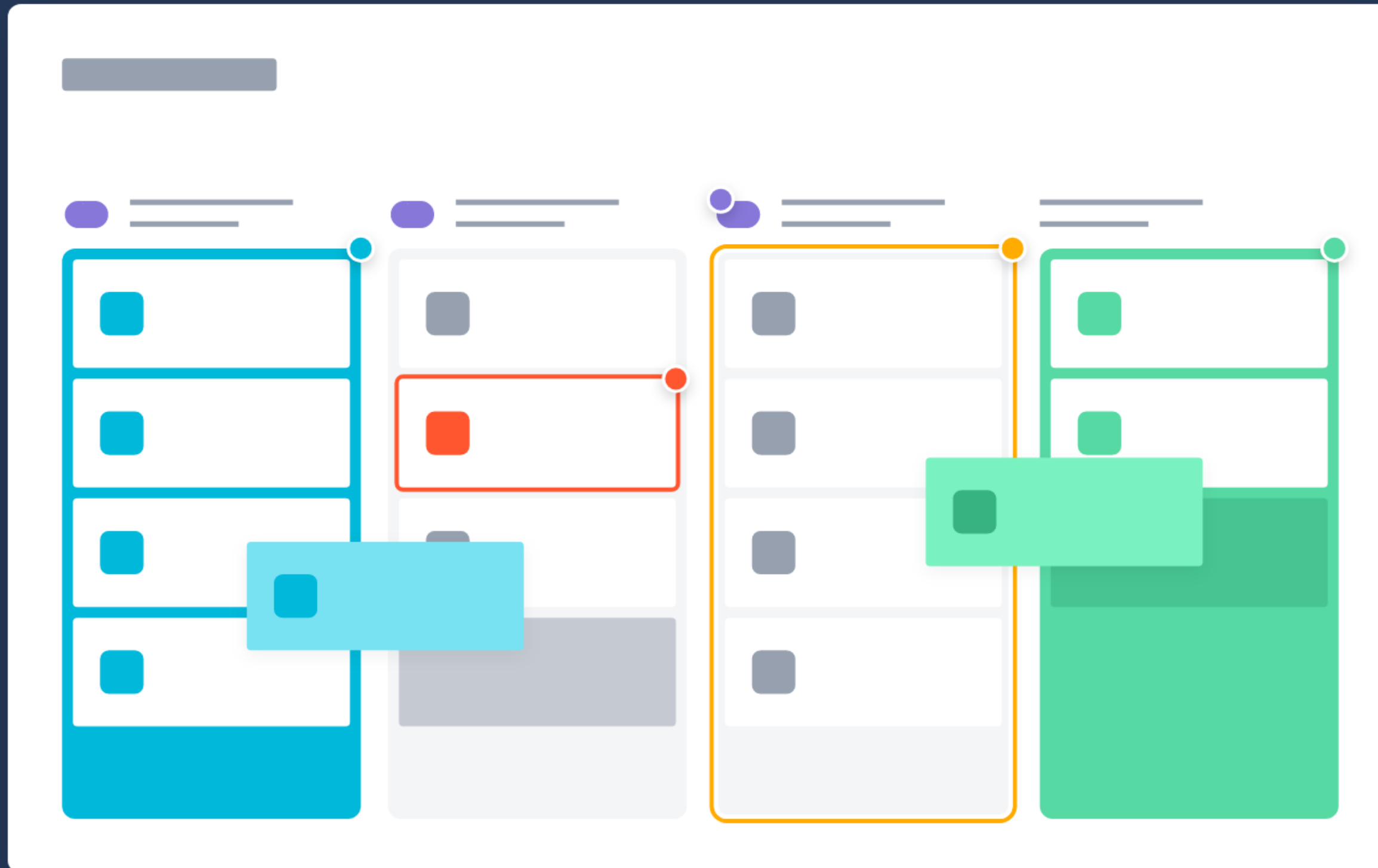
Work-in-progress limit



Commitment point

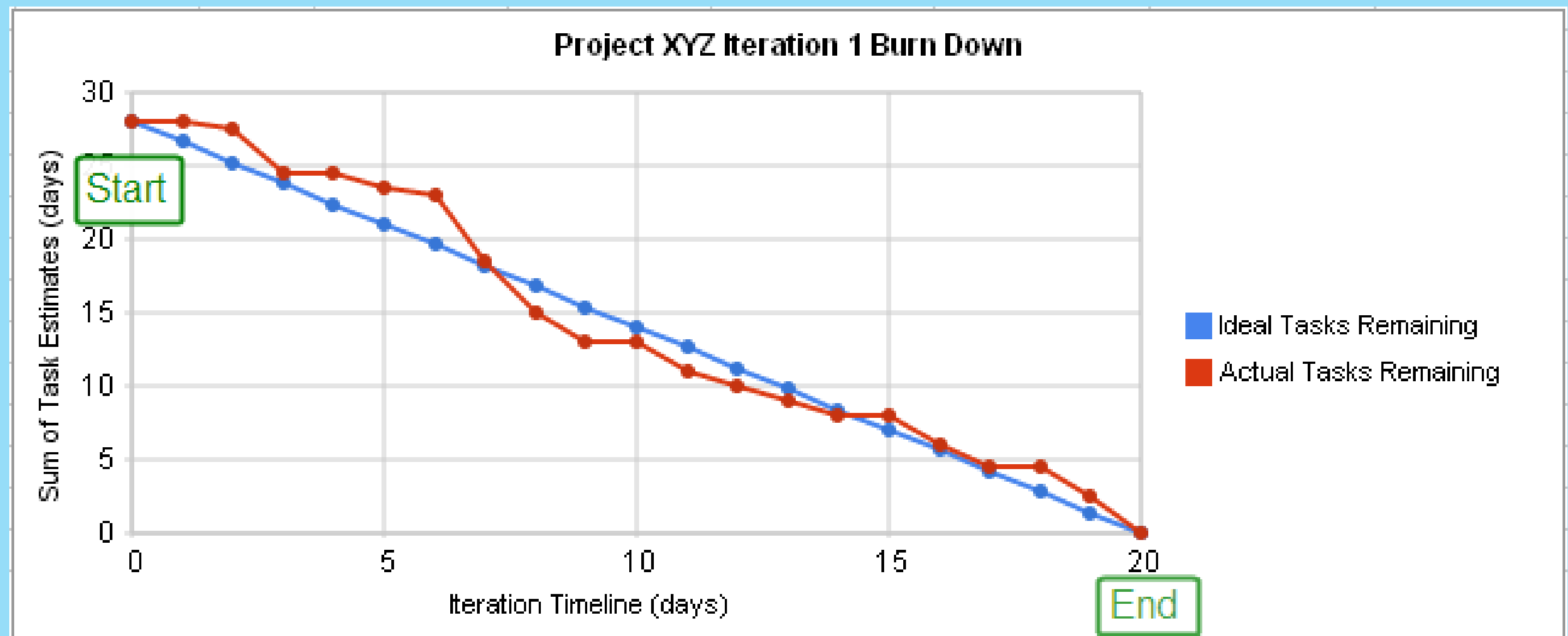


Delivery point



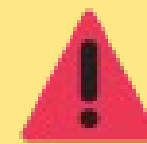
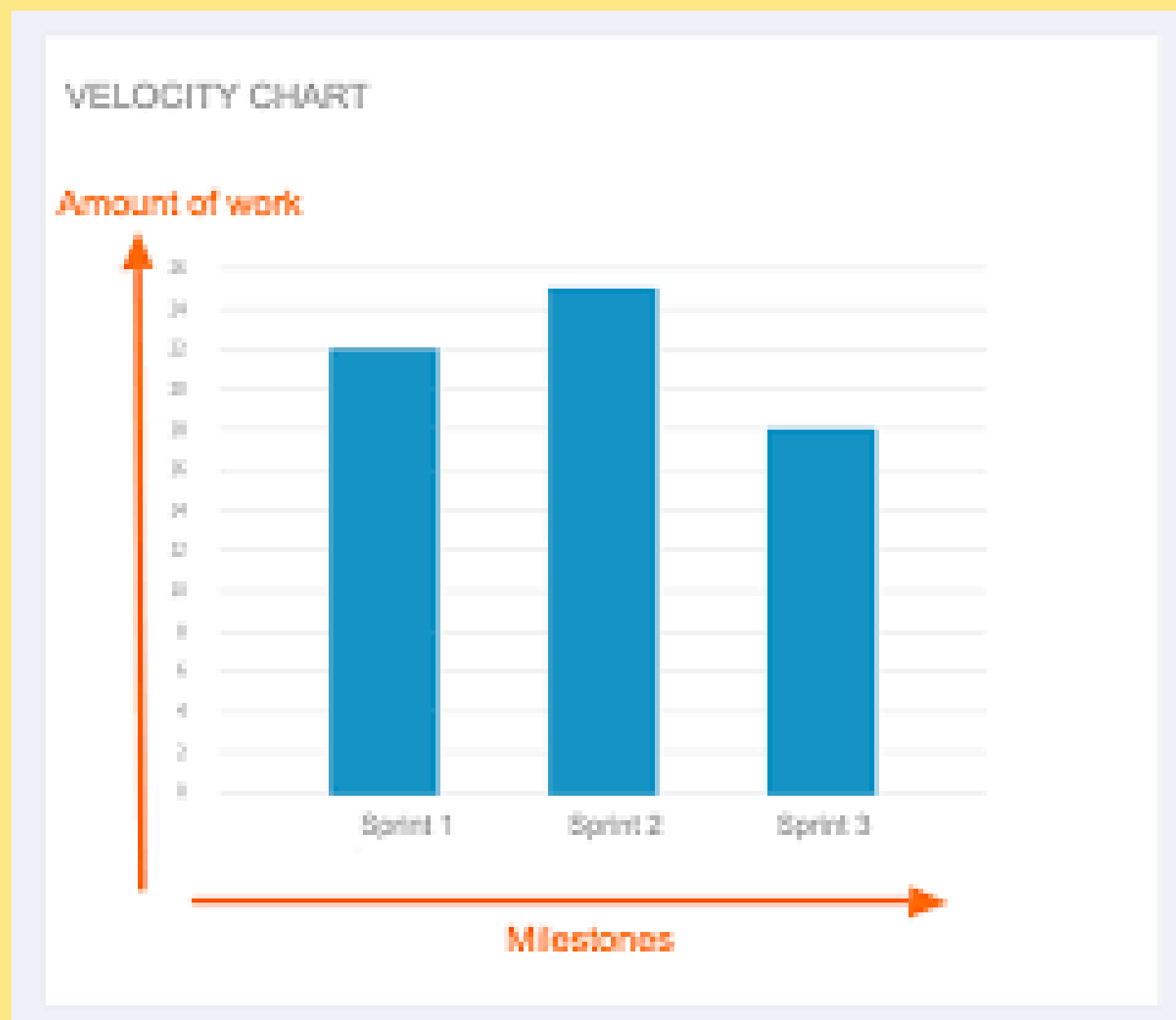
INDICATEURS : BURNDOWN

Le burn-down chart est un graphique souvent utilisé au niveau du sprint. Il permet de visualiser le reste à faire et donne une vision du rythme de développement.



INDICATEURS : VÉLOCITÉ

La vélocité permet de déterminer le nombre de story points réalisables en un Sprint par l'équipe. Pour la mesurer, on fait une moyenne du nombre de points livrés sur plusieurs sprints. La vélocité aide à la planification.



Attention, on ne mesure pas la performance et la productivité d'une équipe avec la vélocité. Ce qui est important c'est la valeur métier livrée, la qualité du logiciel livré.

DIFFICULTÉS

Scrum est simple mais difficile : la courbe d'apprentissage peut être longue et demande beaucoup d'autonomie.

La motivation et l'engagement du Product Owner et du Scrum Master sont des variables critiques dans la réussite d'une méthodologie Scrum, ainsi que l'adoption par tous les membres de l'équipe de développement.

Scrum est uniquement un cadre de gestion de projet. On pourra donc coupler son usage à des méthodes de développement : eXtreme Programming, Software Craftsmanship, ...

Comment démarrer un projet ?

Préparation du projet

Avant de commencer à réaliser techniquement le projet, il faut le préparer :

- Créer et partager une vision du produit final
- Définir et partager les personas, la story map, le plan de release du projet
- Préparer le backlog complet de la 1ère itération
- Préparer le socle technique : langages, frameworks, stratégie DevOps, pipelines CI/CD, serveurs, environnements, ...
- Former l'équipe à la méthodologie choisie
- Définir la gouvernance : qui est responsable de quoi ?
- S'assurer que le budget et les ressources prévus sont adaptés
- Comment déclarer un Go / No go à tout moment ?
- Quelle est pour ce projet la Definition of Ready et la Definition of Done ?

Framing vs Sprint 0 vs Scrum

Le “Sprint 0” est une pratique non-officielle qui consiste à démarrer par un sprint spécial (dit de planning ou de design) d’une durée différente (2-3 jours à un vrai sprint). Il n’y a pas de livrable à la fin de ce sprint.

Le “Framing” est une méthode agile permettant de gérer le lancement d’un projet agile (Scrum ou autre) en itérant sur les étapes nécessaires au démarrage du projet avant de réellement lancer le projet. Cette méthode reprend la phase de cadrage de la cascade.

La méthodologie Scrum “officielle” consiste à démarrer par un vrai sprint similaire à tout sprint Scrum (même durée, production d’un livrable, ...). La construction du produit est réellement empirique et les choix sont faits au moment de leur utilisation.

Comment passer à l'échelle ?

Passer à l'échelle

Scrum of Scrum : délégations de 2-3 personnes se synchronisant

- très léger
- uniquement des points de synchronisation
- à adapter au projet courant
- <https://www.atlassian.com/fr/agile/scrum/scrum-of-scrums>

LeSS

- Produit global : planification et revues générales
- Puis : planification et revues refaites par équipe pour sous-projets
- Plus lourd

D'autres méthodologies existent : scrum@scale, ...

Liens

- <https://www.uv.es/nemiche/cursos/Scrum.pdf> : cours Scrum sous licence CC-BY