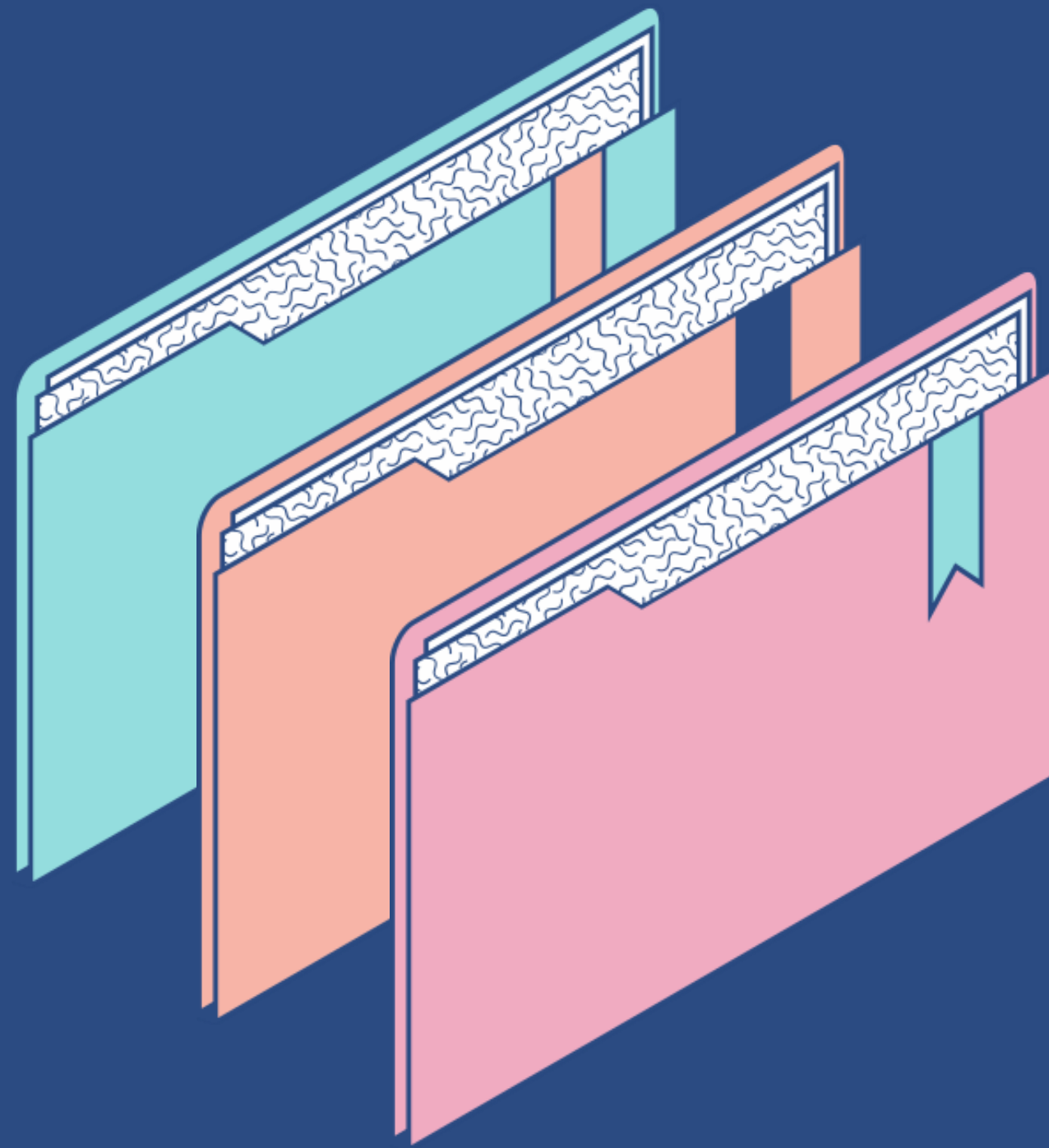


# Gestion de projets



# SOMMAIRE



- Introduction
- Paramètres d'un projet
- Méthodologies traditionnelles
  - Méthodologies Agiles
- Collaboration à distance
  - Outils collaboratifs



# Introduction à la gestion de projets

3





# Pourquoi le Génie Logiciel ?

## Grosses Erreurs :

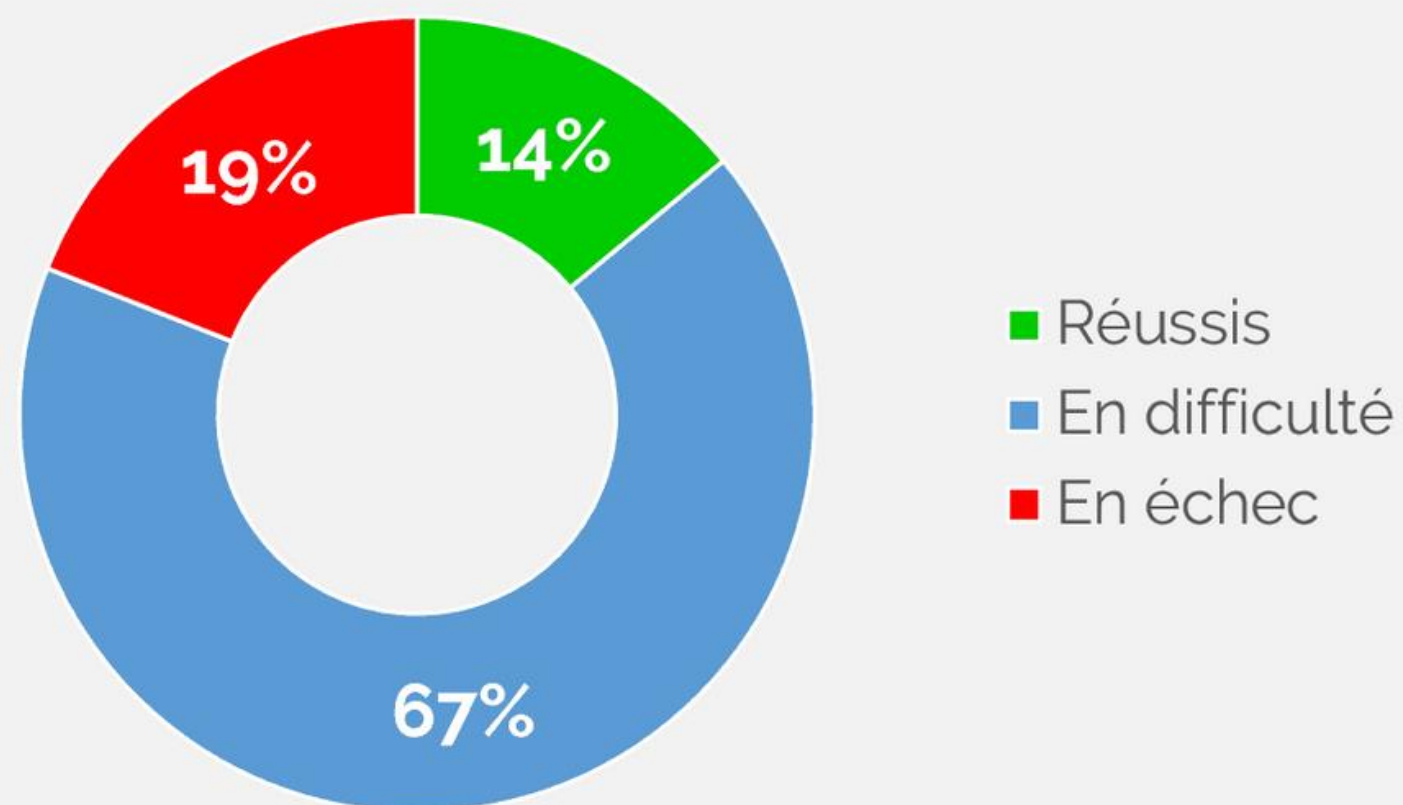
- Missiles patriotes (Erreur d'arrondis)
  - Sondes perdues (Erreur d'unités)
- Bug de l'an 2000 (année sur 2 chiffres)



# Pourquoi le Génie Logiciel ?

## TAUX DE RÉUSSITE DES PROJETS IT

(toutes méthodologie confondues)



[www.laurineautin.fr](http://www.laurineautin.fr)

Source : Standish Group Chaos Studies 2013-2017

## Les projets logiciels :

- Ne livrent pas le produit à temps
- Coûtent beaucoup plus cher que prévu
- délivrent un produit de faible qualité
- Coûtent en moyenne 81 milliards \$ / ans en échecs



# Construire un logiciel comme un pont ?

- crash système habituel
- cause de bug possiblement non identifiable
  - dommages (souvent) mineurs
- anticipation impossible de toutes les situations
  - complexification très rapide
- modification fréquente de fonctionnalités, architectures ...



# Les mythes de la gestion de projet

*Les outils sont la solution ...*

=> Un outil est inutile sans méthodologie





# Les mythes de la gestion de projet

*En cas de retard, on ajoutera des ressources humaines ...*

=> Ajouter du personnel à un projet en retard le fera retarder d'autant plus  
(Loi de Brooks)

Personnel non formé, sans expérience sur le produit ...





# Les mythes de la gestion de projet

*Le logiciel est flexible, il est facile d'y ajouter des changements ...*

**=> Plus un changement est tardif, plus il coûte cher**



# Les mythes de la gestion de projet

*Une fois le programme écrit, le travail est terminé ...*

=> 50 à 70% de l'effort d'un projet est réalisée après sa livraison



# Les mythes de la gestion de projet

*Jusqu'à ce que le programme tourne, on ne peut évaluer sa qualité ...*

=> Inspections & revues de code





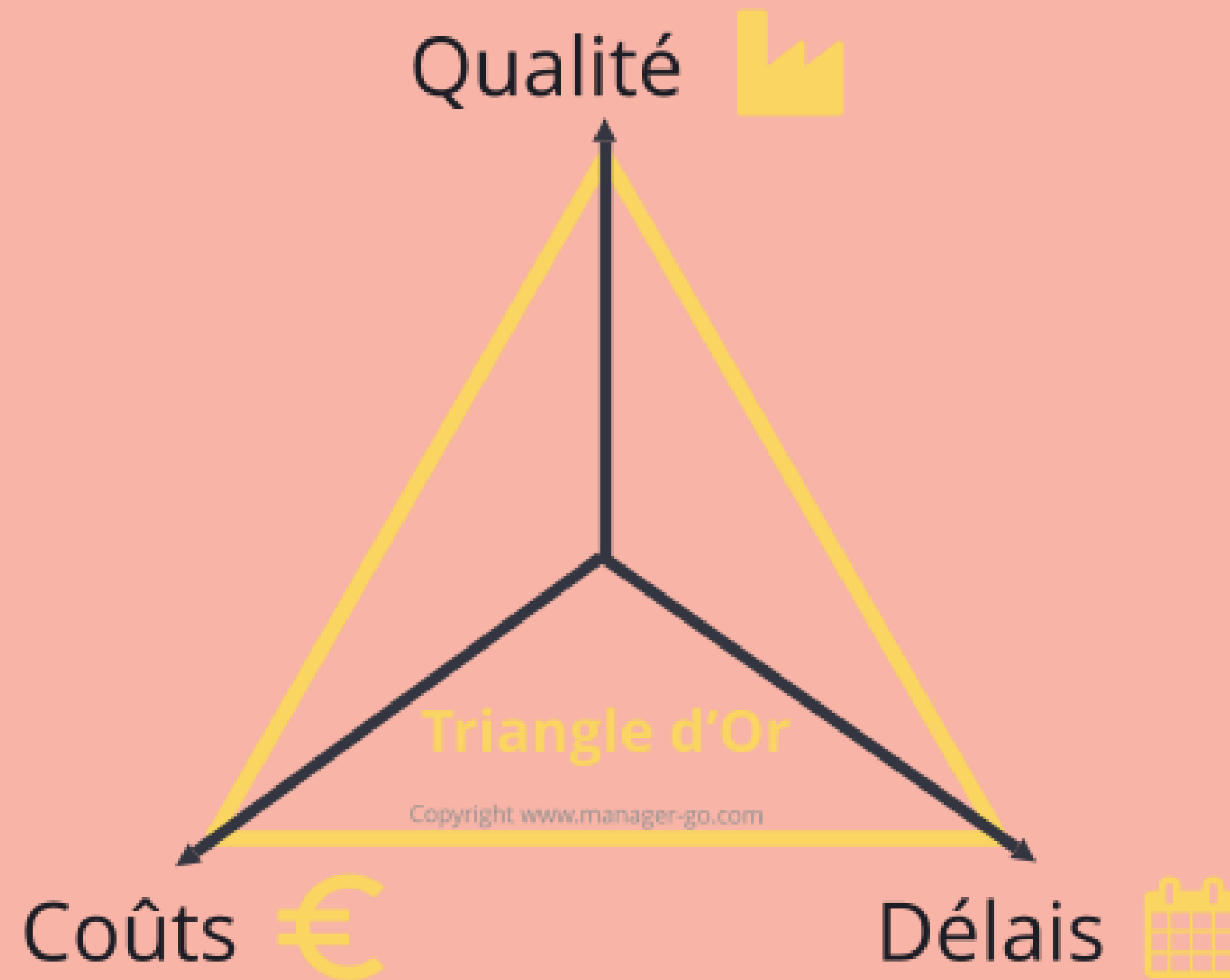
# Les mythes de la gestion de projet

*La seule chose à livrer pour un projet est programme qui tourne ...*

=> Documentation (Utilisateur, Maintenance), tests, intégration

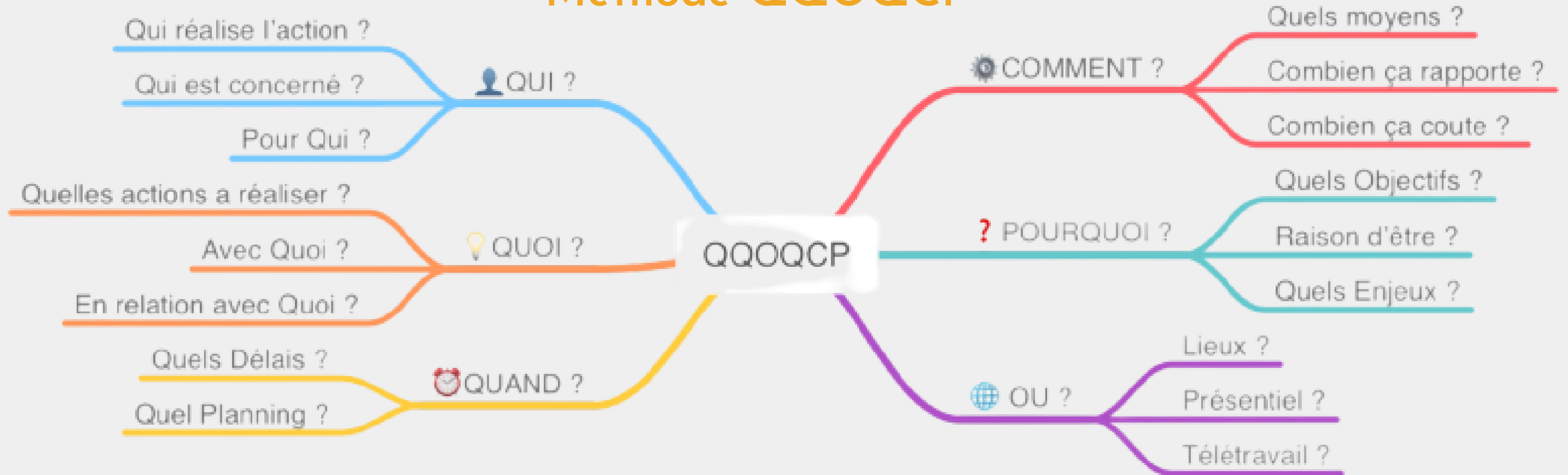
# Les différents paramètres d'un projet

13



# Définition du projet

## Méthode QQQQCP







# Définition du projet

## Méthode QQQQCP

QUESTION	QUESTION SECONDAIRE	ELEMENTS DE REPONSE
Qui ?	Par qui ? Pour qui ? Avec qui ?	Les acteurs impliqués : responsables, victimes, témoins, cibles
Quoi ?	Quel est le problème ?	Les problème lui même : Résultats et symptômes du problème
Où ?	Depuis où et Vers où ?	Le.s lieu.x concerné.s : Espace physique ou virtuel identifiable
Quand ?	Depuis et Jusqu'à quand ?	Date, durée, et fréquence de l'évènement
Comment ?	Quelles conditions ?	Moyens matériels et procédures qui ont engendré le problème
Pourquoi ?	Quelles causes ?	Causes du problème



# Définition du projet

Exercice de groupe :

Les exemples de projets suivants sont ils bien définis ?

- *Mettre en place sous 3 mois un logiciel de gestion comptable au sein du service financier d'une entreprise.*
  - *Concevoir et réaliser le site vitrine d'un restaurant.*
- *Développer une stratégie marketing sous 12 mois afin de gérer plus de prospects.*
  - *Lancer un nouveau modèle de produit existant.*



# Définition du projet

## Exercice individuel :

- *Définissez puis présentez votre projet fil rouge*





# Intervenants d'un projet

## Maitrise d'ouvrage ( MOA )

- *Entité responsable de l'expression du besoin*
  - *Souvent non informaticien*
  - *Définit les besoins et le budget*



# Intervenants d'un projet

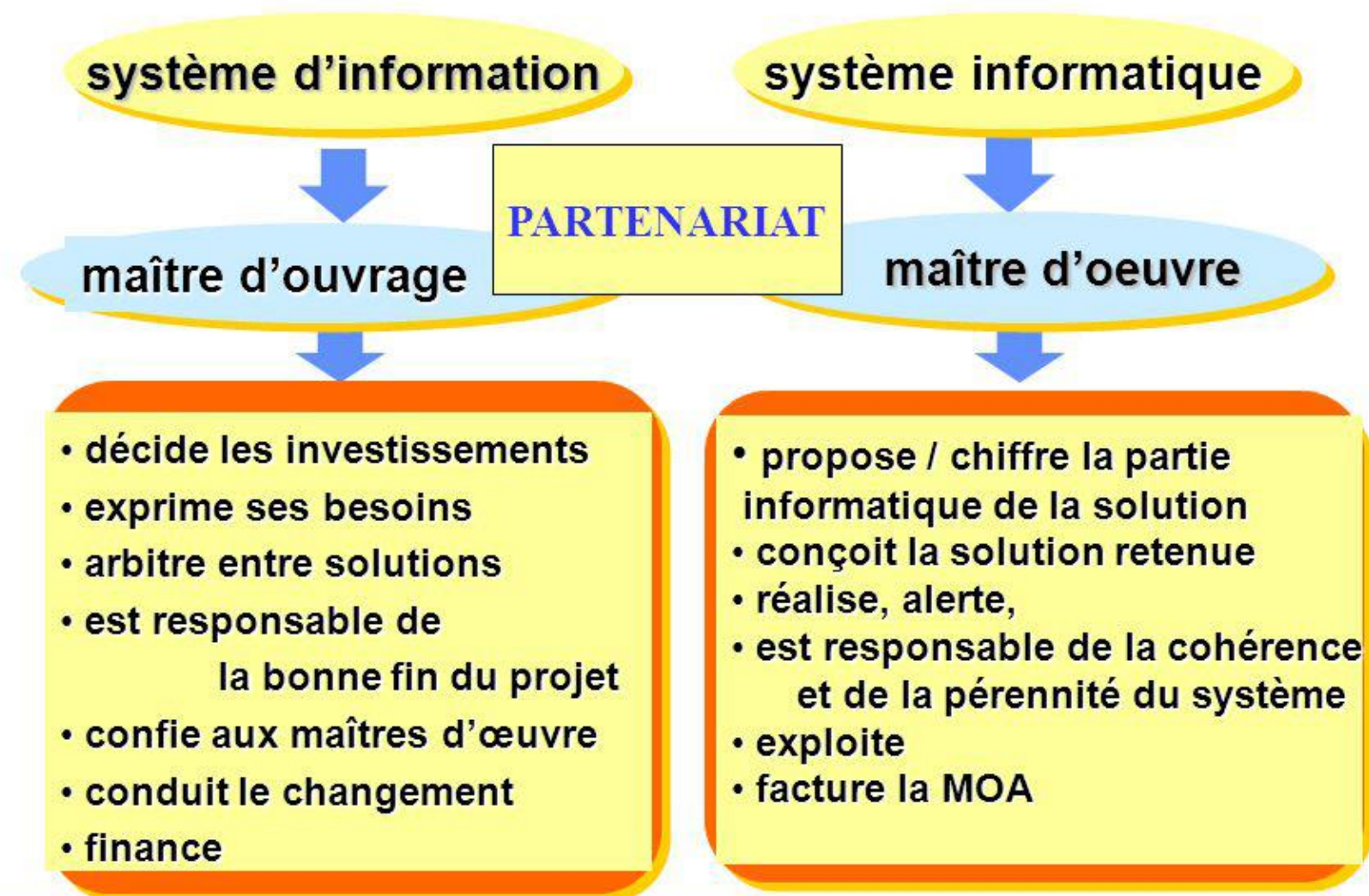
## Maitrise d'œuvre ( MOE )

- *Entité responsable de la concrétisation de l'idée en outil informatique*
  - Souvent pas de connaissance fonctionnelle
  - *Définit les choix techniques, et élabore les délais*



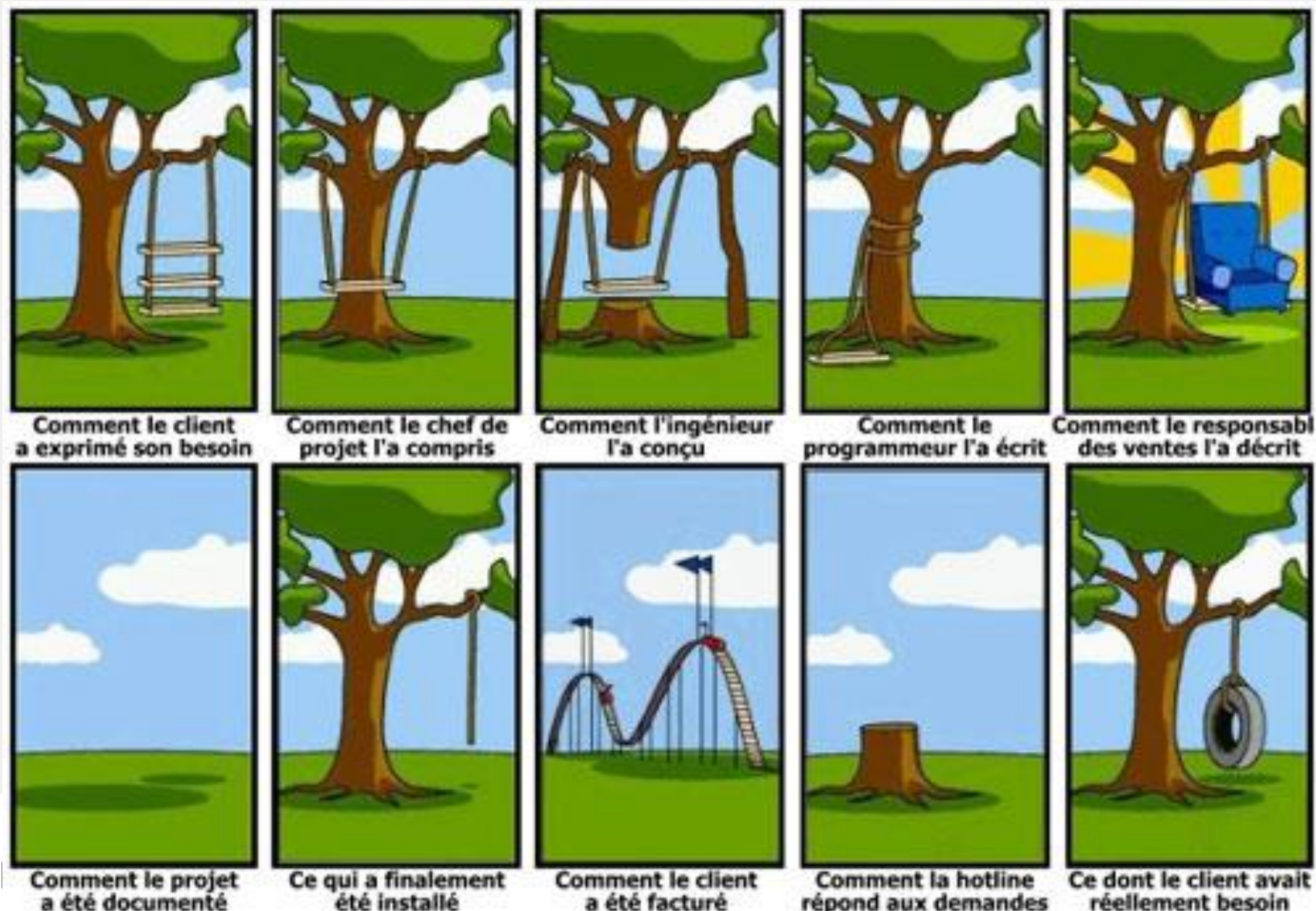
# Intervenants d'un projet

## MAÎTRISE D'OUVRAGE ET SYSTÈME D'INFORMATION





# Visions d'un projet

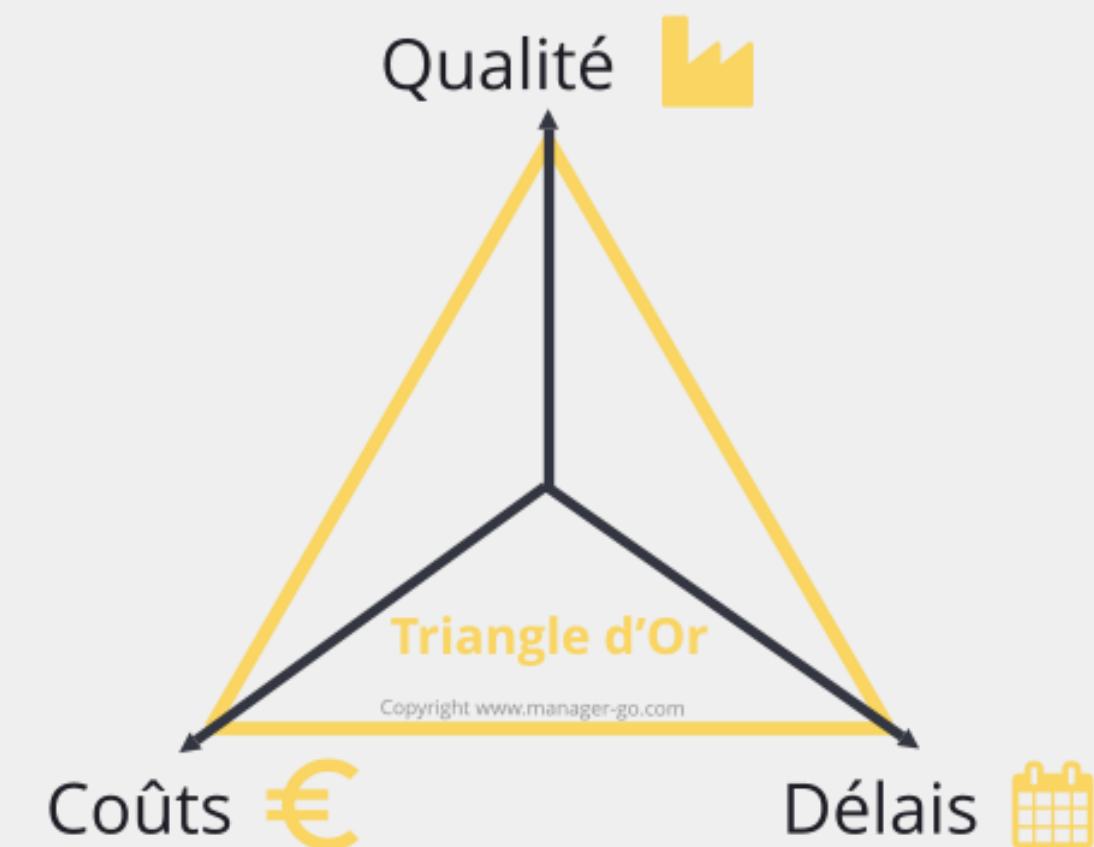




# Paramètres d'un projet

*Un projet est défini par :*

- *Une durée limite : Délai*
- *Des contraintes définies : Objectifs*
- *Un budget alloué : Coût*





# Evaluer les charges

Pourquoi suivre les coûts du projet ?

- *Pour ajuster les prévisions budgétaires (anticiper les tendances de coûts)*
- *Pour décider d'actions correctives éventuelles (C.U.D de fonctionnalités)*
  - *Pour suivre les contraintes de coûts*





# Coûts et recettes

## Estimation des charges en Jours/Homme

- *Facilite la planification des actions / jalons*
  - *Détermine une date de fin au projet*
- Facilite la répartition de la charge de travail
- Estime le budget du projet en ressources humaines



# Chiffrer un projet - Démarche

- *Lister en détails les tâches à mener*
  - *Décomposer les grandes étapes du projet en plusieurs tâches élémentaires*
- *Définir les intervenants pour chaque tâches*
  - *Attribution des tâches en fonction du rôle et des compétences*



# Chiffrer un projet - Démarche

## 3. Evaluer la durée d'une tâche

- Appuyez vous sur les bilans de fin de projet
  - Contactez un expert
  - Cherchez des coûts standards
  - Utilisez la méthode PERT





# Chiffrer un projet - Démarche

## 3. Evaluer la durée d'une tâche : Méthode PERT

- La méthode PERT repose sur une moyenne pondérée entre :
  - O : Evaluation optimiste
  - M : Evaluation moyenne
  - P : Evaluation pessimiste

- $\Rightarrow \text{Estimation} = (O + 4M + P)/6$

Tâches	Intervenant	Temps (j/h)	Coût jours	Total coût (€)
Design				
Charte graphique	DA	1	600	600
Maquettes	DA	2	600	1200
<b>Total Design</b>		<b>3</b>		<b>1800</b>
Fonctionnel				
Arborescence site	Chef de projet web	1	600	600
Fonctionnalités	Chef de projet web	1	600	600
Architectures des données	Chef de projet web	0.5	600	300
<b>Total fonctionnel</b>		<b>2,5</b>		<b>1500</b>
Développement web				
Intégration html	Developpeur	4	500	2000
Development php	Developpeur	4	500	2000
<b>Total développement web</b>		<b>8</b>		<b>4000</b>
...				
<b>Total</b>				

28

# Chiffrer un projet - Démarche

## 4. Estimer les charges



# Chiffrer un projet - Indicateurs de coûts

## Indicateurs de coûts :

- *Coût réel*
- *Coûts non planifiés*
- *Coûts des retards*
- *$\text{Ecart de coût (réel - prévisionnel)} / \text{prévisionnel}$*





# Chiffrer un projet - CJM par profil

- Développeur Junior : 200-400€
- Développeur Senior : 400-800€
  - UX designer : 500-800€
  - Chef de projet : 500-1000€
    - • Manager : 600-1200€



# Délais et planifications

Pourquoi suivre les délais du projet ?

- *Pour ajuster les prévisions de planning (analyser avancement et reste à faire)*
  - *Pour anticiper et décider d'actions correctives (actions à entreprendre)*
  - *Pour respecter les contraintes de délais*



# Délais et planifications

## Indicateurs de délais:

- *Taux de retard (%) : (tâches non réalisées / prévues ) \* 100*
  - *Durée d'une tâche*
- *Ecart de durée (Durée réelle - initiale ) / initiale*
- *Ecart de délai : Temps planifié - Temps utilisé*





# Suivi de qualité

La démarche qualité commence lors du cadrage du projet, par la mise en place d'outils et de méthodes :

Use case, MCD, MLD, diagrammes de classe, d'activité, de séquence, UML, Merise ...



# Suivi de qualité

## Norme ISO 9000

- Le projet doit être tourné client et utilisateur, dans une logique d'amélioration continue,
  - Un suivi de la satisfaction client/utilisateur doit être mis en place,
- La hiérarchie doit définir des orientations claires en terme de qualité (principe de leadership),
- Les objectifs du projet doivent être clairs, réalisables et motivants, selon le principe des objectifs SMART,
- Les comportements au sein de l'équipe projet doivent être bienveillants et doivent respecter les règles éthiques
- Le management de l'équipe-projet doit être motivant et les membres de l'équipe doivent se sentir concernés par les questions de qualité,
- Le processus projet doit être piloté dans une logique d'amélioration permanente : suivi de l'allocation des ressources, indicateurs de performance, et adoption d'actions correctives,
  - Chaque membre de l'équipe-projet doit avoir conscience de sa place et de son rôle dans le projet,
  - Le management de la qualité doit inclure toutes les parties prenantes au projet, y compris externes.



# Objectifs SMART

Acronyme SMART, permettant de définir des indicateurs efficaces :

- *Specific* : Objectifs clairs, précis et compréhensibles de tous
  - *Mesurable* : Objectifs clairement quantifiés
- *Ambitious* : Objectifs suffisamment élevés pour être motivants
  - *Realistic* : Objectifs réalisables pour éviter l'abandon
- *Time-Bound* : Objectifs délimités précisément dans le temps





# Suivi de qualité

## Indicateurs de qualité:

- Satisfaction client
- Nombre d'erreurs
- Nombre de plaintes
- Taux de disponibilité ( période de disponibilité de l'application sur l'année )

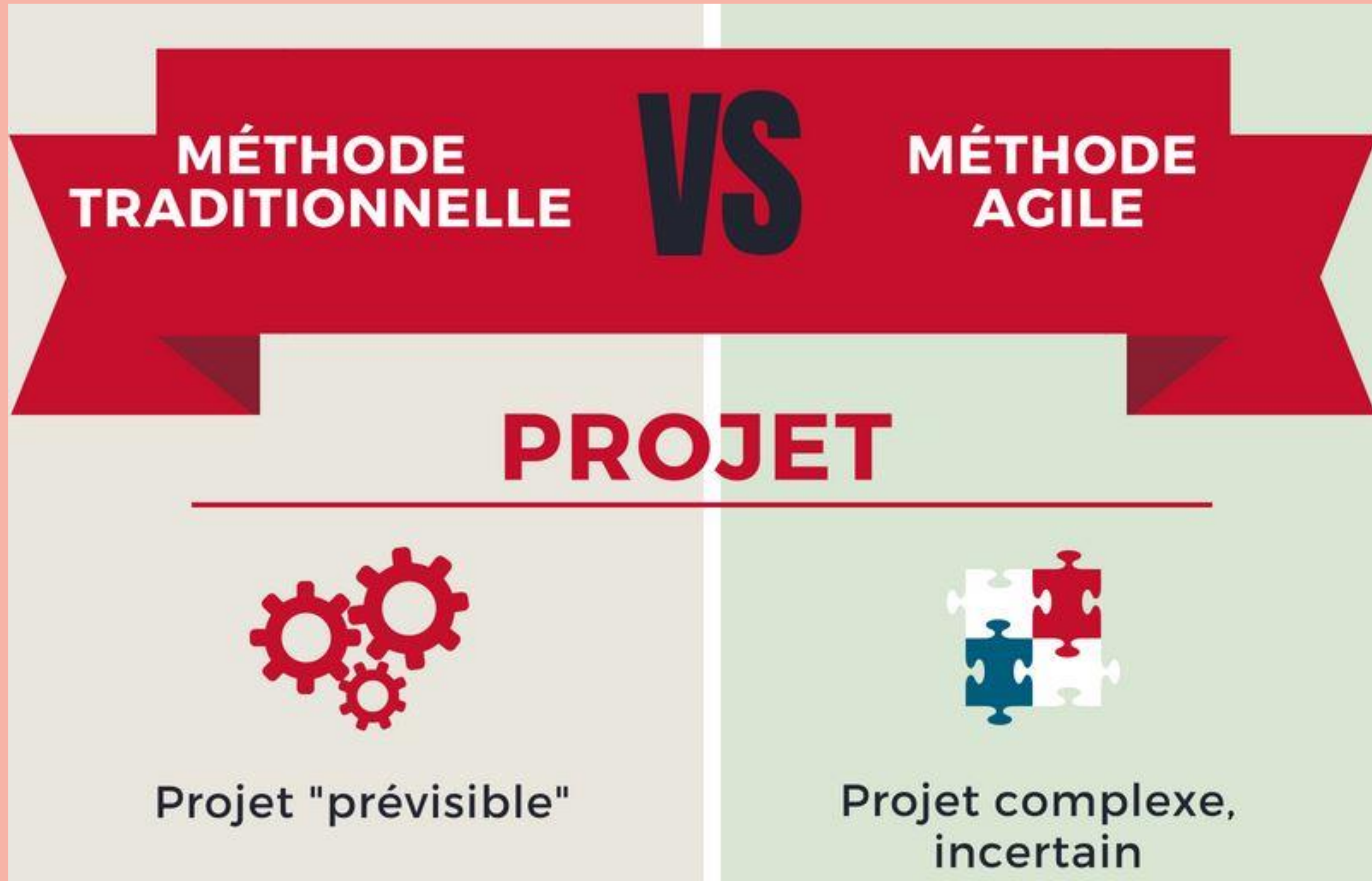


# Exemples d'indicateurs

Indicateurs d'efficacité et d'avancement :

- Taux d'avancement (%) :  $(\text{tâches accomplies} / \text{prévues}) * 100$
- Nombre de tâches / jalons réalisés
- Temps passé sur le projet

# Les méthodes traditionnelles







# Modèle en cascade

Le modèle en cascade est une organisation des activités d'un projet sous forme de phases linéaires et séquentielles où chaque phase :

- Correspond à une spécialisation des tâches
- Dépend des résultats de la phase précédente.



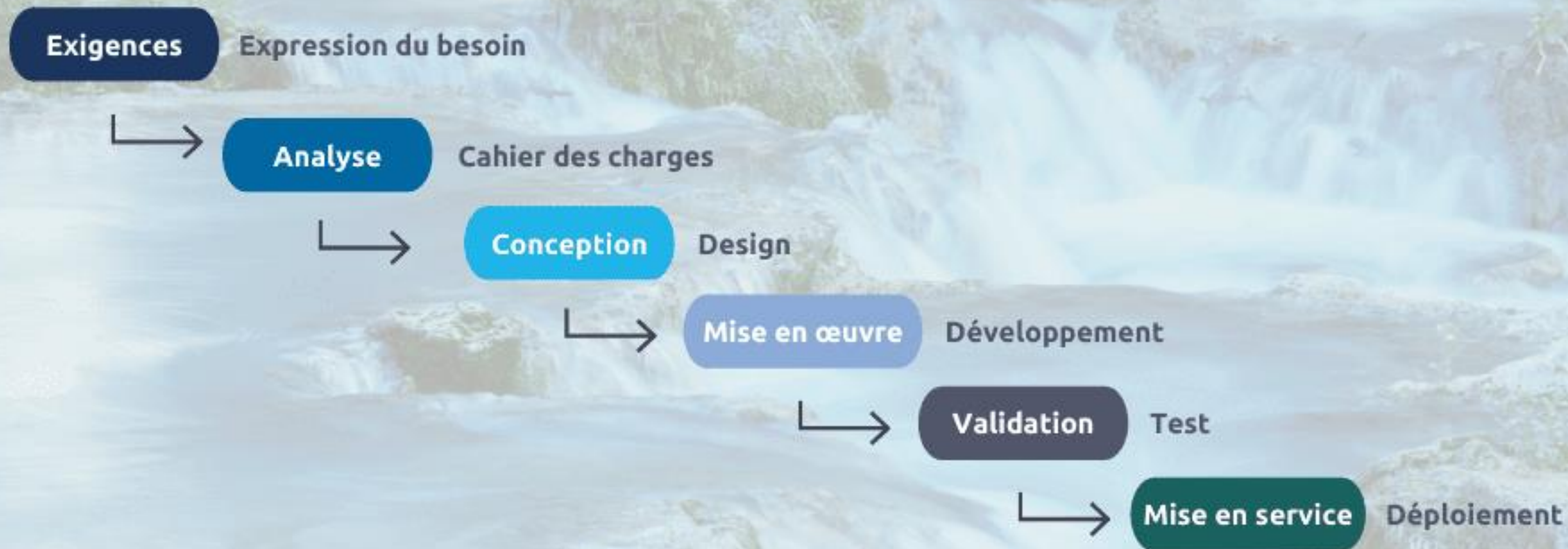
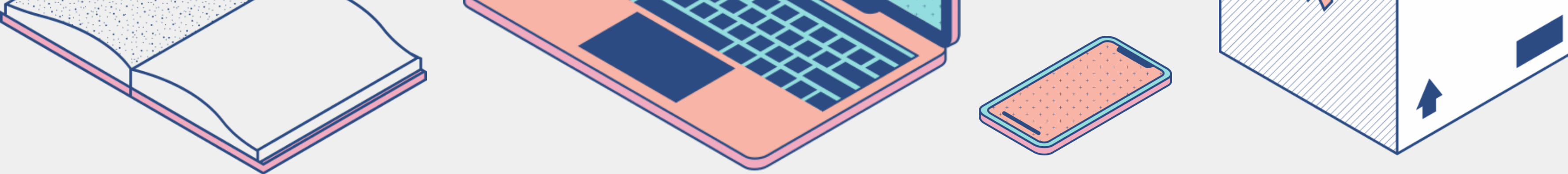
# Modèle en cascade

## Origines

Le modèle en cascade est hérité de l'industrie du BTP, et repose sur les hypothèses suivantes :

- on ne peut pas construire la toiture avant les fondations
- les conséquences d'une modification en amont du cycle ont un impact majeur sur les coûts en aval









# Modèle en cascade

## Phases et livrables

- Exigences : les exigences font l'objet d'une expression des besoins
- Analyse : les exigences sont analysées pour établir un cahier des charges fonctionnel
  - Conception : le produit est conçu et spécifié de sorte à pouvoir être réalisé
    - Mise en œuvre : le produit est réalisé sur la base des spécifications
- Validation : le produit est testé et vérifié et sa conformité aux exigences est validée
- Mise en service : le produit est installé, les préparatifs pour sa mise en service sont organisés, puis le produit est utilisé



# Modèle en cascade

## En pratique

- En pratique, ce modèle n'est plus très utilisé car peu réaliste.
- Cependant, il est à la base des modèles que nous allons aborder par la suite



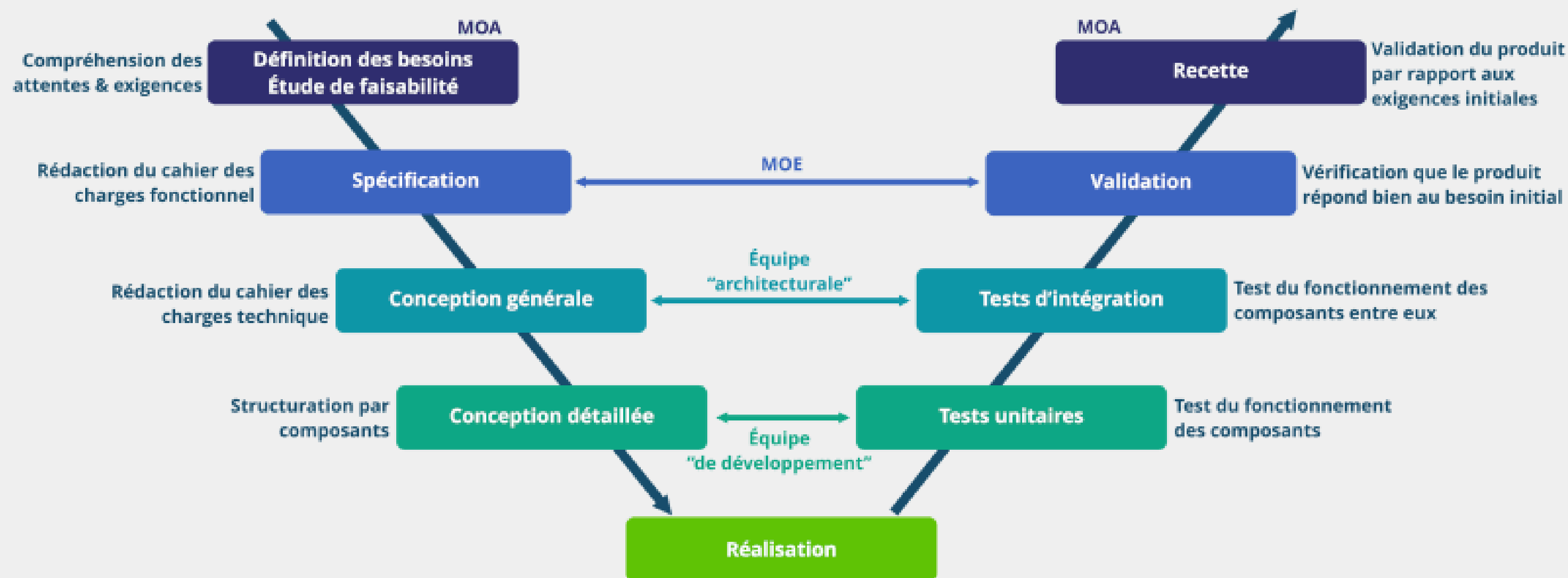
## Cycle en V

Basé sur le modèle en cascade, le cycle en V est :

- un flux d'activité descendant qui détaille le produit jusqu'à sa réalisation
- ET un flux ascendant, qui assemble le produit en vérifiant sa qualité



# Cycle en V



© www.managem-ga.com



# Cycle en V

## Origine :

Issu de l'industrie, le cycle en V est une alternative plus réaliste au modèle de cascade. Il reprend une approche séquentielle et linéaire des phases, mais l'enrichit avec :

- Des étapes d'intégration du système à partir de briques simples
- Chaque phase de production est associée à sa phase de validation



# Cycle en V

## Flux descendant :

Une première série d'étapes, le flux descendant, vise à détailler le produit jusqu'à sa réalisation.

Il comprend l'expression des besoins, l'analyse la conception et la mise en œuvre.





# Cycle en V

## Flux ascendant :

Une deuxième série d'étapes, le flux ascendant, vise à valider le produit jusqu'à sa recette. Il comprend principalement une série de tests, jusqu'à pouvoir valider que le produit répond au besoin et aux exigences.



# Cycle en V

## Etapas du modèle :

- Exigences : objet d'une expression des besoins. Le cas échéant, une étude de faisabilité peut être conduite avant l'engagement des travaux.
- Analyse : élaboration du cahier des charges fonctionnel à partir de l'expression des besoins.
- Conception générale: Conception du système répondant aux exigences, et définition de son architecture.
  - Conception détaillée : Conception de l'ensemble des composants de manière à répondre à l'expression des besoins.



# Cycle en V

## Etapas du modèle :

- Mise en œuvre : réalisation de chaque composants nécessaire.
- Tests unitaires : Vérification du bon fonctionnement de chaque composant.
- Intégration et tests d'intégration : Assemblage du système à partir de l'ensemble des composants, et vérification du fonctionnement d'ensemble du système.
- Tests fonctionnels : Tests que le système est conforme aux exigences.
- Recette : Validation du système par rapport aux besoins exprimés.





# Cycle en V

## En pratique :

L'idée est de découper la validation pour distinguer les tests unitaires, d'intégration, et systèmes.

Ce modèle s'adapte bien aux systèmes complexes composés de multiples composants. Le cycle en V est un standard de l'industrie du logiciel, et probablement le modèle de développement le plus connu.



## Cycle en V

### Risques :

Ce modèle est bien adapté pour les projets stables, clairs et bien compris. Cependant, en cas de changement dans les premières étapes du modèle (changement de besoin, incompréhension des spécifications, risques mal évalués), l'entièreté du modèle est impacté.

Une solution est de travailler en versions successives de cycle en V



## Cycle en W

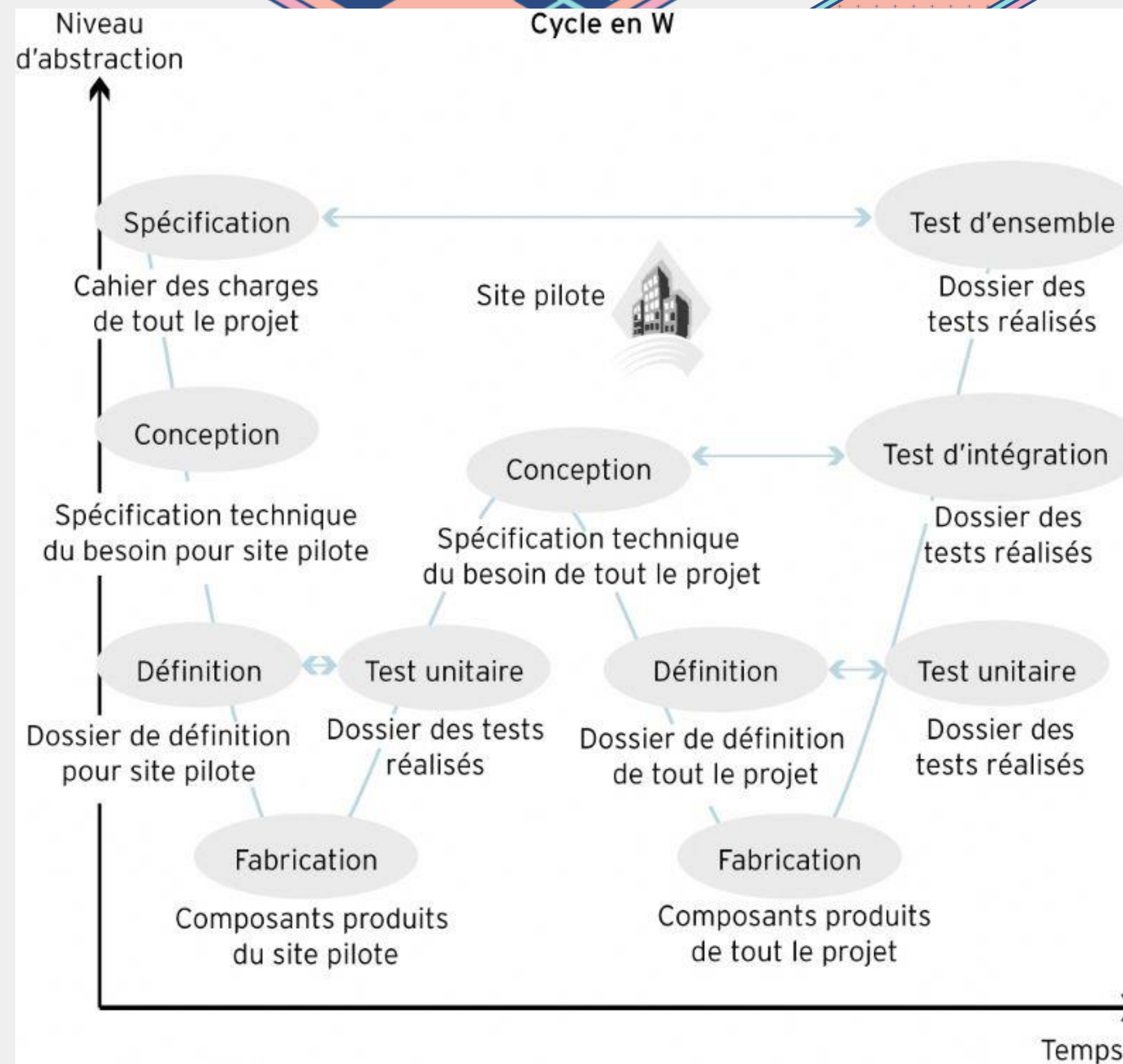
Le cycle en W est un double cycle en V.

Il reprend le principe de ce dernier, en ajoutant la création d'un prototype (1e V) avant la réalisation réelle du produit (2e V).

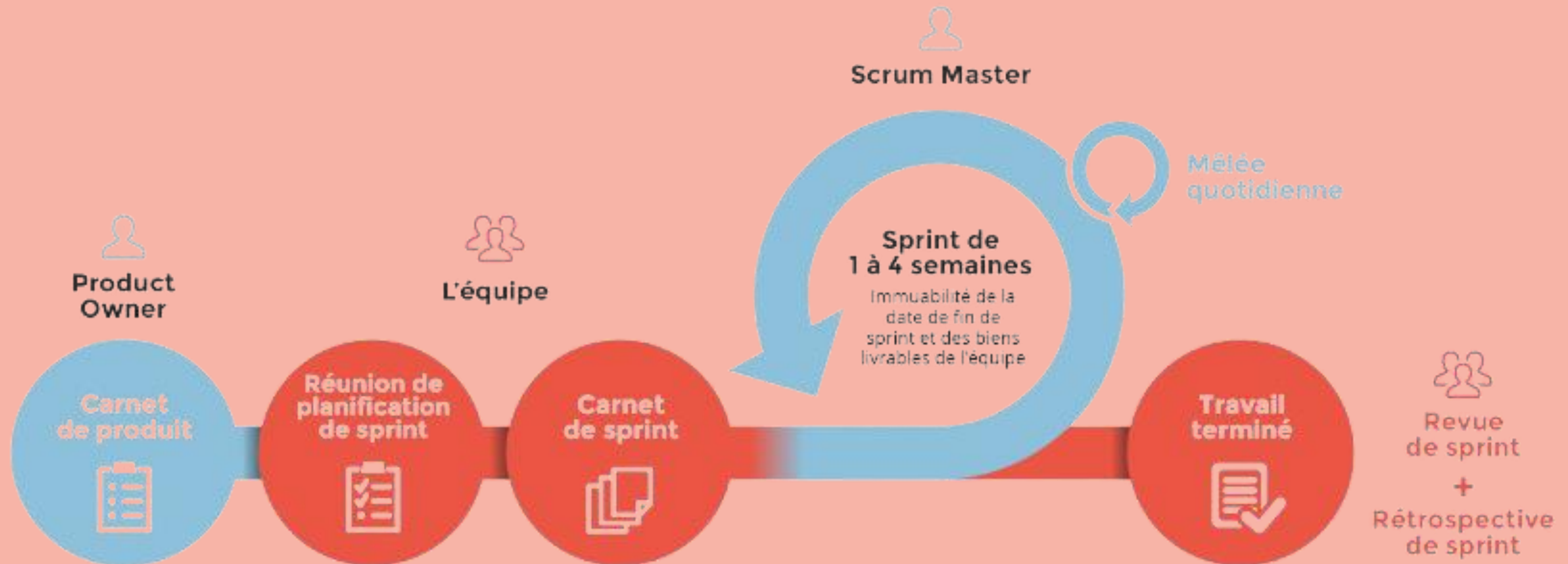
L'intérêt du prototypage est de s'accorder sur les besoins du produit, mieux comprendre les spécifications et les impacts techniques (limite l'effet "tunnel").

La réalisation du prototype induit cependant un coût et une durée de réalisation plus importante.





# La méthodologie Agile





# Méthodologie Agile

L'agilité est une approche de la gestion de projet incluant une méthode de travail et 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 qu'une méthode stricte de travail. Chaque équipe pourra redéfinir sa méthode de travail en utilisant des objectifs, BP et outils fournis par ce Framework.





# Manifeste agile

Le manifeste agile [agilemanifesto.org](http://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.*



# Approche empirique

Au contraire 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 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) et estime le coût de chaque élément de la liste.

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





# Méthodologie Agile

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.



# Méthodologie Agile

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 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. Il économise ainsi son budget et récolte 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).

Il peut ainsi retarder une fonctionnalité dont le besoin n'est pas mûr, ajouter une nouvelle fonctionnalité cruciale en échange du retrait d'une autre ...

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



# Méthodologie Scrum







# Méthodologie Scrum

Scrum est de loin la méthode agile la plus utilisée.

C'est un modèle complet pour gérer des projets à la fois complexes et imprévisibles.

Elle repose sur trois piliers :

- Transparence
  - L'ensemble des acteurs (client, développeurs ...) doit avoir à toutes les informations
- 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.



# 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.
- 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
  - Un développement incrémental : Chaque itération doit être utilisable
- Une pratique agile : Le client et les utilisateurs finaux participent à la gestion du projet





# Les rôles Scrum

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

- Le Product Owner (PO) qui porte la vision du produit à réaliser
- Le Scrum Master, garant de l'application de la méthodologie Scrum
  - L'équipe de développement réalisant le produit

*Ces rôles peuvent être attribués à la même personne*



# Le Product Owner

Les méthodes agiles placent le client au centre des négociations.  
Ce dernier n'est cependant pas toujours disponible ou qualifié pour cela.  
Le Product Owner 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 (!= développeur)
- L'initiateur du projet s'il n'y a pas de client



# 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.

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 !*





# Le Scrum Master

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.



# L'équipe de développement

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.

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



# L'équipe de développement

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.





## 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.

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.

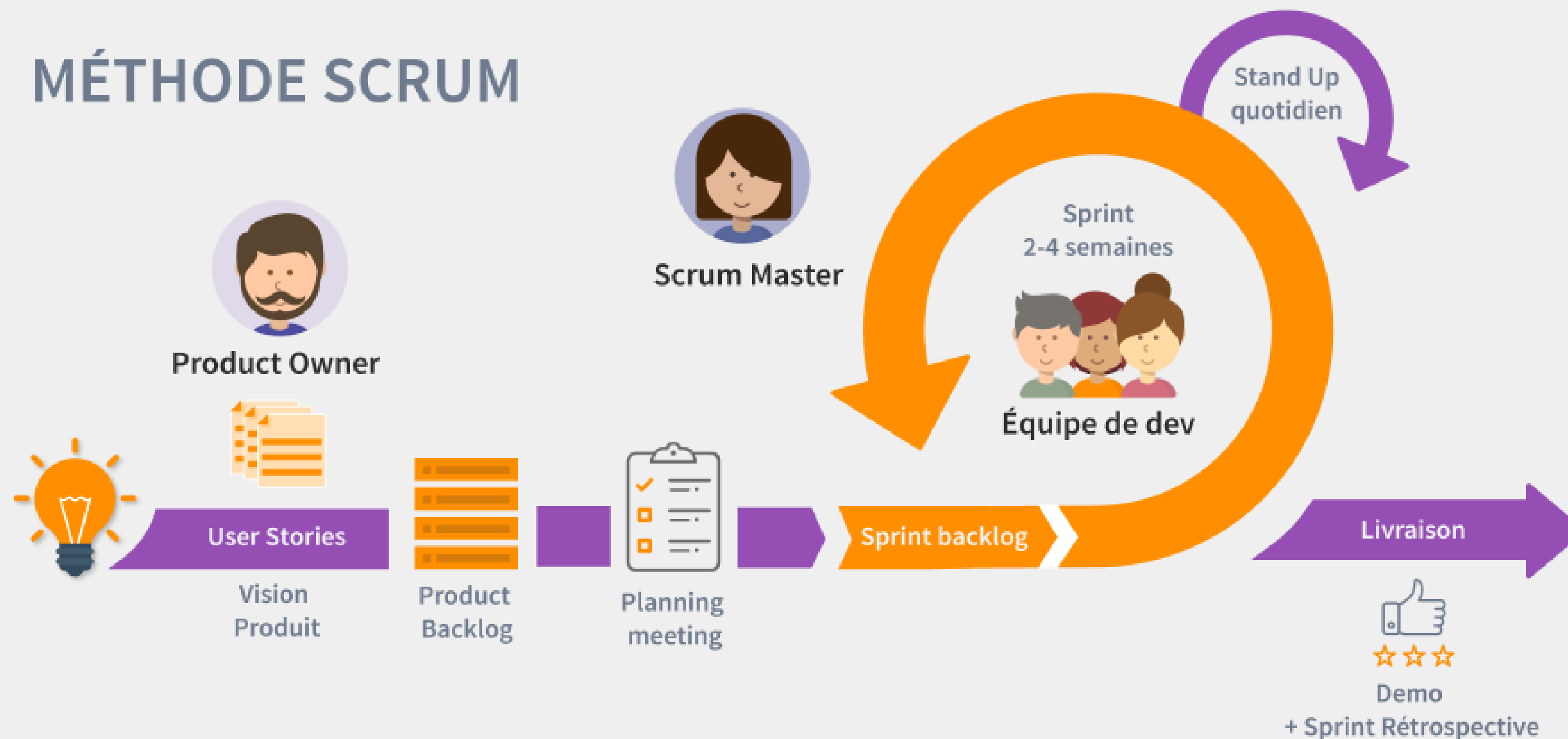


## Les rôles annexes

- 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.

# Scrum en pratique

## MÉTHODE SCRUM







# 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.



# User Stories

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 Story 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)



# 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.





# 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.



# 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.



## 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.





## Stand-Up

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.*



## 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.



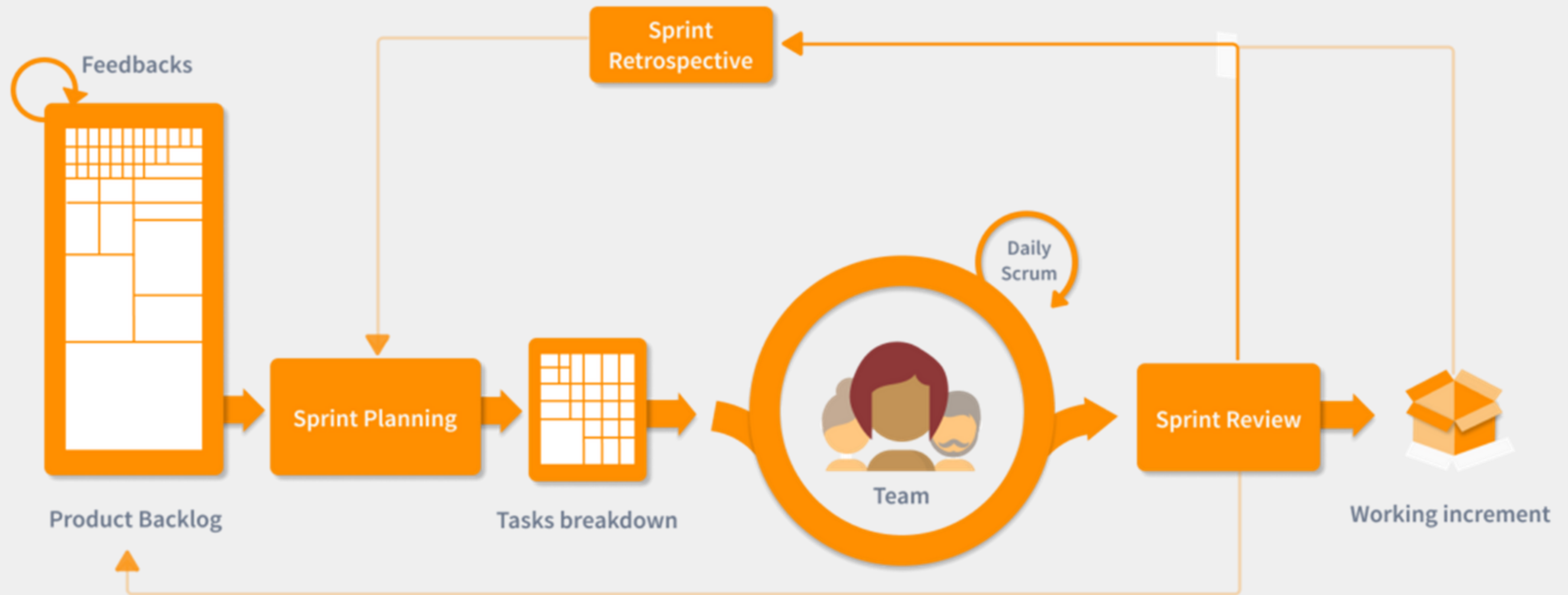
# Rétrospective

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





# Story mapping

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

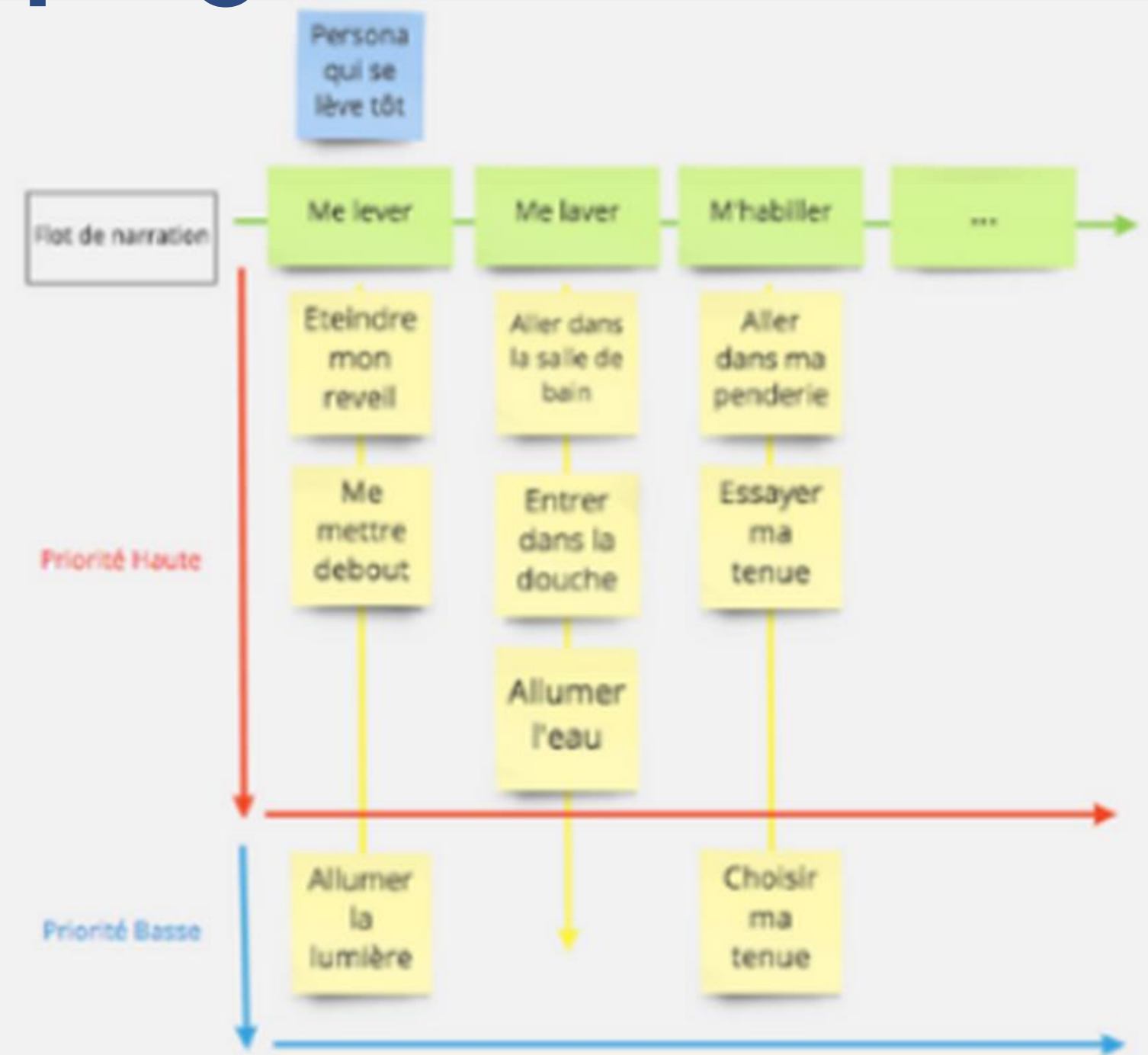
- l'axe horizontal représente des flots de narration : un utilisateur type du produit effectue un ensemble de grandes actions l'une après l'autre (ce sont des Epics)
- l'axe vertical représente les User Stories associées aux Epics de la narration.

Ces User Stories sont triées par priorité.

# Story mapping

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).







# 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.



# Kanban

*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..*

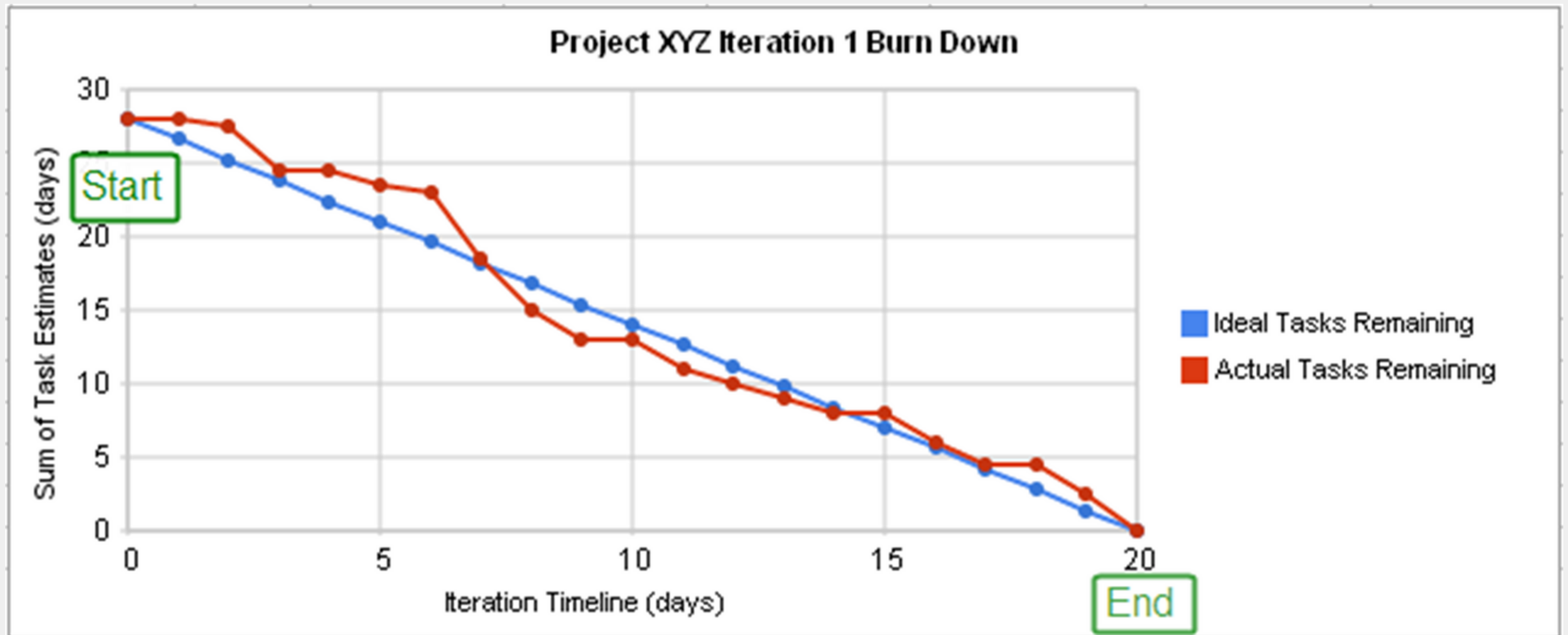


# Kanban





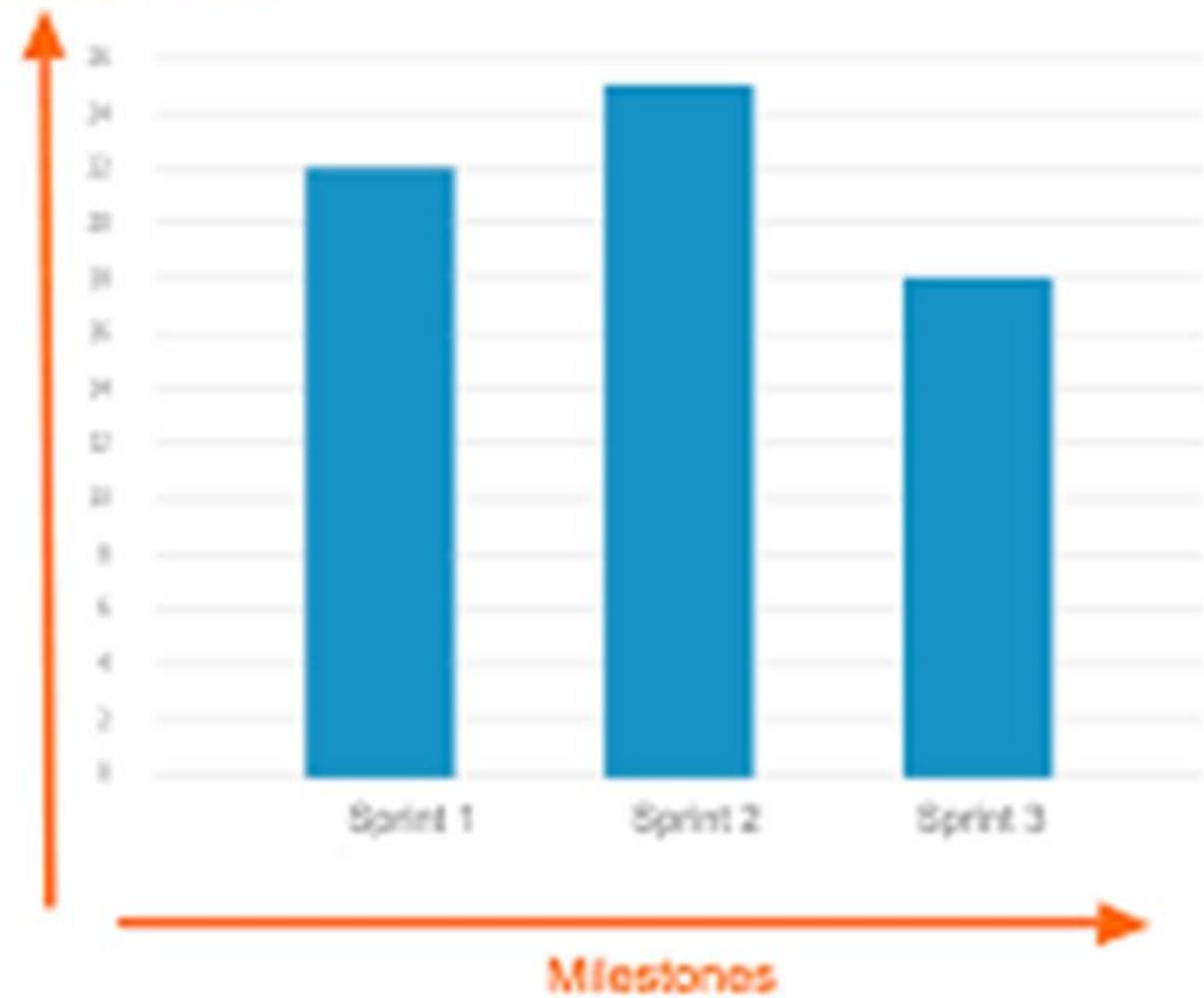
# Indicateurs : Burndown



# Indicateurs : Vitesse

VELOCITY CHART

Amount of work



*La vitesse 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 vitesse aide à la planification, et non à indiquer la performance d'une équipe !*



## Difficultés

*Scrum est easy to learn, hard to master* : 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



# Collaboration à distance

93





# Les télécommunications et le travail collaboratif

- Internet a été créé pour la collaboration
- Apparition des outils collaboratifs en 80 : mails en interne, agendas partagés..
- Intranets en 90 : converser avec ses collègues, systèmes fermés et contrôlés
- Intranets en 2000 : accessibilité et facilité d'utilisation, contribution, blogs, réseaux sociaux...
- Pandémie Covid19 : généralisation du télétravail, collaboration fortement numérique => outils collaboratifs en croissance exponentielle



# La dynamique d'action collective

*Chacun d'eux par lui-même ne peut pas être de bonne qualité ; mais quand ils viennent tous ensemble, il est possible qu'ils puissent surpasser - collectivement et en tant qu'ensemble, mais pas individuellement - la qualité des quelques meilleurs. Aristote, La Politique. Livre III*





# La dynamique d'action collective

*Se réunir est un début ; rester ensemble est un progrès ; travailler ensemble est la réussite.*  
*Henri Ford*



# La partage

Le partage est la valeur clé du travail collaboratif.  
Ensemble, les membres de l'équipe partagent :

- un projet avec des objectifs communs à atteindre
  - des connaissances et des compétences
  - des ressources
- des outils pour coordonner et planifier la réalisation des objectifs
  - des valeurs pour créer des liens et développer la confiance



# L'organisation horizontale

Une structure de travail horizontale prend le contrepied du modèle hiérarchique traditionnel, qui repose sur une verticalité dans les rapports de travail, où ce sont la direction et les managers qui prennent les décisions.

Ces décisions sont ensuite suivies par les collaborateurs dont le niveau hiérarchique est inférieur.





# L'organisation horizontale

Dans une organisation horizontale (ou transversale), le travail se fait par équipes ou projets, dans des cercles de travail, où tous les collaborateurs participent au même niveau. Tout repose sur la coopération et le manager et le chef de projet deviennent des « facilitateurs ».



# L'organisation horizontale

L'idée derrière cette forme d'organisation est que les travailleurs sont bien plus productifs lorsqu'ils sont impliqués directement dans les processus de prise de décision plutôt qu'en étant simplement supervisés par plusieurs niveaux de management.

Cette nouvelle approche de management s'applique bien à de petites structures, de type start-ups, mais peut tout à fait trouver du sens dans des organisations plus importantes, pour faciliter la mise en place de modes de travail collaboratifs.



# L'organisation horizontale

## Avantages :

- La flexibilité est bien supérieure : plus besoin de remonter tous les palliers pour faire changer les choses
  - La cohésion est renforcée grâce à la communication entre tous les acteurs
- Cela permet d'augmenter l'implication des salariés et de les fidéliser : ils sont désormais partie prenante dans le développement de certaines stratégies





# L'organisation horizontale

## Avantages :

- Permet une meilleure transparence auprès des équipes
- Très performant pour les projets transversaux, où un plus grand nombre d'expertises est mis à contribution que dans un modèle vertical



# L'organisation horizontale

## Limites :

- Attention à toujours fixer un cadre de travail (organisation, règles morales et éthiques), la disparition de hiérarchie n'est pas synonyme d'anarchie !
- Les instances autrefois dirigeantes (manager, chef de projet, ...) doivent être formées au métier de facilitateur
  - Les équipes doivent être capables de travailler en autonomie
- L'organisation horizontale peut parfois entraîner un manque de vision claire. Pour éviter cela, les décideurs finaux doivent être clairement identifiés (SPOC : Single Point of Contact).

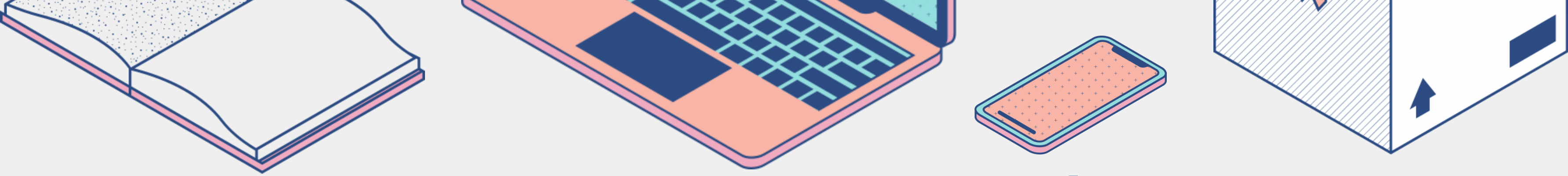


# L'interaction => coopération, productivité, innovation

Dans une gestion de projet classique :

- le chef de projet orchestre le travail des membres de l'équipe.
  - Il distribue les tâches
  - s'assure qu'elles sont bien réalisées dans les temps.
- chaque collaborateur travaille :
  - individuellement
  - et rapporte directement au chef de projet.





# L'interaction => coopération, productivité, innovation

Chaque tâche étant clairement identifiée et séparée des autres, les membres de l'équipe sont peu amenés à communiquer entre eux.

Les réalisations individuelles sont regroupées afin de construire le produit en cours de réalisation.

Le travail de chacun est clairement identifié.

Chaque membre de l'équipe porte la responsabilité individuelle du travail qu'il a fourni.



# L'interaction => coopération, productivité, innovation

Dans un projet collaboratif, l'organisation est radicalement différente :

- Chaque tâche peut avoir un lien avec les autres
- Chacun ajoute en permanence ses réalisations au produit construit en commun
  - Les modifications sont testées et contrôlées en temps réel par tout le monde
- Les membres de l'équipe sont amenés à communiquer en permanence entre eux afin de se coordonner et de s'entraider au besoin
  - Chacun gagne en autonomie et son travail est valorisé



# L'interaction => coopération, productivité, innovation

- Le pouvoir de décision n'est plus centralisé :
  - chacun possède le même niveau d'information
    - chacun peut apporter son avis sur :
      - le déroulement du projet
      - les tâches à réaliser en priorité
      - la répartition du travail.
- On passe d'une responsabilité individuelle à une responsabilité partagée  
=> la coopération doit primer sur la compétition



# Outils collaboratifs

108





# Les outils de collaboration à distance

Ces outils vont permettre à tous de diffuser librement des informations :

- présentations
- documents
- tableaux
- fichiers audios et vidéos

C'est la base de connaissances commune à toute l'équipe !



# Les outils de collaboration à distance

## Exemple :

À l'aide d'un tableau partagé de type Kanban , chacun pourra suivre l'évolution des tâches qui lui sont attribuées, mais également de celles des autres membres de l'équipe.

Cela permet à l'équipe d'avoir une vision d'ensemble du déroulement du projet afin de mieux décider de la suite, des priorités et des actions à mettre en place.





# Les outils de collaboration à distance

## Les outils de gestion du temps :

La gestion du temps est également très importante et le partage d'un calendrier commun va permettre à tous de se tenir au courant des différentes dates butoirs, des événements (réunions, comités de suivi, livraisons, formations, présentations...)



# Les outils de collaboration à distance

## Les autres outils

D'autres outils plus spécialisés (suivi, gestion des anomalies, affichage des absences...) peuvent être mis en place afin de partager toujours plus d'informations, d'augmenter l'autonomie de l'équipe et de faciliter la prise de décision.

L'utilisation d'outils tels que les réseaux sociaux, les messageries instantanées ou les applications hébergées dans le cloud permet également de faire participer à un même projet collaboratif des personnes éloignées géographiquement.



# Les outils de collaboration à distance

Comment choisir un outil collaboratif ?

Le choix des outils utilisés dans le cadre d'un travail collaboratif peut grandement modifier la productivité et la motivation de l'équipe projet :

ces outils sont donc à choisir avec soin !





# Les outils de collaboration à distance

## Comment choisir un outil collaboratif ?

Pour cela, on étudiera les différents paramètres de l'équipe et du projet :

- taille de l'équipe
- périmètre du projet (projet court et simple vs projet long et complexe)
- autonomie de l'équipe vs hiérarchie fortement verticale
- ...



# Les outils de collaboration à distance

Outil unique complet vs multiplication d'outils spécialisés

Il existe 2 grands types d'outils collaboratifs :

- les outils généralistes
- les outils spécialisés dans une tâche unique



# Les outils de collaboration à distance

Les premiers ont l'avantage d'intégrer facilement tous les besoins de l'équipe mais leur généricité peut les limiter ou les rendre exponentiellement complexes (Jira, ...).

Les seconds ont l'avantage de se focaliser sur un besoin précis et de l'optimiser (Slack, philosophie Unix).

Leur contrainte est une difficulté à intégrer tous les composants pour avoir une vue unifiée du projet.





# Les outils de collaboration à distance

Outil spécifique au projet vs outil commun pour l'entreprise

Il existe également 2 visions différentes pour le choix des outils :

- rationaliser les outils au sein de l'entreprise
- permettre à chaque équipe projet de choisir le(s) outil(s) qui lui convient le mieux



# Les outils de collaboration à distance

Dans la pratique, il est courant d'utiliser des outils différents dans une dynamique d'innovation (start-up), puis de rationaliser ces outils lors de l'industrialisation des processus de l'entreprise.

Cependant, les méthodes agiles prônant plus de liberté dans l'équipe, la seconde solution est de plus en plus favorisée quel que soit le contexte :  
attention à la multiplicité des outils !



# Les outils de collaboration à distance

Quelques exemples d'outils collaboratifs les plus utilisés :

- Stockage de documents : Google Drive , Dropbox , Microsoft One Drive , Office 365
  - Messagerie instantanée : Slack, WhatsApp, Discord
  - Courriers électroniques : Gmail , Outlook
- Gestion de notes / documentation : Evernote , Notion.so
  - Gestion de tâches : Trello





# Les outils de collaboration à distance

Quelques exemples d'outils collaboratifs les plus utilisés :

- Management visuel : Mural
- Gestion de projet, planification : JIRA , Monday.com
- Plateformes de développement : Github , Bitbucket , Gitlab
  - Vision conférence : Skype , Teams , Zoom
- Outils facilitant l'engagement des collaborateurs : Klaxoon , Miro , Loomio



# Les outils de collaboration à distance

## En résumé :

- Prendre le temps d'étudier les besoins de l'équipe et du projet, être force de proposition pour améliorer les processus de communication
- Dans une méthodologie horizontale, l'équipe projet doit être autonome et choisir ses outils collaborativement (en respectant les contraintes de l'entreprise)
- La multiplication des supports permettent différents niveaux de communication (spontanéité de la messagerie instantanée, rigueur du mail, ...)



# Les outils de collaboration à distance

## En résumé :

- Attention à ne pas trop multiplier les outils...
- ...Mais couvrir les besoins d'une dynamique collective : suivi, coordination, collaboration, partage, visibilité et transparence
- Un outil ne remplace pas une méthodologie

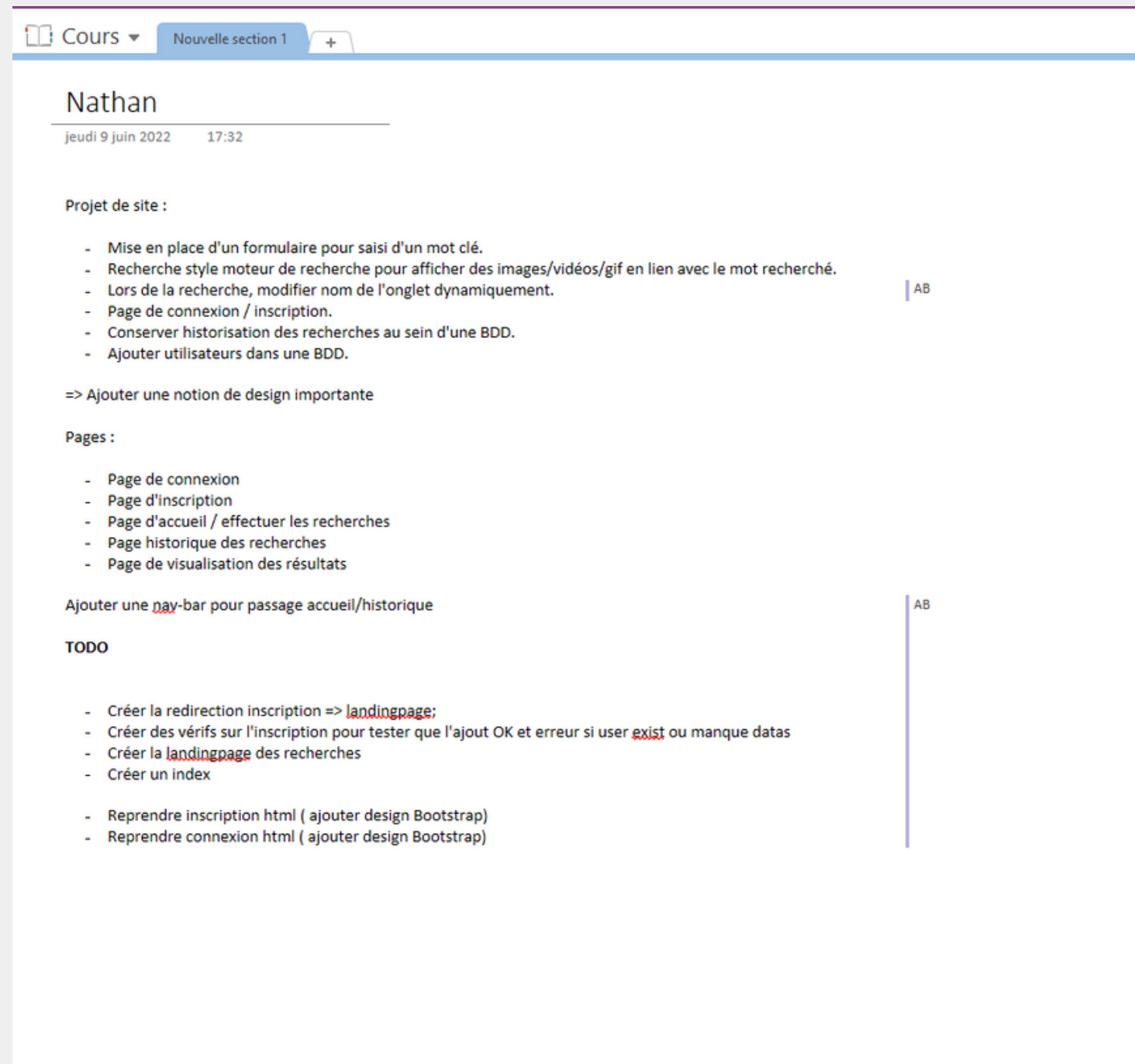


# Evernote : Gestion de notes

The screenshot displays the Evernote web interface. On the left is a dark sidebar with navigation options: Accueil, Raccourcis, Notes, Tâches, Carnets de notes, Étiquettes, Partagé avec moi, and Corbeille. The top header is blue and includes a user profile (Benoît AVENEL), a search bar, and a trial banner for Evernote Personal. The main content area shows a greeting 'Bonsoir Benoît !' and the date 'LUNDI 27 FÉVRIER 2023'. Below this is a 'NOTES' section with tabs for 'Récent' and 'Suggestion'. A large button 'Créer une nouvelle note' is prominent. To the right is a 'BLOC-NOTES' section with a text input field. At the bottom is a 'CAPTURÉ RÉCEMMENT' section with tabs for 'Captures web', 'Images', 'Documents', 'Audio', and 'Emails'. It features a green icon of a document being captured and a button 'Capturer du contenu Web'.

Outil de partage de notes  
collaboratif.  
Gratuit pour un plan personnel.  
Très simple d'utilisation.

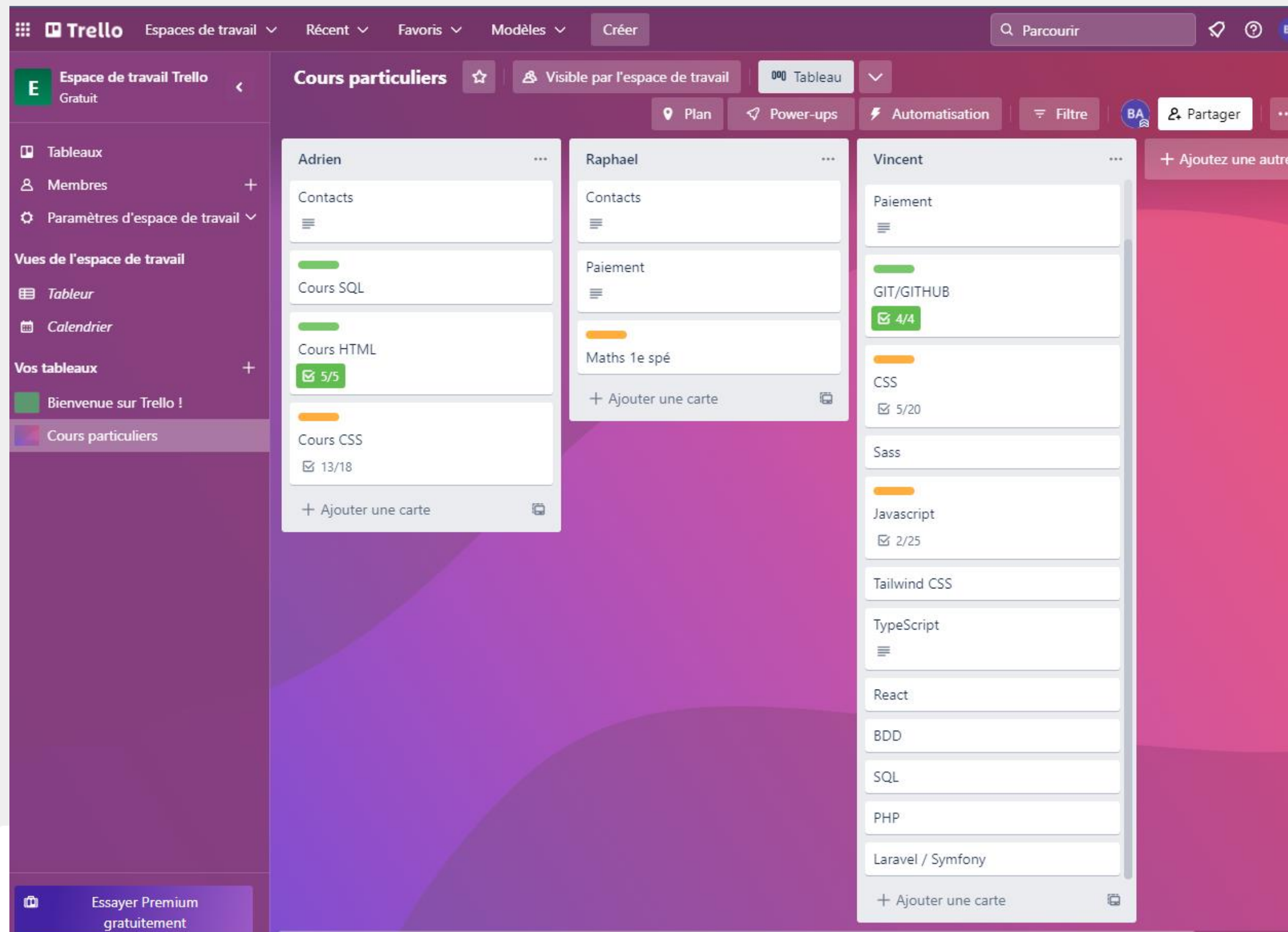
# OneNote : Gestion de notes



Outil de partage de notes  
collaboratif.  
Inclus avec la suite office.  
Très simple d'utilisation.



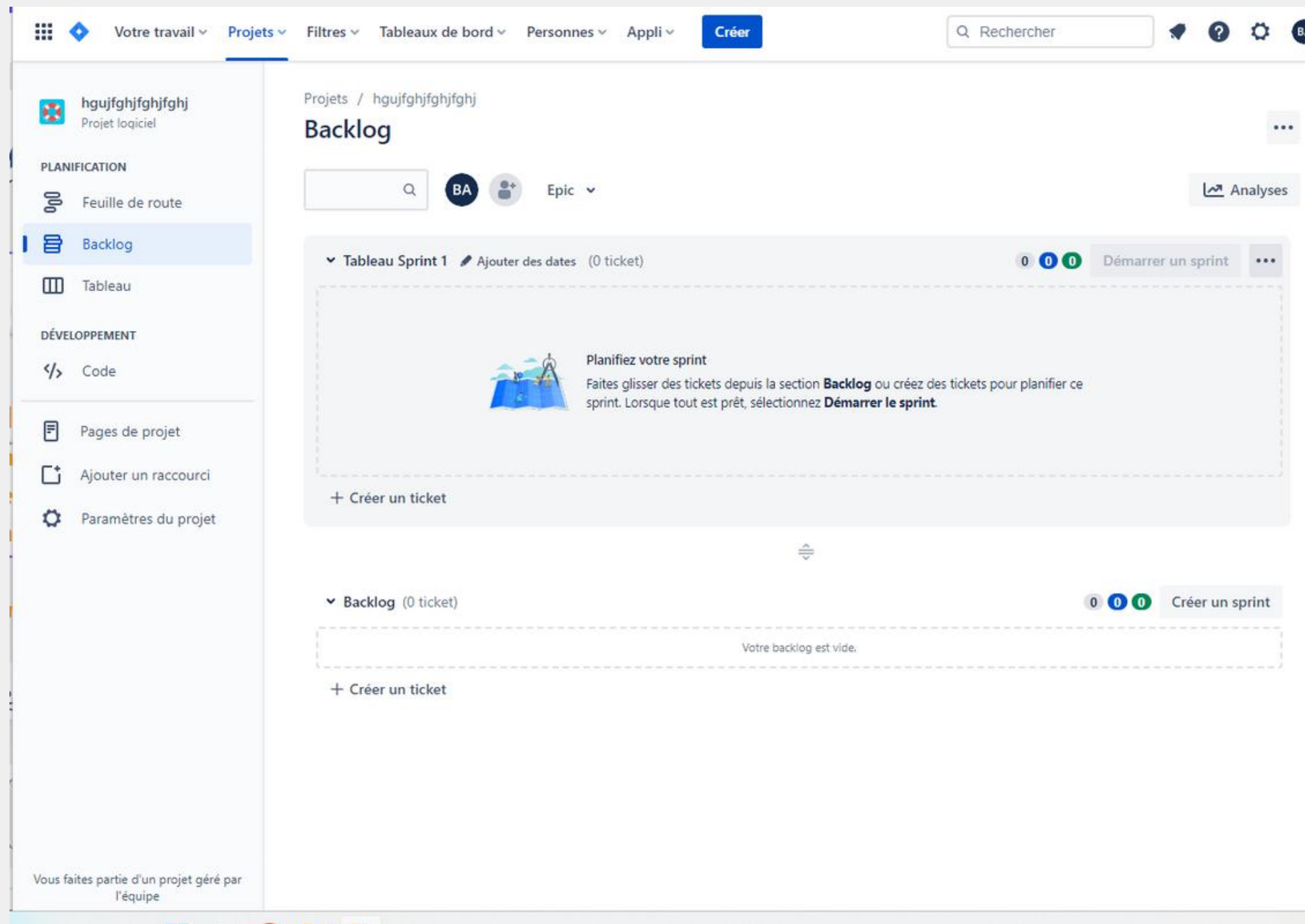
# Gestion de tâches : Trello



Outil de gestion de tâches.  
Fonctionne sous forme de tableaux  
configurable.  
Gratuit pour les offres personnelles  
et petites équipes.



# Gestion de projet : Jira



Outil de gestion de projet.  
Configuration automatique en  
fonction du besoin.  
Connectable à de nombreux autres  
outils.  
Gratuit jusqu'à 10 membres.