

# LES METHODES AGILES

Nassur MHOUMADI  
SM Paris Inside

# SOMMAIRE

- Introduction

- Culture Agile

- Framework Scrum

- Scrum en pratique

# REGLES

01

**ECOUTE**

02

**VOLONTE**

03

**RESPECT**

04

**PARTAGE**

EMOTIONS  
DU JOUR





# 1 - INTRODUCTION



La Banque GiveMeUrMoney  
vous confie un projet  
d'intégration d'un progiciel métier



Tonton comme chef de projet s'engage sur un contenu à livrer en respectant un délai et un budget



Tonton réalise la  
conception du  
projet





Tonton délègue  
les tâches à  
l'Equipe



Chaque  
développeur code  
sa partie dans son  
coin



Après 2 mois,  
l'équipe dit qu'ils sont  
dans les délais, mais  
on ne peut rien  
montrer. La confiance  
s'érode



La fin approche, il  
est temps de mettre  
la pression à  
l'équipe



© Can Stock Photo - csp23351408

Les Heures  
supplémentaires  
s'enchaînent



On arrive à livrer ce  
qui est demandé,  
mais le client n'est  
pas satisfait



De son côté  
Tonton est  
satisfait, car il a  
respecté le  
conten, le  
budget et le  
délai



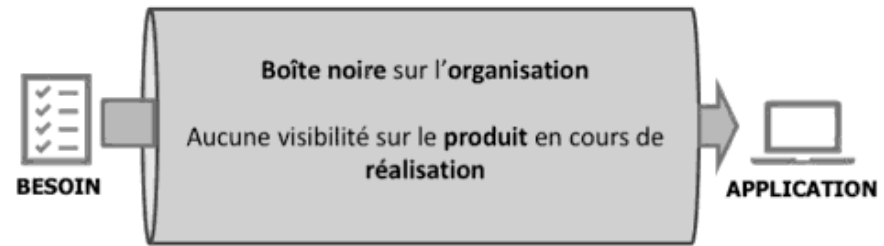
Quel est le  
problème ? La  
demande a été  
respectée



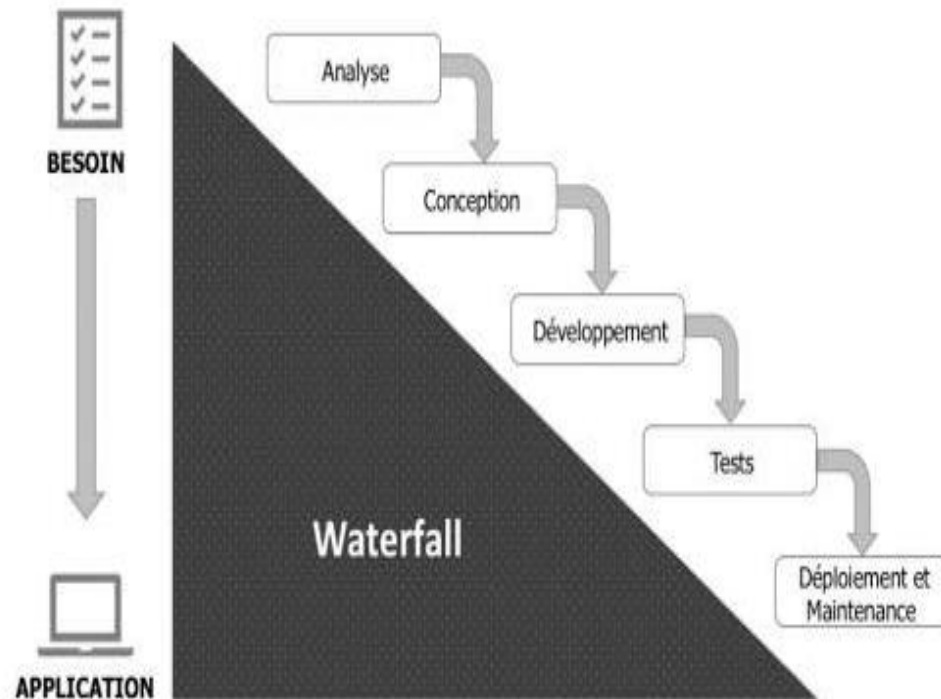
## QUELQUES PISTES

- ❑ L'équipe n'avait pas la parole
- ❑ Les changements ont été mal négociés
- ❑ Paul a décidé seul des tâches pour ses développeurs
- ❑ L'équipe a travaillé à partir d'informations données au fil de l'eau
- ❑ L'équipe a fait des heures sup



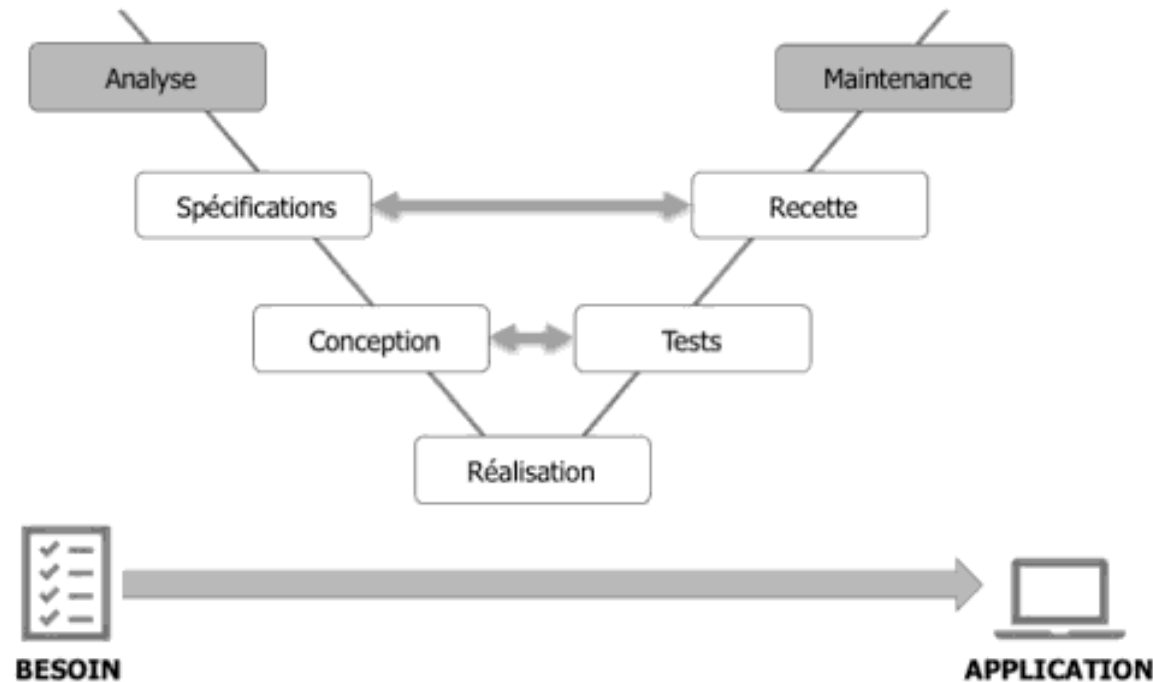


# AGILE VS CLASSIQUES



**Waterfall** : processus linéaire de création logicielle. Succession d'étapes allant du recueil des besoins jusqu'au Déploiement et à la maintenance du produit

# AGILE VS CLASSIQUES



**Cycle en V** : plus élaboré. Les étapes clés sont mises en relation 2 par 2. Il existe un lien de causalité entre les étapes. Par exemple, l'étape de recette se base sur les spécifications

# AGILE VS CLASSIQUES

	MÉTHODES AGILES	MÉTHODES CLASSIQUES
<b>Objectif</b>	Satisfaire l'utilisateur	Respecter le besoin initial et les engagements
<b>Changement</b>	Accepter le changement	Opposé au changement ou, en tout cas, moins enclin à l'accepter compte tenu des livraisons tardives et des processus de gestion lourds
<b>Livraison</b>	Livrer fréquemment	Livrer en une seule fois une application « finalisée »
<b>Équipe</b>	Travailler en synergie	Travailler de façon segmentée (chacun voit sa partie du travail)
<b>Moteur</b>	Stimuler la motivation	Stimuler la productivité
<b>Communication</b>	Communiquer en direct avec les opérationnels	Communiquer de façon verticale en passant par des relais hiérarchiques (par exemple, le chef de projet MOE est relais entre la MOA et ses développeurs)
<b>Indicateurs</b>	Un seul indicateur : les fonctionnalités implémentées	Justifier par les indicateurs (sans livraison intermédiaire, les indicateurs sont les seuls justificatifs de l'avancement, des écarts, etc.)

# AGILE VS CLASSIQUES

	MÉTHODES AGILES	MÉTHODES CLASSIQUES
<b>Rythme</b>	Bannir les <i>rushs</i> de production	Adapter la production aux contraintes projets
<b>Qualité</b>	<i>Que ce soit en méthode agile ou classique, l'excellence technique est essentielle. Ce neuvième principe sert principalement à justifier le fait que les méthodes agiles prônent la qualité des livrables techniques (contrairement à ce que peuvent en dire un certain nombre de détracteurs)</i>	
<b>Livrables</b>	Rester concentré sur l'essentiel : la production	Une documentation précise est essentielle pour assurer les échanges et la validation autour des besoins client
<b>Autonomie</b>	Favoriser une certaine autonomie des équipes	Encadrer scrupuleusement le travail des équipes
<b>Amélioration</b>	Intégrer la notion d'amélioration continue tout au long du projet	Introspection possible mais uniquement en fin de projet



L'argent n'est pas  
un problème pour  
le client. Il vous a  
choisi pour prendre  
le relais





## 2 – CULTURE AGILE

# Atelier 1 -“les clés du succès” ?



# Debriefing ateliers succès

- ❑ Avec l'évolution des technologies les compétences métiers et techniques ne suffisent pas
- ❑ Le leader d'aujourd'hui doit posséder des compétences difficilement mesurables, mais vitales, les soft skills (écoute, partage, empathie, adaptabilité...)
- ❑ Samuel Pierpont Langley Vs Frères Wright
- ❑ La force du Why



# 1 - EXPERIENCE

---



# 1 - EXPERIENCE

---

# CULTURE AGILE

□ Les méthodes agiles découlent d'une succession d'expériences qui remontent au XX<sup>e</sup> siècle

- 1960 – 1990 Amélioration continue et processus industriel
  - Edwards Deming – amélioration de la production de la qualité. PDCA.
- 1991 – RAD, la première vraie méthode itérative et incrémentale liée à la gestion informatique

# CULTURE AGILE

□ 1986 - The new new product development Game (Takeuchi et Nonaka) –  
([http://www.fabrice-aimetti.fr/dotclear/public/mes-documents/TheNewNewProductDevelopmentGame\\_French.pdf](http://www.fabrice-aimetti.fr/dotclear/public/mes-documents/TheNewNewProductDevelopmentGame_French.pdf))

# CULTURE AGILE

- ▣ **1991 – 1999 Naissance d'un écosystème**
  - ▣ De nombreuses méthodes ou cadre de travail voient le jour. Ils présentent des points communs au niveau de la qualité, l'humain, les livraisons fréquentes, cycles courts : DSDM (1994), Scrum (1995), RAD et XP (1999)



# 2001 - AGILE MANIFESTO

- ❑ **Quand** : 11 au 13 février 2001
- ❑ **Où** : Montagnes Wasatch (Utah)
- ❑ **Qui** : 17 experts
- ❑ **But** : promouvoir une approche différente du développement logiciel en s'attachant à livrer du produit de haute qualité

## Depuis 2009 – LE PRESENT DE L'AGILITE

- ❑ **2006 – DAD (Disciplined Agile Delivery – IBM )** Améliore integration de ensemble des équipes techniques et de leur savoir-faire sur les projets
- ❑ **2009 - DEVOPS** – Améliore intégration de ensemble des équipes techniques et de leur savoir-faire sur les projets.

# L'INFORMATIQUE ET AU DELA

❑ **Airbus – Sprints Innovation** Développer de nouveaux projets en 4-6 mois

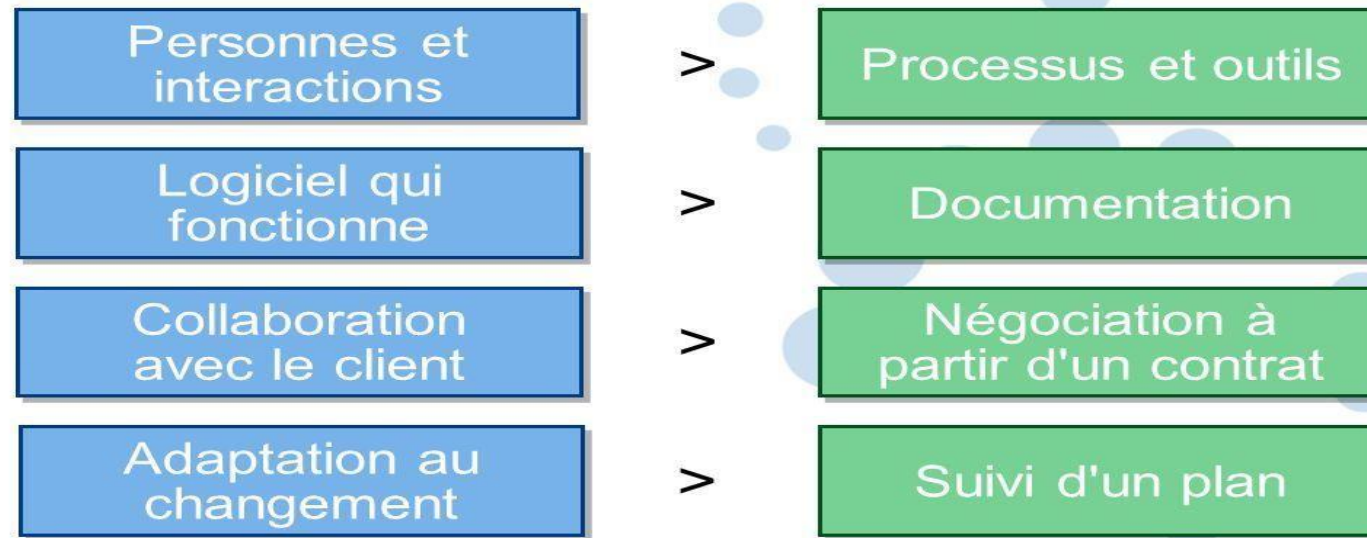
VIDEO → <https://www.dailymotion.com/video/x1xpacc>

❑ **WikiSpeed – Construction automobile Joe Justice.** Mettre au point un prototype de voiture à haute efficacité énergétique (2,3 Litres au 100 km) en moins de 3 mois

❑ **Agile pour les familles**

# LES VALEURS ET LES PRINCIPES DU MANIFESTE AGILE

## Manifeste Agile : valeurs



Source : [www.agilemanifesto.org](http://www.agilemanifesto.org)



# Les individus plutôt que le processus et les outils

- Mettre la communication et l'humain au centre de la démarche.
- La création d'un produit n'est pas à la charge d'équipes cloisonnées. Elle se nourrit des échanges constructifs entre tous les acteurs du projet

# Logiciel qui fonctionne plutôt qu'une documentation

- Focaliser la plus grande partie du temps sur la production et l'outil
- Nous ne remettons pas en cause la documentation et son utilité
- Il faut la rendre plus rationnelle, plus légère, moins chronophage, et plus simple à faire évoluer

# La Collaboration avec le client plutôt que des contrats rigides

- L'implication du client est un prérequis fort qui assure la réussite du projet
- Le client fait partie de l'organisation de manière active. Et, il est impliqué dans toutes les phases de création de son produit
- La relation contractuelle ne doit pas être remise en cause, mais simplement adaptée aux situations

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

- C'est dans l'adaptation d'un logiciel au fil de l'eau que l'agilité prend tout son sens
- Mais être agile ne signifie pas appliquer sans discernement dès que le client le souhaite
- Le plan initial doit servir de ligne directrice



# LES 12 PRINCIPES

1. Satisfaire le client
2. Accueillir le changement
3. Livrer fréquemment
4. Travailler quotidiennement avec les utilisateurs ou leur représentants
5. Créer un environnement qui soutienne l'équipe
6. Communiquer en face à face
7. Mesurer l'avancement sur un logiciel opérationnel
8. Avoir un rythme de développement soutenable
9. Porter une attention continue à l'excellence technique
10. Minimiser la quantité de travail inutile
11. Avoir une équipe auto-organisée pour faire émerger les solutions
12. Inspecter et s'adapter régulièrement

# FACTEURS DE SUCCÈS

- ☐ **Le client sait ce qu'il veut dès le départ**
- ☐ **Les besoins peuvent changer**
- ☐ **L'équipe est auto organisée**
- ☐ **L'utilisateur est impliqué quotidiennement**

# Une multitude de méthodes

## ❑ **RAD – James Martin – 1991**

Modèle de développement dit semi itératif. Les 1ères et seconds phases correspondent à l'expression de besoin et à la conception. Et, la troisième phase correspond aux iterations de développement

## ❑ **Crystal – Alistair Cockburn – 2004.**

Chaque projet est différent, la méthode doit s'adapter aux spécificités de chaque projet

## ❑ **Extrême programming – 1996 – Kent Beck**

Pousser à l'extrême les bonnes pratiques de réalisation de projet.  
Choisir la solution la plus simple et livrer plusieurs fois par jour.

# DevOps UNE EVOLUTION NATURELLE

## ❑ DevOps – 2009 – Patrick Debois

Dev (Développement) et Ops (Opérations ou exploitation en français). L'objectif est de réunir ces 2 mondes afin de garantir le meilleur des produits.

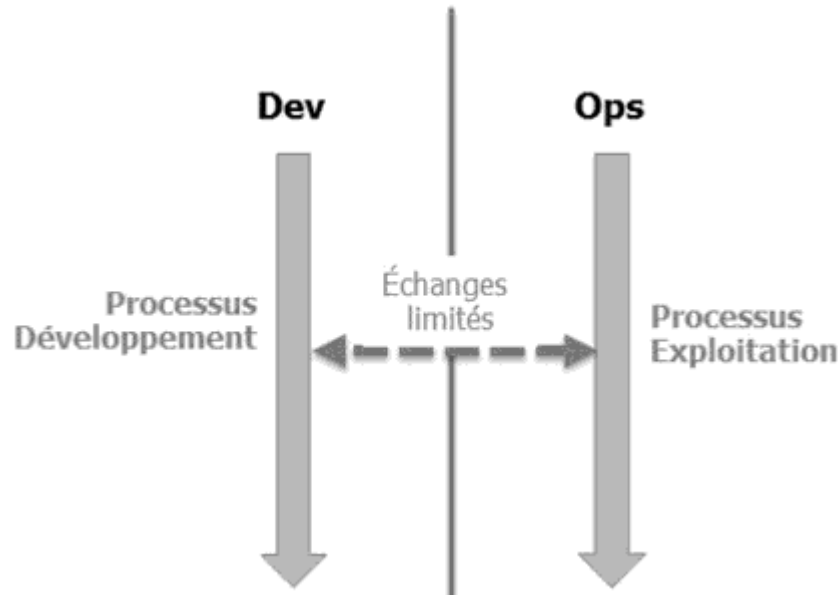
Dev : accent mis sur la réussite de l'application développée

Ops : accent mis sur la stabilité et la sécurité de l'infrastructure

# DevOps UNE EVOLUTION NATURELLE

❑ Ces objectifs distincts ont pour conséquence des différences de pratiques au niveau contractuel, des processus, de la communication, des outils utilisés...

Malgré toutes ces oppositions, le DevOps soulève un paradoxe : les développeurs ne développent rien sans infra et inversement.



# DevOPS, UNE SEULE EQUIPE POUR UN SEUL OBJECTIF

**❑ Pour pallier aux problèmes récurrents de qualité, on rapproche les développeurs aux équipes liées à l'infrastructure à travers le DEVOps**

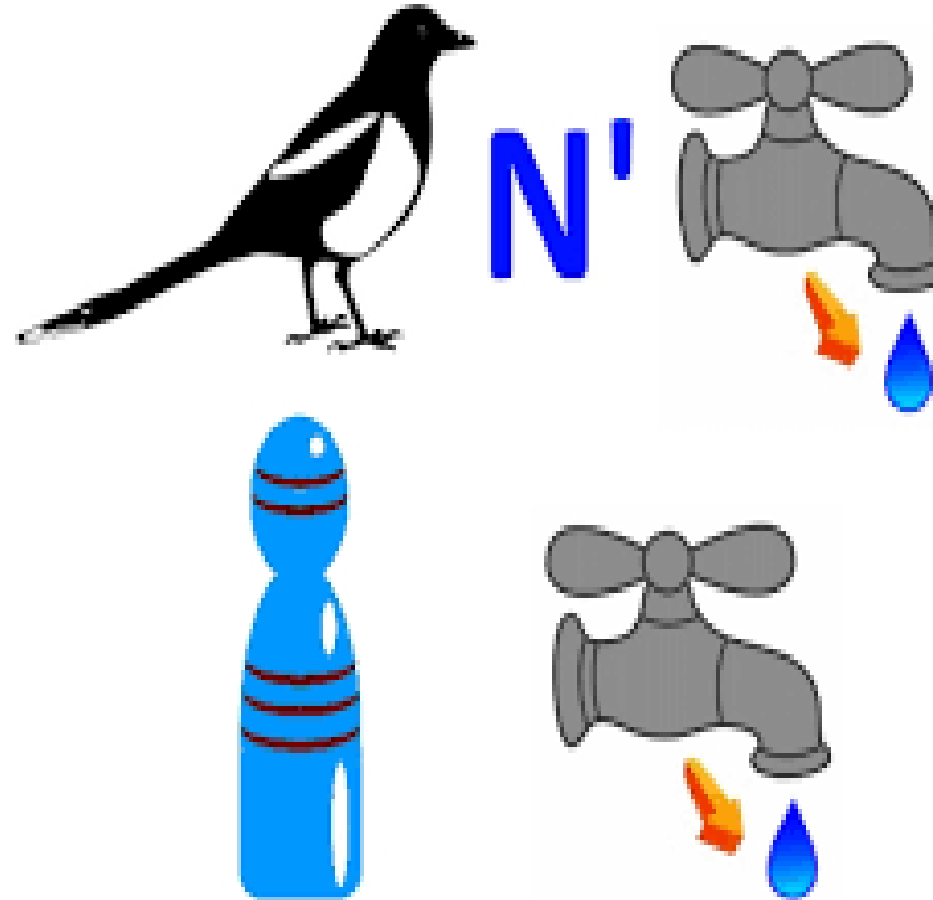
Dev (Développement) et Ops (Opérations ou exploitation en français).  
L'objectif est de réunir ces 2 mondes afin de garantir le meilleur des produits.

Dev : accent mis sur la réussite de l'application développée

Ops : accent mis sur la stabilité et la sécurité de l'infrastructure

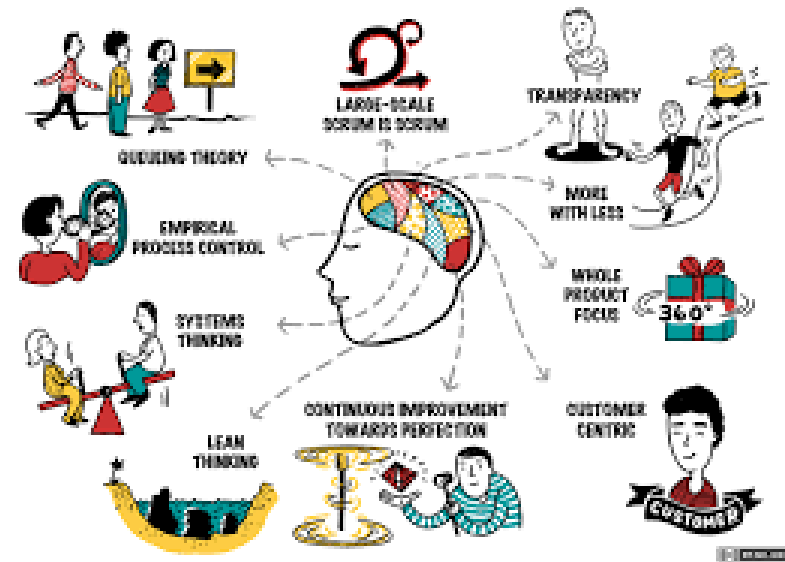
.

## ATELIER 2 – AGILE REBUS



DEVINETTE DU JOUR.COM

### 3 – FRAMEWORK SCRUM





# 3 – BRÈVE PRÉSENTATION DE SCRUM



# Les Rôles dans Scrum

---



# RÔLE SCRUM MASTER – Missions

- ❑ Responsable méthodologie : il s'assure de la bonne application de scrum et de l'agilité d'une manière générale
- ❑ Organisateur : accompagne l'équipe sur l'organisation des différents éléments (daily scrum, sprint planning...). Il contrôle la bonne tenue et le bon déroulement du dispositif

# RÔLE SCRUM MASTER – Missions

- Inspection et adaption au niveau de l'organisation
- Faciliateur : fluidifier la communication entre les acteurs et faire tomber les barrières.

# RÔLE SCRUM MASTER –

## Compétences

- Une base de connaissances scrum et agile
- Une capacité à transmettre et à évaluer les acquis. Le SM doit être capable d'évaluer le niveau des acquis et adapter ses interventions et réorienter la méthode si nécessaire

# RÔLE SCRUM MASTER –

## Compétences

- Une intelligence relationnelle. Résoudre les conflict. Faire travailler les gens ensemble dans le cadre scrum. Créer un synergie.

# RÔLE SCRUM MASTER – Positionnement dans l'équipe

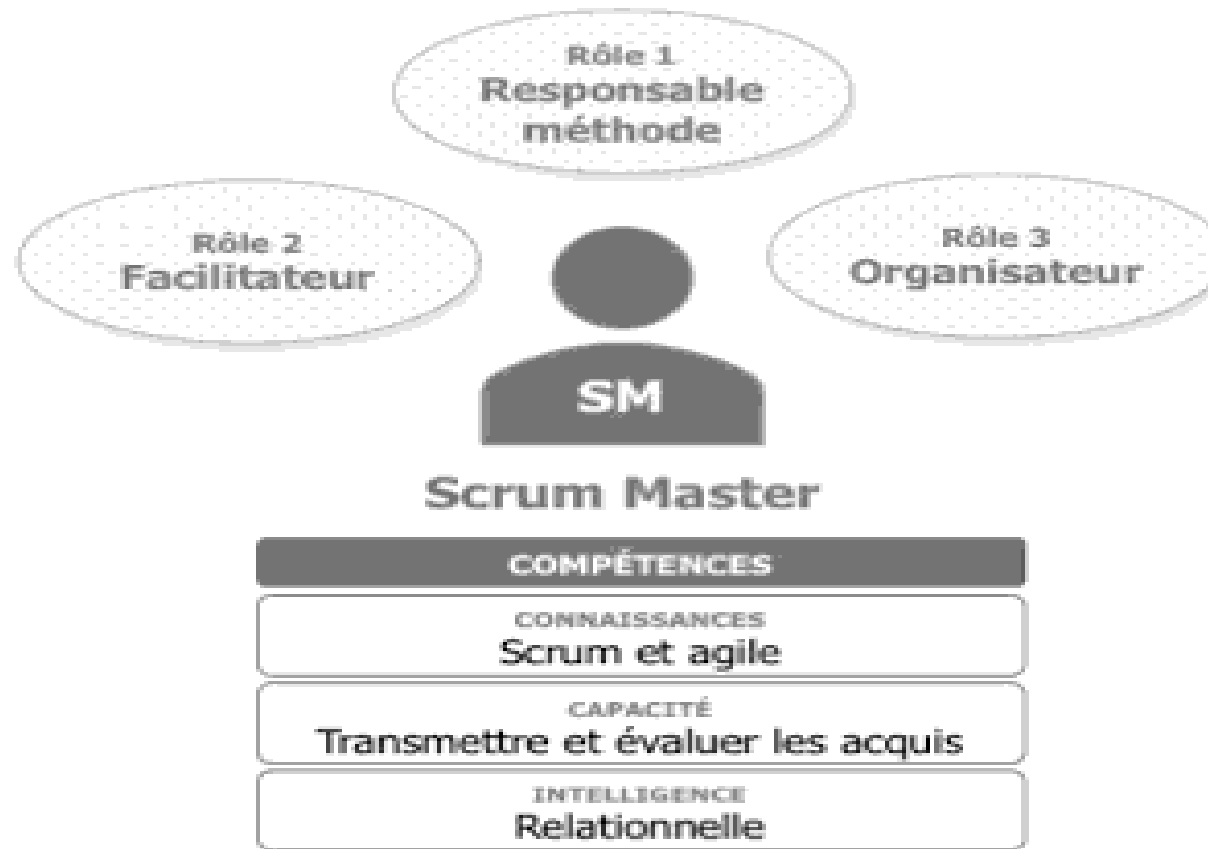
- ❑ Il n'exploite pas les compétences. Il motive les individus. Il tire l'équipe vers le haut en l'aidant à avancer. Il laisse une certaine autonomie aux membres de l'équipe afin que ces derniers donnent le meilleur d'eux même.
- ❑ Il n'est pas au dessus, il est dans l'équipe. Le SM est avant tout un membre de l'équipe. Il n'est pas le responsable de l'équipe.

# RÔLE SCRUM MASTER – Positionnement dans l'équipe

- Il ne dirige pas. Il guide.
- Il ne cherche pas à qui a fait la faute, mais trouve une solution.



# RÔLE SCRUM MASTER



# RÔLE PRODUCT OWNER -

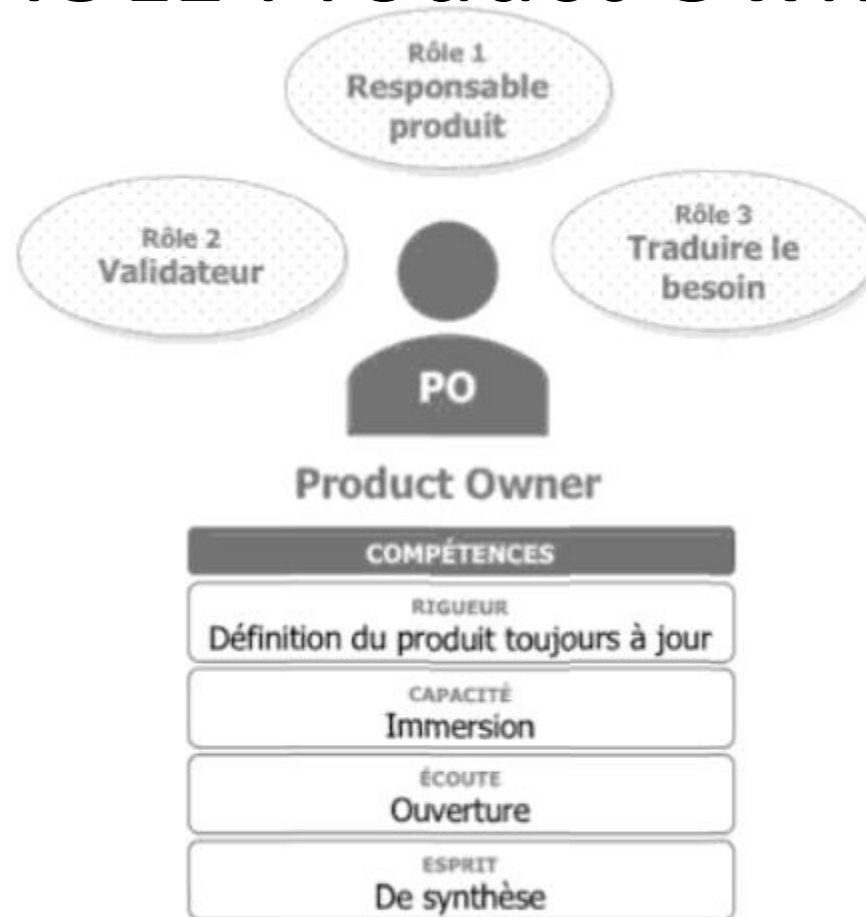
## Compétences

- ❑ Rigueur dans la maintenance et la mise à jour du Product Backlog.
- ❑ Capacité d'immersion.
- ❑ Ecoute et ouverture.
- ❑ Rend explicite les besoins du Product backlog

# RÔLE Product Owner– Positionnement dans l'équipe

- Comme pour le Scrum Master, le PO est représenté par une personne.
  - Le Product Owner est le Chef du produit. Il est le référent sur toutes les décisions et actions qui tournent autour du logiciel.
  - Il est le maillon central de Scrum, car il est situé entre les réalisateurs et les utilisateurs

# RÔLE Product Owner



# Equipe développement

- ❑ Mode de fonctionnement, auto-organisation.  
Principe n°11 “Les meilleures architectures, specifications et conceptions emergent d’équipe auto organisées”. Scrum impose ce type d’organisation.

# Equipe développement – Missions

- ❑ Transformer les besoins transmis par le PO en Nouvelles fonctionnalités dans l'application. 3 rôles incumbent à l'équipe de développement :
  - Responsable de l'increment.
  - Chargée de l'estimation
  - Réalisateur

# Equipe développement – Compétences

- On trouve des compétences qui couvrent l'intégralité de la chaîne de réalisation :
  - Conception technique et fonctionnelle
  - Architecture, développement,
  - Administration système

# Equipe développement – Positionnement

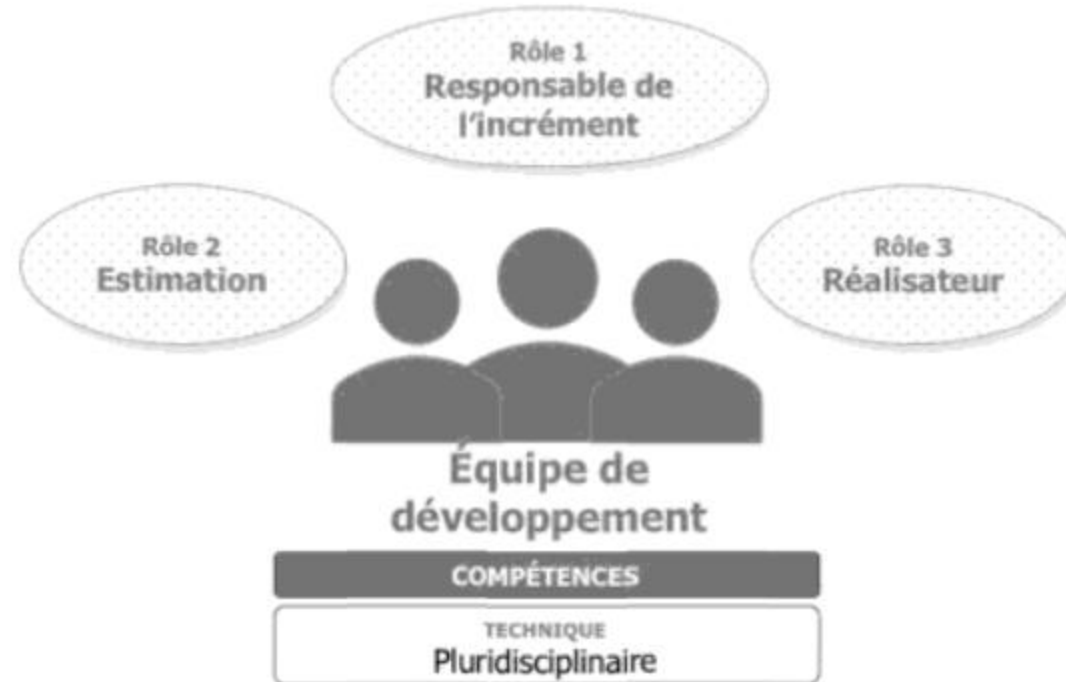
- Taille maximale, 9 personnes. L'objectif est d'optimiser :
  - la souplesse et la réactivité de l'équipe
  - la capacité à produire dans le respect des délais et de la qualité
  - les échanges et la communication qui se complexifient à mesure que l'équipe s'agrandit



# Equipe développement – Positionnement

- Les membres de l'équipe de développement sont particulièrement attentifs et actifs :
  - l'élaboration du planning et du chiffrage en raison de leur responsabilité sur l'estimation du projet
  - la priorisation des éléments du product backlog liée à la contrainte technique
  - les choix technologiques et leur mise en oeuvre

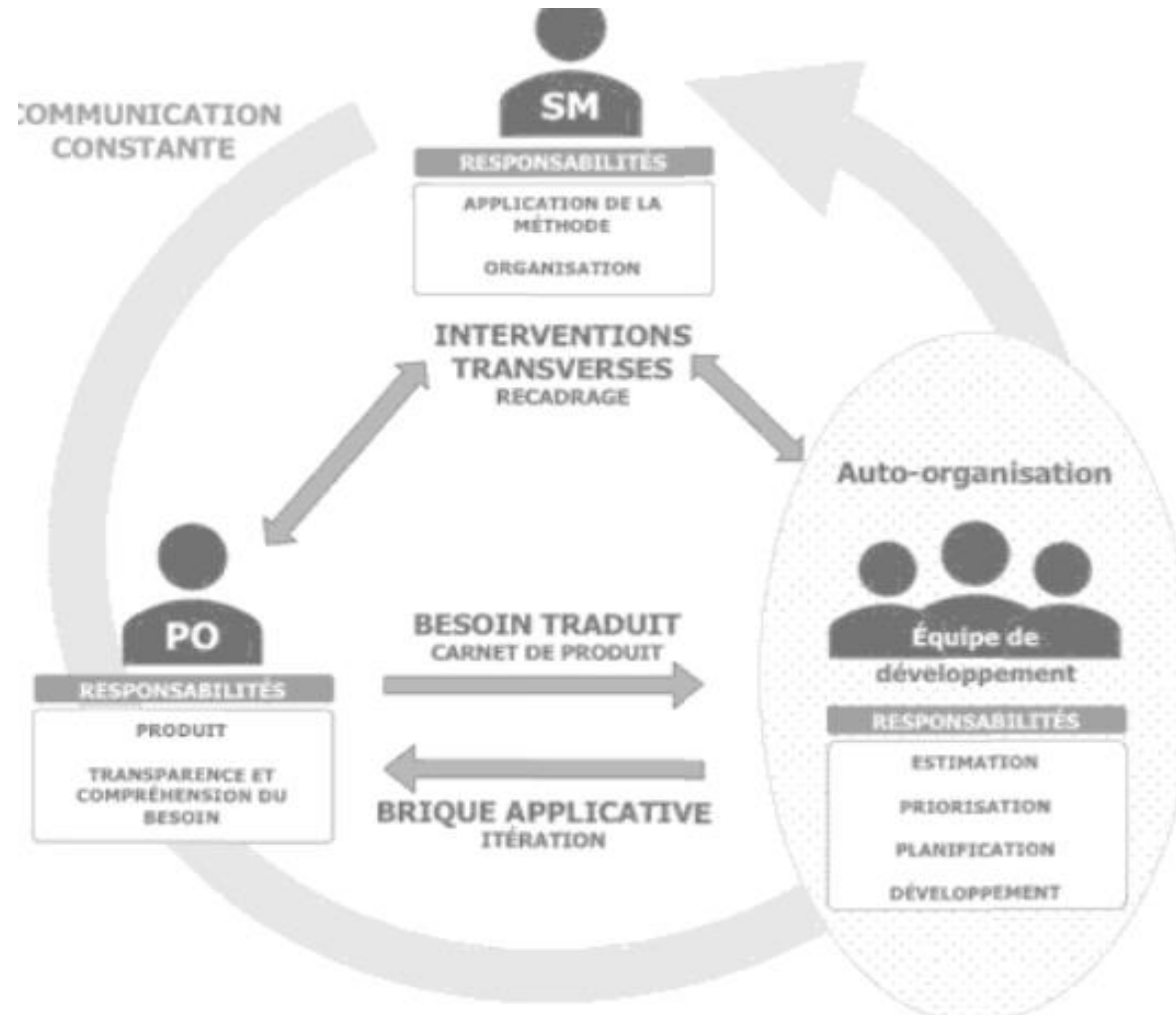
# RÔLE Development Team



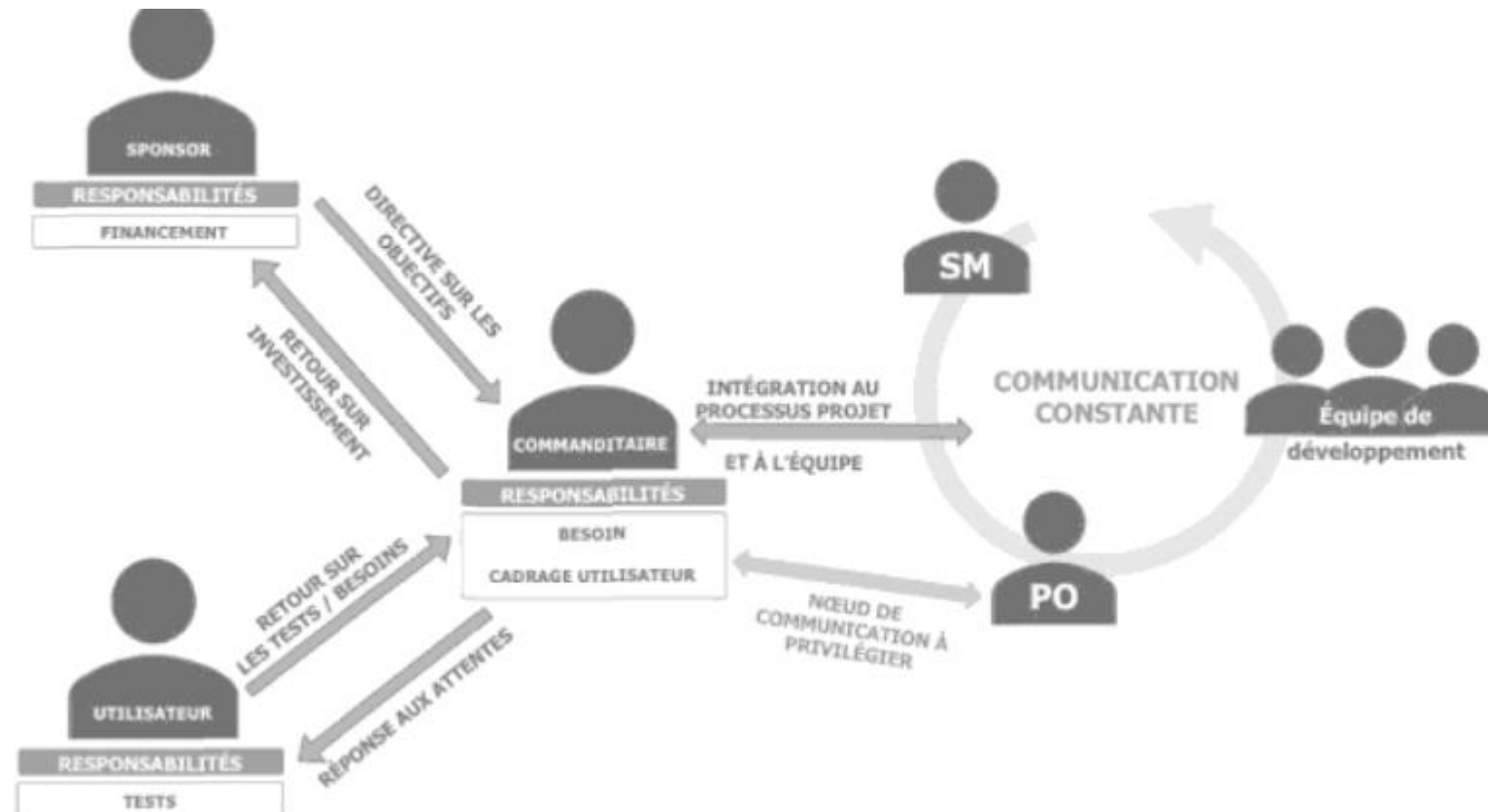
# Les parties prenantes

- Les commanditaires (clients)
  - Les utilisateurs
  - Les financeurs

# Organisation de l'équipe Scrum



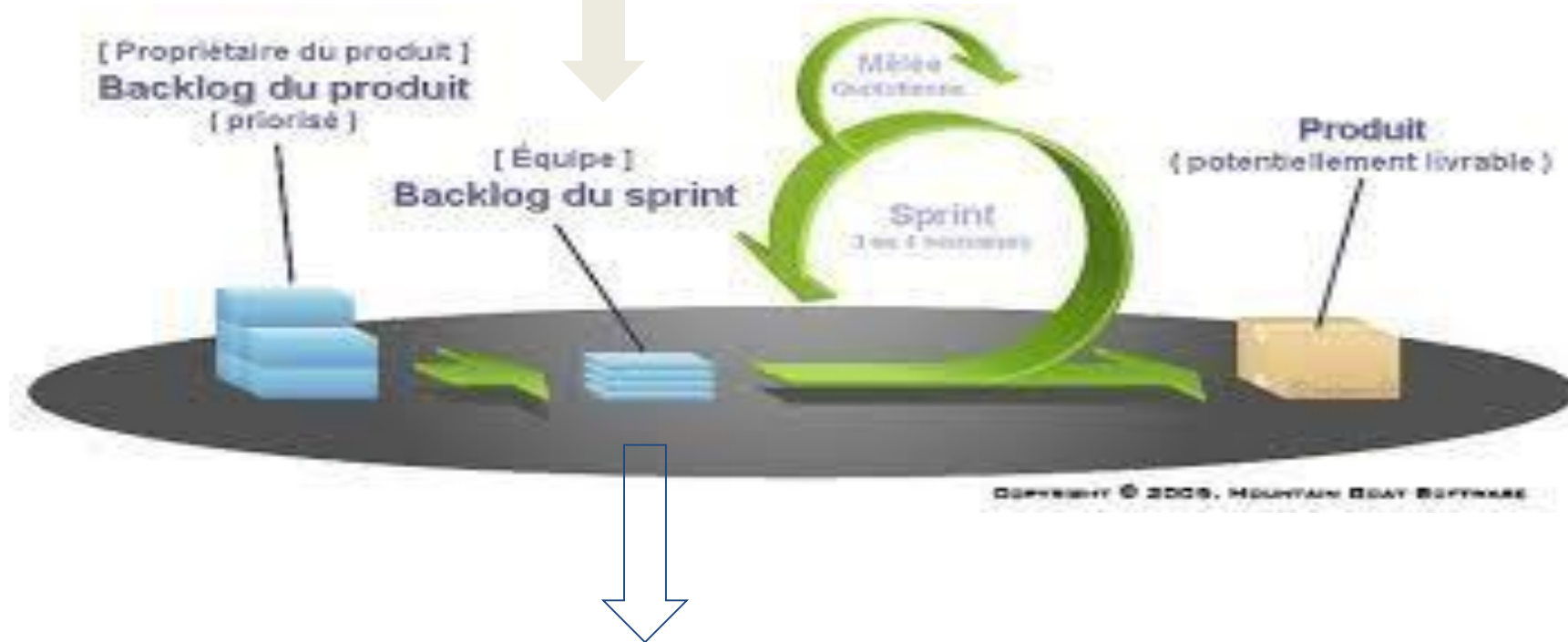
# Organisation de l'ensemble des parties prenantes



# Evenement dans Scrum



# PLANIFICATION SPRINT



# PLANIFICATION SPRINT

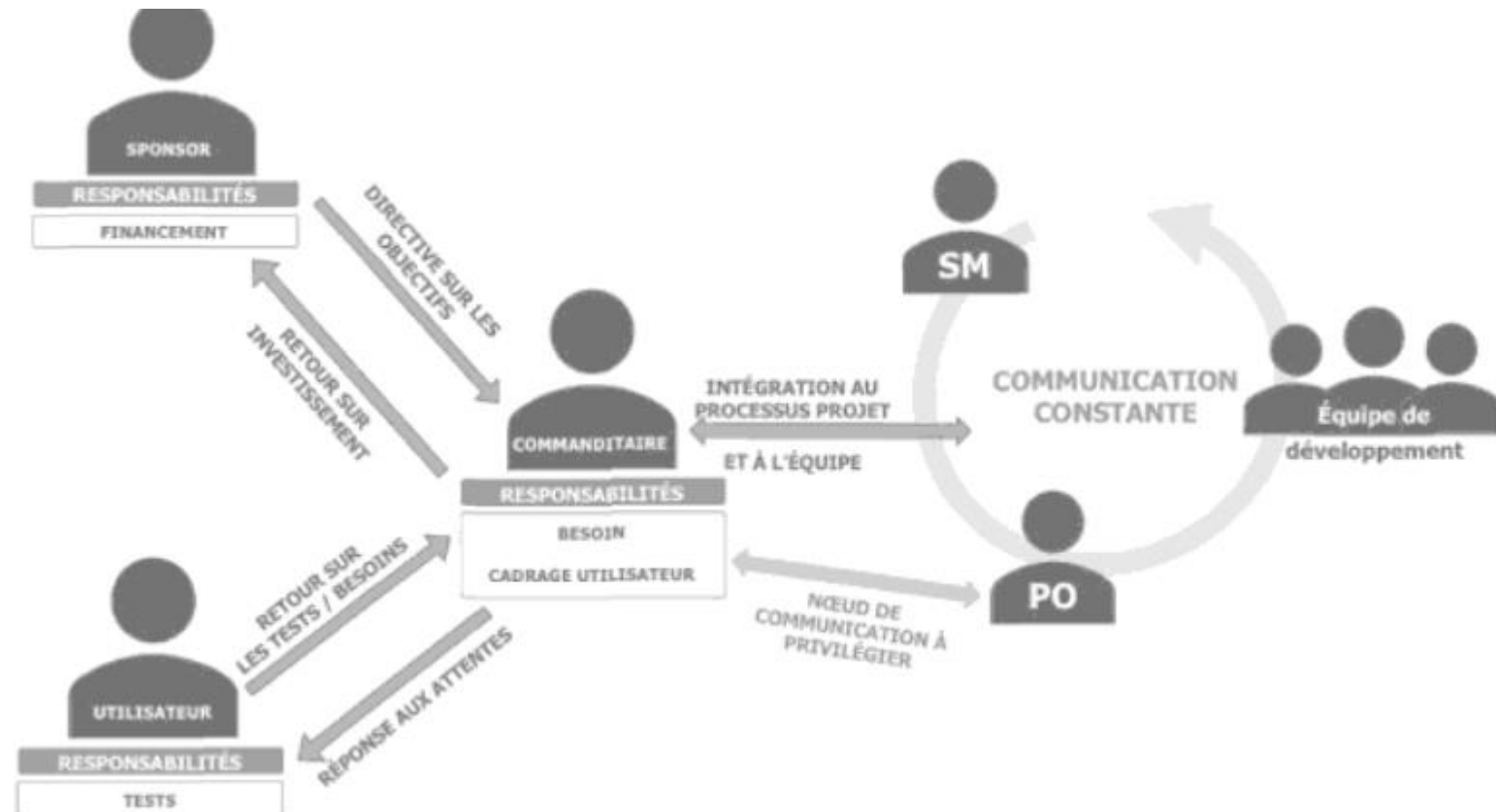
 Choisir et s'engager collectivement

:

- objectif sprint
- les user stories du product backlog embarquées dans le sprint backlog



# Organisation de l'ensemble des parties prenantes

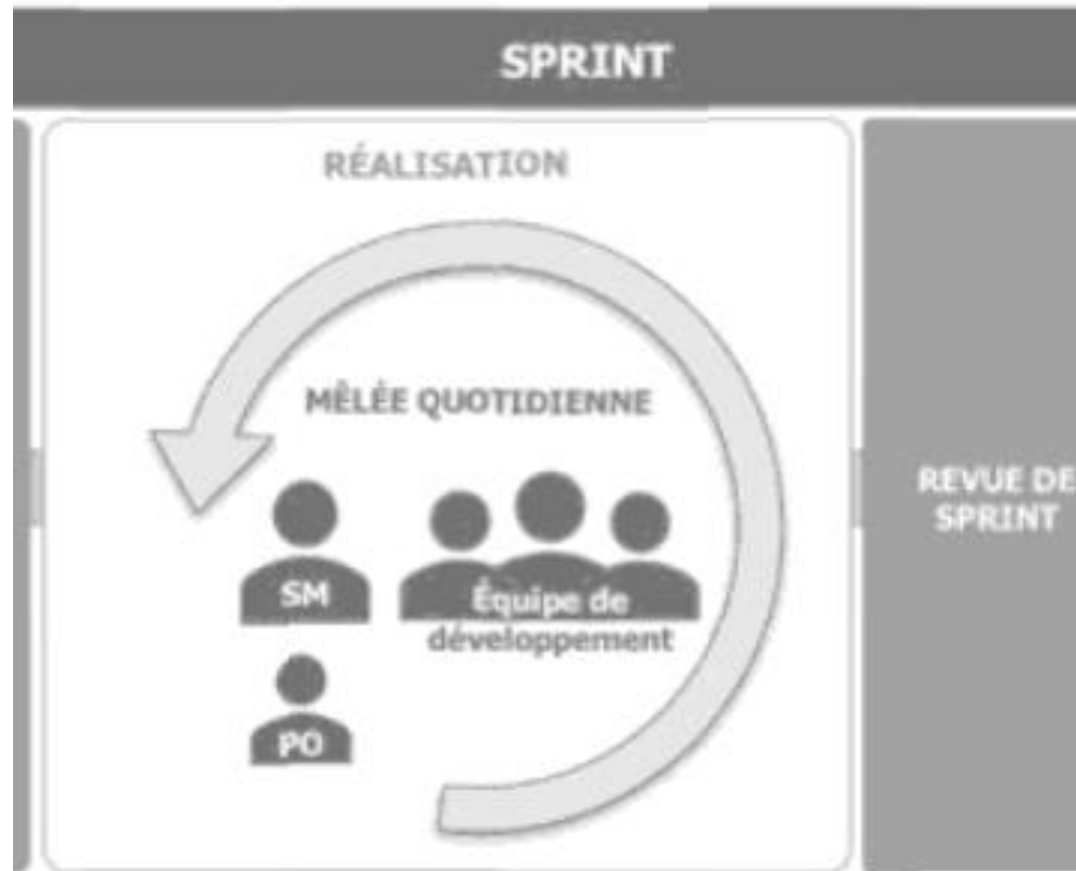


# SPRINT

- ❑ Réalisation des fonctionnalités du sprint backlog avec un PO disponible et une équipe focalisée
- ❑ Management visuel (kanban et burn down chart)

# MÊLÉE QUOTIDIENNE

- Menée par l'équipe
  - 15 minutes
- 3 questions : Qu'aj fait hier ? Que vais je faire ce jour ?
- Problèmes



# SPRINT REVIEW

- ❏ **Démonstration de l'incrément réalisé**
- ❏ **Tous les acteurs participent**
- ❏ **Démonstration**

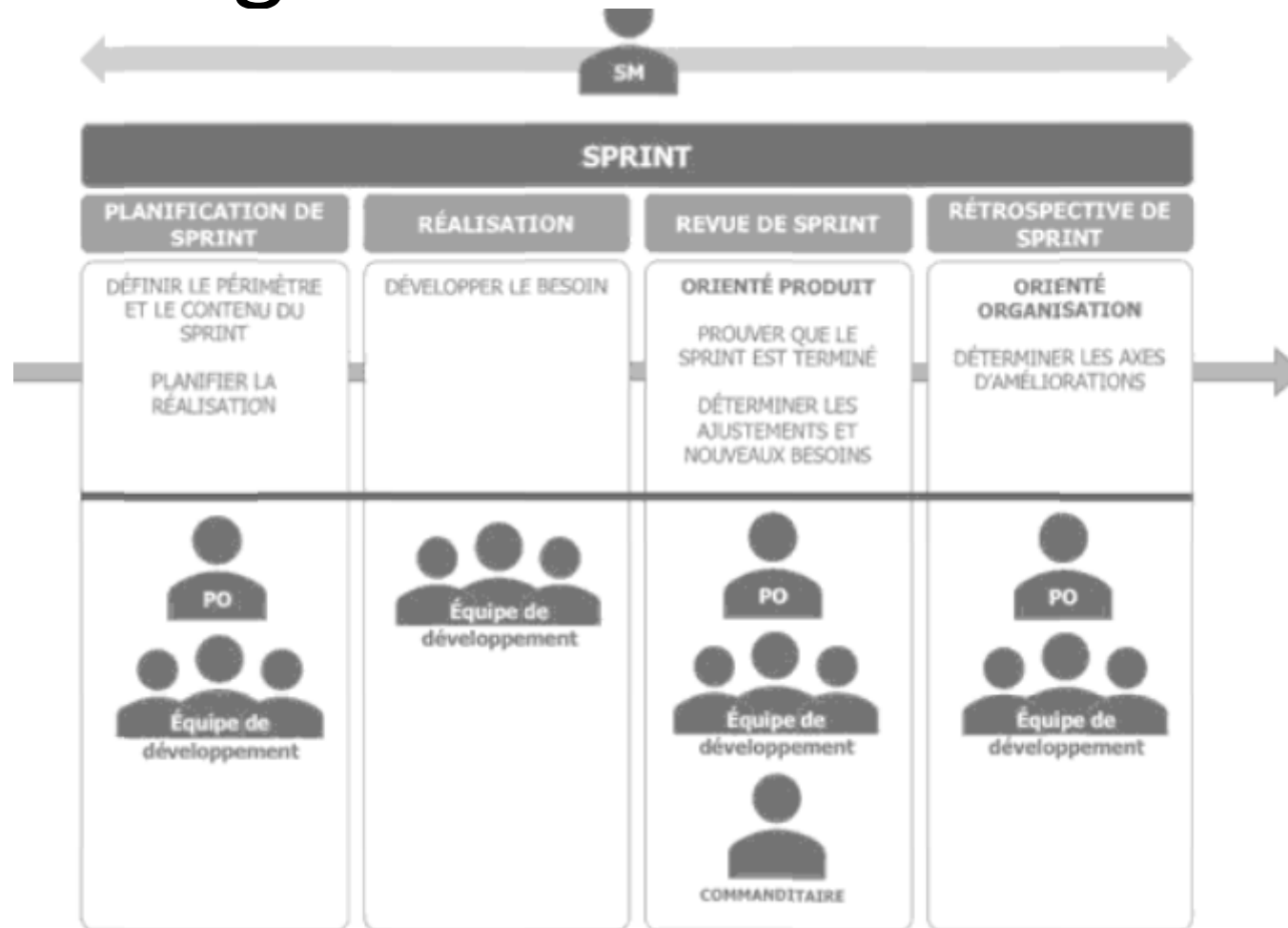
# RETROSPECTIVE SPRINT

 Ateliers d'amélioration continue

-Introspection

-Adaptation

# Organisation evenement



# Artefact dans Scrum



# Product Backlog

- ❑ Il est unique.
- ❑ Il est considéré comme la reference.
- ❑ Il est partagé et disponible pour l'ensemble de l'équipe projet
- ❑ Il parle une langue commune.
- ❑ Il vit avec le projet . Le PB est crée, vit et évolue tout au long du projet



# Exemple Product Backlog

DESCRIPTION	ESTIMATION DE L'EFFORT	ESTIMATION DE LA VALEUR	PRIORITÉ
ITEM 1 : S'AUTHTENTIFIER	10	MOYEN	1
ITEM 2 : GÉRER LES UTILISATEURS	7	MOYEN	1
ITEM 3 : GÉRER LA FACTURATION	20	FORT	2
ITEM 4 : EXPORTER LES FACTURES	3	FAIBLE	2
ITEM 5 : GÉRER LES RAPPORTS STATISTIQUES	12	FORT	3
ITEM 6 : CONTACTER L'ADMINISTRATEUR	15	MOYEN	3
ITEM 7 : CONSULTER L'ANNUAIRE	4	MOYEN	2

# Cycle de vie Product Backlog

- ❑ **Création.** A lieu lors de l'étape du sprint 0 et permet de constituer un Product Backlog dans lequel les items prioritaires sont détaillées.
- ❑ **Affinage.** Consiste à préciser tout au long du projet les items du backlog. Cette étape est favorisée par les planifications de sprint et les mêlées quotidiennes.
- ❑ **Vidange.** Supprime les items déjà réalisées. Cette étape se déroule en fin de sprint.

# Responsabilité du Product Backlog

- ❑ Le PO est en charge de l'élaboration du Product Backlog. Il est le seul et unique garant du product Backlog.
- ❑ Dans le cadre de la définition du Product Backlog, le PO est le représentant du client et ou des utilisateurs.
- ❑ Il doit relever les attentes et besoins du client, prioriser l'ensemble des demandes en fonction de leur souhaits
- ❑ Le backlog ne dispose pas d'un ordre logique de realisation, mais d'un ordre de priorité de réalisation

# Sprint Backlog

- ❏ Il propose un decoupage par sprint des items

- ❏ Il offre une vision détaillé de l'avancement du sprint

# Caractéristiques Sprint Backlog

- ❑ Le Sprint backlog propose un découpage du projet en sprint

- ❑ Pour chaque sprint, on sélectionne les items du product backlog qu'il est possible de réaliser afin de les ajouter au sprint backlog

# Caractéristiques Sprint Backlog

- Une fois mise en place, le sprint backlog sert de reference sur l'avancement de la realisation

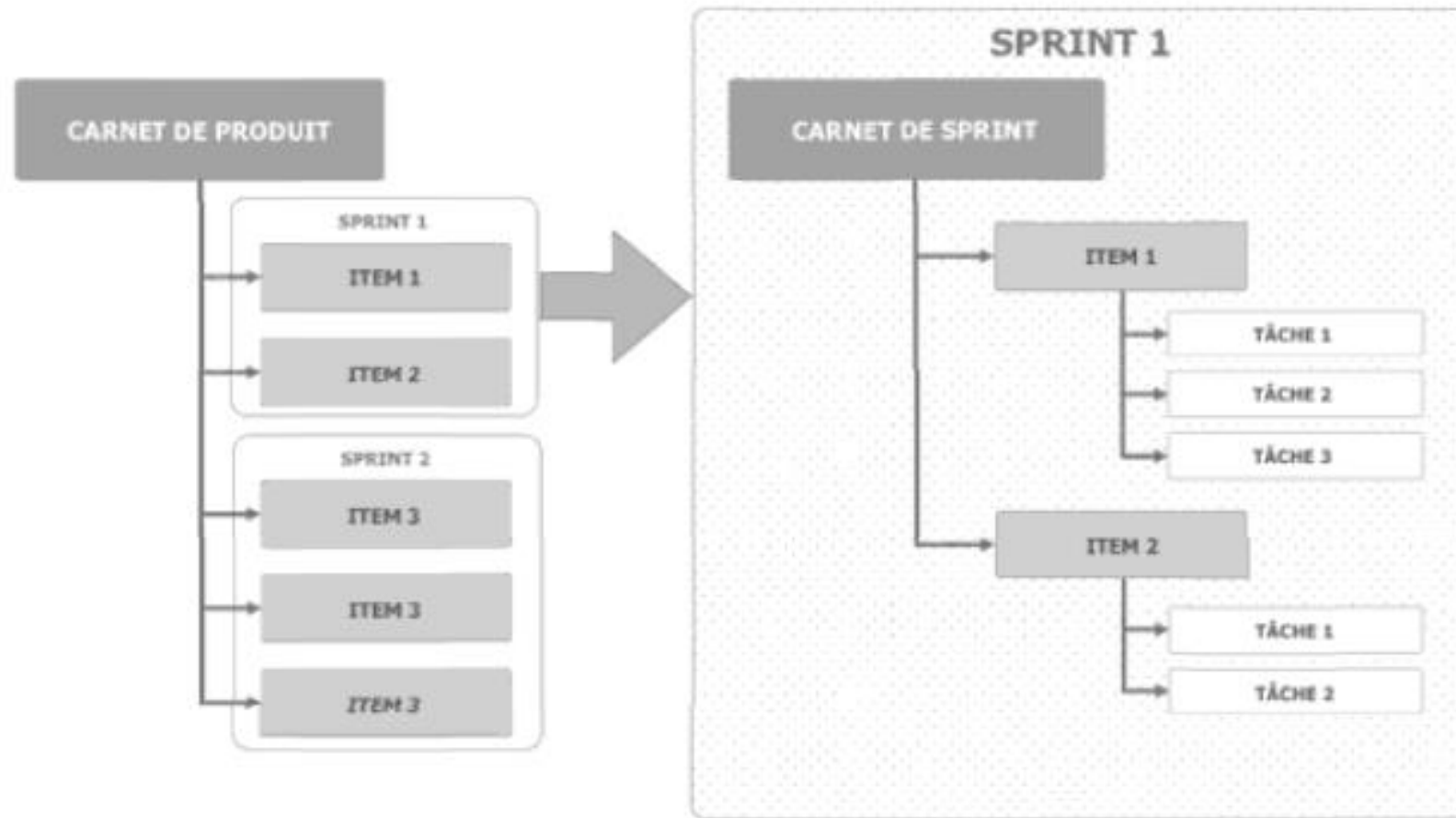
- Pour chaque item, il va proposer :

- un plan de realisation précis qui consiste en un decoupage de l'item en différentes tâches

# Caractéristiques Sprint Backlog

- ❑ Pour chaque item, il va proposer :
  - une mise à jour régulière de l'état de chaque tâche permettant d'être transparent sur l'avancement
- ❑ Comme le product backlog, le sprint backlog est mise à disposition de l'ensemble des acteurs qui ont une visibilité sur ce qui est estimé et réalisé.

# Du Product backlog eu Sprint Backlog





# Du Product backlog eu Sprint Backlog

## **Transformation en cours de développement**

Lorsqu'un élément du backlog est en cours de développement, il faut minimiser le risque de modification. En effet, cela viendrait parasiter l'élan de l'équipe de développement ce qui pourrait nuire au moral des membres de l'équipe.

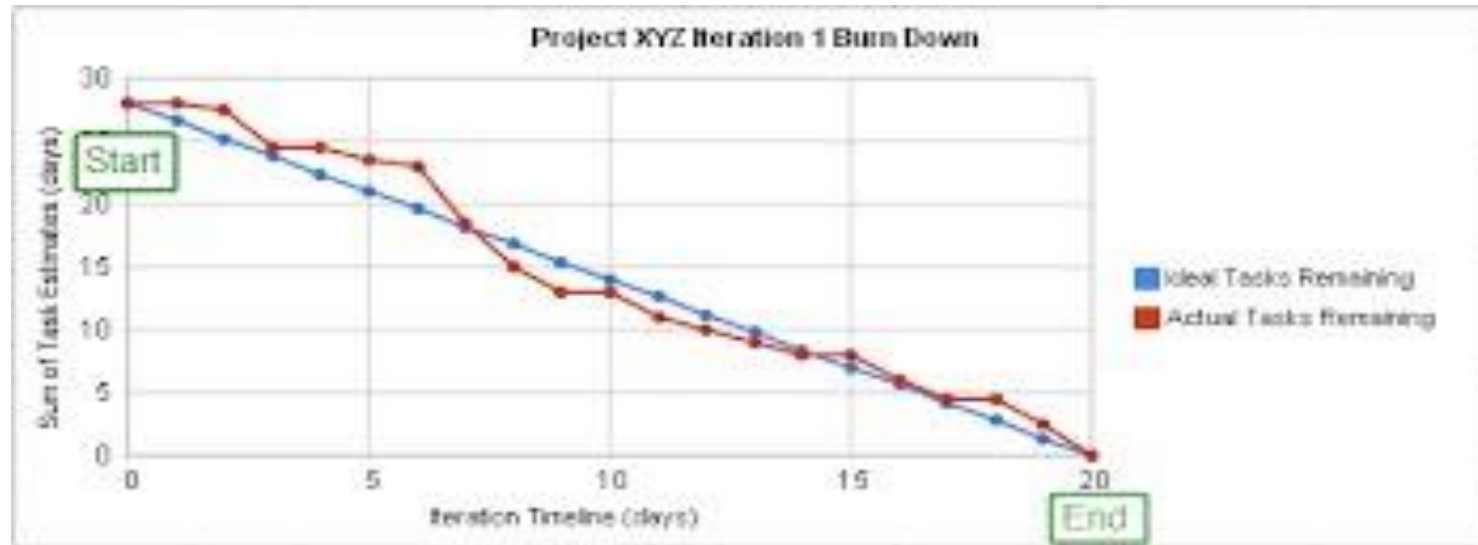
# Cycle de vie Sprint Backlog

- ❑ Le sprint backlog évolue en permanence  
Durant le sprint.
- ❑ Initié durant la phase de planification de  
sprint
- ❑ Puis évolue dans le sprint pendant les  
mêlées quotidiennes
- ❑ La maintenance de cet artefact est à la  
charge de l'équipe de développement.

# Sprint Backlog



# BURN DOWN CHART



# Product Increment

- ❑ Il comprend les items réalisés et terminus lors du sprint
- ❑ Il inclut également l'ensemble des items des précédents incréments
- ❑ Au final il s'agit de l'accumulation d'un ensemble d'item du product backlog qui ont été validés et mise à disposition

# Product Increment

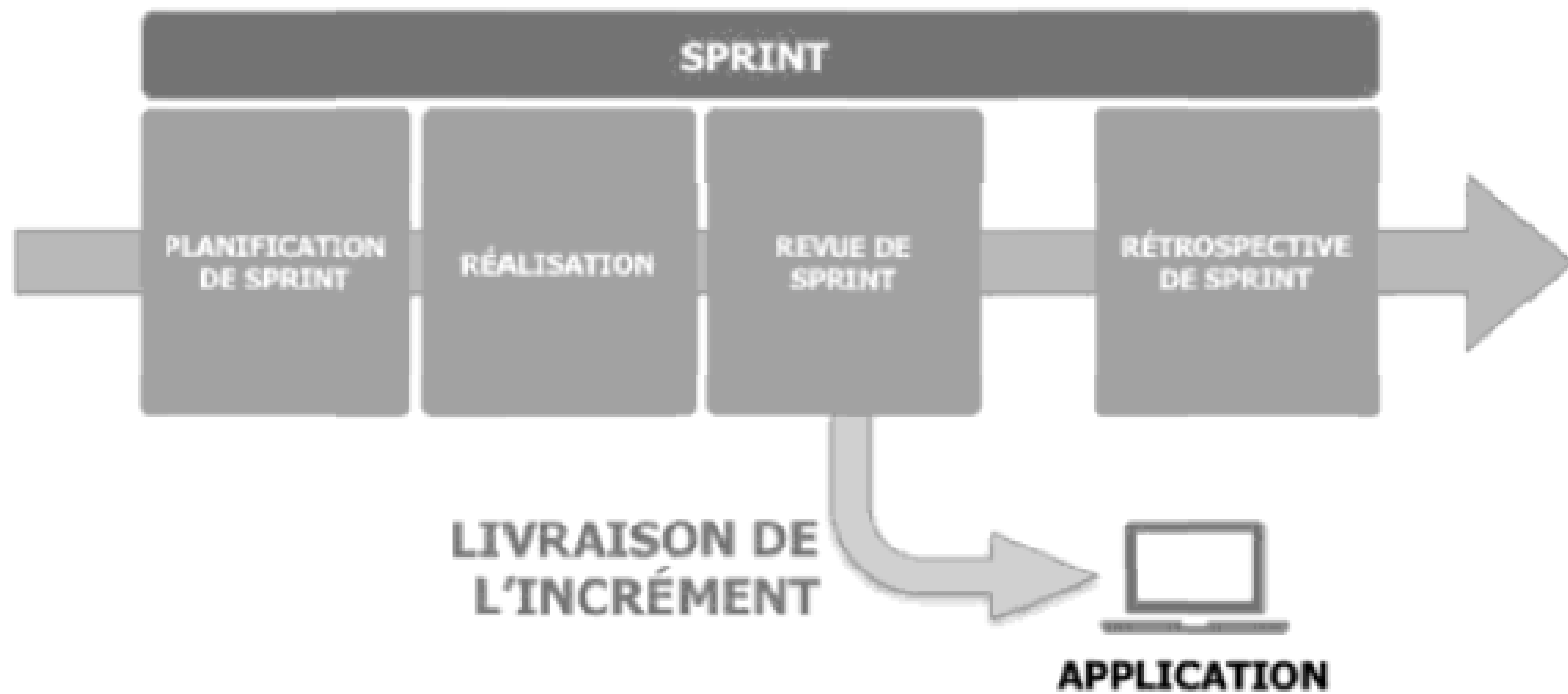
## **La définition de fini**

La « définition de fini » (*Definition of Done* ou DoD) est une notion importante liée au pilier de transparence. Elle implique que tous les acteurs aient la même conception d'une tâche terminée. Cette définition doit donc être claire et comprise par tous les membres de l'équipe.

# Product Increment

- ❑ Il comprend les items réalisés et terminus lors du sprint
- ❑ Il inclut également l'ensemble des items des précédents incréments
- ❑ Au final il s'agit de l'accumulation d'un ensemble d'item du product backlog qui ont été validés et mise à disposition

# Increment Product

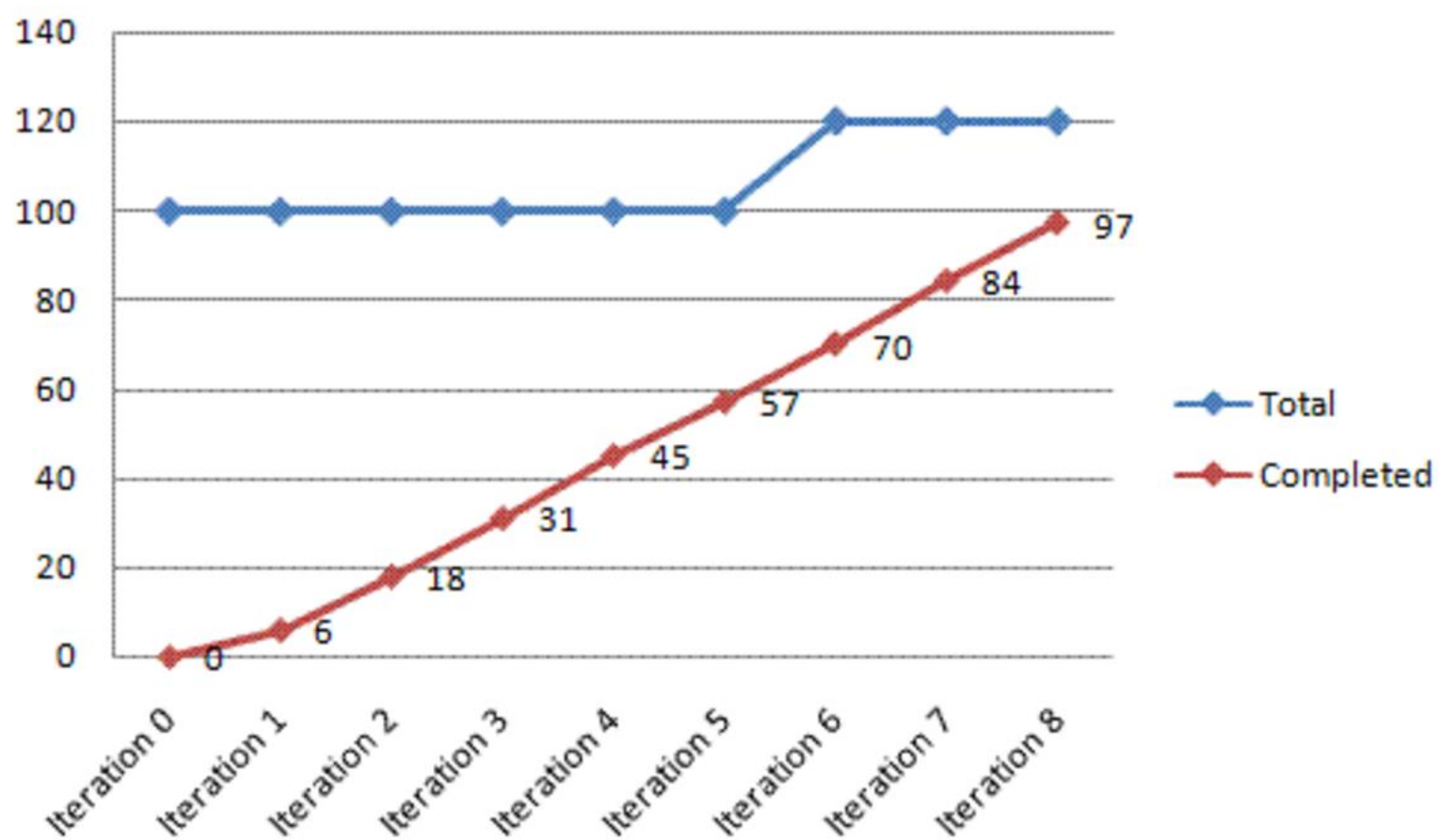




# Realese

- ❑ Il s'agit de la version de l'application
- ❑ Il s'agit d'un ensemble d'items développés au cours de sprint successifs et permettant d'aboutir à une application utilisable dans un environnement réel
- ❑ Au final il s'agit de l'accumulation d'un ensemble d'item du product backlog qui ont été validés et mise à disposition

# Burnup Chart



# CYCLE VIE SCRUM



Sujet 1 : Méthodes classiques Vs Méthodes Agiles

Définition et types des méthodes classiques ? / Différence méthodes classiques et méthodes agiles / Impact des méthodes agiles dans les entreprises (hiérarchie, compétences etc...)

Sujet 2 : Schéma Type Processus Scrum

Sujet 3 : Récapitulatif des différentes phases de processus : 1 – Initialisation (Objectifs, acteurs et artefacts)/ 2 – Préparation du sprint (Objectifs, acteurs, événements et artefacts)/ 3 – Développement Sprint (Objectifs, acteurs, événements et artefact) / 4 Finalisation (Objectifs, acteurs, événements et artefact)