



# Initiation aux méthodes agiles

08/10/2024

# Objectifs du cours

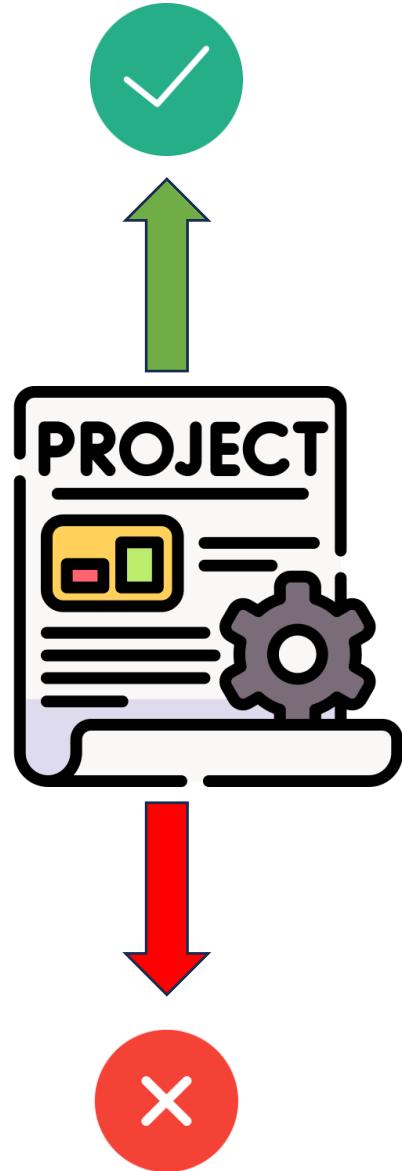
- ✓ Comprendre les concepts fondamentaux des méthodes agiles
- ✓ Appréhender les avantages de l'agilité
- ✓ Comparer les méthodes agiles aux méthodes traditionnelles
- ✓ Acquérir les bases pour intégrer un projet agile





Un peu d'histoire





# Un peu d'histoire...

Méthodes traditionnelles

Gestion de projets

Méthodes "prédictives"

- ✓ Cycle en cascade
- ✓ Cycle en V
- ✓ Cycle en Y
- ✓ Modèle en spirale



Tout est  
planifiable  
avant le début  
du projet



# Un peu d'histoire...

Cycle en cascade

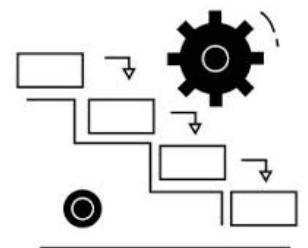


Le cycle en cascade est présenté en 1970 par **Winston W. Royce**, ce modèle est hérité du BTP, où les projets doivent suivre une séquence très précise et rigide.

Ce modèle se base sur deux principes :



Une étape ne peut pas être débutée avec la fin de la précédente



Étapes séquentielles



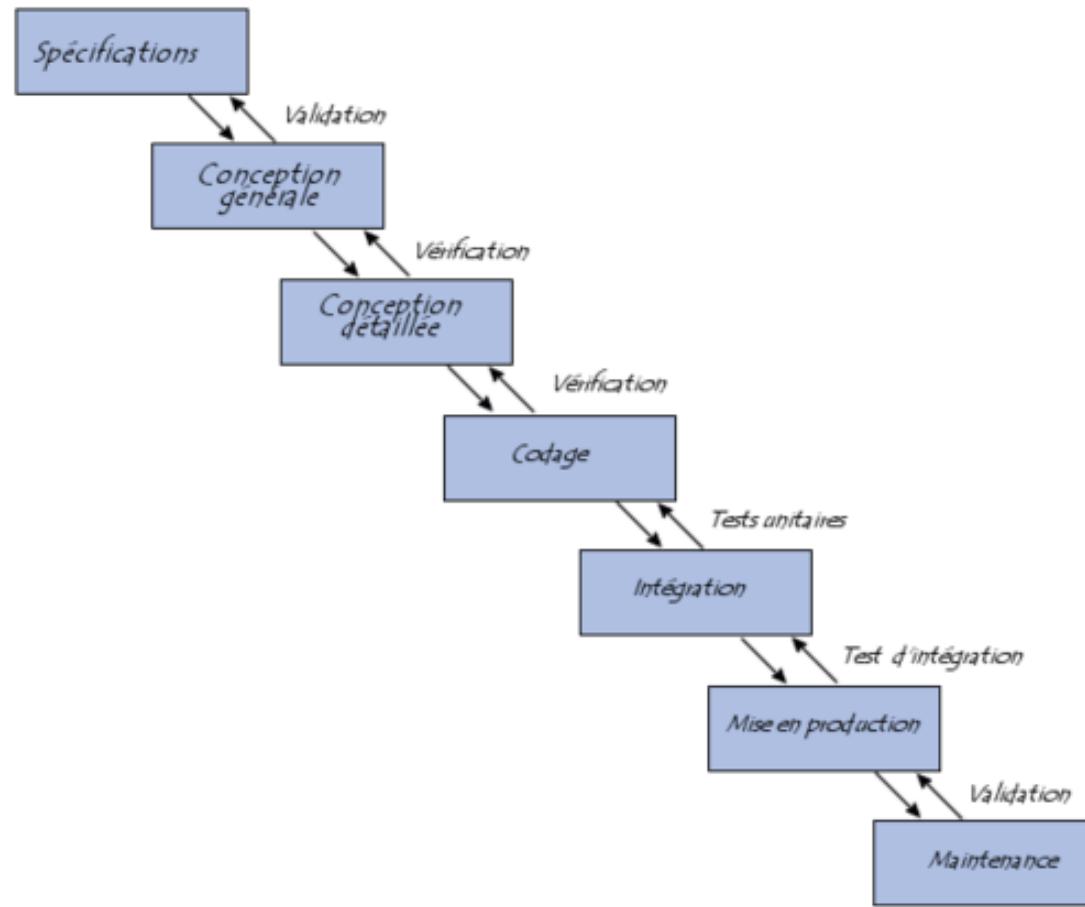
La modification d'une étape du projet a un impact important sur les étapes suivantes



# Un peu d'histoire...

Cycle en cascade

Le modèle en cascade se compose de 7 phases, qui se succèdent :



# Un peu d'histoire...

Cycle en cascade

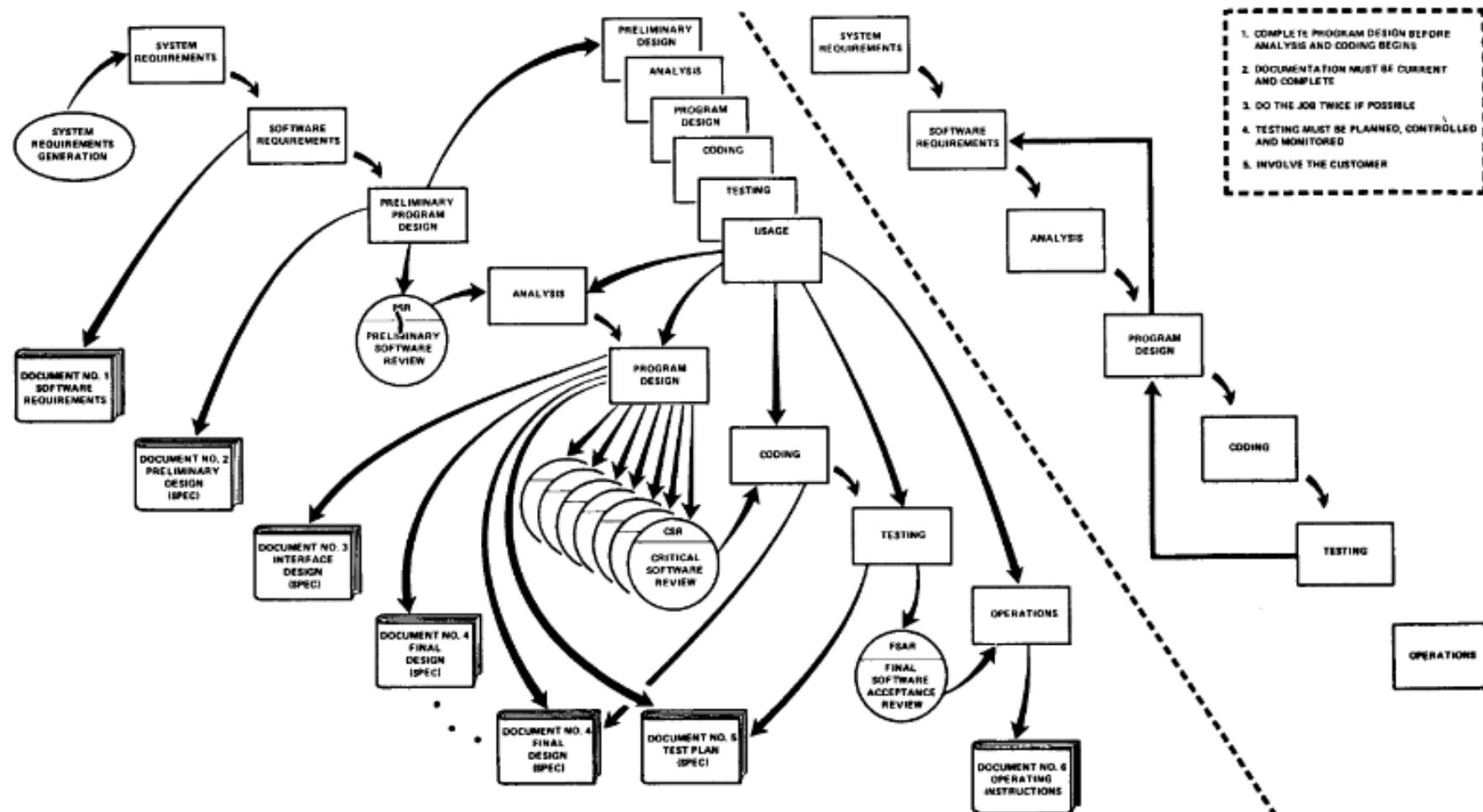


Figure 10. Summary

Module Agilité - eilco



# Un peu d'histoire...

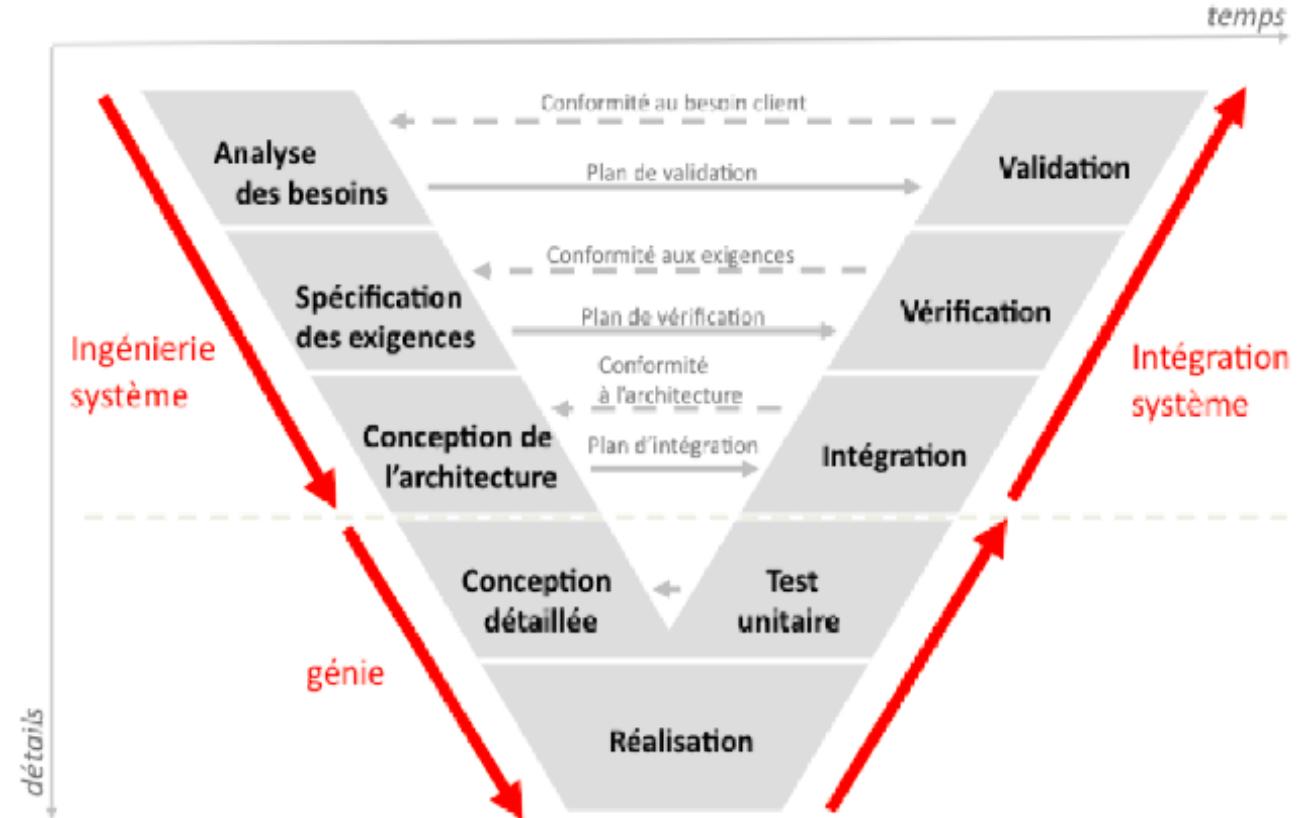
Cycle en V

Le **cycle en V** créé dans les années 80 est une variante du modèle en cascade, qui, tout en s'inspirant de ce dernier, cherche à résoudre certaines de ses limitations majeures

Comment le cycle en V s'inspire du modèle en cascade ?

✓ Le **cycle en V** est une approche **séquentielle**. Chaque phase de conception doit être terminée avant de passer à la suivante

✓ Les étapes de **conception** et de **développement** sont très similaires dans les deux modèles.



# Un peu d'histoire...

Cycle en V

**MOA**

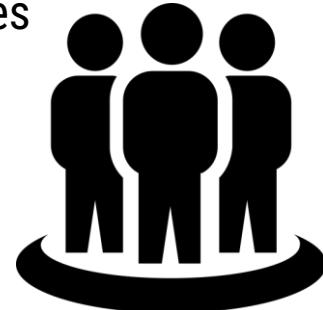
La maîtrise d'ouvrage Elle est l'entité pour qui le projet est réalisée. Elle porte la connaissance métier du projet, c'est à dire à quoi doit répondre la solution. Elle est assistée par l'AMOA.

C'est la MOA qui recueille le besoin auprès des utilisateurs et valide la solution.

C'est le client du projet sans pour autant être l'utilisateur

**MOE**

La maîtrise d'oeuvre, est l'entité qui réalise le projet pour le compte de la MOA.  
Elle est représentée par le chef de projet.



## Le comité de pilotage

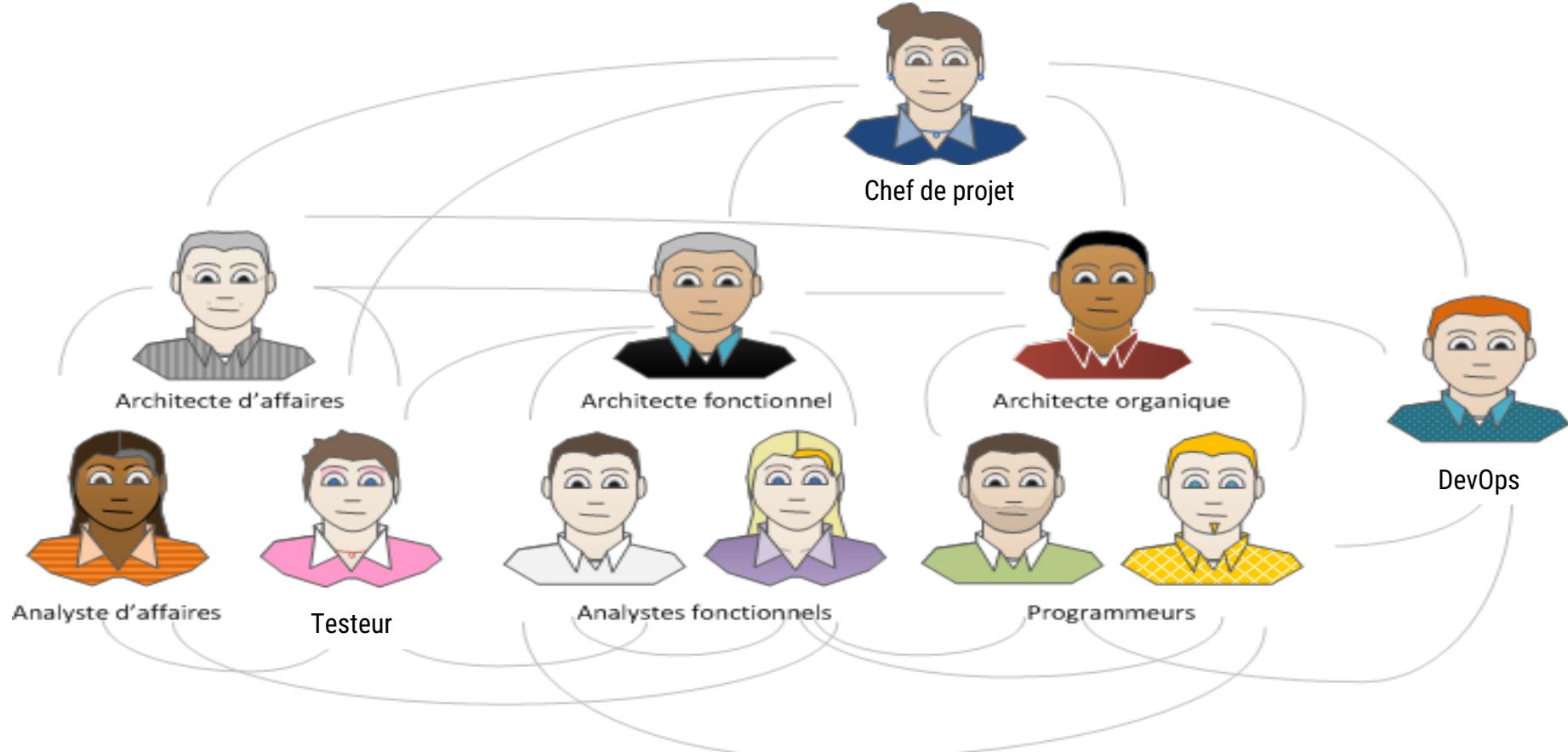
Instance transverse dont le but est de vérifier l'atteinte des différents jalons fixés pour le projet. Il est constitué par un ou des représentants clients, MOA et MOE



# Un peu d'histoire...

Cycle en V

Les différents acteurs du projet ont des rôles et des responsabilités différentes

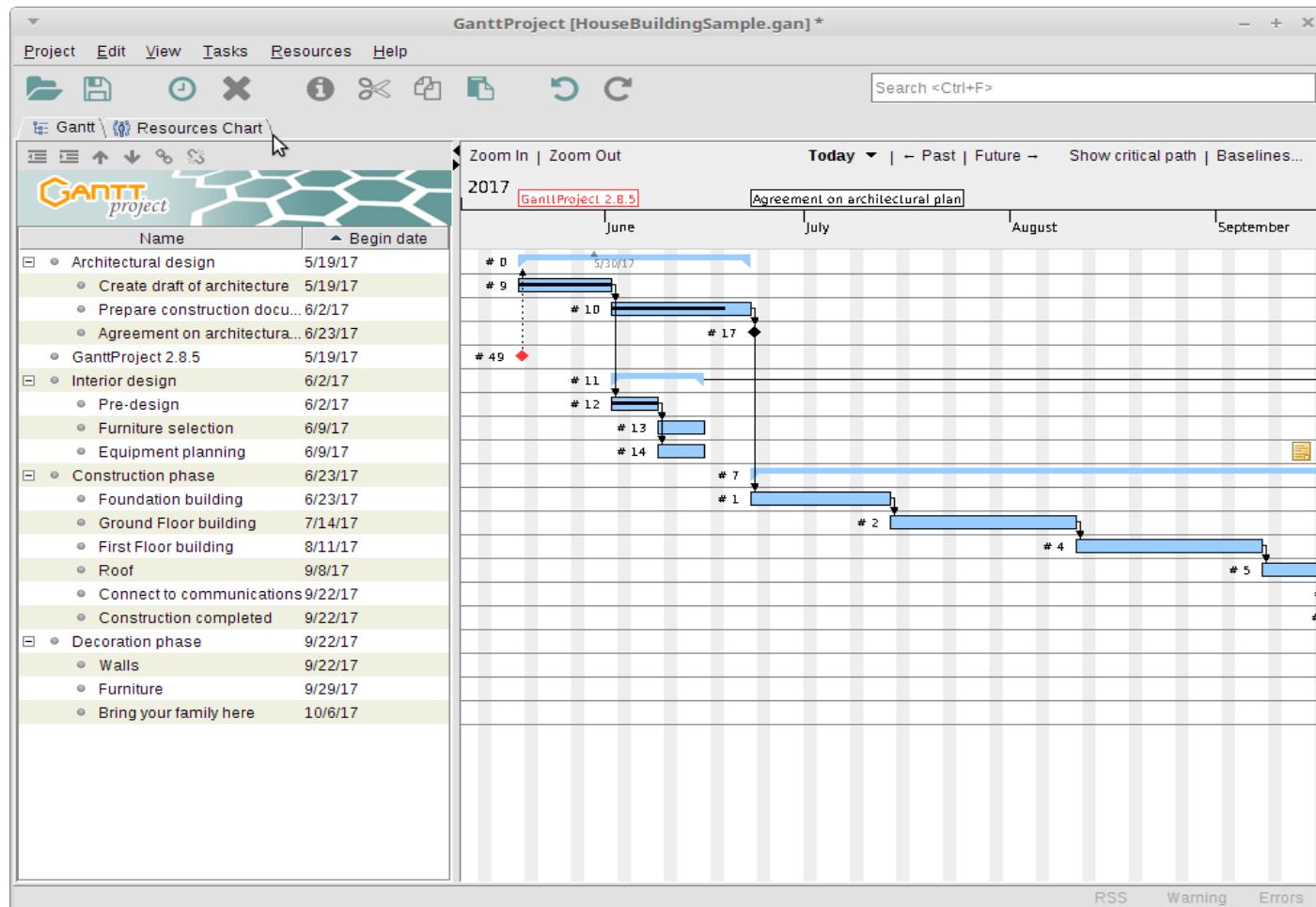


# Un peu d'histoire...

Cycle en V

Exemple de planning réalisé à partir du logiciel GanttProject.

Ici, tout doit être planifié: les dates de début et de fin de réalisation, la personne (ou "ressource") réalisant les travaux.



# Un peu d'histoire...

Avantages et inconvénients

✓ Simplicité du modèle

✓ Le planning est prévu à l'avance

✓ Facilement contractualisable

✓ Documentation complète

✓ Idéal pour des projets à exigences stables



# Un peu d'histoire...

Avantages et inconvénients

- ✓ Faible tolérance à l'erreur
- ✓ Tout est anticipé, aucune possibilité de s'adapter
- ✓ Aucune innovation possible
- ✓ Réalisation qui peut s'éloigner de l'objectif du client
- ✓ Risque de découverte tardive de spécifications fausses, incomplètes



# Un peu d'histoire...

Quizz

**<https://quizizz.com/admin/quiz/67031394e4aded9b7c5da3fd>**





Pourquoi l'Agilité



# C'est quoi l'agilité ?

Le **modèle agile** a été introduit pour répondre aux **limites** des modèles traditionnels comme le cycle en cascade.

L'agilité est une approche de gestion de projet qui favorise la **flexibilité, l'adaptation continue** aux changements et la **collaboration** active avec les parties prenantes. Elle s'appuie sur des cycles itératifs courts (sprints) pour livrer des produits de valeur de manière incrémentale, tout en recueillant régulièrement les retours d'expérience.



# Pourquoi l'Agilité

Le **modèle agile** a été introduit pour répondre aux **limites** des modèles traditionnels comme le cycle en cascade.

L'agilité est une approche de gestion de projet qui favorise la **flexibilité, l'adaptation continue** aux changements et la **collaboration** active avec les parties prenantes. Elle s'appuie sur des cycles itératifs courts (sprints) pour livrer des produits de valeur de manière incrémentale, tout en recueillant régulièrement les retours d'expérience.



# Manifeste pour le développement Agile de logiciels

Bible des équipes agiles

Nous découvrons comment mieux développer des logiciels  
par la pratique et en aidant les autres à le faire.  
Ces expériences nous ont amenés à valoriser :

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 privilégiions les premiers.

# Les 4 valeurs fondamentales



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

- ✓ Équipe idéalement colocalisée
- ✓ Communique au maximum et bonne entente
- ✓ Pluridisciplinaire
- ✓ Collaboration au quotidien
- ✓ Amélioration continue



# Les 4 valeurs fondamentales



## Des logiciels opérationnels plutôt qu'une documentation exhaustive

- ✓ Livraisons fréquentes et incrémentales
- ✓ Documentation légère mais suffisante
- ✓ Améliorations continues basées sur les retours
- ✓ Prototypes utilisables



Fonctionnel et répond au besoin client  
(ni plus ni moins)

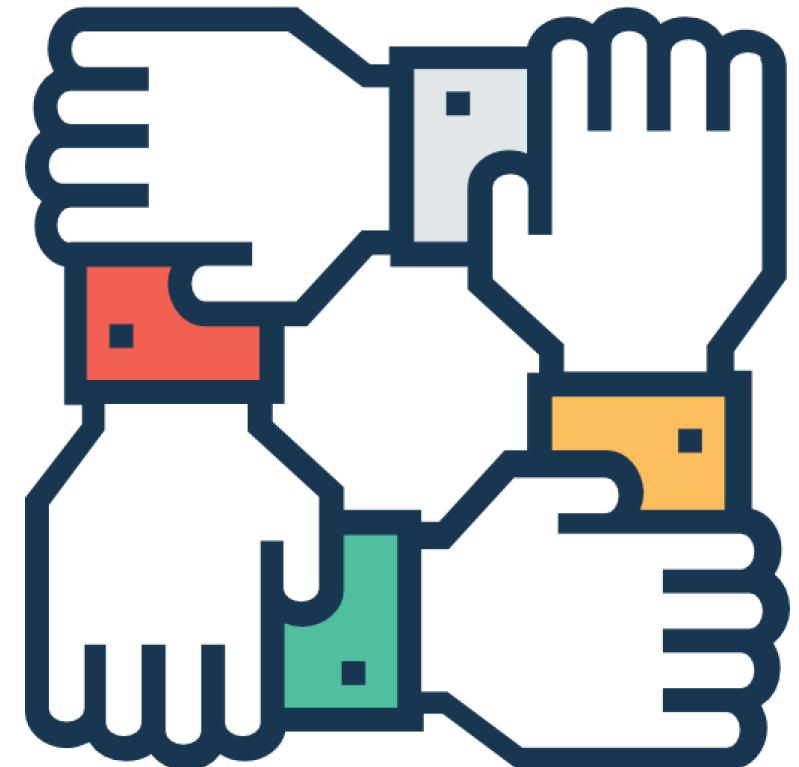


# Les 4 valeurs fondamentales



## La collaboration avec les clients plutôt que la négociation contractuelle

- ✓ Échanges réguliers et constructifs
- ✓ Adaptation aux priorités changeantes
- ✓ Partenariat plutôt que relation fournisseur-client
- ✓ Validation continue

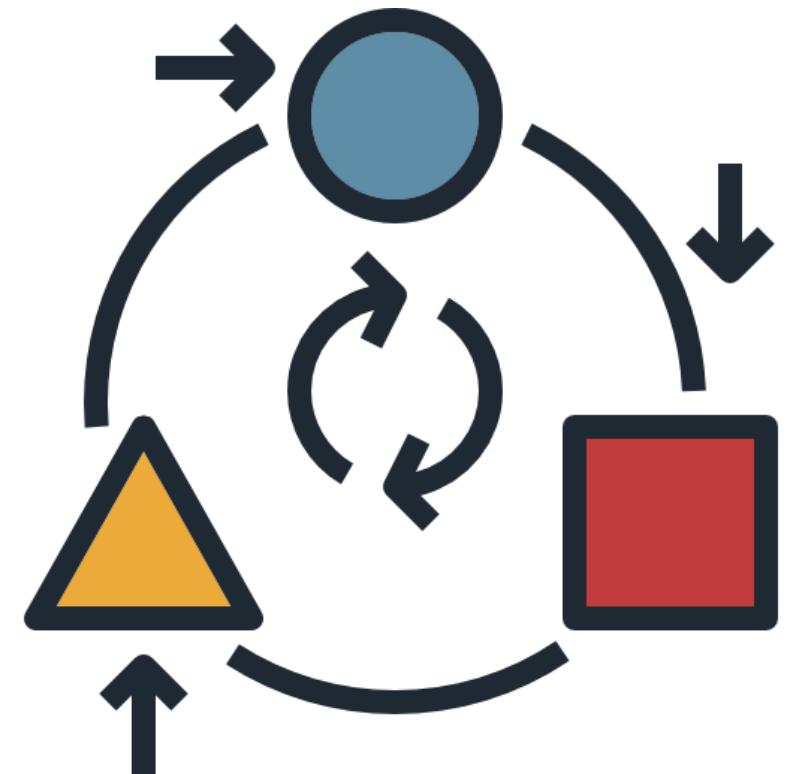


# Les 4 valeurs fondamentales



## L'adaptation au changement plutôt que le suivi d'un plan

- ✓ Revoir les priorités à chaque itération
- ✓ Flexibilité face aux imprévus
- ✓ Anticipation des risques
- ✓ S'adapter rapidement aux nouvelles opportunités
- ✓ Plans flexibles et révisables



# Les 4 valeurs fondamentales

Personnes et interactions

>

Processus et outils

Logiciel qui fonctionne

>

Documentation

Collaboration avec le client

>

Négociation à partir d'un contrat

Adaptation au changement

>

Suivi d'un plan



# Pourquoi l'Agilité ?

## Notre plus haute priorité est de satisfaire le client

en livrant rapidement et régulièrement des fonctionnalités à forte valeur ajoutée.

**A intervalles réguliers,**



## Nous accueillons positivement les changements

de besoin, même tard dans le projet. Les processus agiles exploitent le changement pour donner un avantage compétitif au client.

## Les meilleures architectures spécification et conceptions émergent d'équipes auto-organisées.



## La simplicité



**Une attention continue à l'excellence technique**  
et à une bonne conception renforce l'agilité.



**Les processus agiles encouragent un rythme de développement soutenable.** Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.



 **Les utilisateurs ou leur représentants** et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.

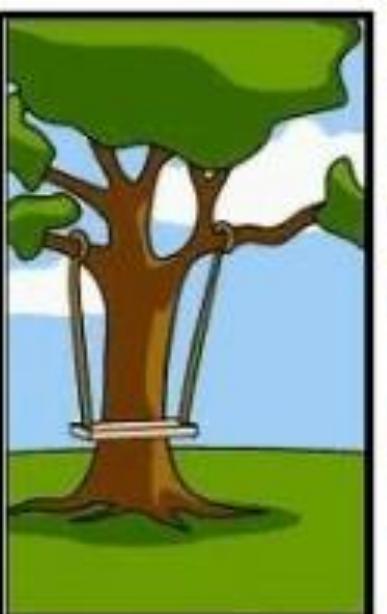
**Nous sommes motivés dans la réalisation du projet.**  
Nous disposons de l'environnement et du soutien dont nous avons besoin ; L'organisation nous fait confiance dans l'atteinte des objectifs fixés.

**La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.**

**Un logiciel opérationnel**  
est la principale mesure d'avancement.



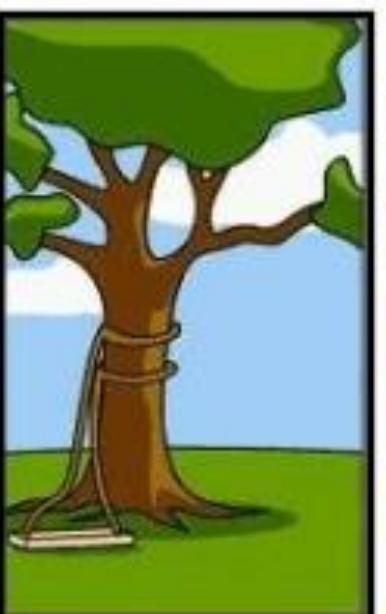
Comment le client  
a exprimé son besoin



Comment le chef de  
projet l'a compris



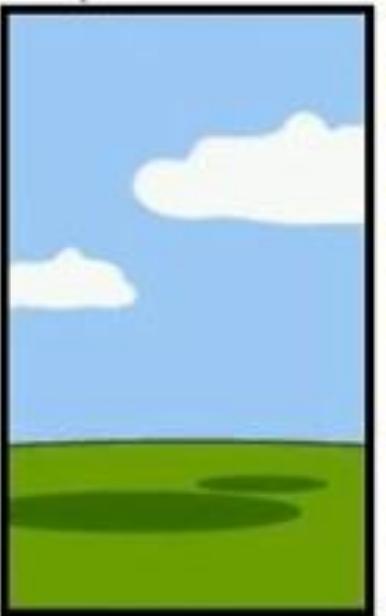
Comment l'ingénieur  
l'a conçu



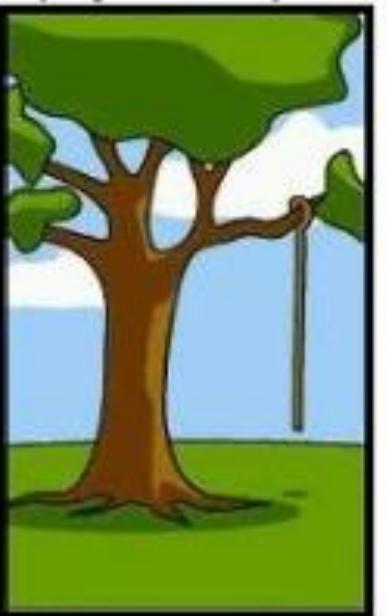
Comment le  
programmeur l'a écrit



Comment le responsable  
des ventes l'a décrit



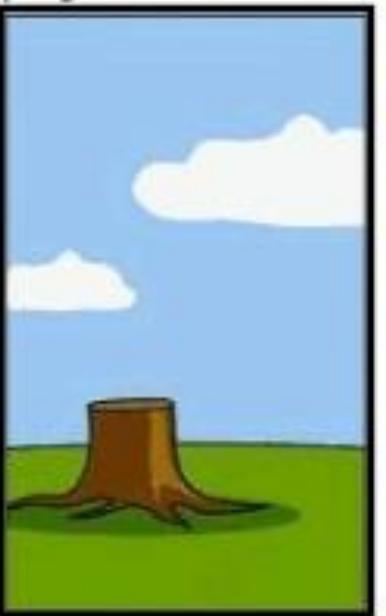
Comment le projet  
a été documenté



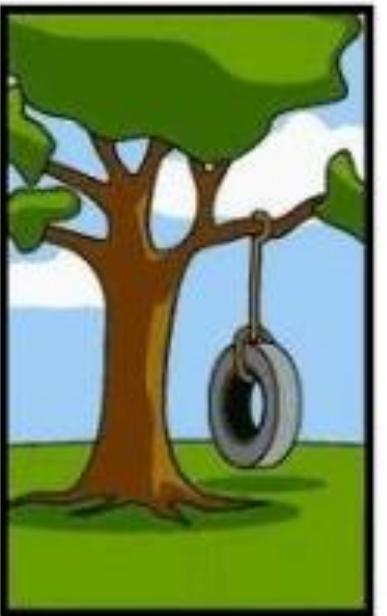
Ce qui a finalement  
été installé



Comment le client  
a été facturé



Comment la hotline  
répond aux demandes



Ce dont le client avait  
réellement besoin

# Pourquoi l'Agilité

## Un changement de paradigme

### Itératif



- 👎 L'équipe adresse l'ensemble du périmètre en même temps, ce qui crée du risque sur la livraison finale.
- 👍 On produit une version « grossière » que l'on fait valider et que l'on améliore à chaque itération.



# Pourquoi l'Agilité

## Un changement de paradigme

### Incrémental



👎 L'approche incrémentale est très proche du cycle en V.  
L'équipe connaît tout depuis le début (possible grâce à une grande analyse)

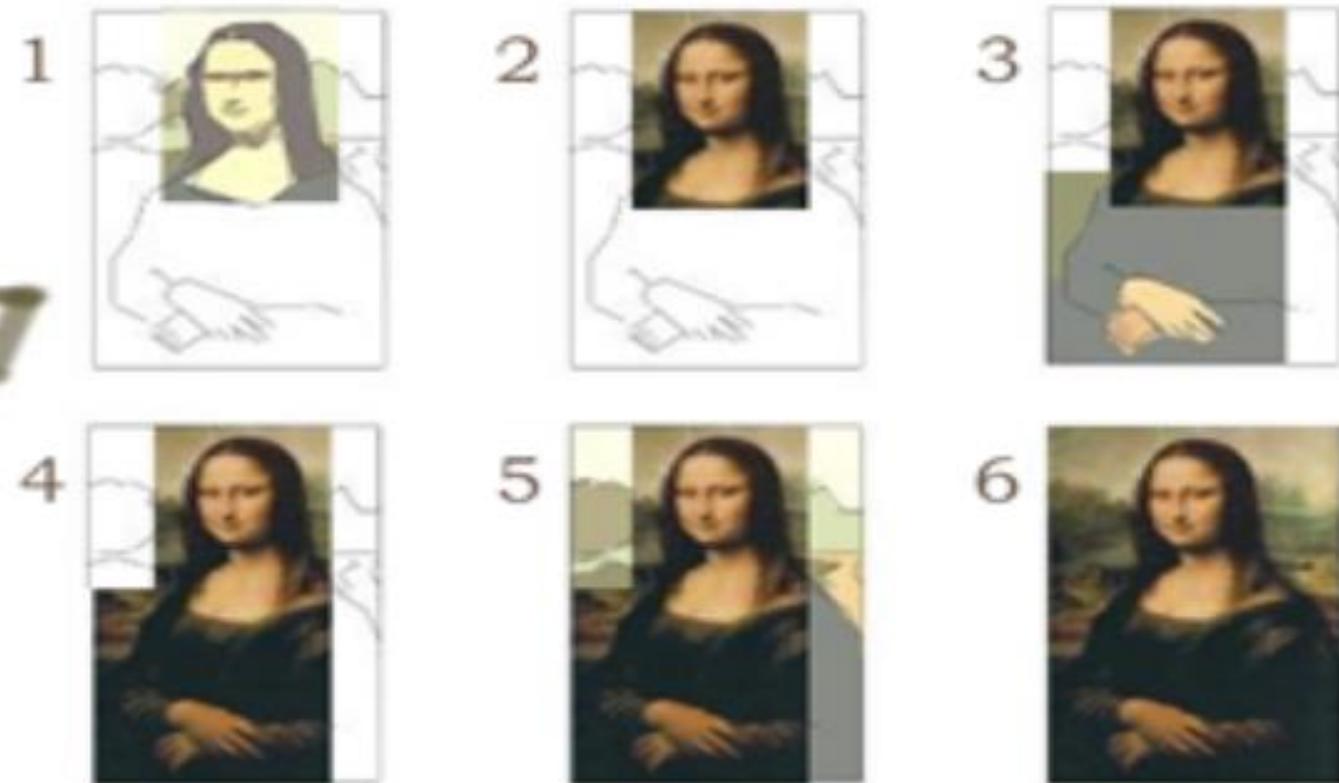
👍 Puis commence la construction des morceaux. C'est de la livraison échelonnée



# Pourquoi l'Agilité

## Un changement de paradigme

Agilité = Incremental & Itératif



# Pourquoi l'Agilité

## Un changement de paradigme

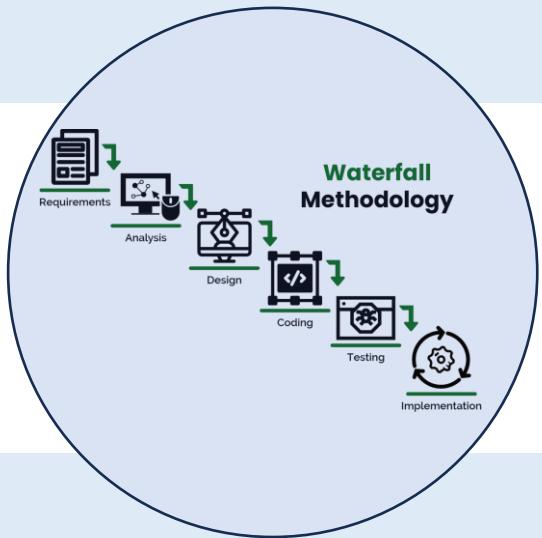
Mais attention...



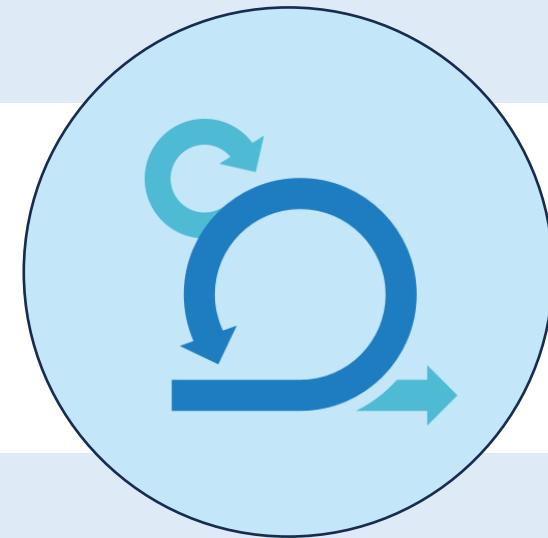
Avoir une vision est primordiale !



# Comparaison des méthodes prédictives et des méthodes agiles

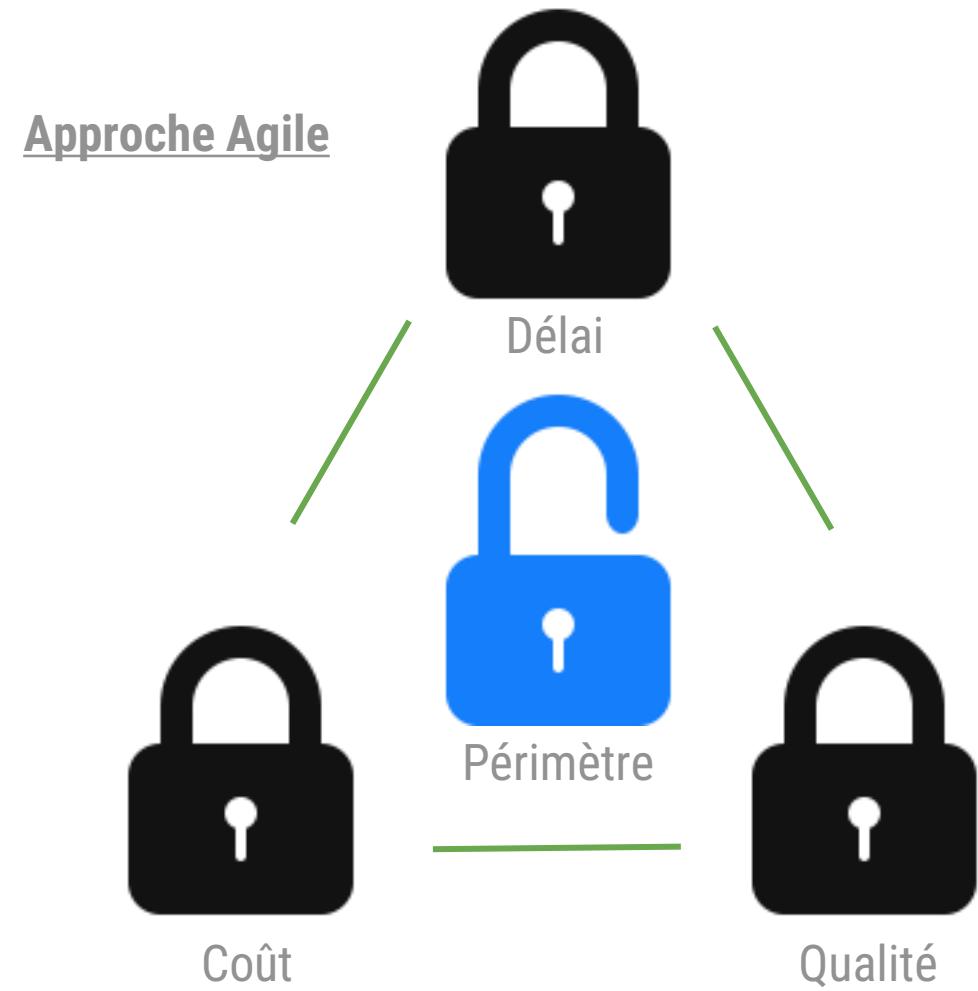
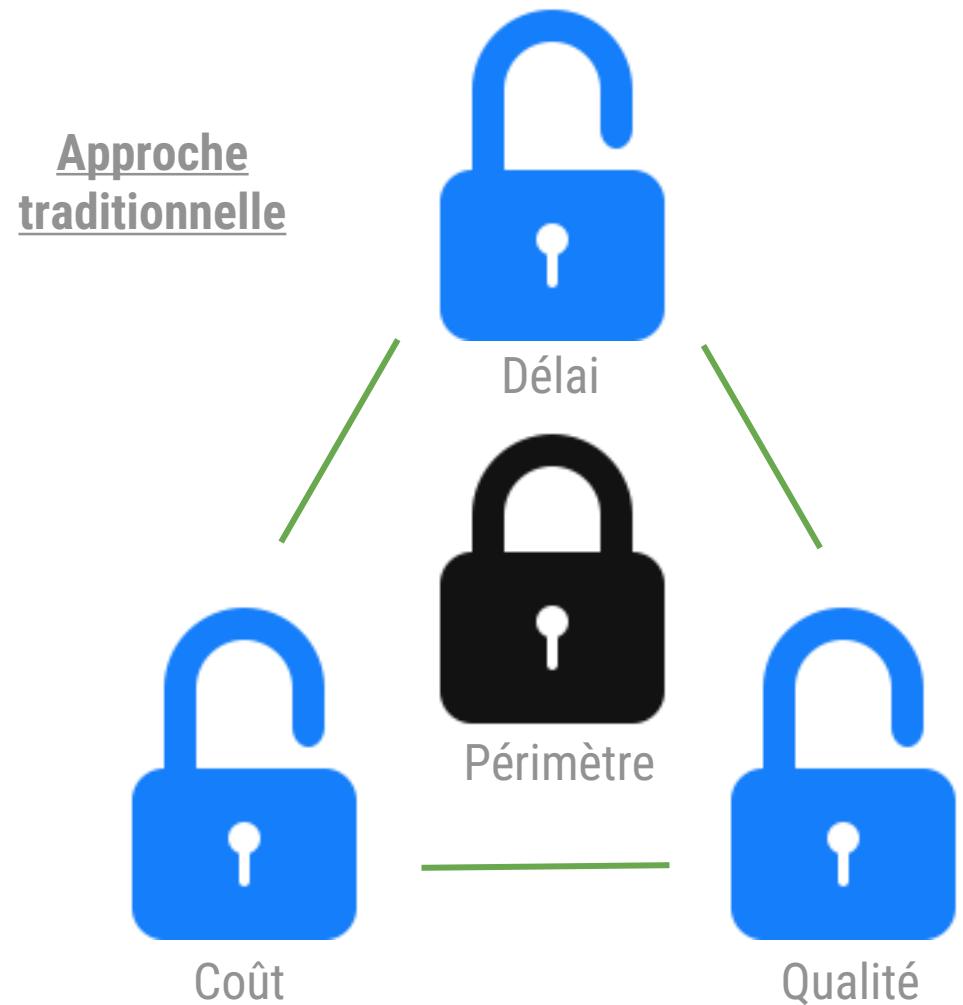


Méthodes Prédictives

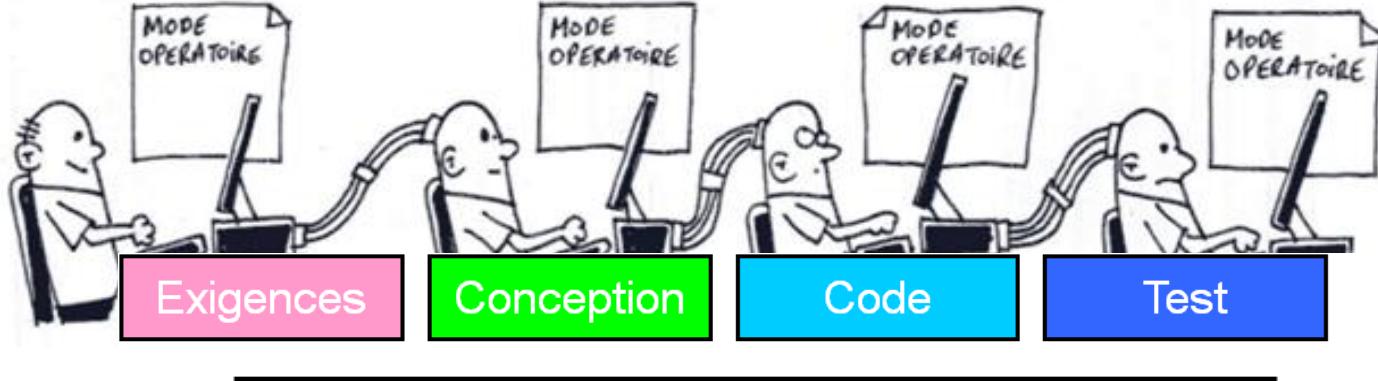


Méthodes Agiles

# Agilité vs cascade



# Agilité vs cascade



Approche traditionnelle

Plutôt que de faire toute une discipline d'un coup...

...Les équipes Scrum font un peu de tout tout le temps

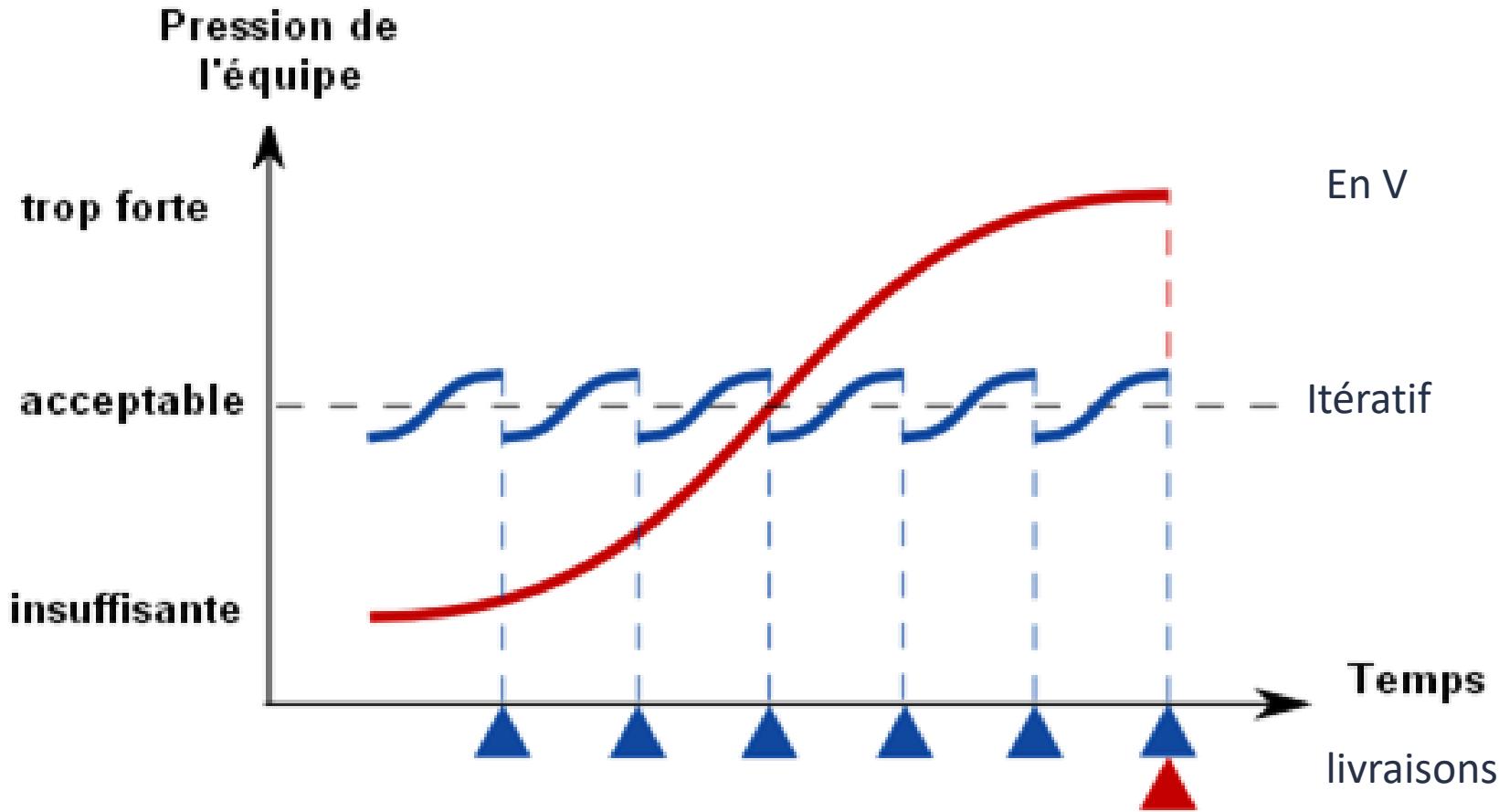
Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.



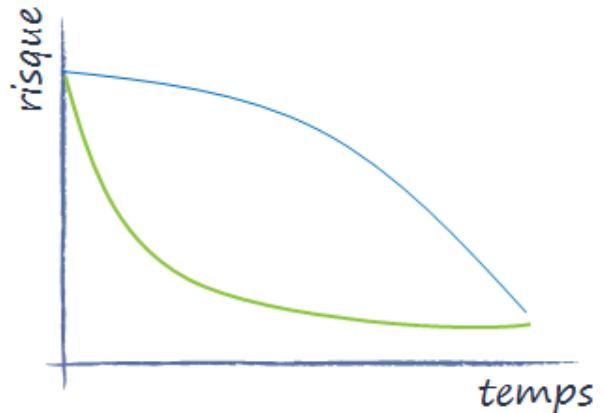
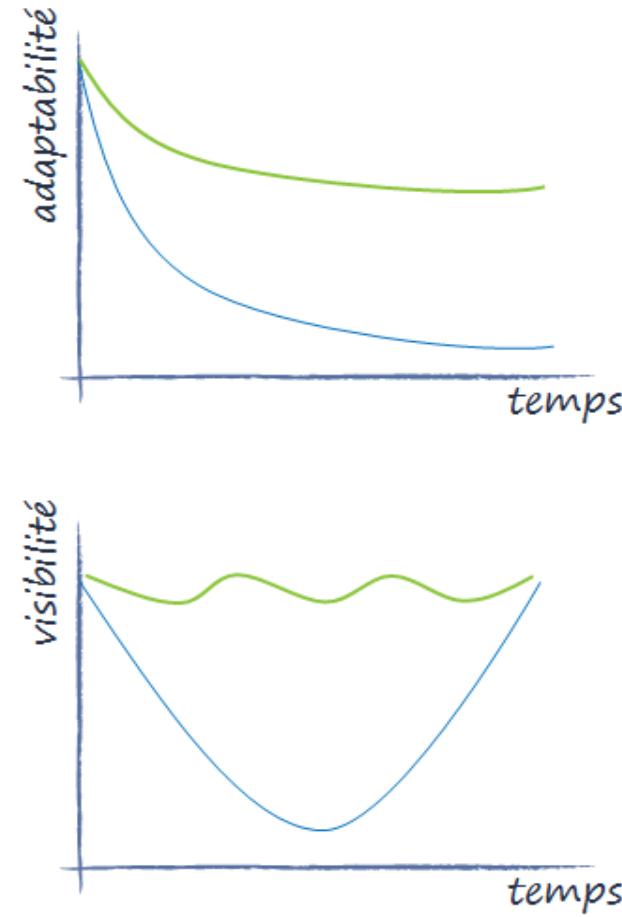
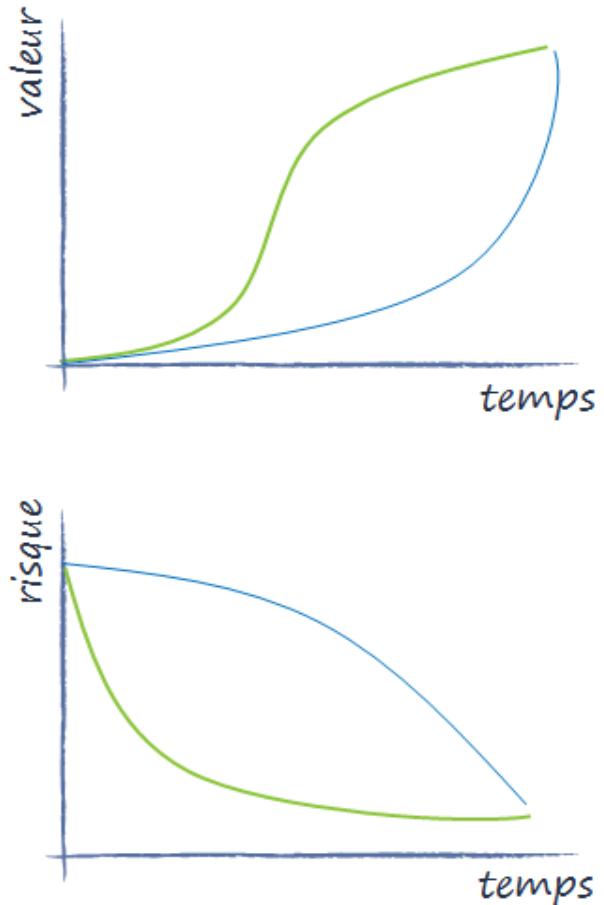
Approche Agile



# Agilité vs cascade



# Agilité vs cascade

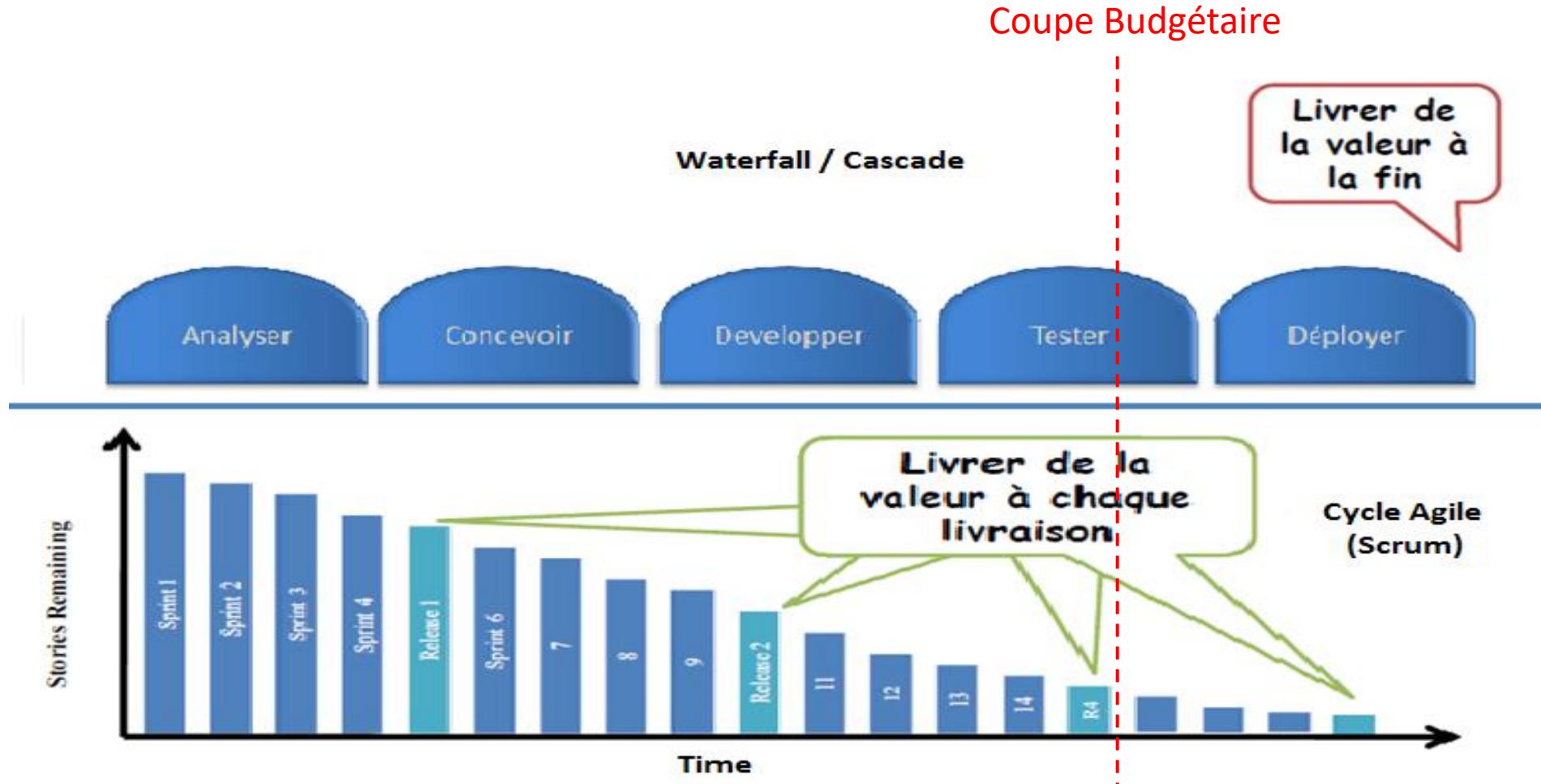


— agile — cascade

Module Agilité - eilco



# Agilité vs cascade



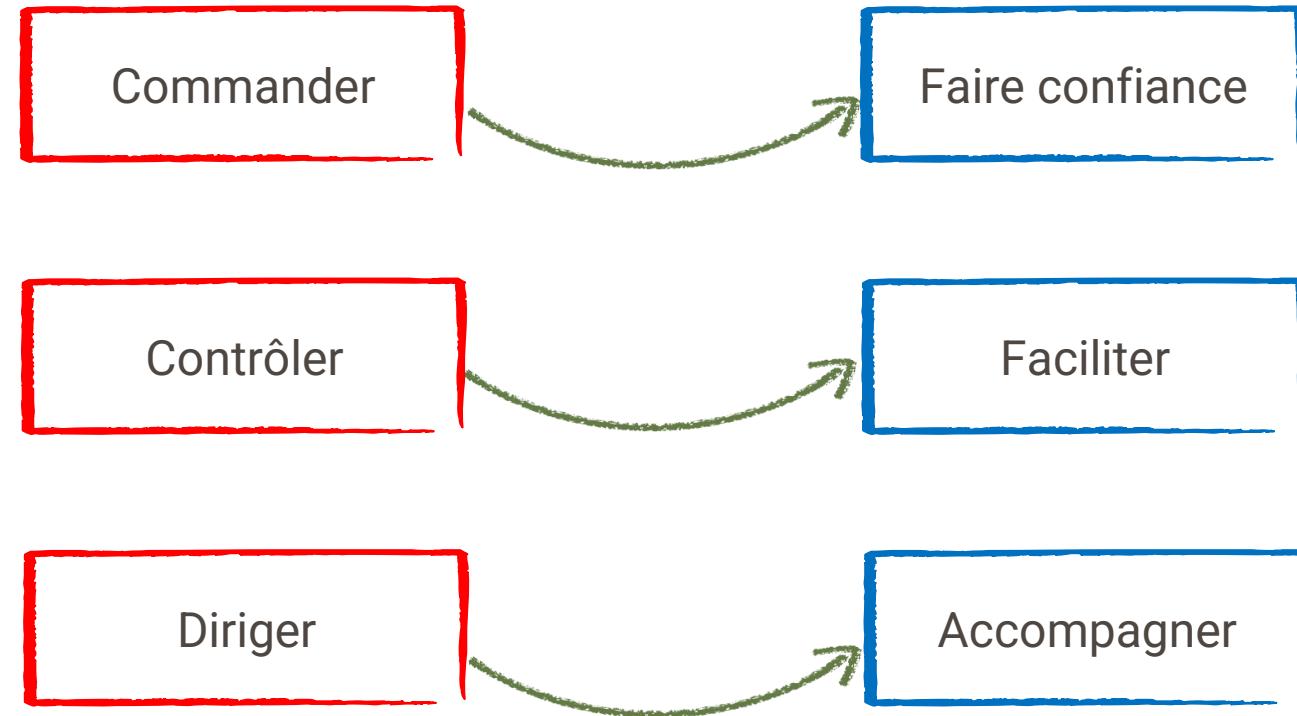
Pour passer d'un centre de coût à un centre de profit ...

Pour construire la confiance du Métier ...



# Agilité vs cascade

Transformation du rôle de 'chef' vers un rôle de coach



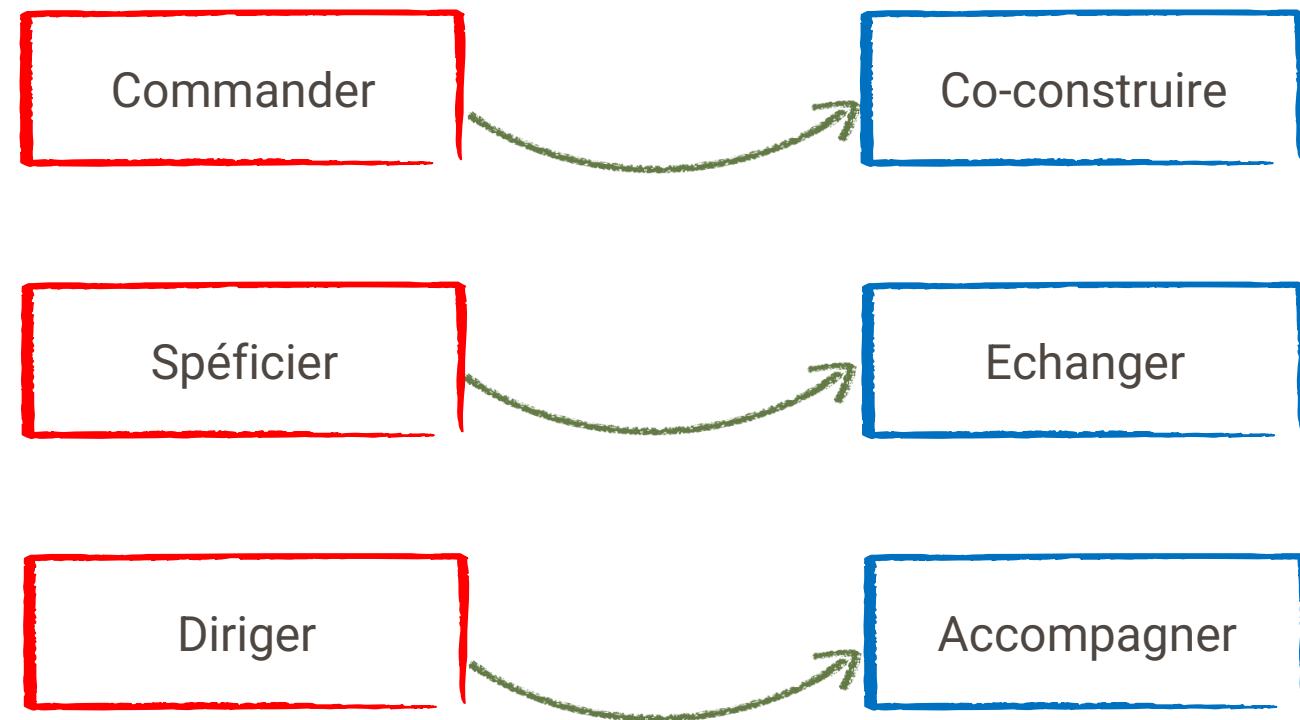
Représentation de l'évolution  
du rôle de 'chef' vers un rôle de coach

Cette transition se fait par un changement de **culture**, l'adoption d'un **nouveau paradigme**.



# Agilité vs cascade

Changement dans le rôle du client



Cette transition se fait par un changement de **culture**, l'adoption d'un **nouveau paradigme**.

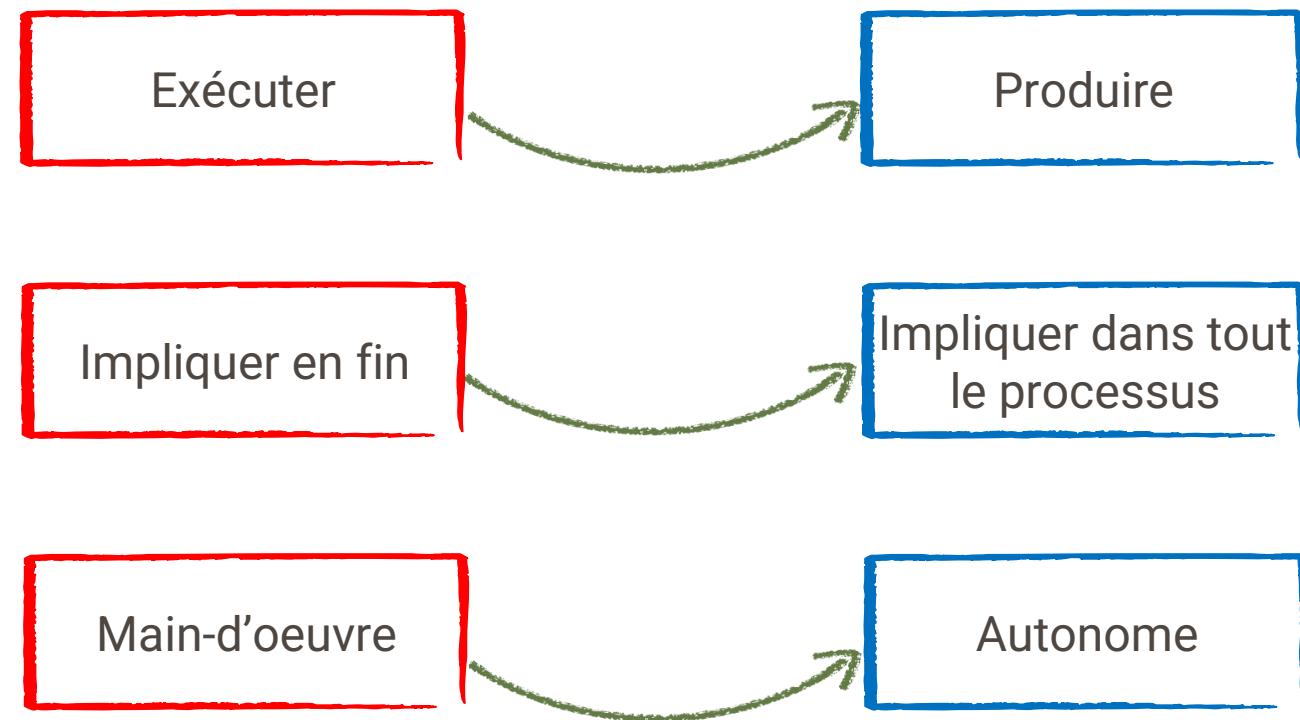


**Le client est présent tout au long de la production**



# Agilité vs cascade

## Transformation du rôle de développeur



## Software Developer



**L'équipe de développement est Pluridisciplinaire**

Cette transition se fait par un changement de **culture**, l'adoption d'un **nouveau paradigme**.



# Agilité vs cascade

Comparaison Méthodes Agiles vs Traditionnelles

	Agiles	Traditionnelles
<b>Processus</b>	Itératif, incrémental et adaptatif	Linéaire, étapes séquentielles, sans rétroaction
<b>Livraison</b>	Livraison continue de petites parties fonctionnelles	Livraison unique en fin de projet
<b>Flexibilité</b>	Très flexibles, adaptation à chaque itération	Peu flexible, difficile à modifier
<b>Implication du client</b>	Participation active et régulière	Participation limitée au début et à la fin du projet
<b>Planification</b>	Planification évolutive, adaptée à chaque sprint	Planification rigide définie dès le départ
<b>Structure de l'équipe</b>	Pluridisciplinaire et autoorganisée	Hiérarchique, rôles clairement définis



# Agilité vs cascade

Comparaison Méthodes Agiles vs Traditionnelles

	<b>Agiles</b>	<b>Traditionnelles</b>
<b>Gestion des changements</b>	Facile, intégrée à chaque itération	Difficile, changements couteux
<b>Qualification/Test</b>	Le plus régulièrement possible (chaque fin de sprint)	À la fin du projet
<b>Documentation</b>	Documentation minimale, juste suffisante	Documentation complète et détaillée
<b>Objectifs</b>	Satisfaction du client par livraison de valeur métier	Respect de l'engagement



# Agilité vs cascade

L'agilité face à l'échec des projets



**Echec d'un projet**

=

**Perte d'argent pour l'entreprise**

- ✓ Mauvaise définition des besoins
- ✓ Manque de flexibilité face aux changements
- ✓ Problèmes de communication dans l'équipe
- ✓ Découverte tardive des erreurs
- ✓ Risque de sur-engagement des ressources
- ✓ Difficultés à gérer les priorités



# Agilité vs cascade

Quizz

**<https://quizizz.com/admin/quiz/67031b9c3706e815310290de>**





# Les principales approches agiles



# Les principales approches agiles



# Les principales approches agiles

## Kanban

Kanban est une méthode agile de gestion de projet qui utilise un **tableau visuel** pour suivre et gérer le flux des tâches à accomplir.

Les tâches sont représentées sous forme de **cartes**, déplacées à travers différentes colonnes qui reflètent les **étapes du processus**.

Par défaut chaque tâche peut franchir les phases suivantes :

- **À faire (Backlog ou To Do)**
- **En cours (Work In Progress - WIP ou Doing)**
- **Terminé (Done)**



# Les principales approches agiles

Comment fonctionne le Kanban?

TODO	DEV	TEST	DEPLOY
1			
2			
3			
4			



# Les principales approches agiles

Comment fonctionne le Kanban?

TODO	DEV (WIP = 2)	TEST (WIP = 1)	DEPLOY (WIP =1)
	4	2	1
5	3		



# Les principales approches agiles

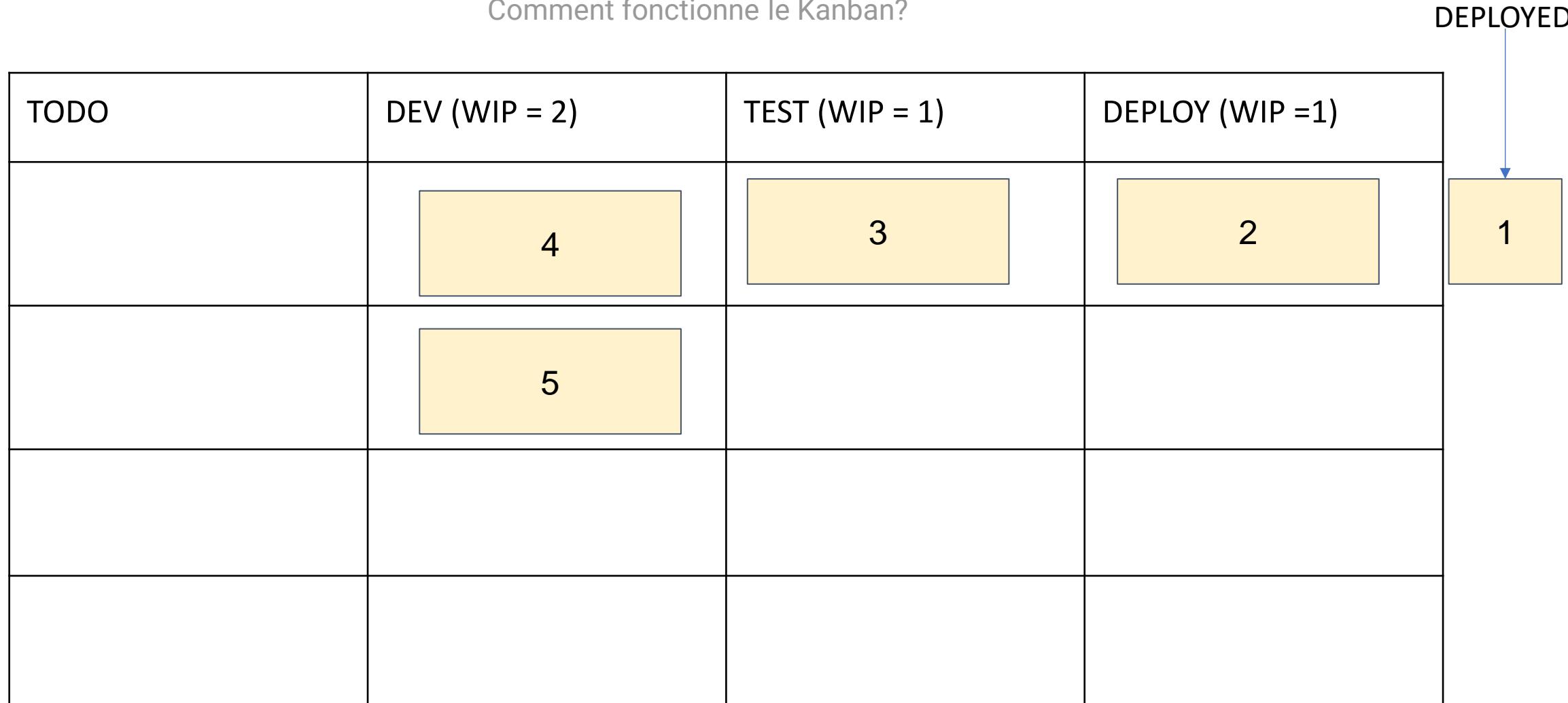
Comment fonctionne le Kanban?

TODO	DEV (WIP = 2)	TEST (WIP = 1)	DEPLOY (WIP =1)
	3	2	1
5	4		



# Les principales approches agiles

Comment fonctionne le Kanban?

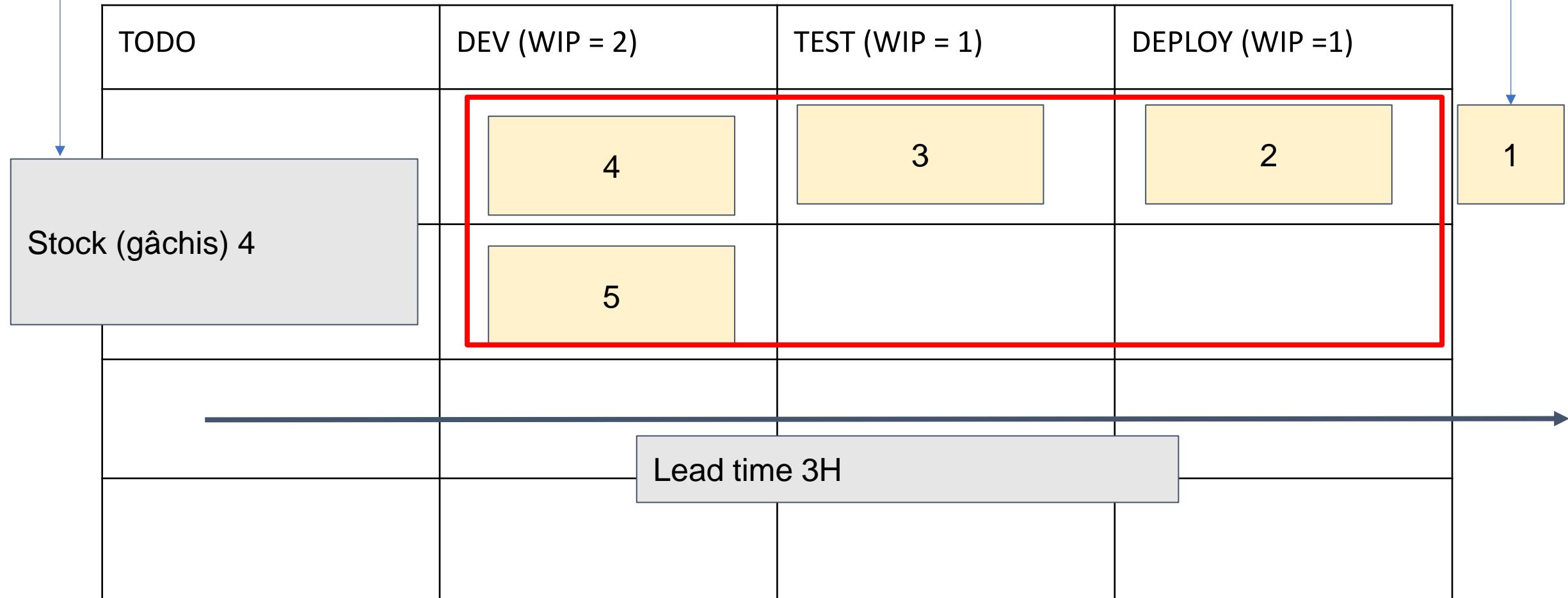


# Les principales approches agiles

Accumulation du stock

Comment fonctionne le Kanban?

DEPLOYED



# Les principales approches agiles

Exemple d'un Tableau Kanban dans un vrai projet

The image shows a digital Kanban board with the following columns:

- READY TO START (2/0)**: Contains two cards: one for creating a power point presentation and another for a new product launch.
- IN PROGRESS (2/6)**: Contains two cards: one for running an NPS campaign and collecting results, and another for branding materials for a conference. A red callout box with the text "Blocked Card" points to the first card.
- REVIEW (2/3)**: Contains three sub-columns:
  - READY FOR REVIEW (1/0)**: One card for running a campaign on the website and blog.
  - IN REVIEW (1/0)**: One card asking which software to use for the campaign.
  - EXPEDITE**: One card for deciding and developing brand architecture.
- PUBLISH/DEPLOY (1/0)**: Contains one card for initial research.
- DONE (2/0)**: Contains two cards: one for corporate re-branding and another for customer satisfaction score collection.

**PLANNED WORK** section at the bottom:

- Card 1604: Create power point presentation with value proposition and key differentiators.
- Card 1595: Notify important business stakeholders - Collect Feedback.
- Card 1610: Run the campaign on the website and blog.
- Card 846: Which software should we use to run the campaign?
- Card 1596: Decide and develop your brand architecture.



# Les principales approches agiles

Objectifs et principes de Kanban

## Objectifs

- ✓ Améliorer l'efficacité du flux de travail
- ✓ Maximiser la productivité
- ✓ Amélioration continue
- ✓ Réduire les temps de cycle

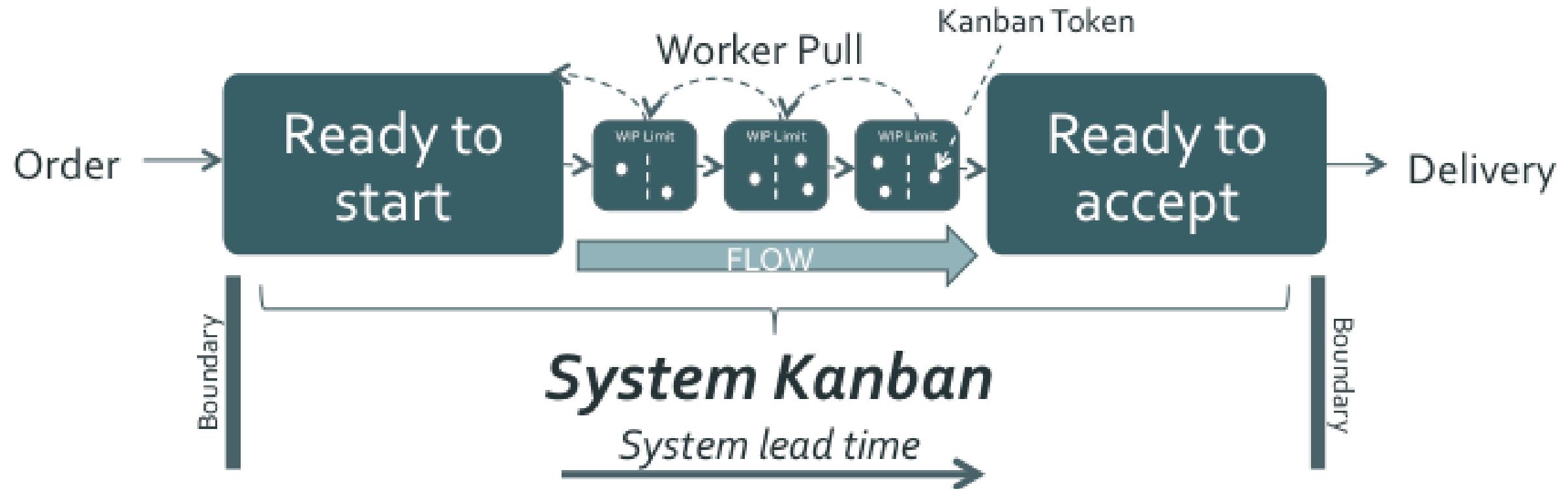
## Principes

- ✓ Commencer avec ce que vous faites actuellement
- ✓ S'engager dans une amélioration continue
- ✓ Respecter les processus et les rôles actuels
- ✓ Visualiser le travail
- ✓ Limiter le travail en cours (WIP) et gestion du flux



# Les principales approches agiles

Workflow global de la méthode kanban



# Les principales approches agiles

Scrum fondateurs



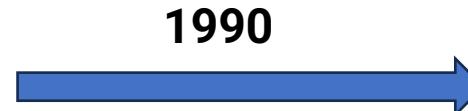
**Ken Schwaber**

L'un des fondateurs  
de l'Agile Alliance



**Jeff Sutherland**

Co-rédacteur du  
manifeste agile



**Scrum.org**



# Les principales approches agiles

Qu'est ce que c'est Scrum ?

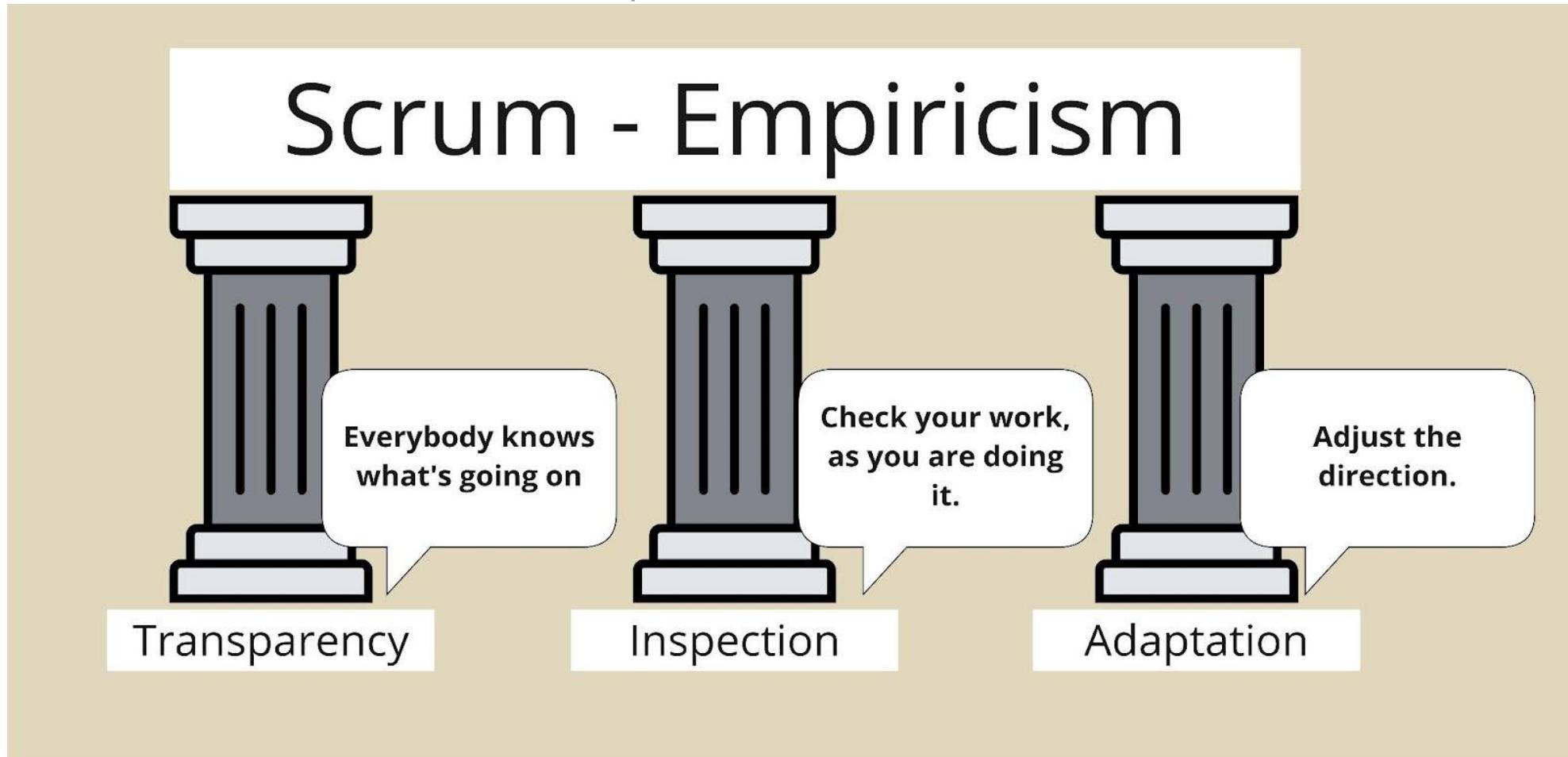
## Scrum en 100 mots

- Scrum est un processus agile qui permet de produire la plus grande valeur métier dans la durée la plus courte.
- Du logiciel qui fonctionne est produit à chaque sprint (**toutes les 2 à 4 semaines**).
- Le métier définit les priorités. L'équipe s'organise elle-même pour déterminer la meilleure façon de produire les exigences les plus prioritaires.
- A chaque fin de sprint, tout le monde peut voir fonctionner le produit courant et décider soit de le livrer dans l'état, soit de continuer à l'améliorer pendant un sprint supplémentaire.



# Les principales approches agiles

Les piliers de Scrum ?



# Les principales approches agiles

Les piliers de Scrum ?

## Transparence :

- Les aspects importants du processus doivent être **visibles**.
- La transparence implique que soit défini un **standard commun**, une définition commune de « Fini » (Definition of Done).
- Scrum met l'accent sur le fait d'avoir un **langage commun** entre l'équipe et le management.



# Les principales approches agiles

Les piliers de Scrum ?

## Inspection :

- Les utilisateurs de Scrum doivent régulièrement passer en revue les artefacts et l'état d'avancement par rapport aux objectifs afin de détecter les écarts indésirables.



# Les principales approches agiles

Les piliers de Scrum ?

## Adaptation :

Si une dérive est constatée pendant l'inspection, le processus / produit doit alors être **adapté**.

Scrum fournit des rituels, durant lesquels cette adaptation est possible.

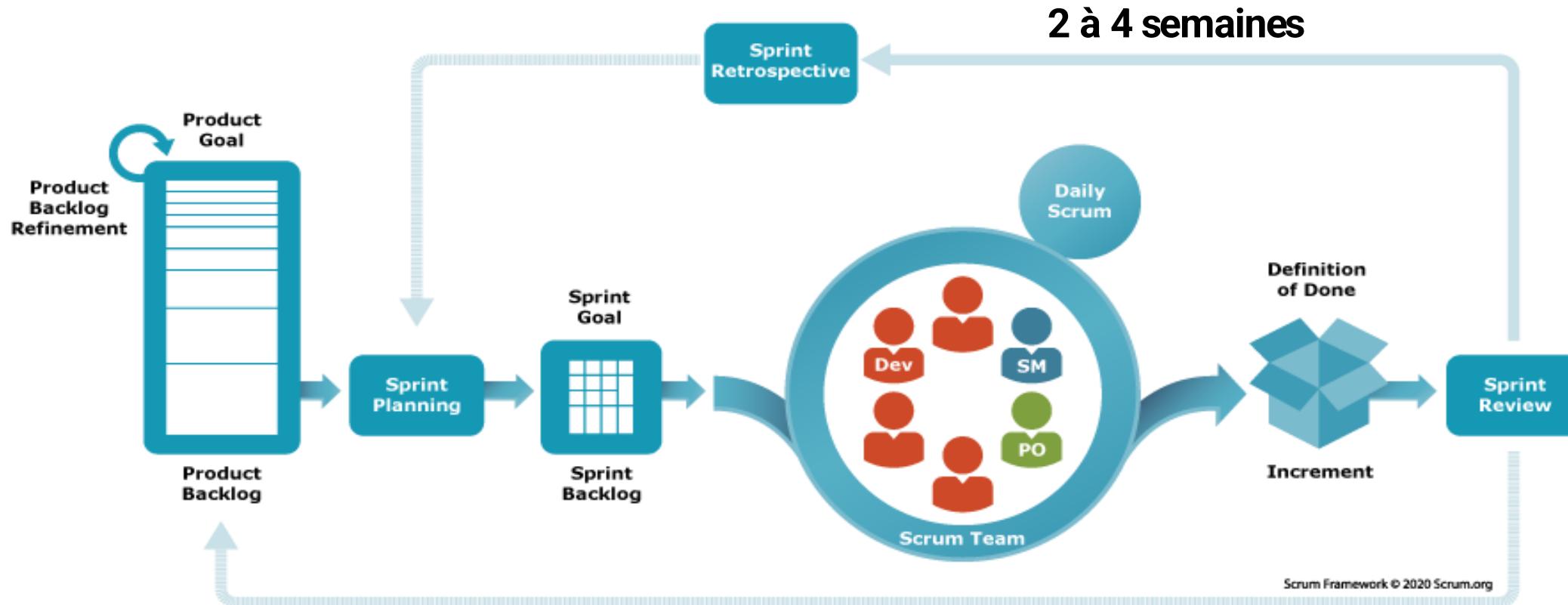
Il s'agit de :

- La réunion de **planification de sprint** (sprint planning),
- La **mêlée** quotidienne (daily scrum),
- La **revue de sprint** (sprint review),
- La **rétrospective** de sprint (sprint retrospective).



# Les principales approches agiles

Cycle Scrum



# Les principales approches agiles

## Rappel User Story

### User Story

As a <role>  
I want <goal>  
so that <benefit>

Acceptance criteria:  
(Conditions of Satisfaction)

...

...

As an Account Manager  
I want a sales report of my account  
to be sent to my inbox daily  
So that I can monitor the sales  
progress of my customer portfolio

#### Acceptance criteria:

1. The report is sent daily to my inbox
2. The report contains the following sales details: ...
3. The report is in csv format.

- Une **User Story** (ou "histoire utilisateur") est une description **simple** et **concise** d'une fonctionnalité ou d'un besoin du point de vue de l'utilisateur final
- Elle est utilisée dans les méthodologies agiles, notamment **Scrum**, pour capturer ce que l'utilisateur souhaite accomplir avec le produit
- Les user stories sont généralement écrites dans un format standard suivant cette structure : "En tant que **[rôle d'utilisateur]**, je veux **[objectif]** afin de **[bénéfice]**"

#### Exemple :

"En tant qu'utilisateur enregistré, je veux pouvoir changer mon mot de passe afin de sécuriser mon compte."



# Les principales approches agiles

Le cadre Scrum

## Rôles

- Product Owner
- Scrum Master
- Dev Team

## Cérémonies

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

## Artefacts

- Backlog de produit
- Backlog du sprint
- Incrémentation du produit



# Les principales approches agiles

Le cadre Scrum

## Rôles

- Product Owner
- Scrum Master
- Dev Team

Scrumies

Planning

- Daily Scrum
- Sprint Review
- Sprint Retrospective

## Artefacts

- Backlog de produit
- Backlog du sprint
- Incrémentation du produit



# Les principales approches agiles

Le rôle de PO dans Scrum

## Maximiser la valeur du produit et de ce qui sera fait par la DevTeam

Cela se fait au travers de la gestion du product backlog, dont le PO est le seul responsable.

### Que comprend la gestion du product backlog ?

Exprimer clairement les items.

Prioriser les items de la meilleure manière pour atteindre les objectifs et missions.

S'assurer que le backlog est **visible, transparent et clair pour tous** et qu'il montre ce sur lequel l'équipe travaillera ensuite.

S'assurer que la DevTeam a un niveau de compréhension suffisant des items du backlog.



# Les principales approches agiles

Le rôle de PO dans Scrum

## **Le PO est responsable du “Quoi”, pas du “Comment”**

C'est la DevTeam qui est pleinement compétente pour assumer le "comment".

**Par conséquent, en tant que PO, il ne faut pas...**

Estimer les items à la place de la devTeam .

Décider de la conception technique / du découpage en tâches à la place de la devTeam.

Décider de l'ordre dans lequel les tâches du sprint seront réalisées (elles appartiennent au Sprint Backlog).

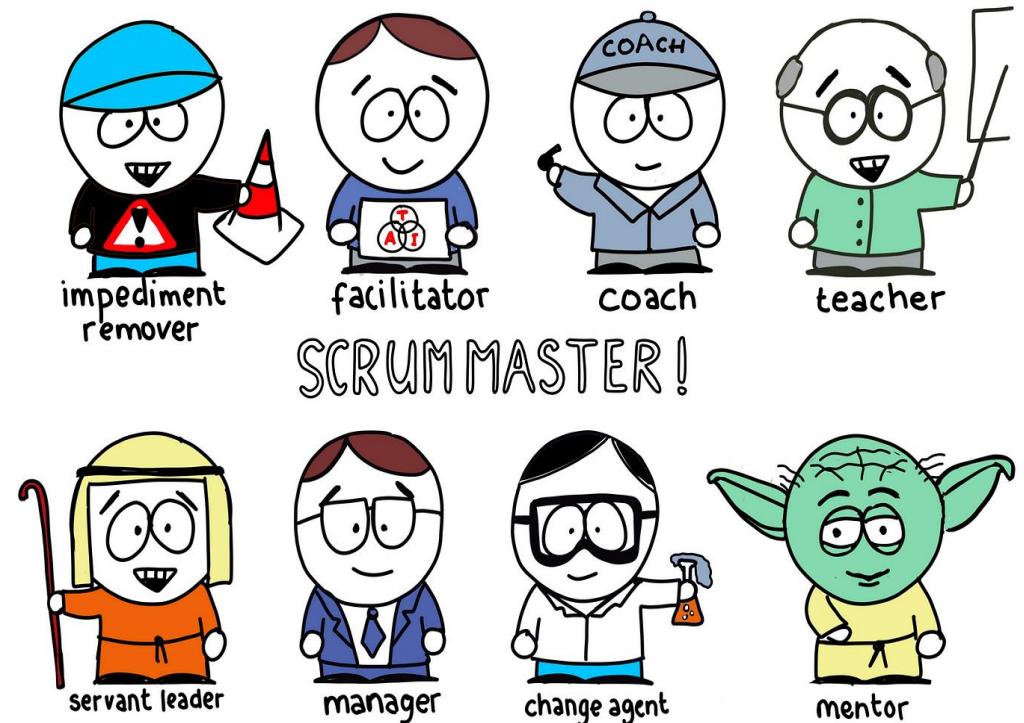
Manager l'équipe (elle est auto-organisée !).



# Les principales approches agiles

## Le rôle du Scrum Master dans Scrum

- Il comprend complètement l'agile et Scrum (La référence )
- Représente le management du projet, il manage le process Scrum. Il ne manage pas la Dev team (il n'est donc pas hiérarchiquement supérieur).
- Responsable de faire appliquer par l'équipe les valeurs et les pratiques de Scrum
- Élimine les obstacles qui empêchent la Dev Team d'aller à sa vitesse optimale
- S'assure que l'équipe est complètement fonctionnelle et productive
- Facilite une coopération poussée entre tous les rôles et fonctions
- Protège l'équipe des interférences extérieures
- Le Scrum Master veille à ce que la Dev Team réalise les cérémonies agiles correctement, en respectant le time-box, et s'assure que celles-ci ont lieu et se déroulent dans de bonnes conditions.  
(Animateur des cérémonies agiles)



# Les principales approches agiles

## La dev team dans Scrum

- L'équipe de développement est composée de professionnels qui livrent un incrément « Fini » et potentiellement publiable à la fin de chaque Sprint.
- Un incrément « Fini » est requis pour la revue de Sprint, et seuls les membres de l'équipe de développement sont responsables de sa création.
- L'équipe est autonome, structurée et habilitée à s'organiser et à gérer son propre travail, ce qui permet d'optimiser son efficacité et son efficience.



# Les principales approches agiles

## La dev team dans Scrum

- Elle se partage la responsabilité de la partie technique du projet
- Cross fonctionnelle & Pluridisciplinaires
- Auto-organisée
- Effectue les estimations des fonctionnalités
- Sollicite le Product Owner dès que nécessaire
- Résout ses propres conflits
- Pas de hiérarchie , pas de structure , pas de rôle prédéfinis
- Chaque membre peut potentiellement être amené à effectuer toutes les activités de l'équipe



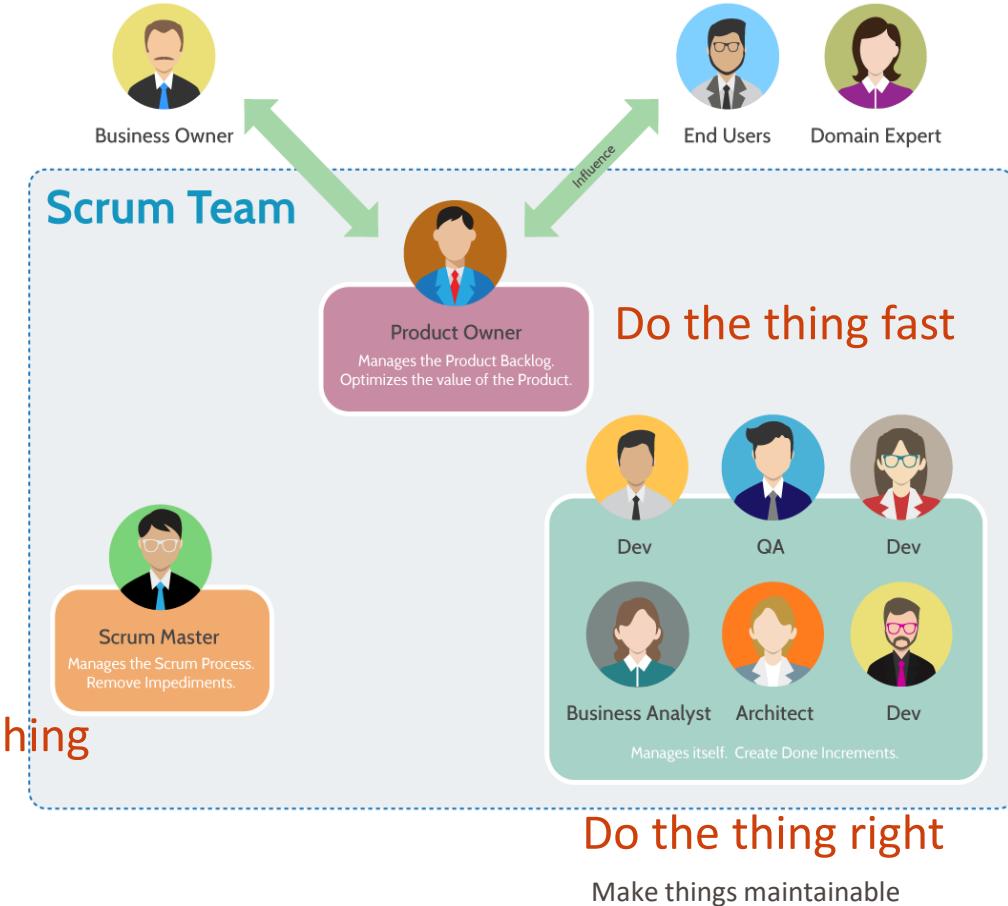
# Les principales approches agiles

Spécificités de l'équipe agile

- De 5 à 10 personnes
- Regroupant tous les rôles
- Architecte, concepteur, développeur, spécialiste IHM, testeur, etc.
- A plein temps sur le projet, de préférence
- L'équipe s'organise par elle-même
- La composition de l'équipe ne doit pas changer pendant un Sprint

**Do the right thing**

Make things lovable



# Les principales approches agiles

Le cadre Scrum

## Rôles

- Product Owner
- Scrum Master
- Dev Team

## Cérémonies

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

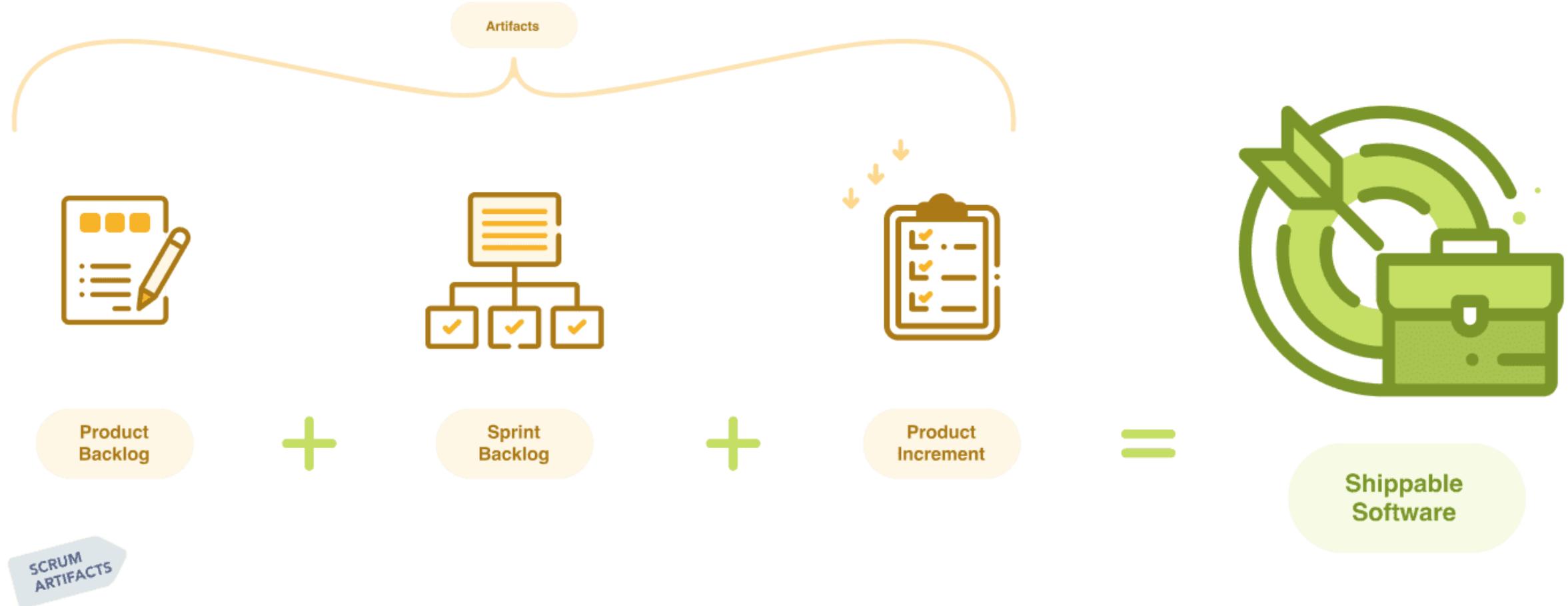
## Artefacts

- Backlog de produit
- Backlog du sprint
- Incrémentation du produit



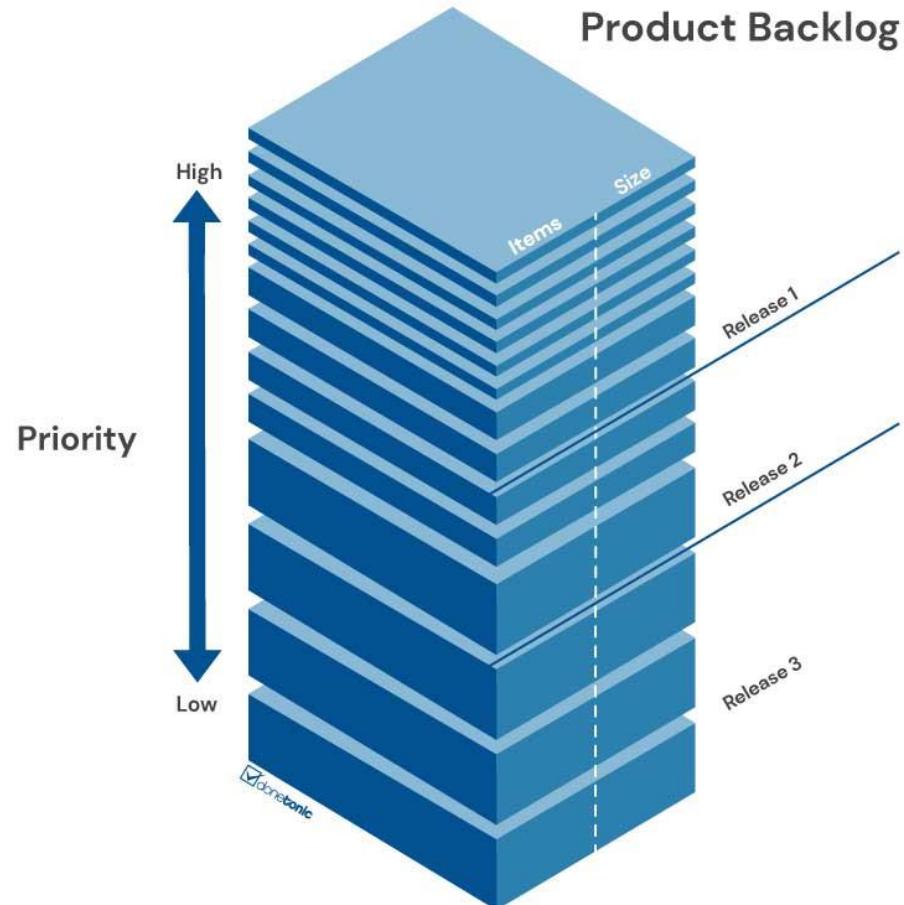
# Les principales approches agiles

Les artefacts



# Les principales approches agiles

Les artefacts



## Backlog Produit

- Le Backlog Produit est une liste ordonnée de tous les éléments identifiés comme nécessaires au produit.
- Il constitue l'unique source d'exigences pour tout changement à apporter au produit.
- Exprimé de telle façon que chaque élément apporte de la valeur aux utilisateurs ou clients du produit
- Le Product Owner est responsable du Backlog produit, y compris son contenu, sa disponibilité et son ordonnancement.



# Les principales approches agiles

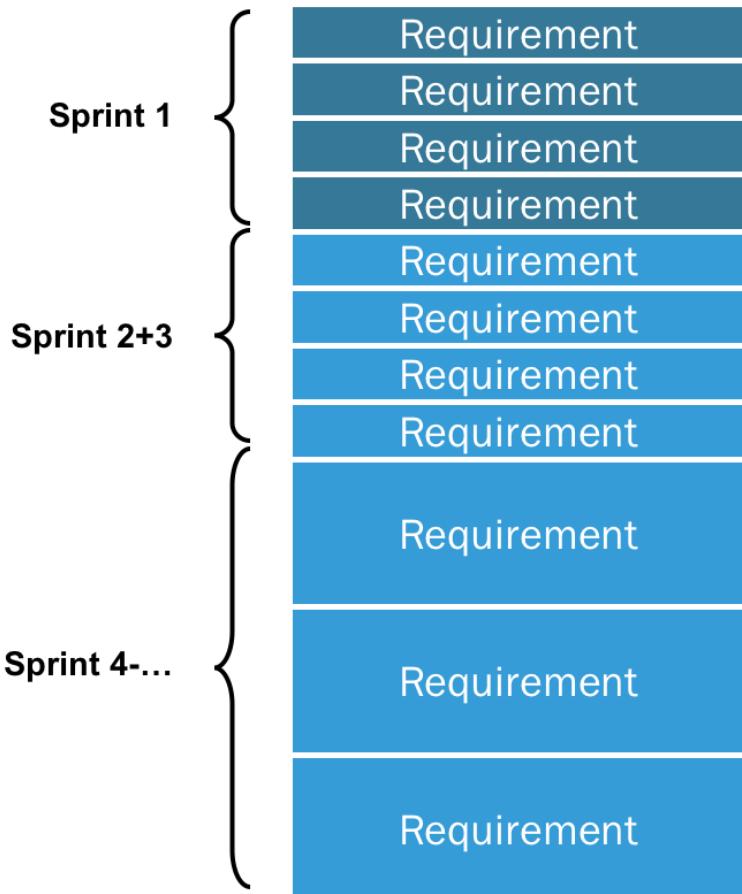
## Les artefacts

Elément de backlog	Estimation
Un invité peut faire une réservation	3
En tant qu'invité, j'annule une réservation	5
En tant qu'invité, je change les dates d'une réservation.	3
En tant qu'employé de l'hôtel, je produis les rapports de revenu par chambre	8
Améliorer la gestion des exceptions	8
...	30
...	50



# Les principales approches agiles

Les artefacts



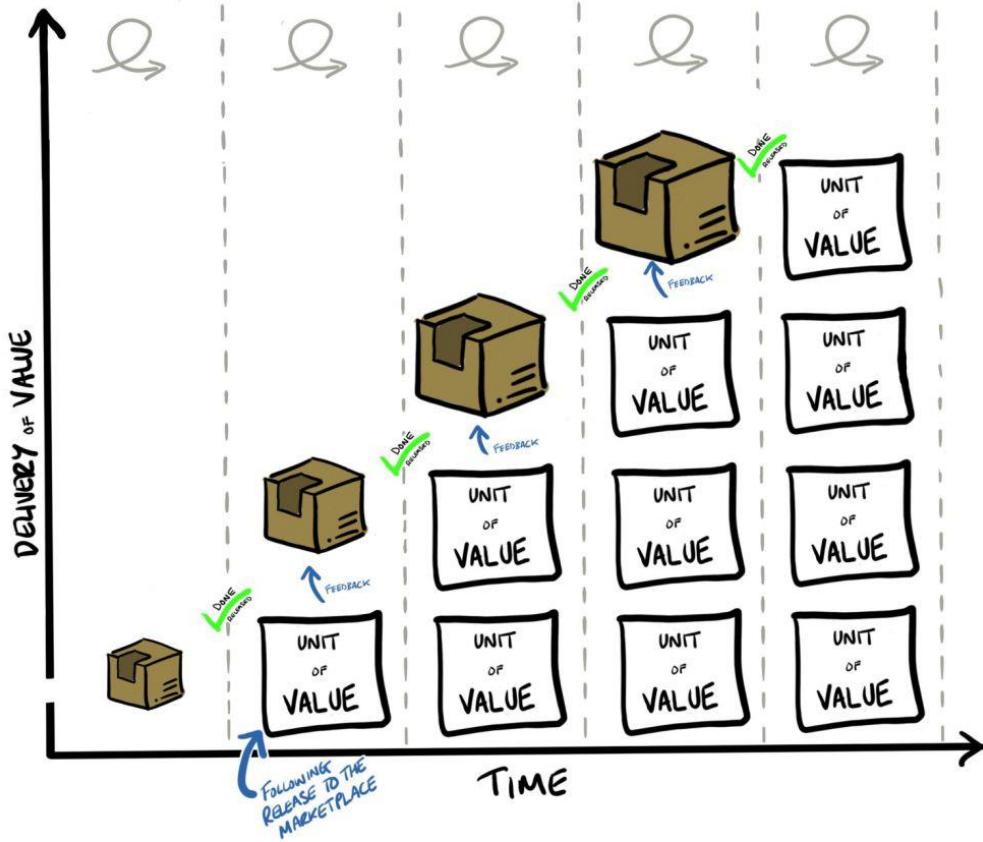
## Backlog Sprint

- Le Backlog Sprint est l'ensemble des éléments sélectionnés pour le Sprint plus un plan pour livrer l'incrément du produit et réaliser l'objectif du Sprint.
- Le Backlog Sprint est une prévision faite par l'équipe de développement concernant les fonctionnalités à intégrer dans le prochain incrément et le travail nécessaire pour les livrer dans un incrément « Fini ».
- Le Backlog Sprint rend visible tout le travail que l'équipe de développement identifié comme nécessaire pour atteindre l'objectif du Sprint.



# Les principales approches agiles

## Les artefacts



## Incrément du produit

- Un incrément dans Scrum représente un résultat tangible ou un ensemble de fonctionnalités complètes qui ont été développées et testées au cours d'un sprint.
- Chaque incrément doit être un ajout au produit, qui est potentiellement livrable, ce qui signifie qu'il peut être livré aux utilisateurs ou aux parties prenantes même s'il n'est pas encore le produit final.
- L'intérêt de l'incrément c'est qu'à chaque fin de sprint on fait une livraison qui a une valeur par rapport au métier.



# Les principales approches agiles

Le cadre Scrum

## Rôles

- Product Owner
- Scrum Master
- Dev Team

## Cérémonies

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

## Artifacts

- Backlog de produit
- Backlog du sprint
- Incrément du produit



# Les principales approches agiles

## Les cérémonies

Officiellement il existe 4 cérémonies en agiles qui sont :

- **Sprint Planning** (planification du sprint) : Cette réunion permet de planifier les tâches du sprint à venir.
- **Daily Scrum** (mêlée quotidienne) : Un stand-up meeting quotidien de 15 minutes pour synchroniser l'équipe.
- **Sprint Review** (revue de sprint) : Une réunion pour examiner l'incrément produit à la fin du sprint et recueillir des feedbacks.
- **Sprint Retrospective** (rérospective de sprint) : Un moment où l'équipe réfléchit sur ce qui a bien fonctionné et sur ce qui doit être amélioré.

Le **Product Backlog Refinement** (ou affinage du backlog produit) ne fait pas partie des cérémonies formelles de Scrum. Cependant, il est recommandé de le réaliser régulièrement pour maintenir le backlog à jour et bien priorisé



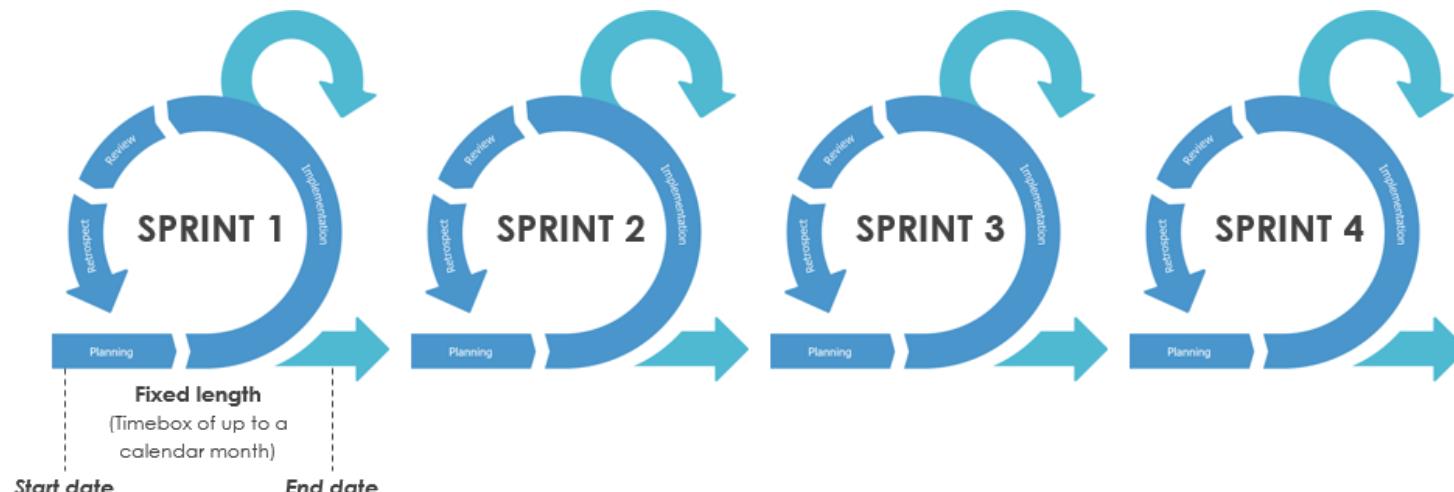
# Les principales approches agiles

Définition d'un sprint

## Le Sprint

Le cœur de Scrum est le Sprint, qui a une boîte de temps (time-box), une durée, **de 2 à 4 semaines** au cours de laquelle un **Incrément Produit « Fini »** fonctionnel et potentiellement publiable est **créé**. Les sprints ont une durée **cohérente** durant la phase de développement. Un nouveau Sprint commence immédiatement après la conclusion du Sprint précédent.

Durant un Sprint on fait toutes les cérémonies qui seront présentés par la suite.



# Les principales approches agiles

Les cérémonies

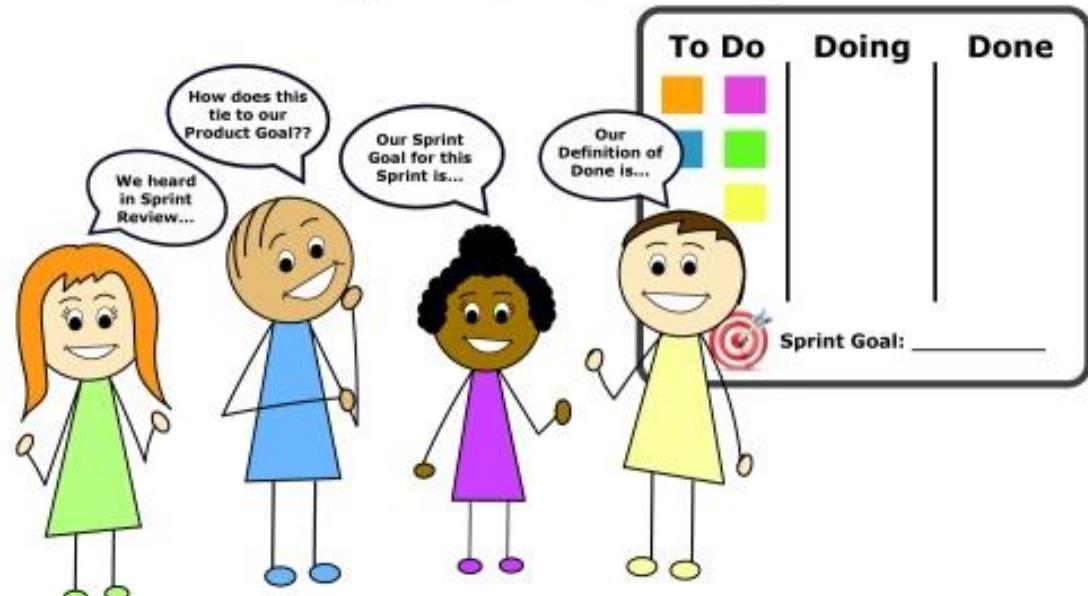
## Sprint Planning

Le **Sprint Planning** (ou planification du sprint) est une cérémonie essentielle dans Scrum qui marque le début de chaque Sprint. Son objectif est de définir **ce qui sera accompli** pendant le Sprint à venir et **comment** l'équipe va y parvenir.

Durant ce meeting l'équipe se réunit afin de :

- Définir les objectifs du Sprint
- Créer le Sprint Backlog
- Estimer le travail nécessaire
- Déterminer comment le travail sera accompli

## Sprint Planning



# Les principales approches agiles

Les cérémonies

## Daily Scrum (mêlée quotidienne)

- Daily Scrum est un événement de **15 minutes** (time-boxé) destiné à l'équipe de développement.
- La mêlée quotidienne est tenue **tous les jours** du Sprint. L'équipe de développement planifie le travail pour les prochaines 24 heures. Cela optimise la collaboration et la performance de l'équipe tout en inspectant le travail depuis la dernière mêlée quotidienne et envisageant le travail restant durant le Sprint.
- La mêlée quotidienne est tenue à la même heure et au même lieu chaque jour pour réduire la complexité.
- Tout le monde est invité mais seuls les membres de l'équipe peuvent parler.
- Pas fait pour résoudre les problèmes
- Permet d'éviter l'organisation d'autres réunions

**Chacun répond à 3 questions**

Qu'as-tu fait hier ?

Que vas-tu faire aujourd'hui ?

Y a t-il un obstacle qui te freine ?



# Les principales approches agiles

Les cérémonies

## Revue de Sprint

- Une revue de Sprint (Sprint Review) est tenue à la fin du Sprint pour inspecter l'incrément réalisé et adapter le Backlog Produit si nécessaire.
- Pendant la revue de Sprint, l'équipe Scrum et les parties prenantes échangent sur ce qui a été fait durant le Sprint et présente une démo des nouvelles fonctionnalités ou de l'architecture
- C'est un point qui est informel, dure moins de deux heures et ne nécessite pas de préparation.
- Toute l'équipe est invitée et nous pouvons inviter les managers et d'autres équipes afin de présenter l'état d'avancement.



# Les principales approches agiles

Les cérémonies

## Rétrospective de Sprint

- La rétrospective de Sprint (Sprint Retrospective) est une opportunité pour l'équipe Scrum de s'auto-inspecter et de créer un plan d'améliorations à adopter au cours du prochain Sprint.
- Réfléchir régulièrement à ce qui marche et ce qui ne marche pas
- Dure en général de 15 à 30 minutes
- Fait à la fin de chaque sprint
- Toute l'équipe participe : SM, PO, DevTeam

Toute l'équipe collecte du feedback et discute sur ce qu'elle aimeraient :

**Commencer à faire**

**Arrêter de faire**

**Continuer à faire**

Juste une façon  
parmi d'autres de  
faire une  
réetrospective.



# Les principales approches agiles

## Les cérémonies

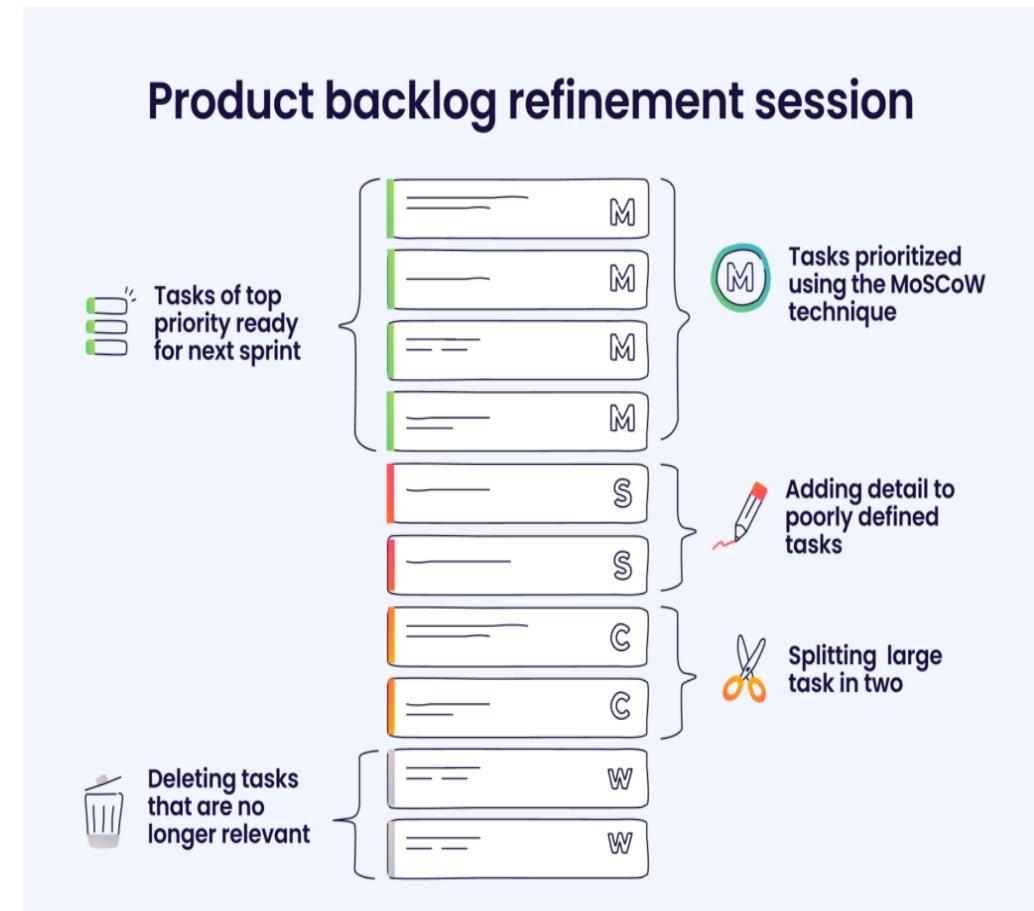
### Product Backlog Refinement

Le raffinement du backlog de produit (Product Backlog Refinement) consiste en l'ajout de détails, d'estimations et de l'ordonnancement des éléments du Backlog Produit.

Il s'agit d'une activité régulière dans laquelle le Product Owner et l'équipe de développement collaborent pour détailler les éléments du Backlog Produit.

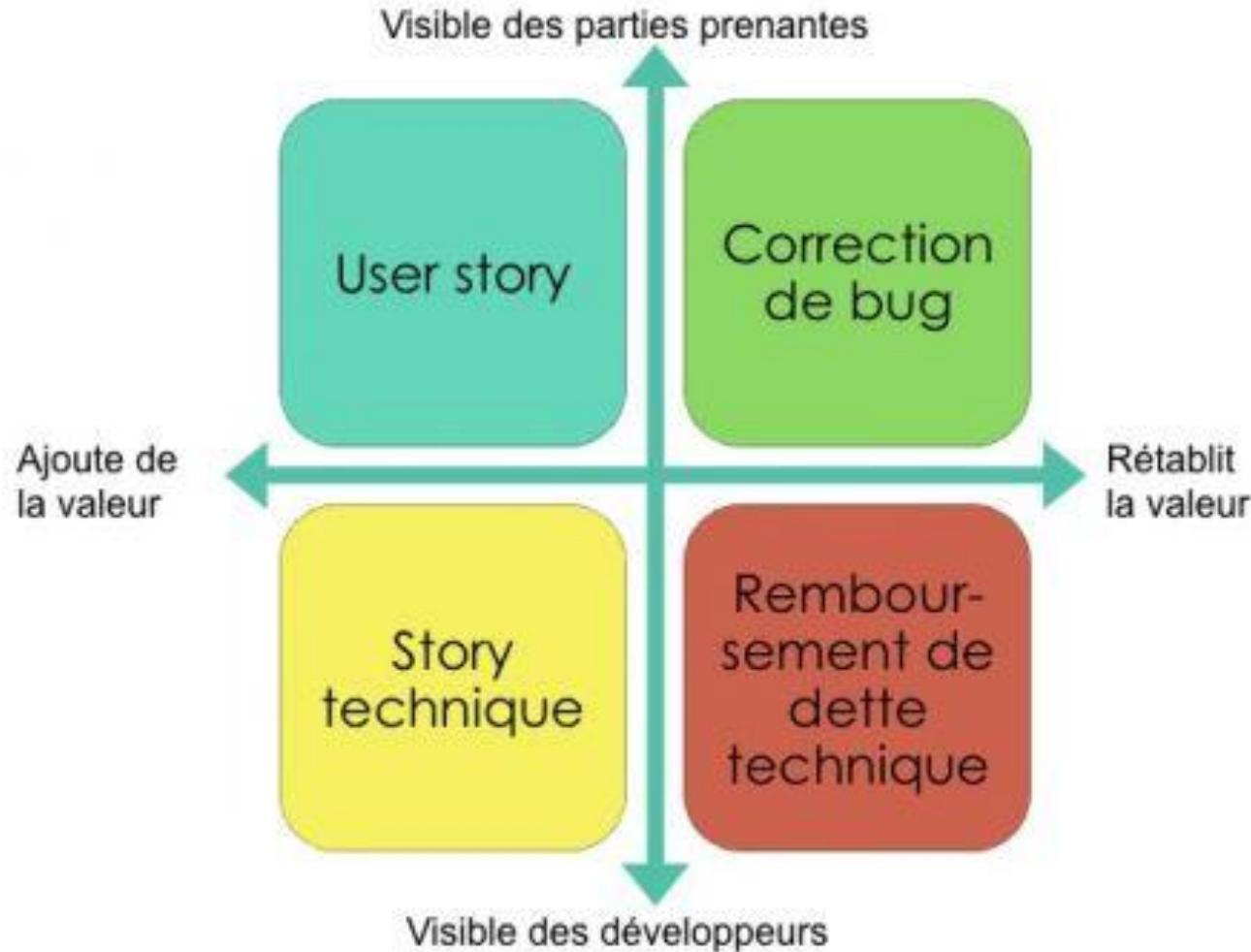
Durant le raffinement du Backlog Produit, les éléments sont revisités et révisés et ceci par le biais des actions suivantes :

- **Clarification** des éléments du backlog
- **Priorisation** des éléments
- **Estimation** des efforts
- **Division** des éléments trop larges
- **Mise à jour** des détails techniques
- **Révision** des éléments obsolètes ou non pertinents



# Les principales approches agiles

De quoi est constitués le product backlog



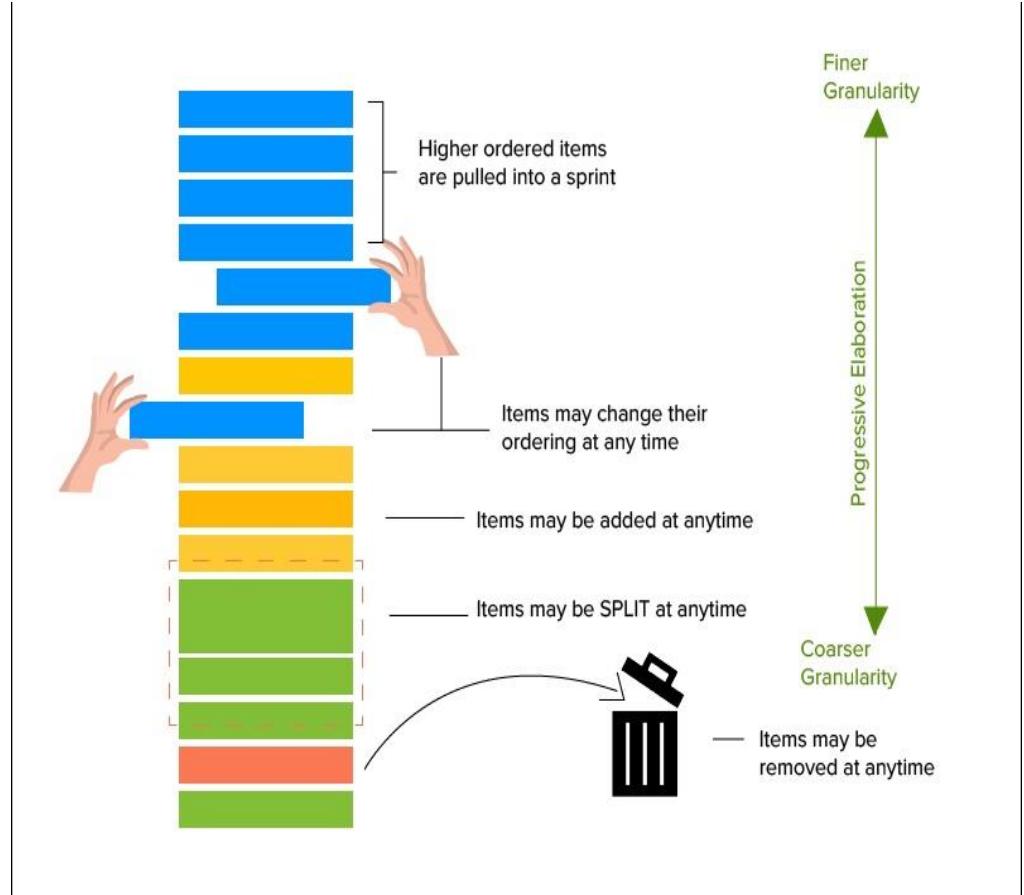
# Les principales approches agiles

## Comment constituer le product backlog

Constituer le **Product Backlog** nécessite une approche structurée pour identifier, prioriser, et organiser les éléments de travail nécessaires au développement du produit.

Le PO s'occupe de :

- **Collecter les besoins**
- **Créer des items dans le backlog**
- **Prioriser les éléments**
- **Estimer l'effort nécessaire**
- **Évaluer continuellement**
- **Impliquer les parties prenantes**



# Les principales approches agiles

Les user stories doivent être prêtes afin de pouvoir être embarquées efficacement lors du planning de sprint.

Alimenter

**Sprint +1**

La granularité la plus grosse est la User Story. Ces sujets sont ceux à avancer en backlog refinement.

**Version +1**

Ces sujets ne doivent pas consommer d'énergie à l'équipe pour le moment. Ils constituent le fond du backlog et ne doivent pas faire l'objet d'une revue collective poussée.

**Au delà**

# Vie du product backlog

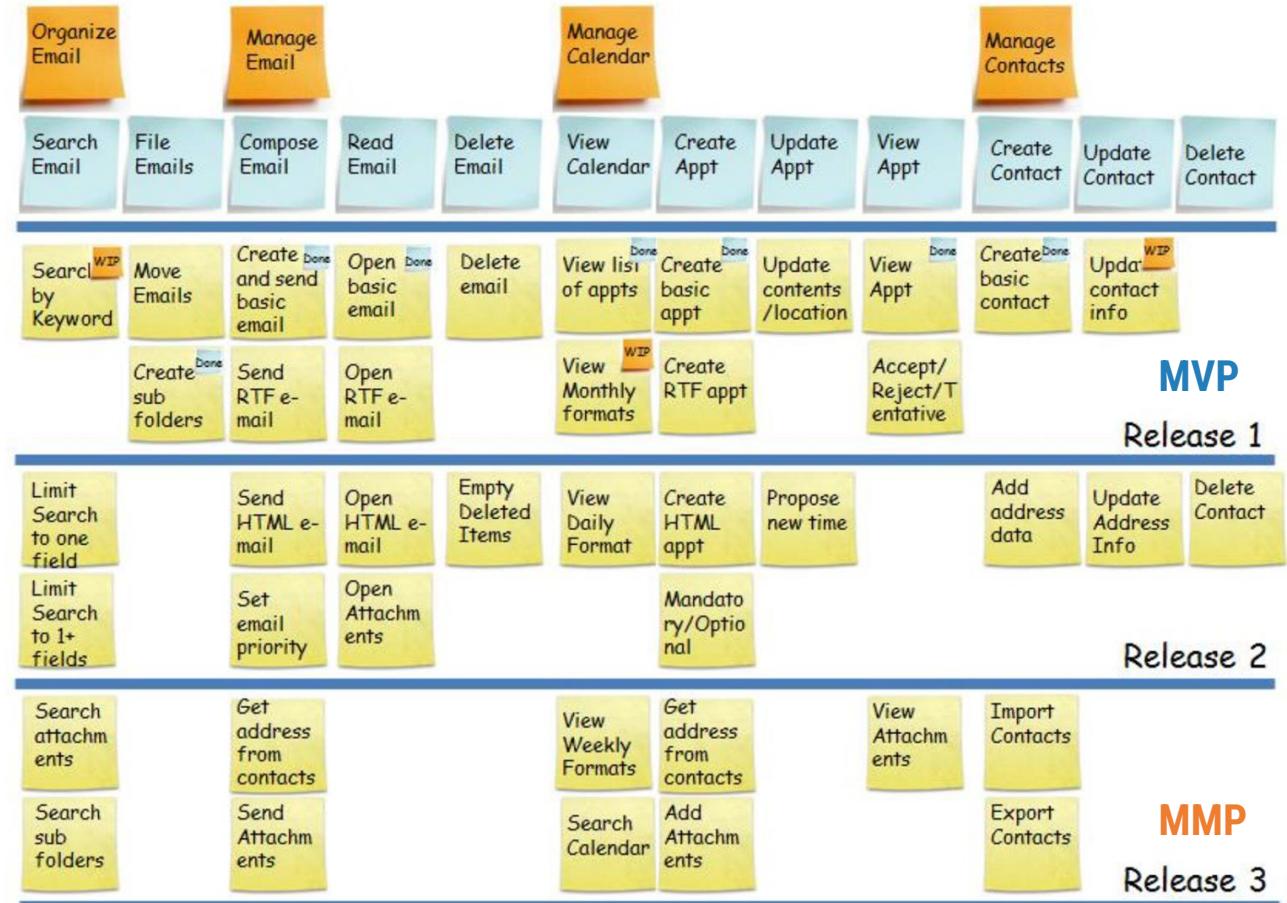
Prioriser : MVP / MMP

## Minimum Viable Product (MVP)

L'ensemble minimal de fonctionnalités permettant de rendre un service / portant une valeur ajoutée. Il est plutôt destiné à des utilisateurs pilotes / volontaires.

## Minimum Marketable product (MMP)

L'ensemble minimal de fonctionnalités permettant de lancer le produit sur le marché / d'être vendu. Il est destiné à n'importe quel utilisateur potentiel.



# Les principales approches agiles

Quizz

**<https://quizizz.com/admin/quiz/670419b5ad9847ea23f6b9b0>**



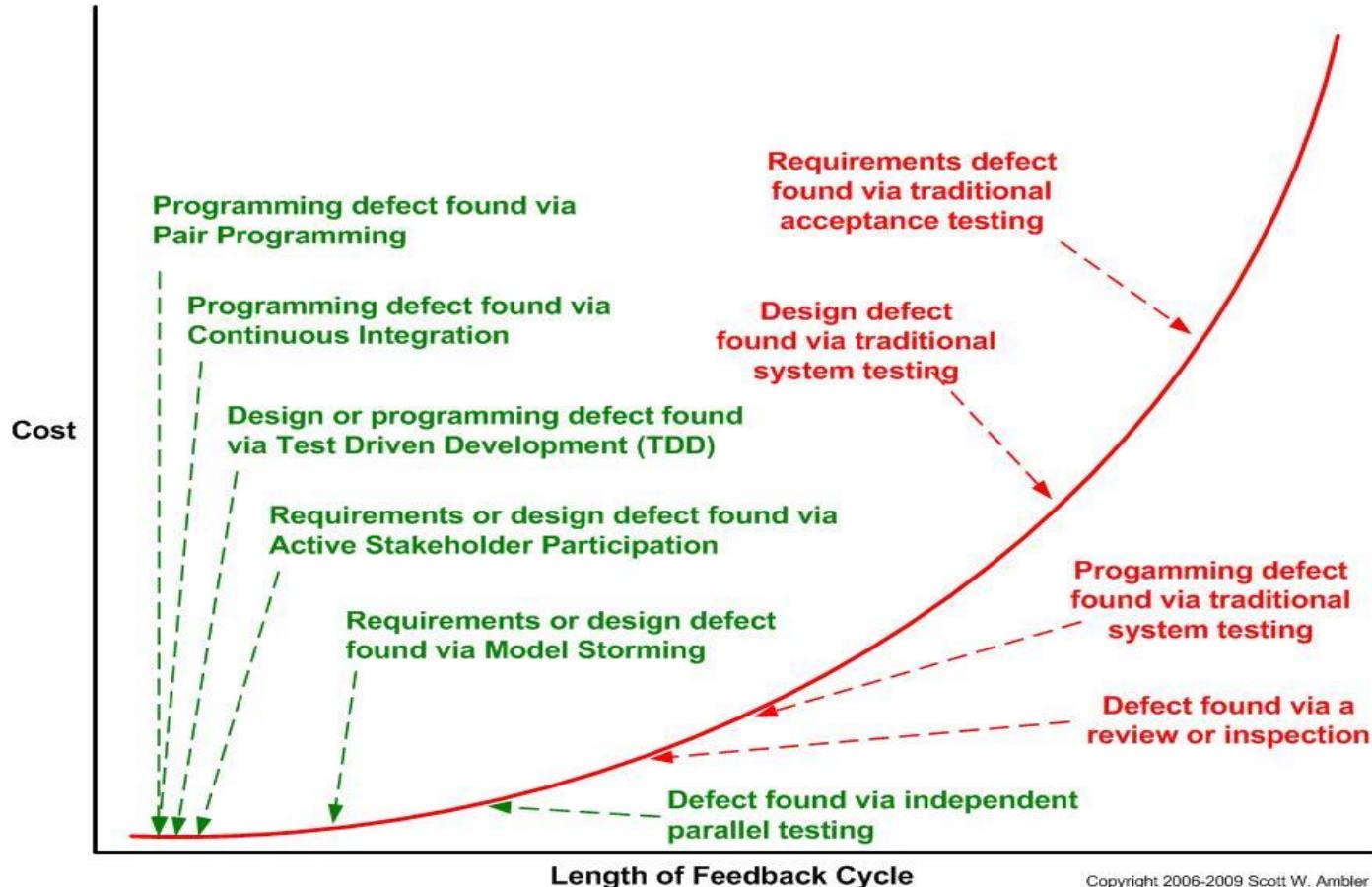


La qualité



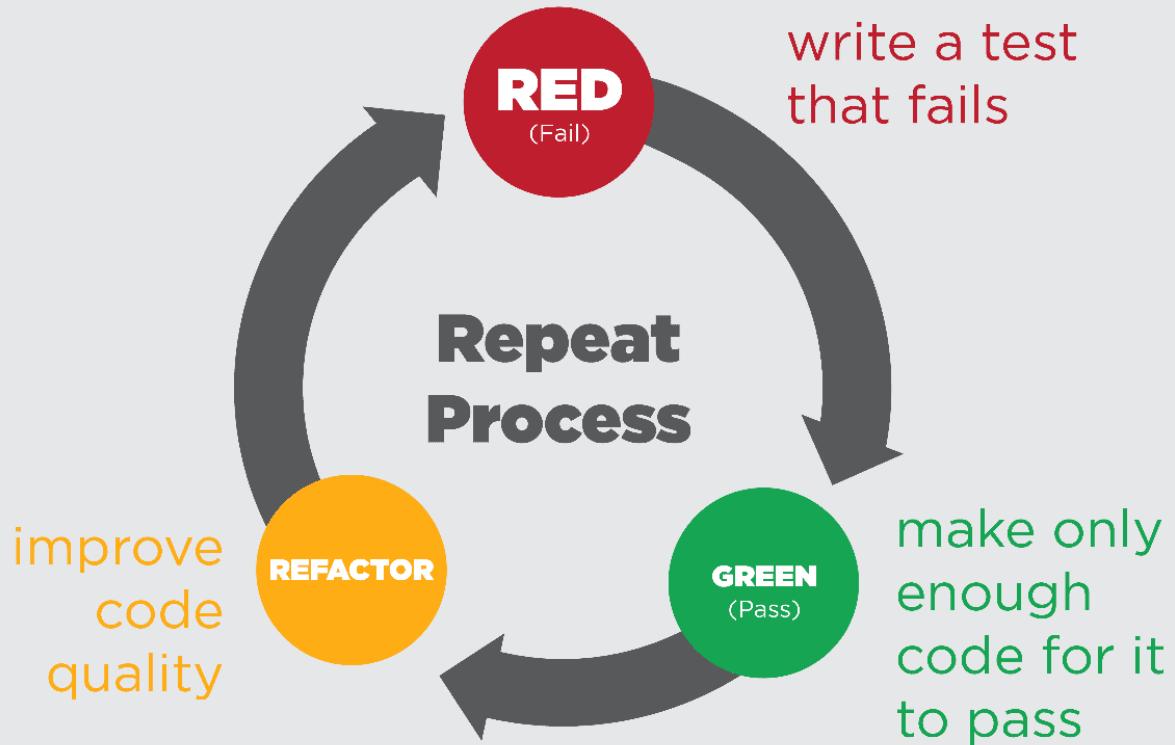
# La qualité

Différence des tests en agile et en cascase



# La qualité

TDD : Tests driven development



On écrit d'abord les tests, ensuite l'implémentation est réalisée.

- Grâce à cela, on s'assure d'avoir une fonctionnalité opérationnel,
- On peut refactorer sans crainte,
- On s'aperçoit des effets de bord au plus tôt (feedback)



# La qualité

## TDD : Tests driven development

```
GreetingTest.java  Greeting.java
1 import static org.junit.Assert.*;
2 import static org.hamcrest.CoreMatchers.*;
3
4 import org.junit.Test;
5
6 public class GreetingTest {
7
8     @Test
9     public void test() {
10
11         Greeting greeting = new Greeting();
12         assertThat(greeting.getMessage(), is("Hello world!"));
13     }
14
15 }
16
```

Problems @ Javadoc JUnit Declaration

Finished after 0.123 seconds

Runs: 1/1 Errors: 0 Failures: 1

GreetingTest [Runner: JUnit 4] (0.039 s) Failure Trace

test (0.039 s) java.lang.AssertionError:  
Expected: is "Hello world!"  
but: was null  
at org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:20)  
at GreetingTest.test(GreetingTest.java:12)

**Ecrit un test qui échoue**

**La classe Greeting est vide**

```
GreetingTest.java  Greeting.java
1
2     public class Greeting {
3
4         public Object getMessage() {
5             // TODO Auto-generated method stub
6             return "Hello world!";
7         }
8
9     }
10
```

Problems @ Javadoc JUnit Declaration

Finished after 0.122 seconds

Runs: 1/1 Errors: 0 Failures: 0

GreetingTest [Runner: JUnit 4] (0.049 s) Failure Trace

**Ecrire le minimum de code possible pour faire fonctionner le test**

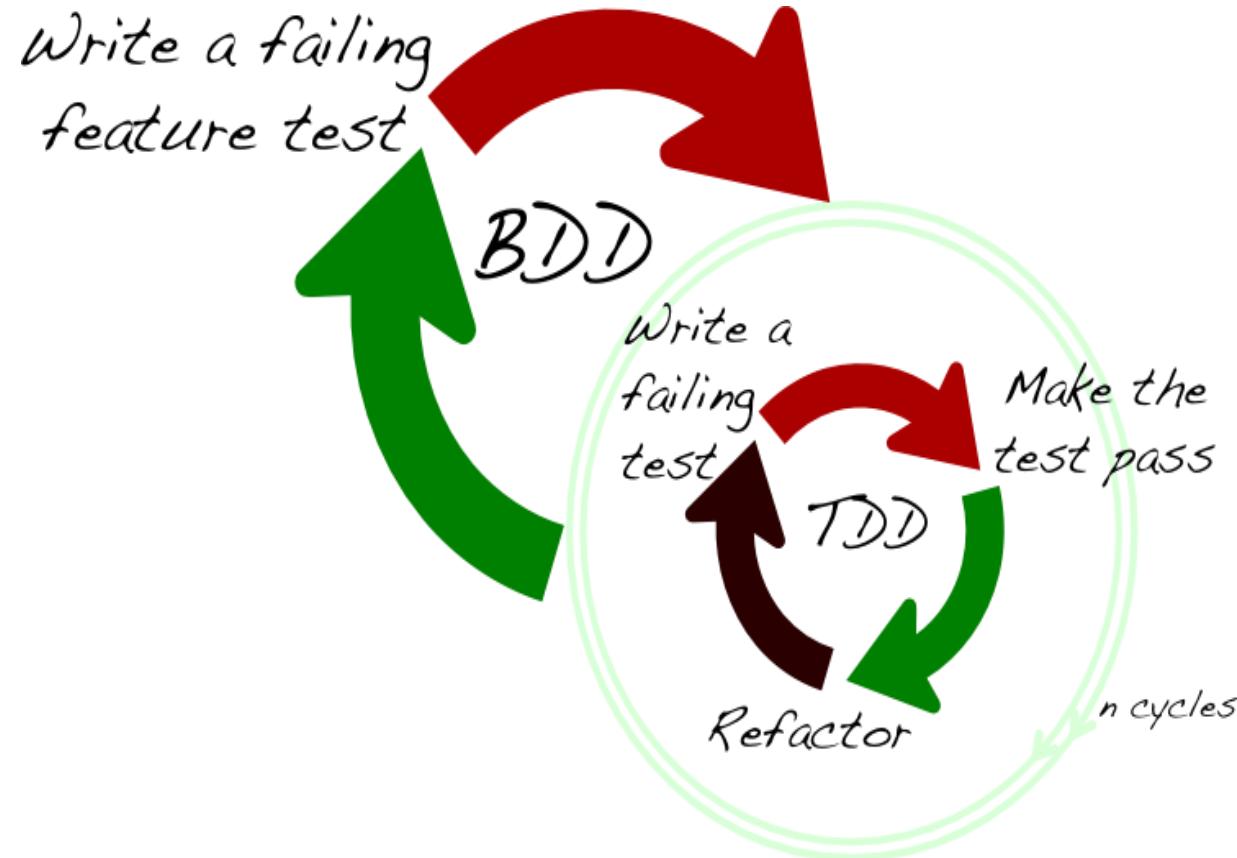
**Refactoriser le code mais jamais le test**



# La qualité

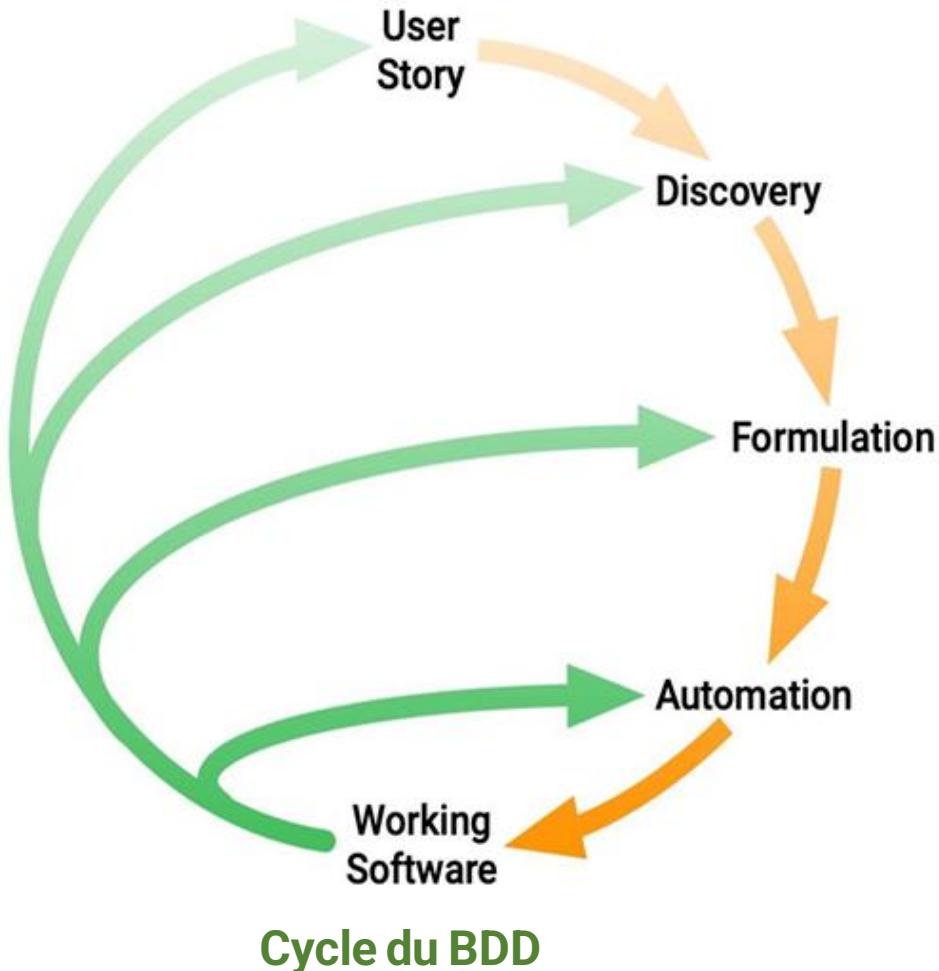
BDD : Behavior Driven Development

- Behavior Driven Development (BDD) est une méthodologie de **développement logiciel** qui met l'accent sur la collaboration entre les **développeurs**, les **testeurs** et les **représentants métier** (comme les Product Owners).
- L'idée principale de BDD est d'écrire des scénarios qui décrivent le comportement attendu d'une fonctionnalité du point de vue de l'utilisateur final.
- Ces scénarios sont rédigés dans un langage commun (souvent proche du langage naturel appelé **Gherkin**), compréhensible par toutes les parties prenantes, et ils servent de base à la création des tests.



# La qualité

BDD cycle et avantages



## Avantages

- ✓ Améliore la collaboration
- ✓ Exigences claires et compréhensibles
- ✓ Facilite l'automatisation des scénarios de test et la réutilisation du code dans les tests
- ✓ Alignement sur les besoins métiers,
- ✓ Détection précoce des bugs
- ✓ Documentation vivante



# La qualité

BDD exemple

Un exemple de scénario BDD serait rédigé ainsi :

- **Étant donné** une certaine situation,
- **Quand** une action spécifique est effectuée,
- **Alors** un résultat attendu doit se produire.

**Feature:** Get greeting

As a consumer of the greetings resource  
I should be able to get a greeting

**Scenario Outline:** Get greeting using appropriate caller

**Given** I use the caller <caller>

**When** I request a greeting

**Then** I should get a response with HTTP status code <status>

**And** The response should contain the message <message>

**Examples:**

caller	status	message
Duke	200	Hello World, Duke
Tux	200	Hello World, Tux

**Scenario:** Get greeting using caller 0xCAFEBAE

**Given** I use the caller 0xCAFEBAE

**When** I request a greeting

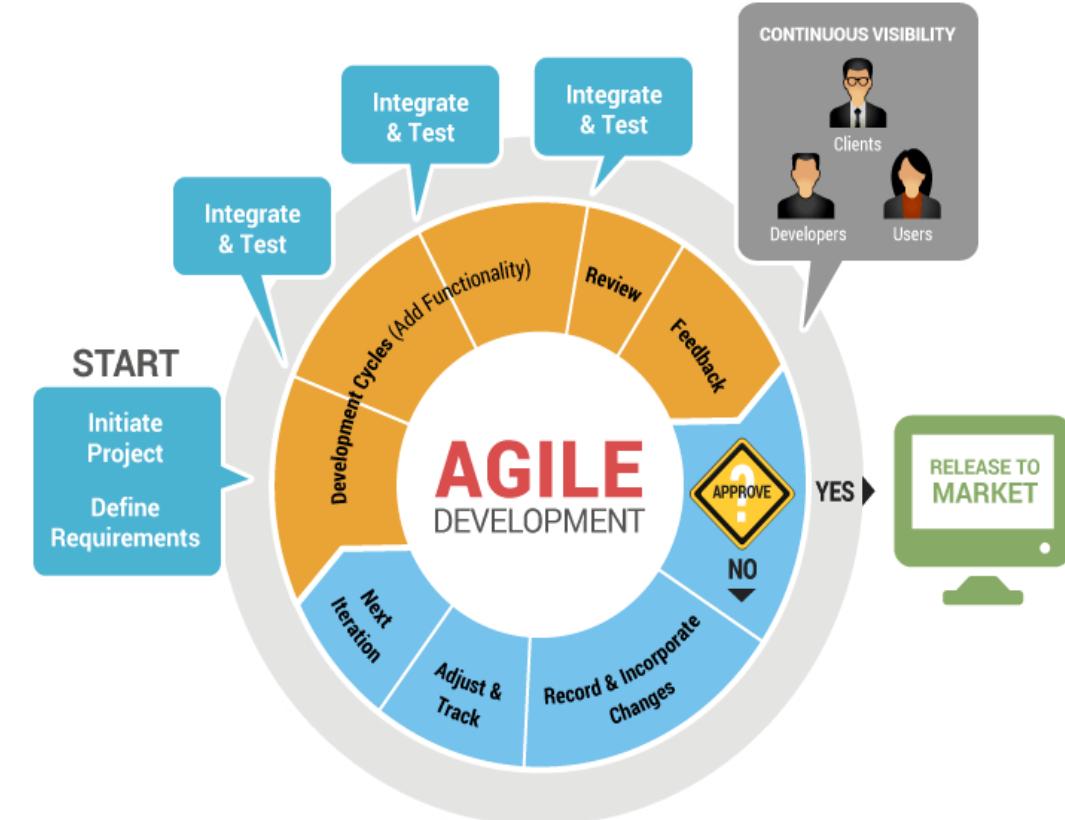
**Then** I should get a response with HTTP status code 418



# La qualité

L'importance des tests

- Les tests permettent de s'assurer que la solution fournit la réponse attendue dans les conditions attendues (temps de réponse, format de sortie ....).
- Les tests permettent de donner un feedback sur les réalisations.
- La stratégie de test du produit représente l'assurance qualité.

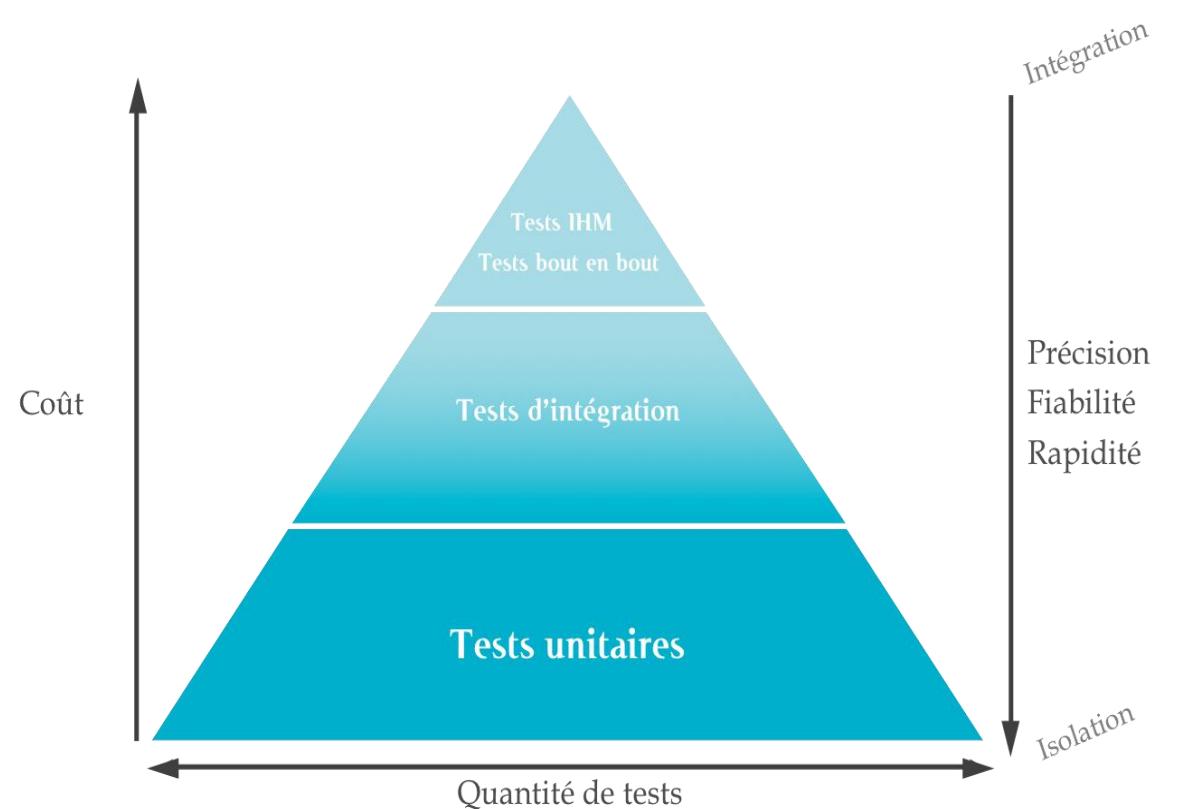


# La qualité

## Types de tests

Il existe une multitude de type de tests,  
aussi bien manuel que automatique:

- unitaire
- d'intégration
- E2E tests
- d'acceptance
- ....



# La qualité

Couts des tests

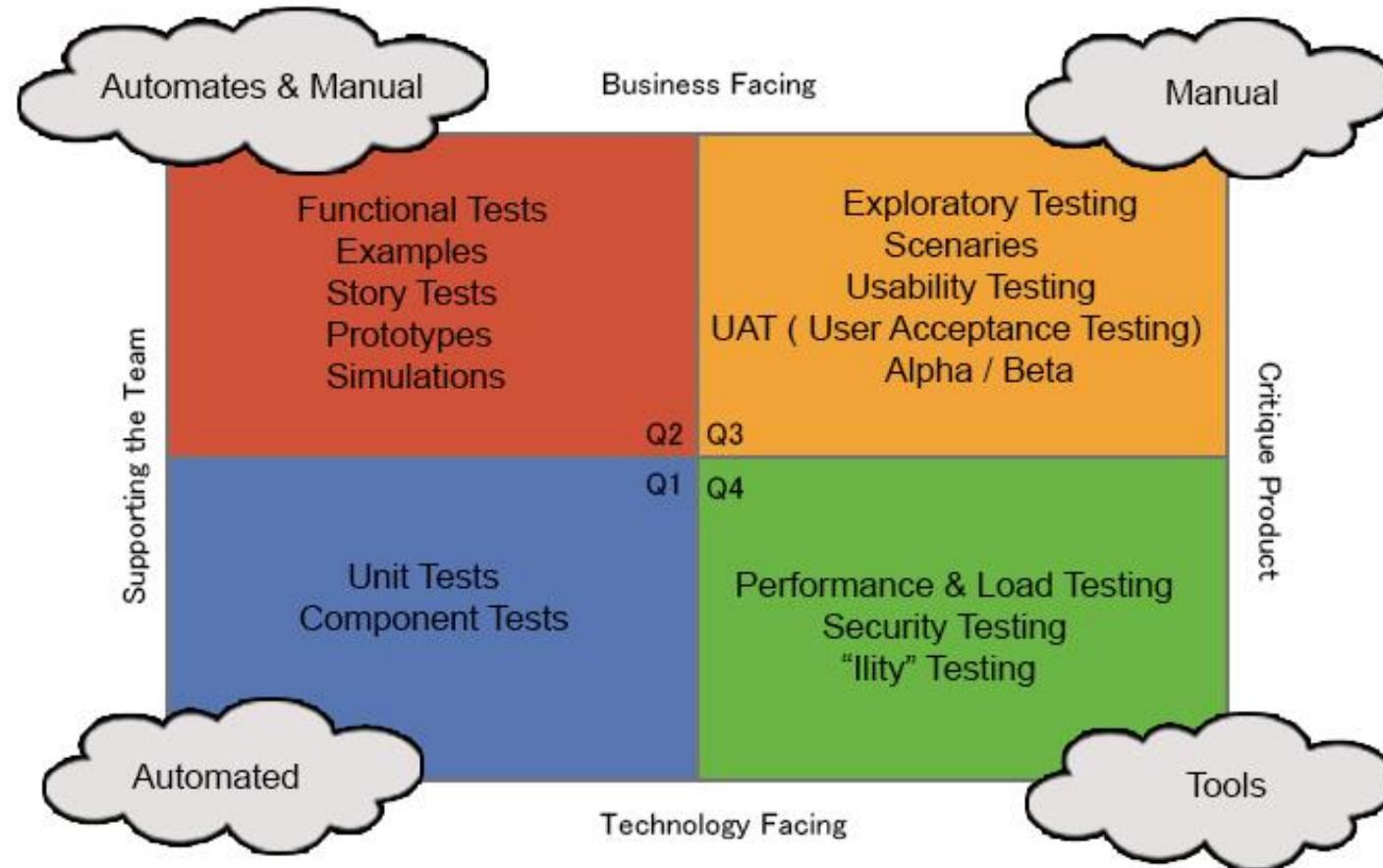
Type de test	Feedback			Coût (création + maintenance)
	Précision	Fiabilité	Rapidité	
Test Unitaire	Très fine (au niveau fonction)	Très fiable (répétable à l'infini)	Très rapide (quelques secondes)	
Test d'Intégration Test Fonctionnel	Moyenne (partie de logiciel intégré)	Fiable (les dépendances peuvent provoquer des échecs)	Assez rapide (quelques minutes)	
Test de bout en bout Test d'IHM	Faible (logiciel pleinement intégré)	Peu fiable (relativement instable)	Lent (plusieurs dizaines de minutes)	



# La qualité

Quadrants des tests

## The Agile Testing Quadrants



Source: Lisa Crispin, Brian Marick

[source](#)

Module Agilité - eilco





# Conclusion



- **Question 1 :**  
Quelles sont les différences entre les paradigmes des méthodes en cascade et des méthodes agiles ?
- **Question 2 :**  
Que pouvez-vous mettre en place pour améliorer la pertinence du produit et le fonctionnement de l'équipe ?
- **Question 3 :**  
Définissez le product backlog, les éléments le constituant, et la manière dont il est géré par le PO. Comment le PO estime la valeur d'une US ?
- **Question 4 :**  
Vous venez d'être nommé CEO d'une startup, les dernières fonctionnalités de votre nouveau phare prennent énormément de retard, en plus de cela les premiers tests sont catastrophiques. Quelles sont les mesures que vous appliquez afin d'améliorer la situation ?
- **Question 5 :**  
Vous devez réaliser un nouveau produit pour un nouveau client, il a déjà une liste de fonctionnalités et il souhaite les avoir à disposition dans 4 mois pour le lancement. Que lui proposez-vous avec les accords avec les méthodes agiles ?



# TP

## Objectif du TP

- L'objectif de ce projet est de concevoir une application de votre choix en adoptant la méthode agile (Scrum).
- Vous aurez l'opportunité de mettre en pratique les principes et techniques agiles étudiés en cours, tout en travaillant en équipe pour développer les fonctionnalités essentielles de l'application dans un délai limité.

## Instructions du TP

- Vous disposerez de 7 heures pour réaliser ce projet.
- Formez une équipe de 5 à 6 membres, en assignant les rôles prévus par la méthode Scrum :
  - 1 Product Owner (PO)
  - 1 Scrum Master
  - 3 à 4 Développeurs
- Chaque tranche de 2 heures correspond à un sprint.
- Au début de chaque sprint, effectuez une planification (sprint planning) qui servira de guide pour les 2 heures de travail suivantes.
- Avant la fin de chaque sprint, tenez une réunion quotidienne (Daily) d'une durée maximale de 5 minutes pour faire le point sur l'avancement.
- Favoriser la création des US (user story).
- La rétrospective de sprint n'est pas obligatoire, mais elle est fortement conseillée en fonction de votre avancement ; un bonus sera attribué aux équipes qui intègrent cette cérémonie.
- À la fin de chaque sprint fournir un incrément « fini » en fonction de ce qui a été embarqué dans le sprint backlog.
- À la fin du projet, fournir un document récapitulatif de votre progression, basé sur les éléments collectés durant les sprint planning et les Daily (il est important de bien décrire ce qui a été fait dans chaque sprint) il faut également expliquer votre organisation avec la méthode Scrum et présenter le projet réalisé, en mettant en avant les rôles, pratiques Scrum et fonctionnalités développées.



# TP

## Instructions du TP

- Tous les membres de l'équipe doivent participer activement au projet, en respectant pleinement les responsabilités de leur rôle. Chaque rôle doit s'en tenir à ses propres tâches : par exemple, seul le Scrum Master est chargé d'animer les réunions et de fournir les documents à la fin de chaque sprint/ Le PO gérer les tâches à développer et la priorité.
- La collaboration et la communication entre les membres sont essentielles.
- Respectez les principes agiles, tels que l'itération rapide, les retours réguliers et l'adaptabilité face aux changements.
- Il est recommandé d'utiliser des outils de gestion de projet agile, comme Jira ou AgileTaskBoard, pour visualiser les tâches, suivre le progrès et coordonner le travail de l'équipe.
- Le temps alloué pour ce TP est de 7 heures, il est donc important d'utiliser efficacement le temps disponible pour maximiser les résultats.
- PS : Je suis disponible pour vous aider, Bon courage

