

# UFR des Sciences – Formation MIAGE

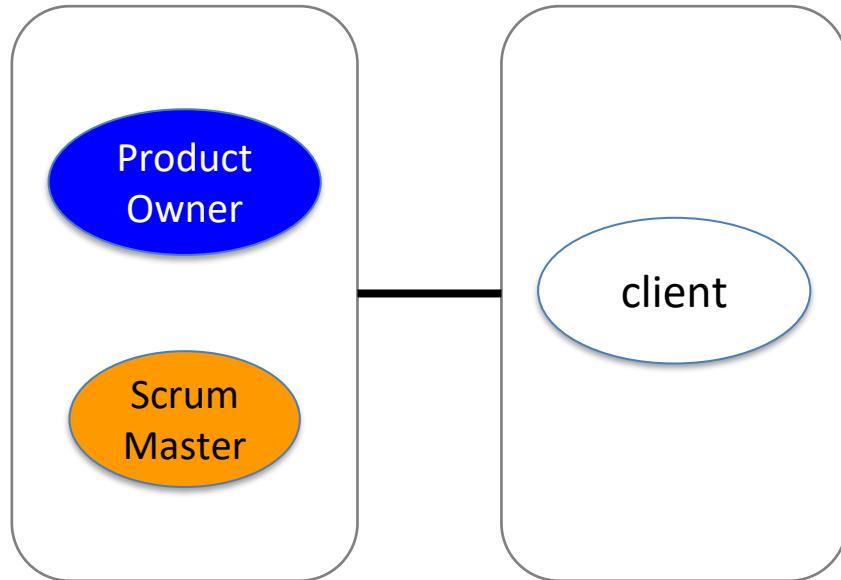
Méthode  
SCRUM



**Conduite de projet agile**  
Les rôles

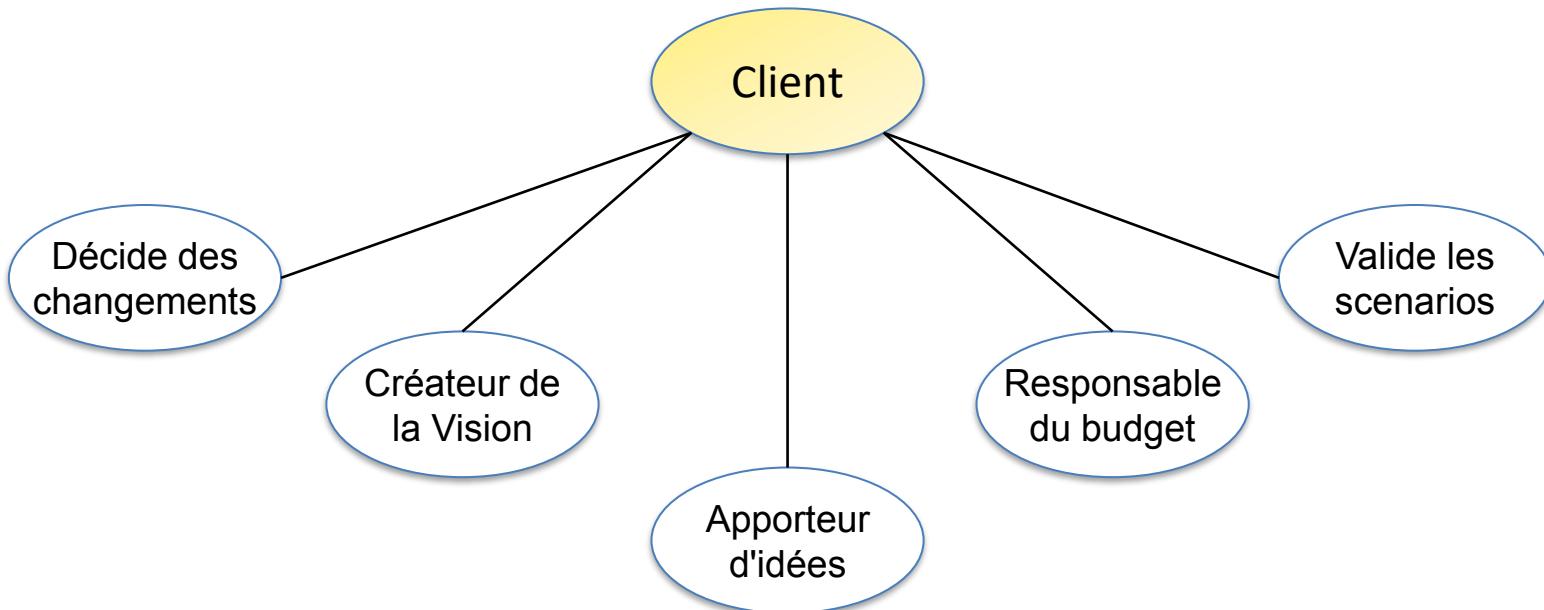


# Les rôles clés

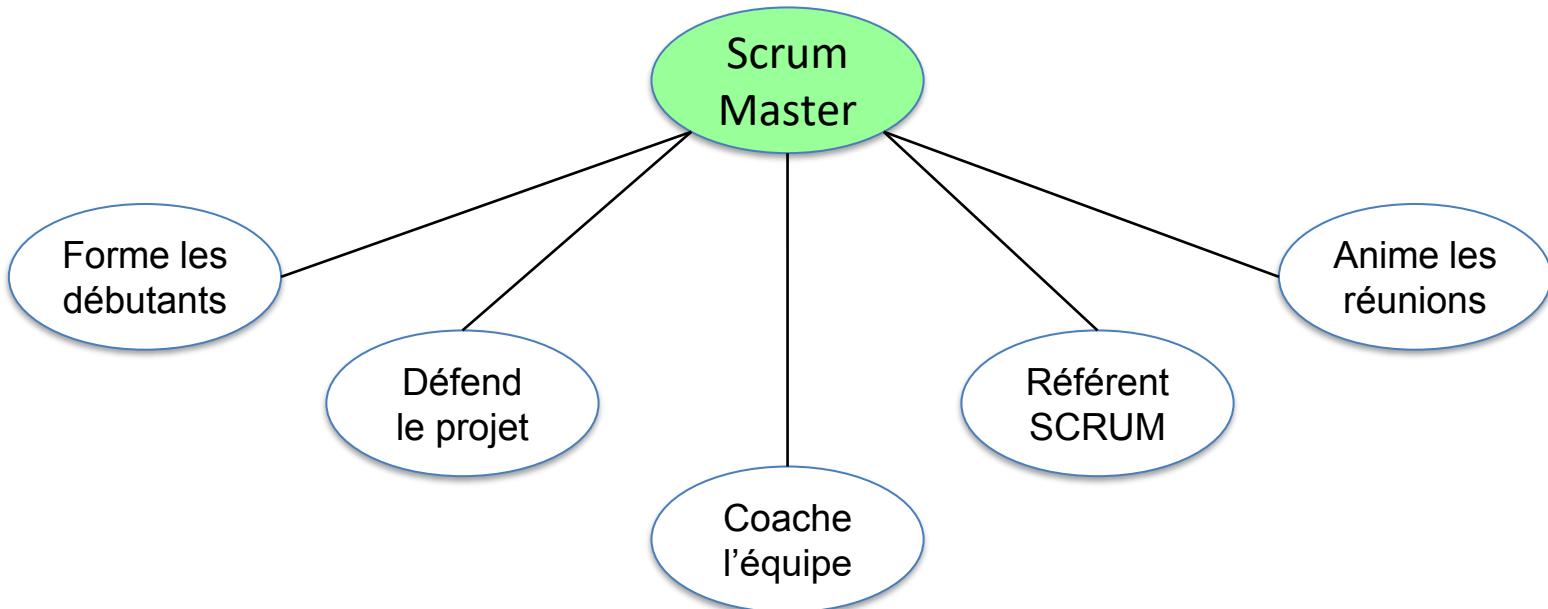


Rappel : le chef de projet n'existe pas pour un projet agile !

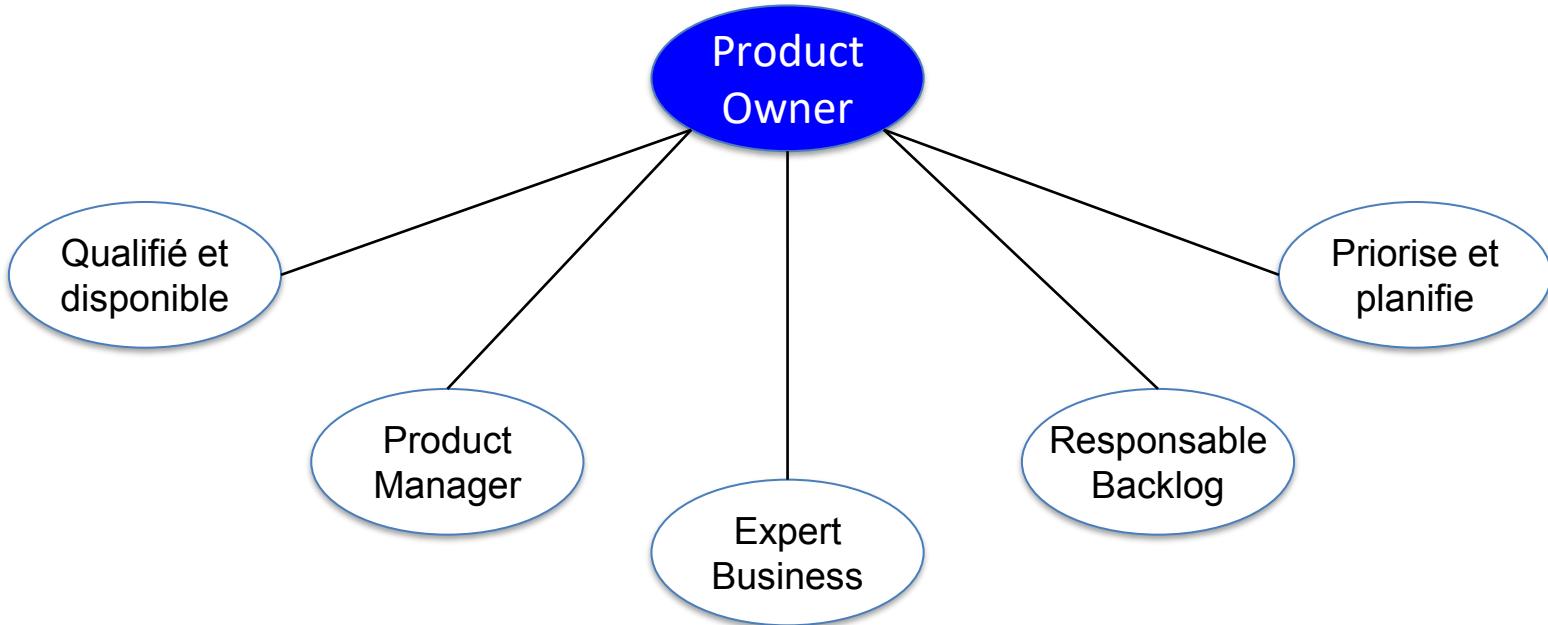
# Le client



# Le Scrum Master



# Le Product Owner



# Le client

---



Ce qui est attendu du client

- des objectifs clairs pour le projet
- des dates pour les évènements clés
- des décisions pour les changements
- une information claire sur le budget
- une terminaison annoncée du projet

# Le client

- des objectifs clairs pour le projet
- des dates pour les évènements clés
- des décisions pour les changements
- une information claire sur le budget
- une terminaison annoncée du projet

# Comment

- ➔ décide des objectifs du MVP
- ➔ décide de la priorité des prochains objectifs
  
- ➔ décide des dates cibles importantes
- ➔ demande une planification à date fixe
  
- ➔ analyse le marché et la concurrence
- ➔ anticipe et propose les changements
  
- ➔ demande une estimation des travaux
- ➔ valide le lancement des travaux
  
- ➔ décide les objectifs qui ne seront pas traités
- ➔ décide la date de fin du projet agile

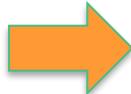
# Pour qui



Product  
Owner

# Le Scrum Master

---



## Ce qui est attendu du Scrum Master

- la défense des valeurs agiles
- une maîtrise des cérémonies agiles
- une évangélisation de la communauté agile
- un coach ressource pour l'équipe agile
- un soutien pendant les situations de crise

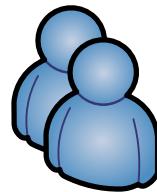
# Le Scrum Master

- la défense des valeurs agiles
- une évangélisation de la communauté agile
- un soutien pendant les situations de crise
- une maîtrise des cérémonies agiles
- un coach ressource pour l'équipe agile

# Comment

- ➔ agit localement en tant que coach du projet agile
- ➔ montre les bénéfices de l'agilité
- ➔ explique les clés du fonctionnement agile
- ➔ propose des actions d'acculturation à l'agilité
- ➔ met en œuvre des réunions de type Rétrospective
- ➔ soutient une démarche d'amélioration continue
- ➔ anime les réunions agiles
- ➔ propose des serious games
- ➔ encourage le développement personnel
- ➔ propose des espaces temps d'approfondissement

# Pour qui



Communauté agile

# Le Product Owner

---



## Ce qui est attendu du Product Owner

- une vision partagée avec la communauté
- des user stories pour le Product Backlog
- des propositions pour les prochains sprints
- des critères du Done pour les user stories
- une animation efficace des groupes de travail
- une communauté de sachants mobilisée

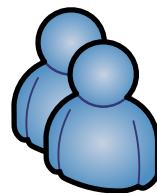
# Le Product Owner

- une vision partagée avec la communauté
- des user stories pour le Product Backlog
- des propositions pour les prochains sprints
- des critères du Done pour les user stories
- une animation efficace des groupes de travail

# Comment

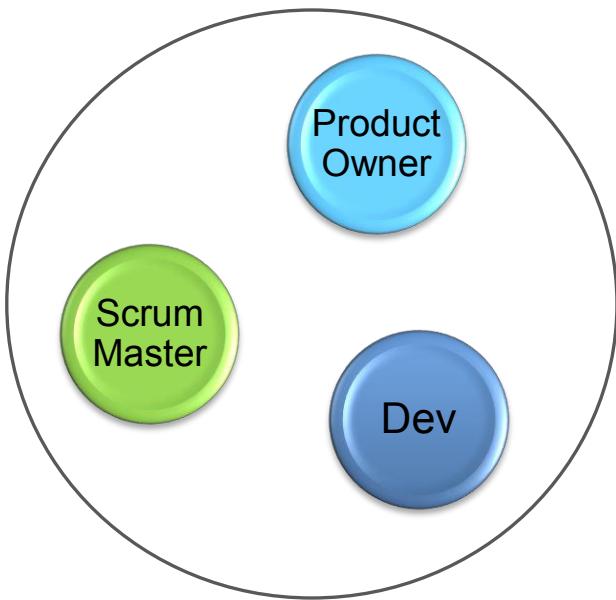
- ➔ donne du sens au projet agile
- ➔ fait adhérer la communauté à la Vision
- ➔ décompose les besoins et les exigences
- ➔ rédige des stories de qualité (mémo INVEST)
- ➔ priorise le backlog en fonction des objectifs
- ➔ sélectionne les stories de la prochaine itération
- ➔ rédige des mini tutoriels utilisateur pour les tests
- ➔ met en œuvre les règles de gestion pour les tests
- ➔ encourage la participation et la prise de parole
- ➔ utilise les Post-It et les serious games

# Pour qui



Client  
Métiers  
Développeurs

# Core Team



# Scrum Master

Le Scrum Master organise le planning de travail des développeurs à l'issue du Sprint Planning.

Le Scrum Master anime la mêlée quotidienne et relaie les questions fonctionnelles au Product Owner.

Il identifie les risques et peut proposer des problématiques d'étude au Product Owner.

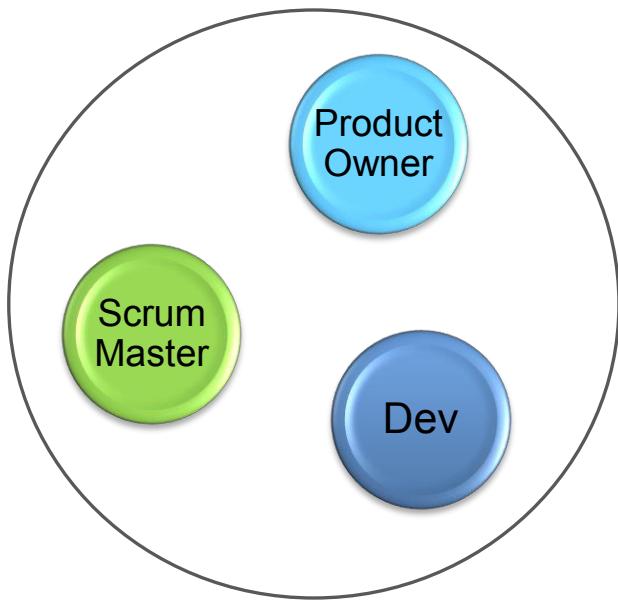
Le Scrum Master est le référent de la méthode SCRUM.

Il protège l'équipe des perturbations extérieures.

Il gère les conflits internes.

Il maximise le focus de l'équipe vers la production de valeur.

# Core Team



# Product Owner

Le Product Owner gère la relation avec le client. Il collecte les besoins auprès des représentants métier et des représentants utilisateur.

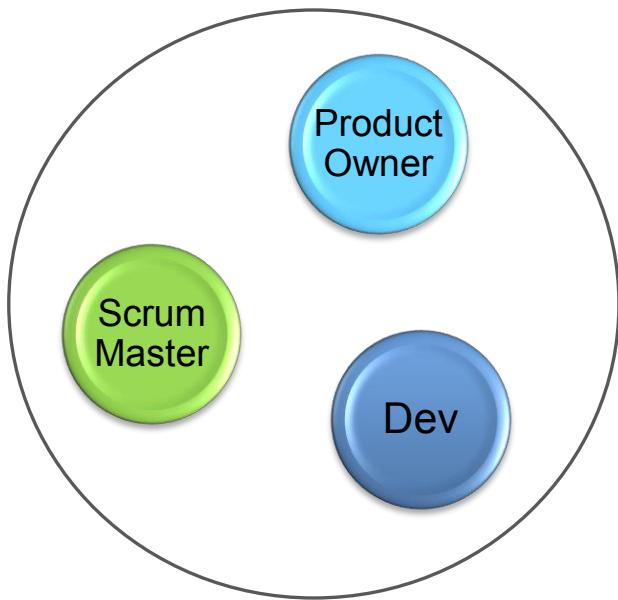
Le Product Owner alimente le Product backlog avec des user stories et des demandes d'étude.

Le Product Owner priorise le Product Backlog en fonction des priorités fonctionnelles du client afin de définir les travaux des prochaines itérations.

Le Product Owner présente les user stories à réaliser aux développeurs lors du Sprint Planning et réponds à leurs questions.

Le Product Owner a la responsabilité du quoi (solution fonctionnelle).

# Core Team



# Développeurs

Les développeurs réalisent les travaux de codage en privilégiant la qualité. Ils rédigent des propositions de solution pour donner suite aux demandes d'étude.

Les développeurs s'engagent à réaliser la livraison d'un Produit de qualité, stable et opérationnel au fil des itérations.

Le projet agile peut s'arrêter à la demande du client à la fin d'une itération.

Les développeurs sont attentifs aux indicateurs de mesure de la performance (\*) afin de satisfaire le client et les utilisateurs.

Les développeurs ont la responsabilité du comment (solution technique).

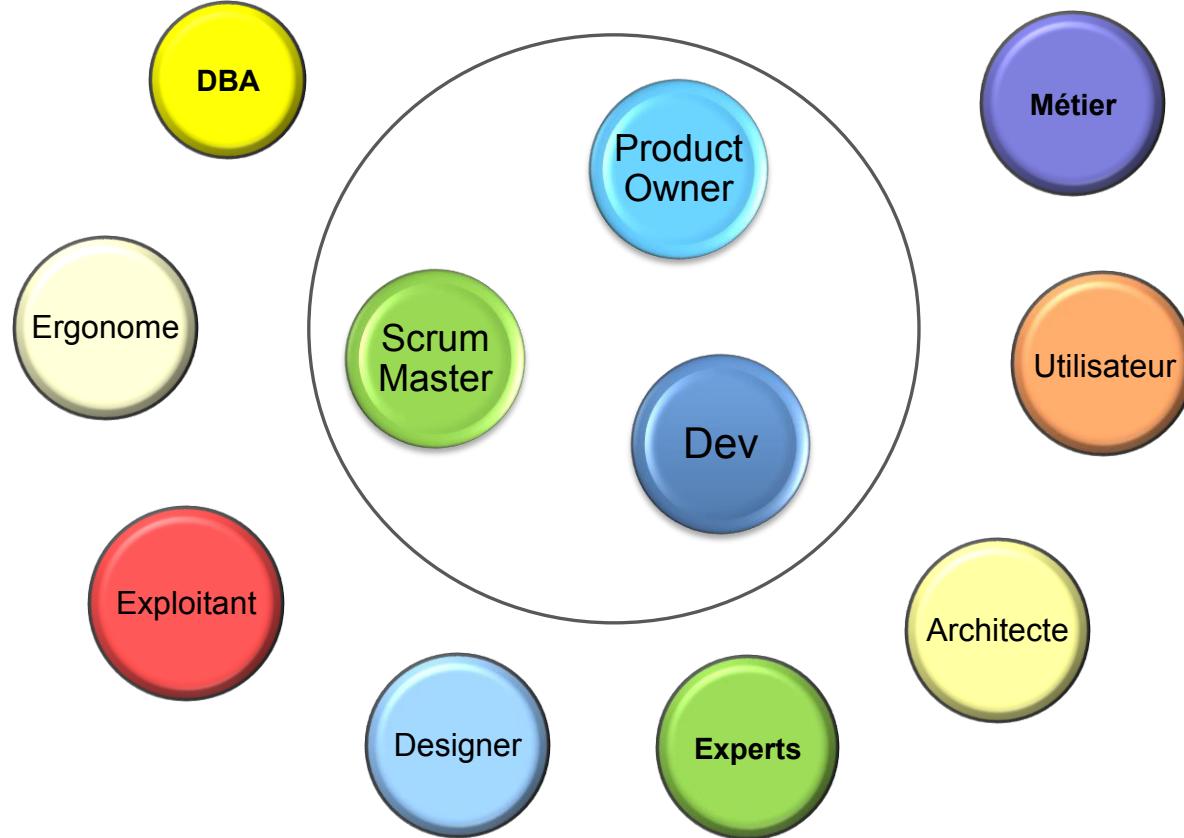
- (\*) Les indicateurs de performance sont :
- Mesure de la performance du Done
  - Mesure des coûts de non-qualité

# Communauté agile élargie

---

Des contributeurs externes à la Core Team participent aux travaux du projet.

Leur contribution est ponctuelle et s'effectue en fonction des besoins du projet.



# Ce qu'il faut retenir

---

Transformation des organisations et des groupes de travail

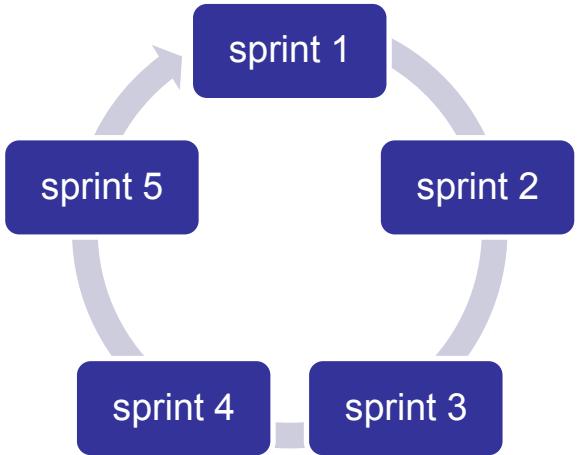
Entreprise apprenante

Knowledge Management

- introduction progressive des méthodes agiles dans les organisations
- remplacement des équipes managées par des équipes agiles
- remplacement progressif du modèle hiérarchique
- redistribution des activités, en particulier pour les managers d'équipe
  
- introduction des nouveaux rôles : Scrum Master et Product Owner
- formation continue des collaborateurs
- acquisition de nouvelles compétences
- création de communauté de pratiques
  
- introduction de la problématique de la conservation des savoir-faire
- exploration et sauvegarde des informations tacites et non explicites
- rôle du Product Owner dans la transformation de l'information
- mise à jour et sauvegarde des connaissances métier et des savoir-faire
- standardisation et transmission des connaissances aux collaborateurs

# Méthodes agiles

Merci



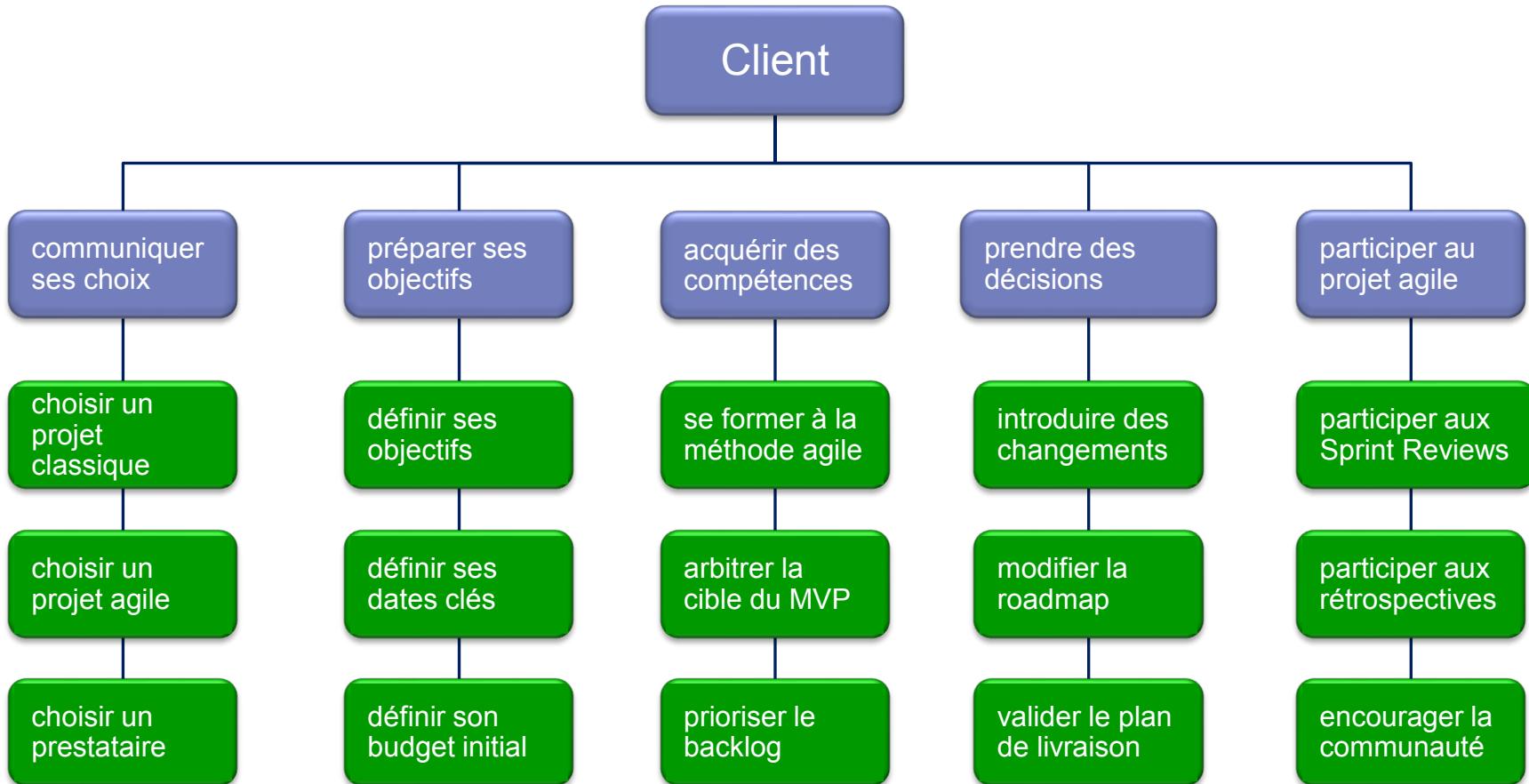
# Méthodes agiles

Méthode  
SCRUM



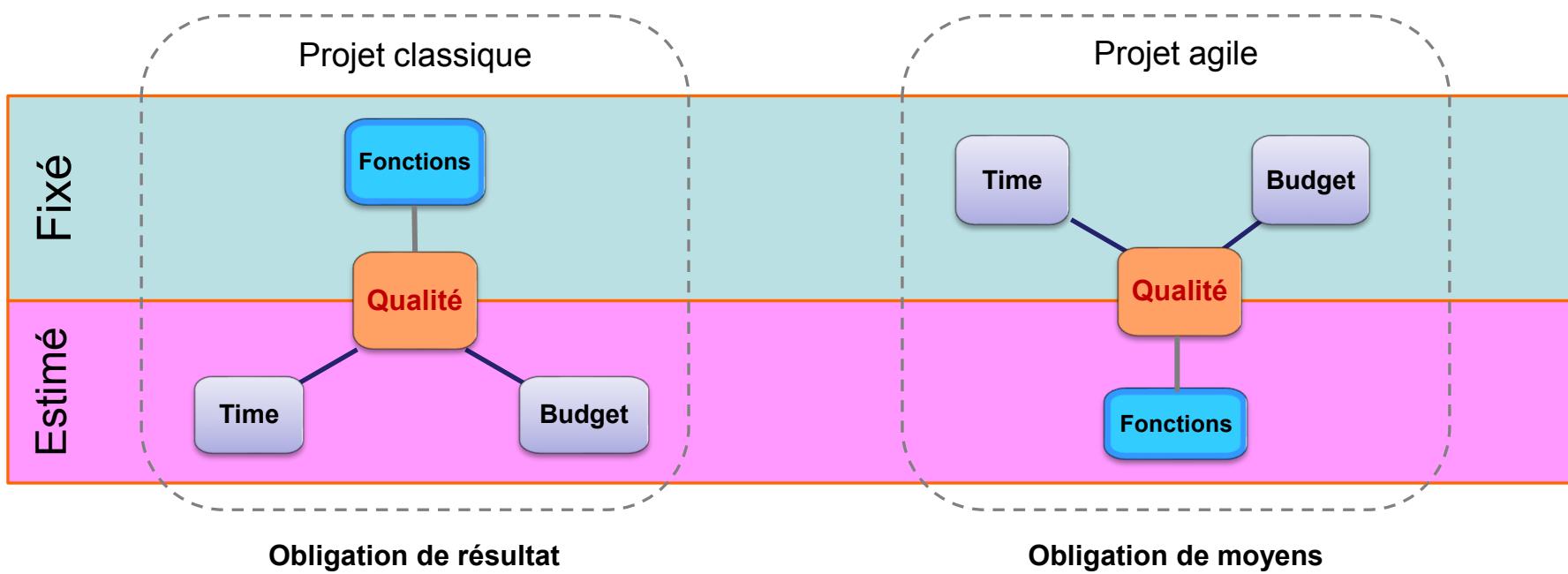
**Conduite de projet agile**  
Collaboration avec le client





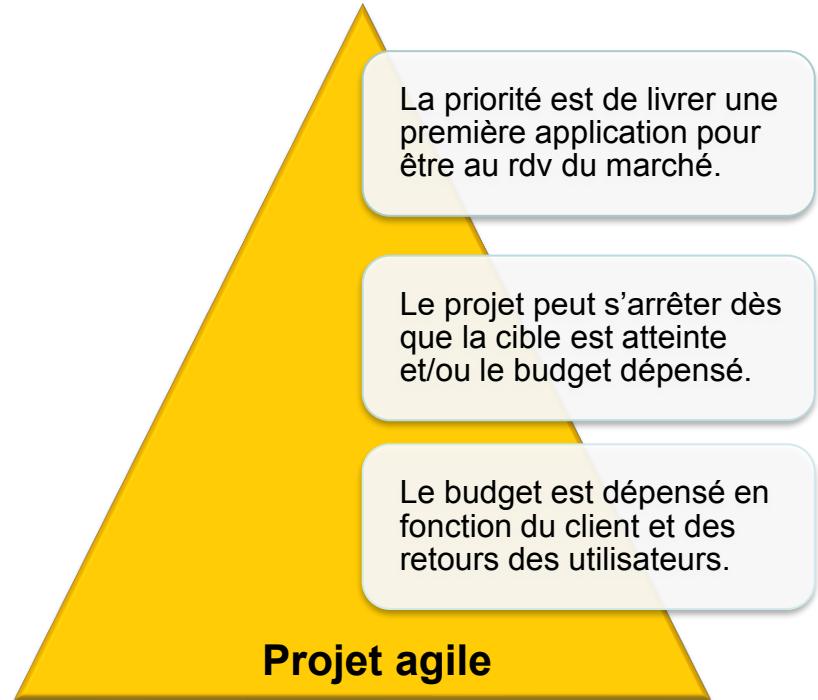
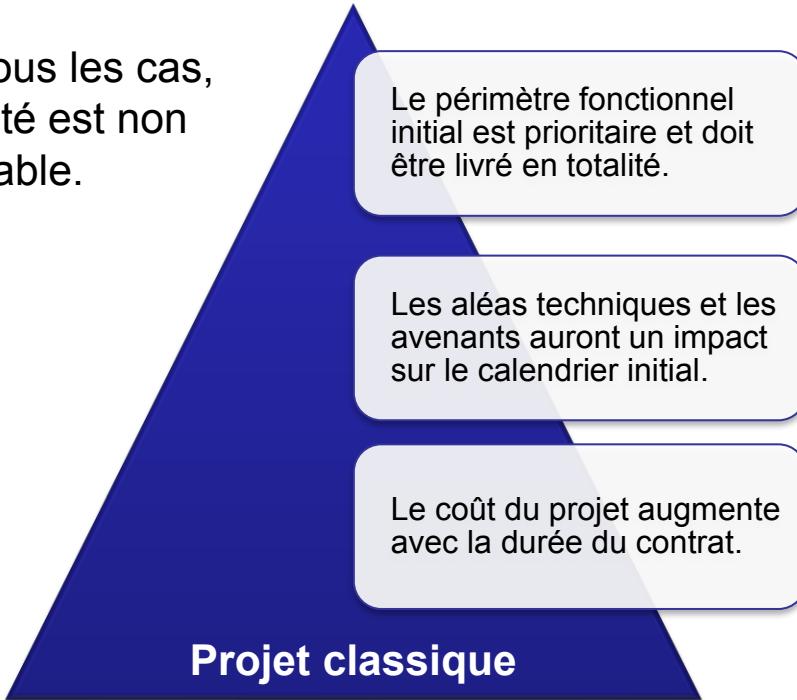
# La collaboration avec le client

Comment aider le client à choisir un projet classique ou un projet agile ?



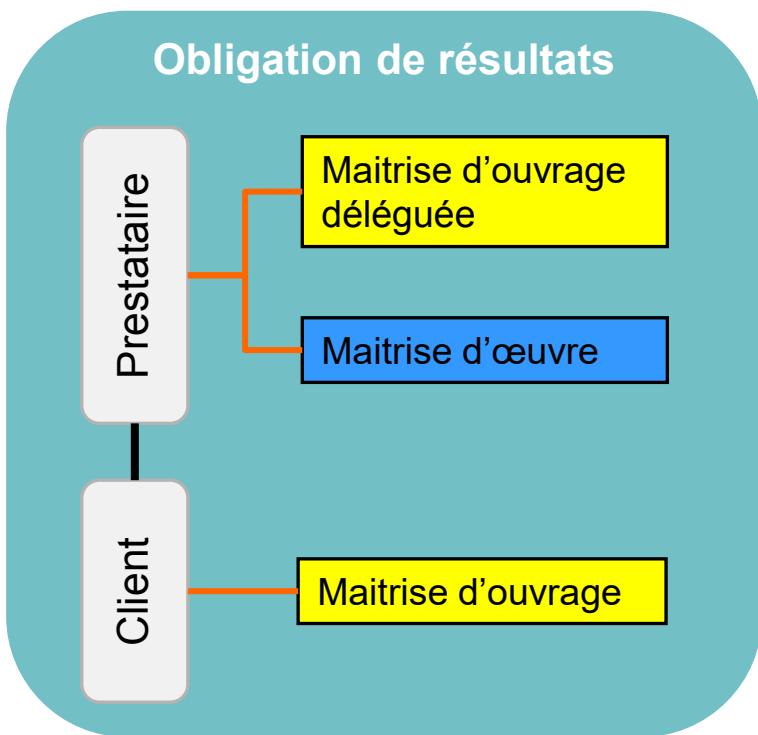
# La collaboration avec le client

Dans tous les cas,  
la qualité est non  
négociable.

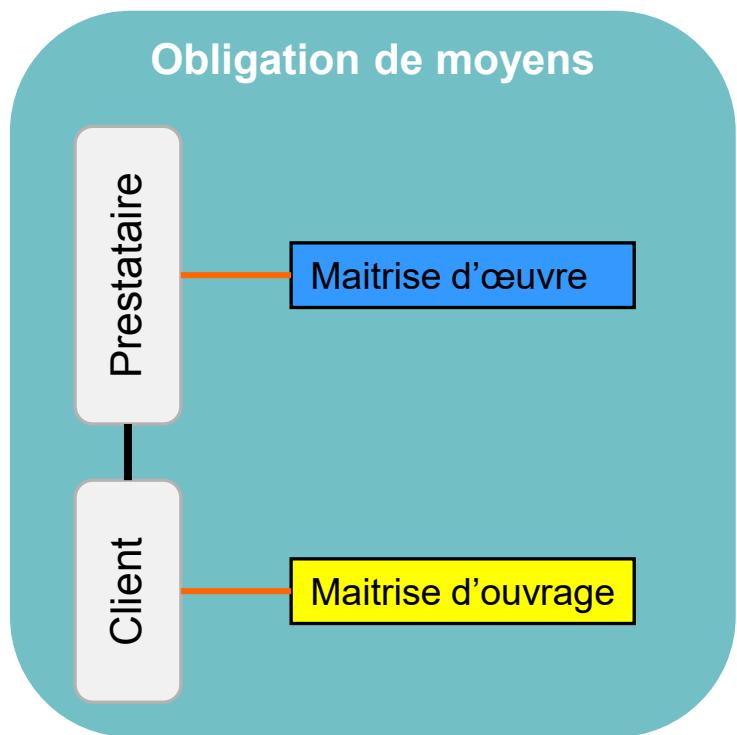


# La collaboration avec le client

## Conduite de projet classique



## Conduite de projet agile



# Projet classique avec obligation de résultat

---

■ Les parties prenantes sont liées par un contrat qui fixe les résultats et les objectifs à atteindre :

- Le contrat inclut la livraison d'une description fonctionnelle du système basé sur le cahier des charges du client.
- Le cahier des charges inclut les exigences en termes d'architecture, d'interfaces, de sécurité et de performance.
- Le contrat fixe un échéancier de livraison et de paiement en fonction de la réalisation des étapes du projet.

■ La contribution du client est la suivante :

- Le client valide les spécifications générales du système rédigées par le prestataire.
- Le client participe au Comité de direction.
- Le client désigne son représentant qui participera au Comité de pilotage.
- Le client et/ou le représentant du client participe aux tests de recette du système.
- Le client valide le PV de recette du système.

■ Le contrat décrit les règles de fonctionnement :

- Le Comité de direction permet au projet de prendre des décisions.
- Le Comité de pilotage permet de réaliser le suivi mensuel du projet.
- Le contrat inclut un Plan Qualité (avec les règles à respecter).
- Le contrat inclut un Plan de production (avec la liste des livrables attendus).
- Le contrat inclut une description des pénalités (retard de livraison, livrable absent, etc.).

# Projet agile avec obligation de moyens

---

■ Les parties prenantes sont liées par un contrat qui fixe les obligations du client et du prestataire :

- Le client a la responsabilité de définir les besoins nécessaires au projet en termes de ressources.
- Le prestataire a l'obligation de fournir les ressources nécessaires en termes de développeurs et de plateformes.
- Le client a la responsabilité de l'emploi de ces moyens.

■ La contribution du client est la suivante :

- Le client désigne un Product Owner et un Scrum Master pour ce projet agile.
- Le prestataire peut allouer un Product Owner et/ou un Scrum Master à la demande du client.
- Le client a la responsabilité de prendre les décisions permettant d'engager les travaux à réaliser.
- Les travaux sont réalisés à concurrence de la capacité à faire de l'équipe de réalisation (pas de pénalités).

■ Le client dispose des prérogatives suivantes :

- Le client peut introduire de nouveaux besoins dans la roadmap du Produit.
- Le client peut choisir un scénario de priorisation parmi ceux présentés par le Product Owner.
- Le client participe aux réunions de validation de fin d'itération (Sprint Review).
- Le client participe aux réunions d'amélioration continue (Rétrospective).
- Le client valide et/ou modifie le plan de livraison présenté par le Product Owner.
- Le client peut allouer un budget supplémentaire au budget initial.

# Décision du client pour un projet classique

---

L'agilité est-elle finalement le bon choix pour le client ?



Le client peut ne pas être volontaire pour s'engager dans un projet agile car un tel projet va requérir une participation active de sa part.

En effet, une conduite de projet classique sera plus confortable pour le client :

- l'engagement avec l'obligation de résultat est supporté par le prestataire
- le client participe essentiellement au début et à la fin du projet
- le client est peu sollicité pendant la phase de réalisation qui demeure sous la responsabilité du prestataire



Attention le client ne peut pas ou très difficilement modifier son cahier des charges initial.

En effet, l'introduction de changements remet en cause l'engagement initial du prestataire et cela n'est pas prévu au contrat. La gestion d'avenants au contrat initial est toujours possible mais elle sera une source de complications et elle sera coûteuse pour le client.

# Décision du client pour un projet agile

---

L'agilité est-elle finalement le bon choix pour le client ?



Une conduite de projet agile requiert une participation active du client.

L'introduction de changements sera possible par le client car il pourra modifier la trajectoire fonctionnelle de son projet en modifiant la priorisation du Product backlog.



Le client reste maître de son budget et il est le seul responsable de son emploi.

Le client communique les dates clés de son projet.

Le client fixe les livraisons fonctionnelles attendues.

Le client peut annoncer l'arrêt du projet à la fin d'une prochaine itération.

Le client peut décider d'allouer un budget supplémentaire à son projet.

# Ambitions du projet agile

---

Placer le client au cœur du projet Agile

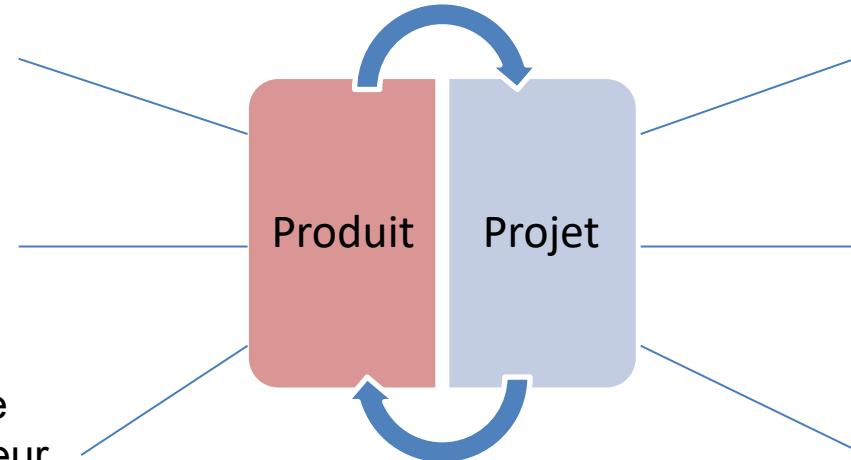
Intégrer le feedback utilisateur

Développer l'expertise en expérience utilisateur de la communauté Agile

Dialoguer efficacement avec les représentants métier

Mettre le client en situation de décider

Maximiser la satisfaction des utilisateurs



# Défis du projet agile

---

Réussir la collecte des besoins

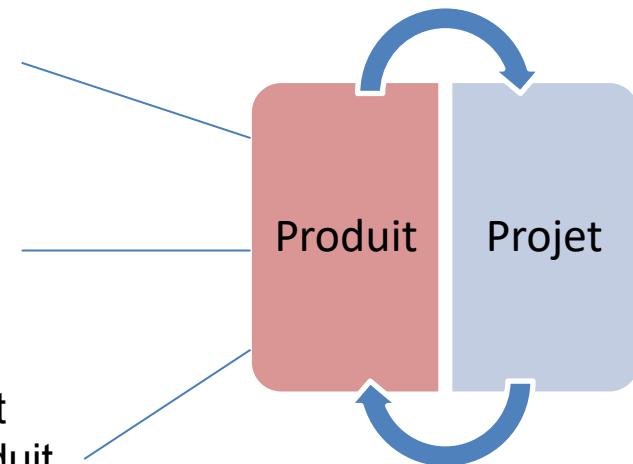
Identifier les rôles du Produit

Décomposer finement les exigences du Produit

Prioriser les versions du Produit au bénéfice des utilisateurs

Accepter les embellissements

Arrêter proprement le projet quand le client le décide



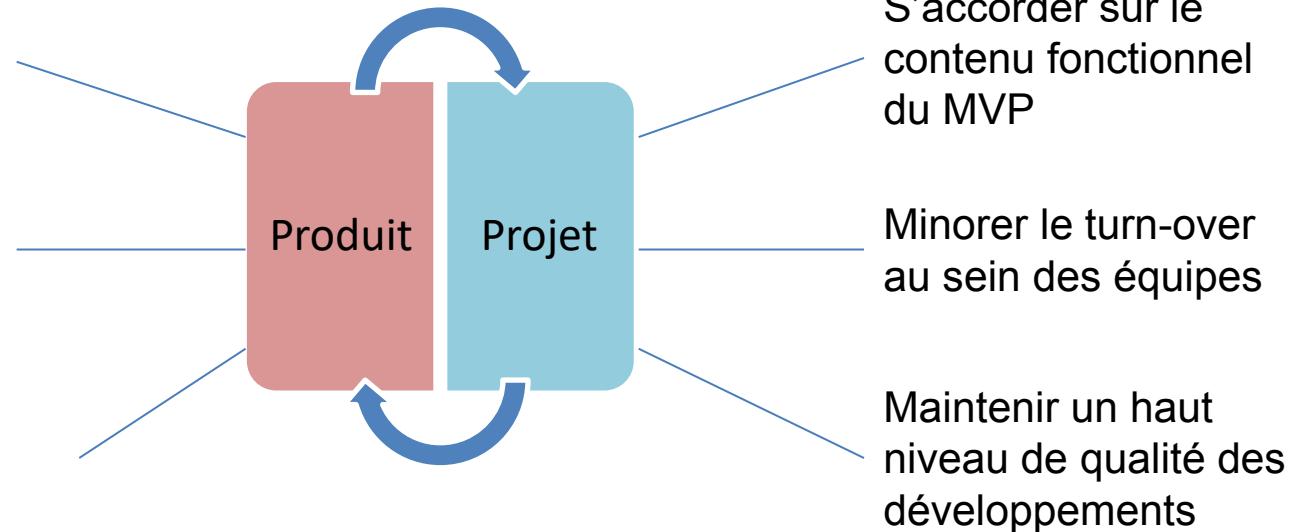
# Défis du projet agile

---

Conjuguer les versions et le budget

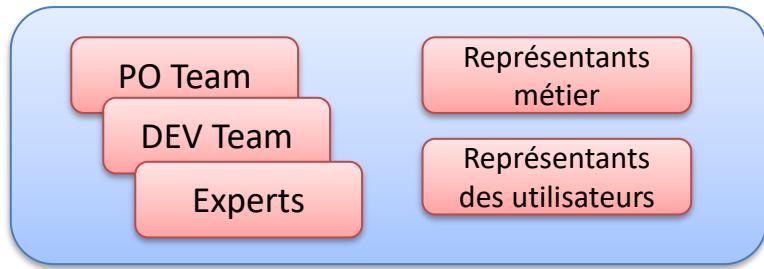
Demander un budget supplémentaire

Vendre le produit aux pragmatiques

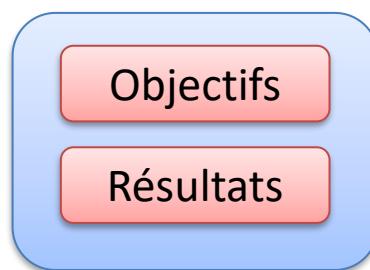


# Vue de synthèse

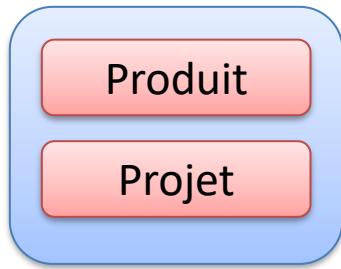
## La Communauté



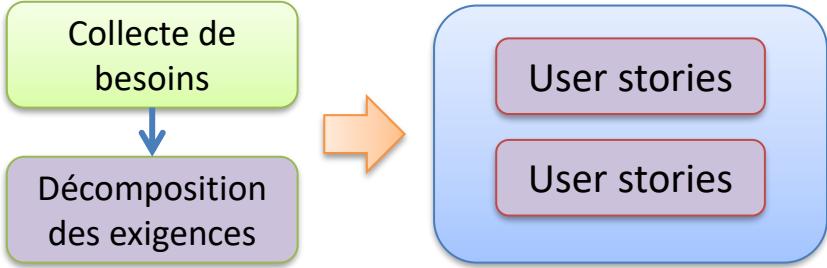
## La Vision



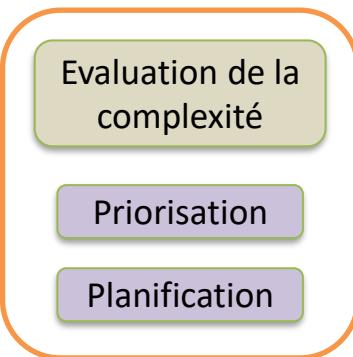
## La Vision



## Le Product backlog

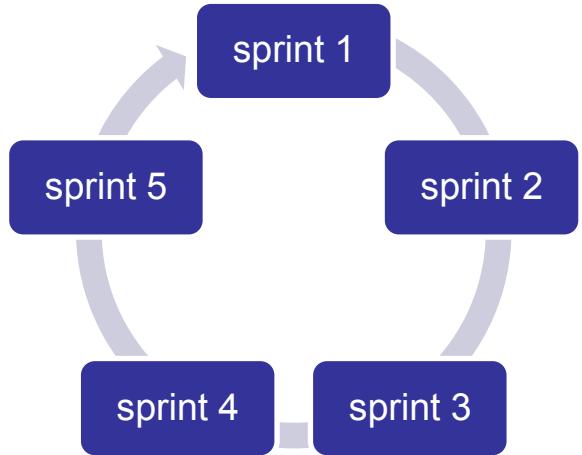


## La Vision



# Méthodes agiles

Merci

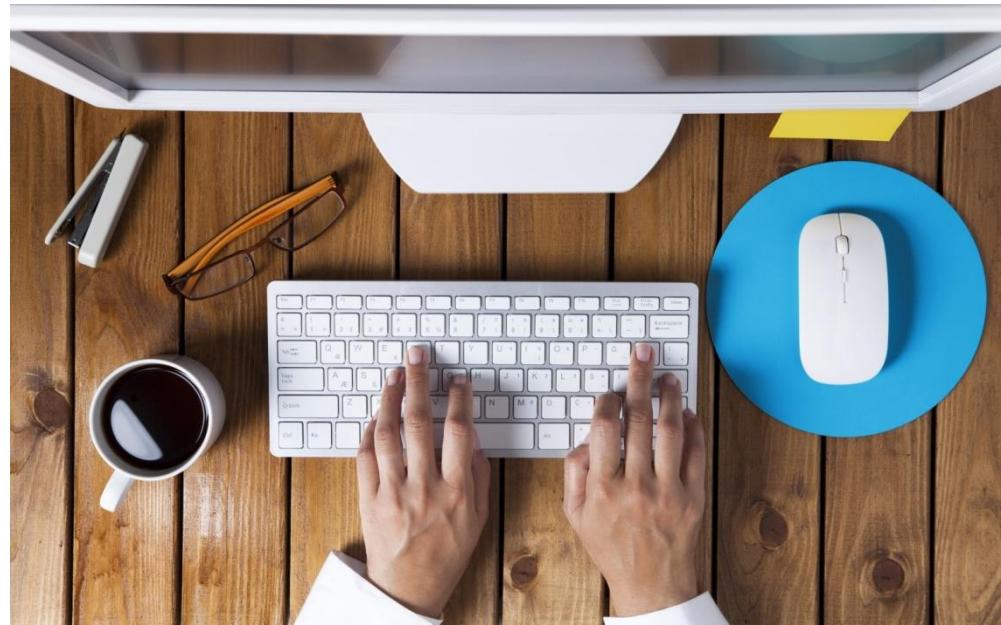


# Méthodes agiles

Méthode  
SCRUM



**Conduite de projet agile**  
La Vision du client



# Stratégie et Vision

---

- La Vision représente la cible fonctionnelle et le but stratégique que le client souhaite atteindre.
- Ces étapes représentent les périmètres fonctionnels qui requièrent des réponses en termes de dates et de calendrier.  
Le Product Owner est en charge des travaux de planification visant à exprimer les dates clés attendues par le client.
- La Vision permet d'aider le client à construire les étapes intermédiaires qui seront nécessaires pour atteindre sa cible fonctionnelle.
- Ces réponses en termes de dates et de calendrier peuvent être demandées au Product Owner si ce dernier dispose d'un Product backlog priorisé et évalué.

## Stratégie et Vision

---

■ La profondeur de la description fonctionnelle du backlog et la qualité de rédaction des stories sont les prérequis pour le calcul de ces réponses.

Ces calculs sont des calculs de planification.

■ L'absence de profondeur du Product Backlog rend difficile, voire impossible la détermination des dates clés du projet agile.

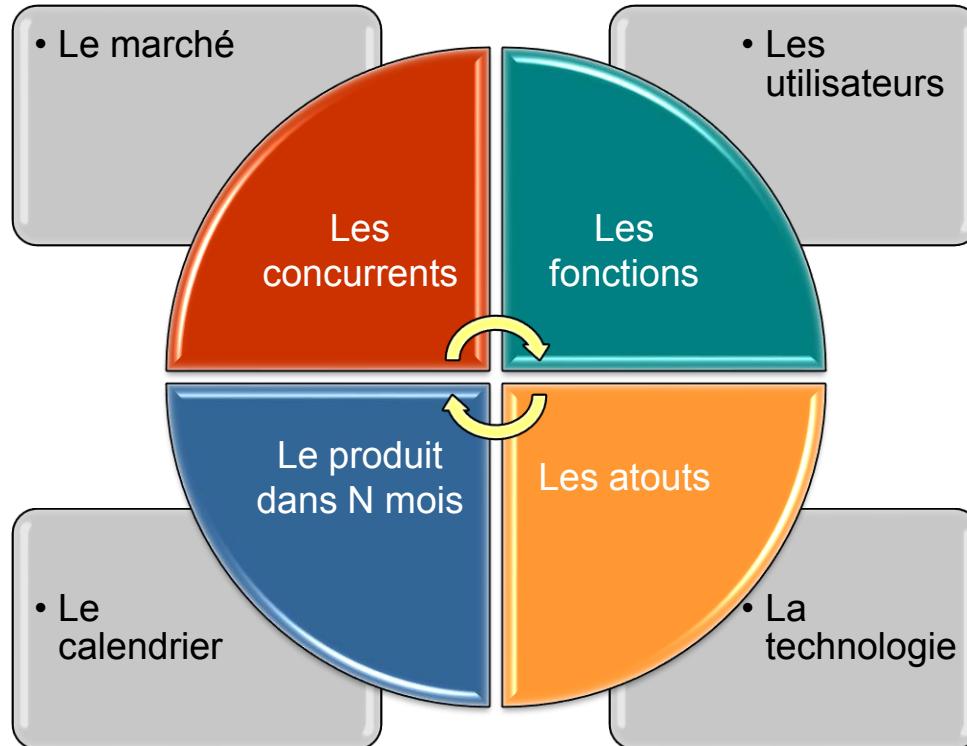
■ La planification est un exercice que le Product Owner doit maîtriser afin de permettre au client de prendre les bonnes décisions.

■ Un Product Backlog sans profondeur fonctionnelle ne donne aucune visibilité et rend aveugle le PO et le client. Seule une approche sur le court terme sera possible sans pouvoir se projeter au-delà.

# Stratégie et Vision

---

## La Vision



# Stratégie et Vision

---

La Vision	Le marché	Les utilisateurs	Le calendrier	La technologie
Aujourd'hui	Quelle est la stratégie du marché ?	Quelle est ma roadmap pour satisfaire les utilisateurs ?	Quelle est ma stratégie de déploiement ?	Comment puis-je tirer profit de la technologie ?



Le Product Owner, les représentants métier et les représentants utilisateurs contribuent à alimenter la Vision du client en complément de sa lecture du marché et de ses concurrents.

# Stratégie et Vision

---

<b>La Vision</b>	<b>Le marché</b>	<b>Les utilisateurs</b>	<b>Le calendrier</b>	<b>La technologie</b>
Aujourd'hui	Quelle est la stratégie du marché ?	Quelle est ma roadmap pour satisfaire les utilisateurs ?	Quelle est ma stratégie de déploiement ?	Comment puis-je tirer profit de la technologie ?
6 mois	Que font mes concurrents ?	Quels sont les premiers feedbacks des utilisateurs ?	Mon premier Rdv est-il adapté ?	Mes choix en termes de technologie ont-ils été confirmés ?

# Stratégie et Vision

---

<b>La Vision</b>	<b>Le marché</b>	<b>Les utilisateurs</b>	<b>Le calendrier</b>	<b>La technologie</b>
Aujourd'hui	Quelle est la stratégie du marché ?	Quelle est ma roadmap pour satisfaire les utilisateurs ?	Quelle est ma stratégie de déploiement ?	Comment puis-je tirer profit de la technologie ?
6 mois	Que font mes concurrents ?	Quels sont les premiers feedbacks des utilisateurs ?	Mon premier Rdv est-il adapté ?	Mes choix en termes de technologie ont-ils été confirmés ?
12 mois	Le marché a-t-il évolué depuis 1 an ?	Le Produit est-il devenu mature pour les utilisateurs ?	Ma stratégie de déploiement est-elle encore adaptée ?	Une innovation est-elle disponible pour mon Produit ?

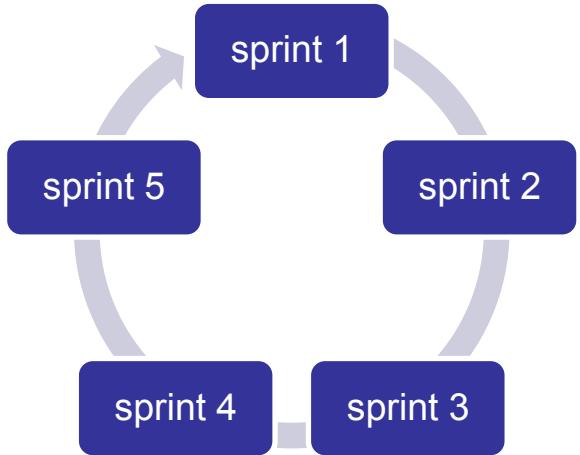
# Stratégie et Vision

---

<b>La Vision</b>	<b>Le marché</b>	<b>Les utilisateurs</b>	<b>Le calendrier</b>	<b>La technologie</b>
Aujourd'hui	Quelle est la stratégie du marché ?	Quelle est ma roadmap pour satisfaire les utilisateurs ?	Quelle est ma stratégie de déploiement ?	Comment puis-je tirer profit de la technologie ?
6 mois	Que font mes concurrents ?	Quels sont les premiers feedbacks des utilisateurs ?	Mon premier Rdv est-il adapté ?	Mes choix en termes de technologie ont-ils été confirmés ?
12 mois	Le marché a-t-il évolué depuis 1 an ?	Le Produit est-il devenu mature pour les utilisateurs ?	Ma stratégie de déploiement est-elle encore adaptée ?	Une innovation est-elle disponible pour mon Produit ?
3 ans	Que sont devenus mes concurrents ?	Les utilisateurs sont-ils passés à autre chose ?	Quelle est ma nouvelle Vision ?	Mon Produit est-il devenu obsolète ?

# Les méthodes agiles

Merci



# Méthodes agiles

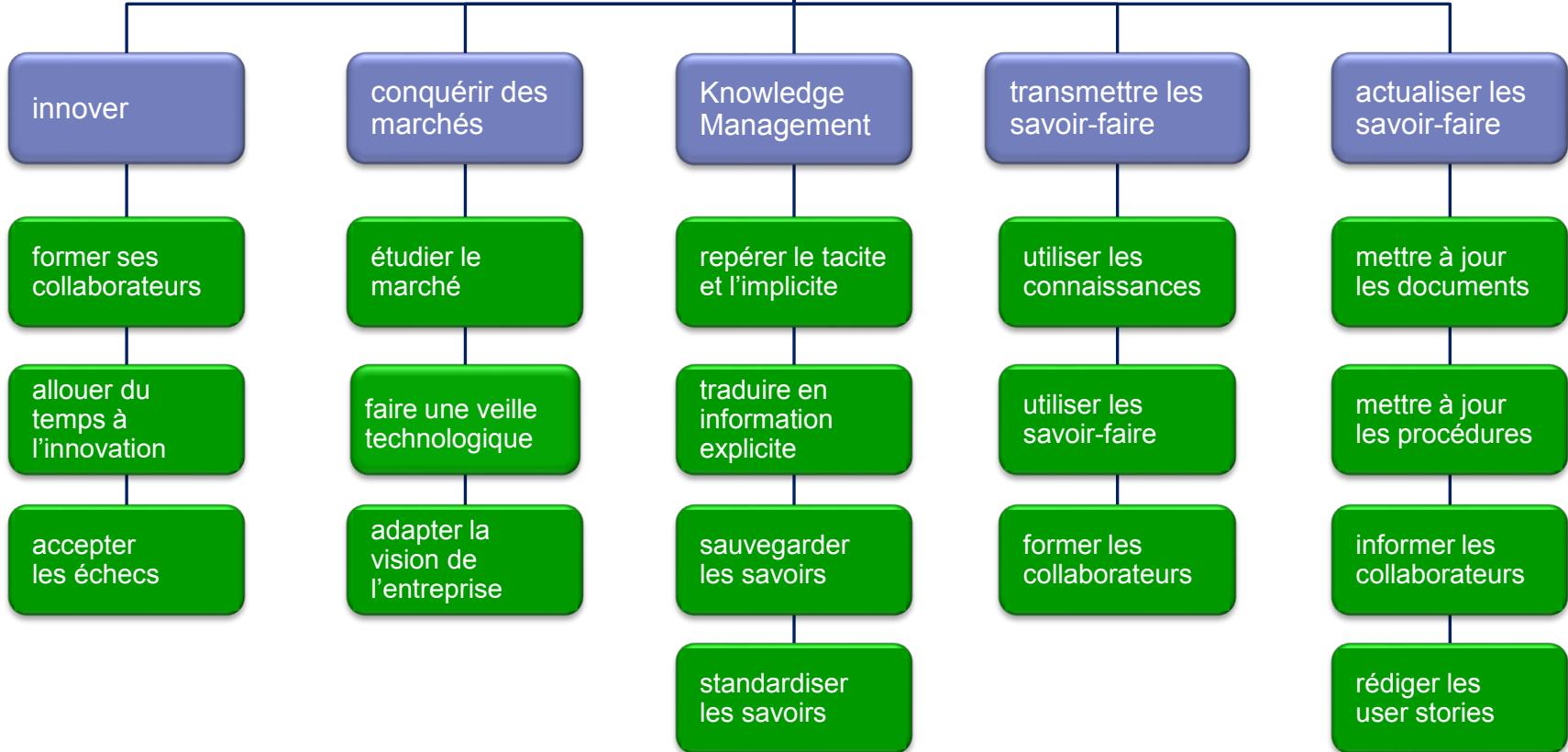
Méthode  
SCRUM



**Conduite de projet Agile**  
Les entreprises



# Entreprise



# Lettre de mission

---

Entreprise

Conquérir de nouveaux marchés

Innover avec de nouveaux produits

Gérer le calendrier des évènements

L'entreprise est confrontée au changement et à la transformation des métiers.

La gestion des connaissances est un objectif stratégique qui requiert le traitement et la transmission des informations de l'organisation.

La gestion des connaissances regroupe l'ensemble des initiatives, des méthodes et des techniques permettant de percevoir, identifier, analyser, organiser, mémoriser, partager les connaissances des membres d'une organisation ainsi que les savoirs créés par l'entreprise elle-même.

# Knowledge Management



Gestion des connaissances



Savoir reconnaître, sauvegarder et faire vivre les connaissances

Apprendre les secrets tacites

Traduire les secrets tacites en savoir explicite

Standardiser ce savoir en procédure ou manuel

Utiliser ce savoir au niveau individuel

# Knowledge Management

---



La bataille des  
mots aura lieu.



Découvrir le tacite et l'implicite  
et explorer ces informations.

1

implicite

3

simplification

2

explicite

4

omission

# Knowledge Management



# Transformation de l'information

# Rôle du Product Owner

Le Product Owner collecte les besoins auprès des métiers et du client.



Le Product Owner accomplit un travail de transformation de l'information.

1

Décomposition  
des besoins

2

Rédaction  
des stories

3

Critères  
du Done

4

Validation  
en groupe



1

## Décomposition des besoins

Existe-t-il encore des besoins à traiter ?  
L'arbre de décomposition des besoins doit alors être modifié. Il doit être maintenu à jour en vie de solution.

2

## Rédaction des stories

Informations explicites et implicites  
Le niveau de connaissance est-il ok ?  
Le sujet est-il maintenant épuisé ?  
Les règles de gestion sont-elles validées ?

3

## Critères du Done

La clé du succès de la Sprint Review : fournir aux développeurs un mini tutoriel utilisateur des instructions d'exécution pas-à-pas de chaque cas de test.

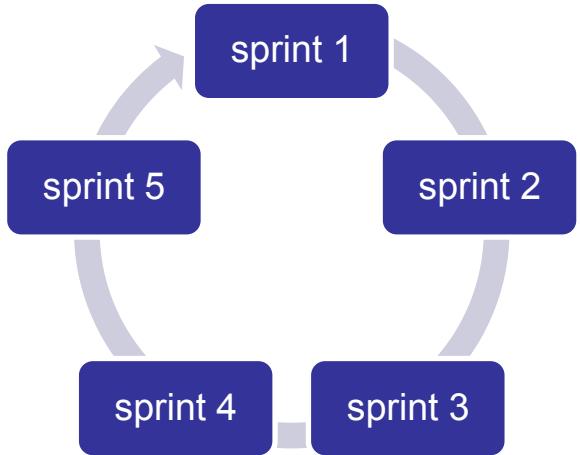
4

## Validation en groupe

La validation des noms, des règles de gestion, des cycles de vie et des scénarios d'usage a-t-elle été acquise auprès des représentants métier et utilisateur ?

# Méthodes agiles

Merci



# Méthodes agiles

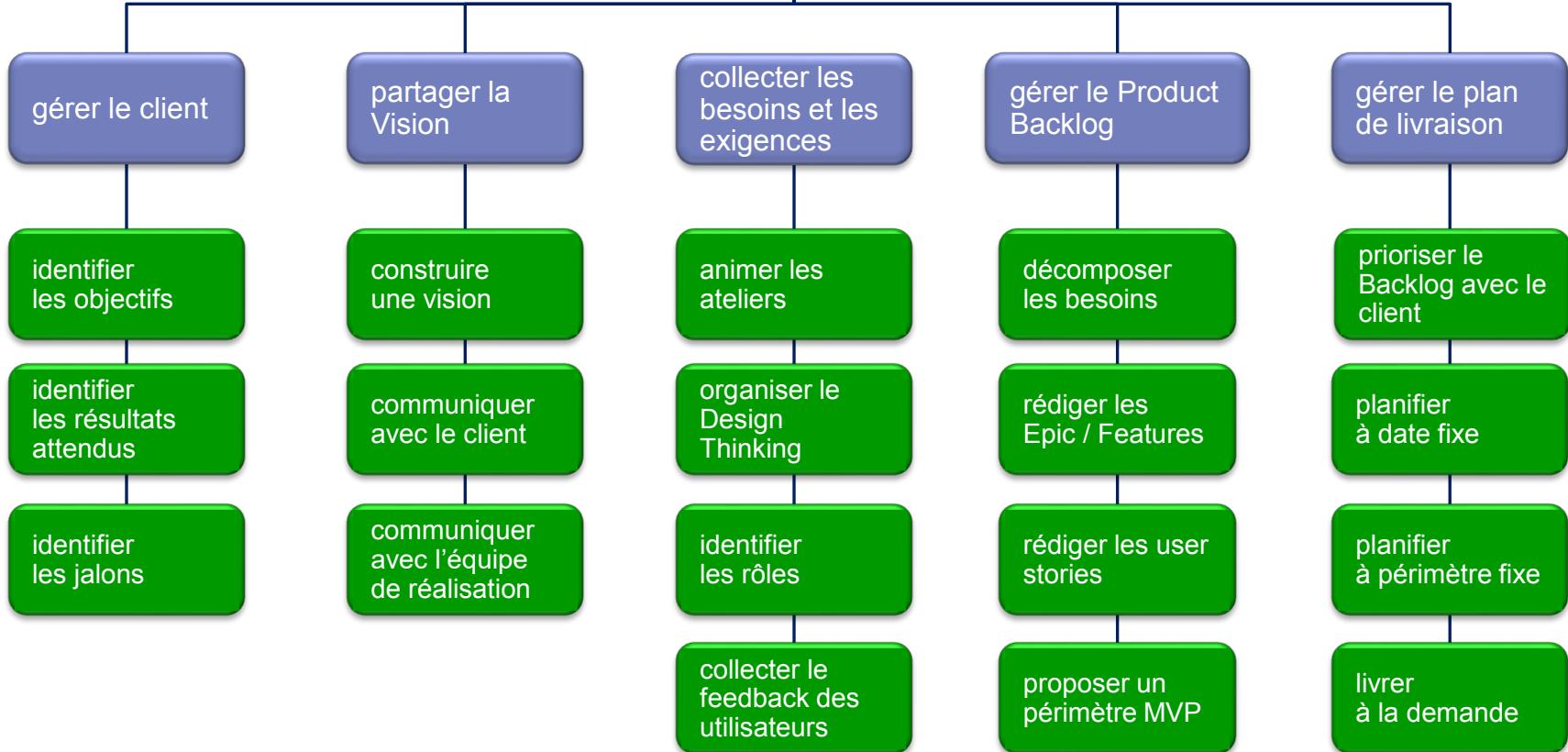
Méthode  
SCRUM



**Conduite de projet agile**  
Product Owner



# Product Owner



# Lettre de mission

(1/2)

---

Product  
Owner

Responsable  
fonctionnel du  
projet agile

Maximise la valeur  
du Produit

Maximise la valeur  
du travail de  
l'équipe DEV

Le Product Owner est une personne.  
Ce n'est pas un comité de direction.

Les décisions du Product Owner doivent  
être respectées. Elles sont visibles grâce  
au contenu et à la priorisation du Product  
backlog.

Toute demande de changement doit être  
adressée au Product Owner.

Personne d'autre que le Product Owner  
n'est légitime pour donner des instructions  
à l'équipe des développeurs.

# Lettre de mission

(2/2)

## Product Owner

Responsable de la collecte des besoins

Responsable du Product Backlog

Interlocuteur unique du client

Interlocuteur unique des développeurs

Le Product Owner gère la relation avec le client.

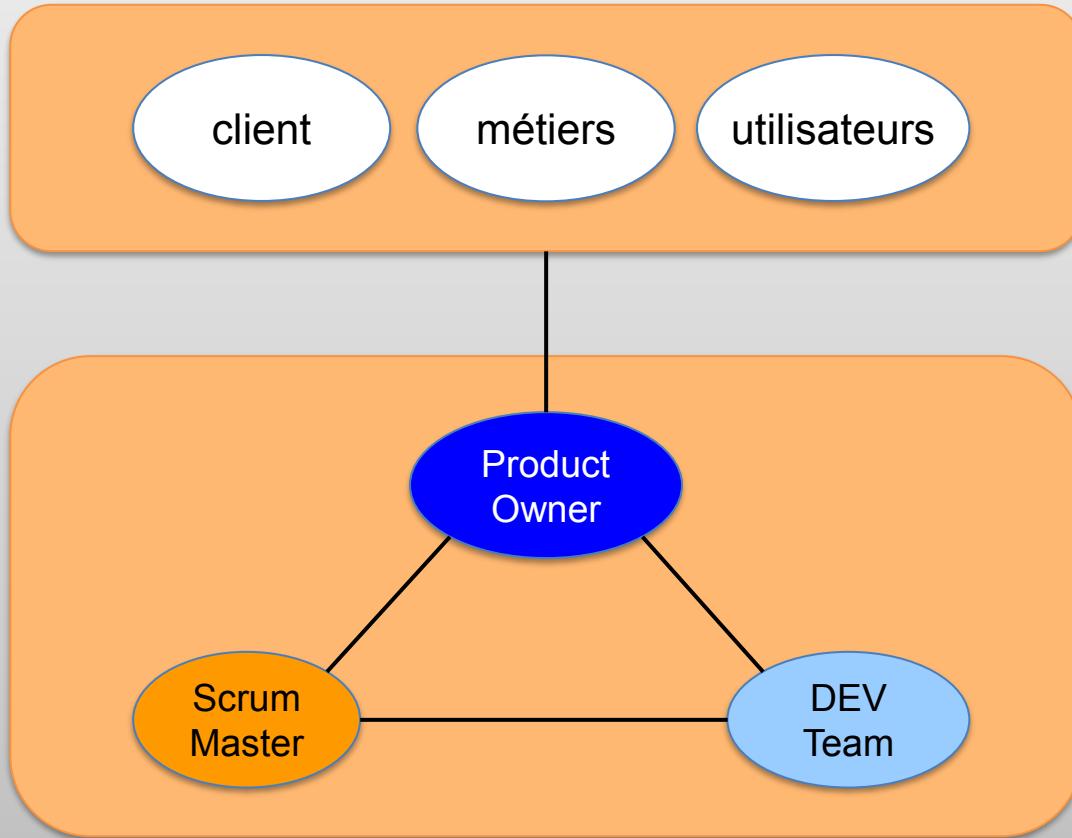
Le Product Owner collecte les besoins fonctionnels et les besoins transverses (non-fonctionnels) auprès des représentants métiers.

Le Product Owner s'appuie sur les compétences et les bonnes pratiques agiles du Scrum Master.

Le client ne s'adresse pas aux développeurs. Le client contacte le Product Owner afin d'exprimer sa demande.

Le Product Owner collecte le feedback des utilisateurs.

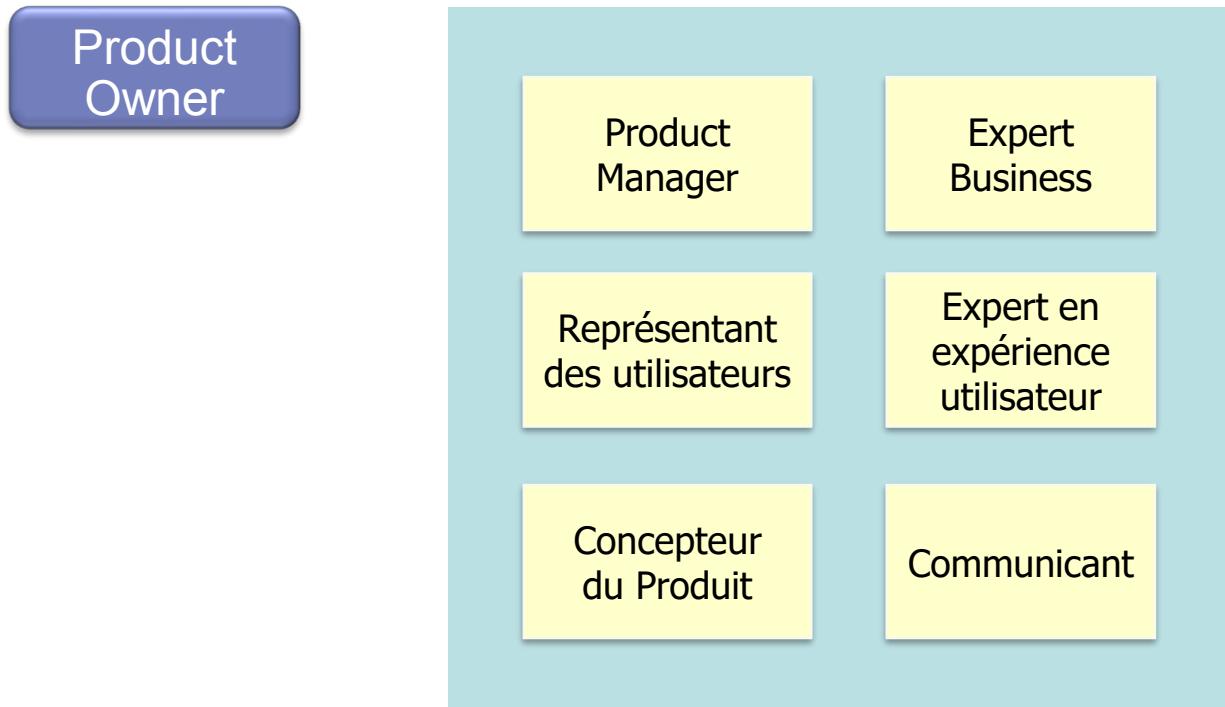
Le Product Owner est le contact fonctionnel unique des développeurs.



Espace de  
coaching du  
Scrum Master

# Le rôle du Product Owner

---



# Le rôle du Product Owner

---

Product  
Owner

Les responsabilités du Product Owner sont importantes.  
Il s'agit d'un rôle difficile.

Dans le monde réel, le Product Owner collabore avec d'autres personnes.  
En effet, le Product Owner ne doit pas être seul en charge de toutes ces activités.

Voici quelques exemples de collaborateurs qui peuvent aider le Product Owner :

- ▶ Un Expert en Expérience Utilisateurs et un correspondant Marketing peuvent aider le PO à recueillir les besoins des Utilisateurs.
- ▶ Un Business Analyst peut aider le PO à formaliser les besoins métier de manière simple et compréhensible.
- ▶ Les Développeurs peuvent proposer des solutions techniques d'implémentation et évaluer les coûts et les bénéfices de chaque solution.

# Le rôle du Product Owner

---

Les compétences attendues du PO



Les causes d'échec du Product Owner



- Incapable de prendre les décisions.
  - Renvoie généralement sur d'autres acteurs.
  - Ralentit fortement le travail de l'équipe.
- 
- Non reconnu par les représentants métier et les utilisateurs comme étant un interlocuteur de confiance.

# Le rôle du Product Owner

---

Les compétences attendues du PO



Les causes d'échec du Product Owner



- Incapable de répondre rapidement aux sollicitations de l'équipe et des utilisateurs
- souvent en responsabilité sur plusieurs projets à la fois
  
- Non expérimenté dans le domaine métier
- Non expérimenté dans les processus et les pratiques du client
- Manque de certaines compétences clés pour la gestion du projet (communication, écoute, synthèse ...).

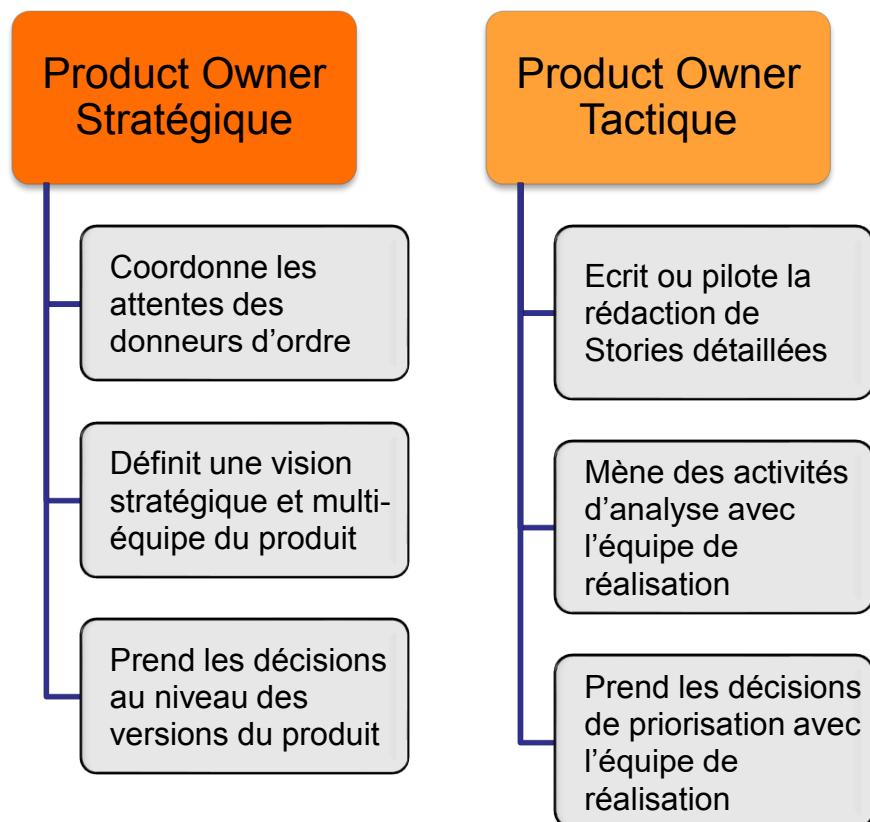
# Le rôle du Product Owner

---

## Armée de PO

Compte tenu des enjeux, ne pas hésiter à demander des ressources supplémentaires afin de disposer d'une « Armée de PO » et ainsi ordonner une répartition réaliste des activités.

Le nombre de PO doit être suffisant en particulier au lancement du projet afin de gérer toutes les activités prévues au calendrier.



# Le rôle du Product Owner

---

Product  
Owner

1

Leadership

Avoir une vision entrepreneuriale  
Accueillir le changement  
Adopter une culture du feedback

3

Décision

Décider du contenu des itérations  
Décider du contenu des livraisons  
avec le client

2

Communication

Etre un ambassadeur du projet agile  
Obtenir l'adhésion de l'équipe à la Vision

4

Disponibilité

100% disponible dès le début du projet  
Pas d'engagement sur plusieurs projets

# Le rôle du Product Owner

Communicant

1

Le Product Owner doit **dialoguer** avec les différents groupes en respectant leurs caractéristiques (langage, métier, processus, calendrier).

2

Le Product Owner doit **partager** la Vision du Produit afin de faire adhérer la communauté agile au Projet.

3

Le Product Owner doit **partager** le contenu des versions afin de réduire les frustrations face au souhait de livrer tout le Produit tout de suite.

4

Le Product Owner doit **partager** le contenu des itérations après chaque Poker Planning afin de réduire les frustrations face au souhait de vouloir embarquer un nb toujours plus grand de stories.

# Le rôle du Product Owner

## Gestion des changements

1

Un projet agile accueille par définition les **changements**.

2

Le Product Owner doit partager les **changements** qui interviennent dans l'intérêt des utilisateurs afin de faire adhérer la communauté agile aux décisions qui modifient la trajectoire prévisionnelle du projet.

3

Le Product Owner doit accueillir les représentants métier et les représentants utilisateurs afin de partager les **décisions** de priorisation qui intègrent les changements.

4

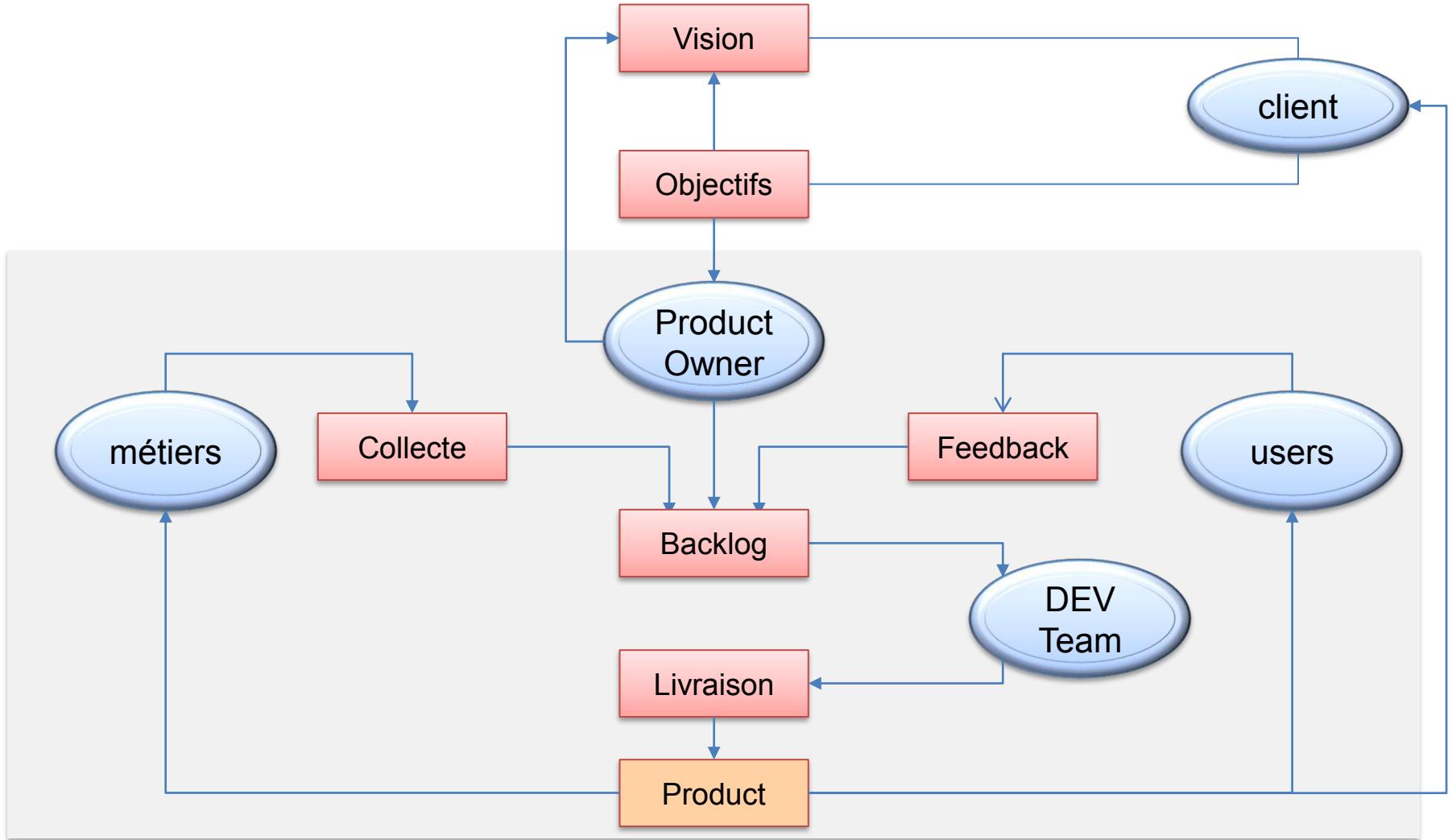
Le Product Owner doit faire preuve de courage lors des prises de **décisions**.

# Le rôle du Product Owner

---

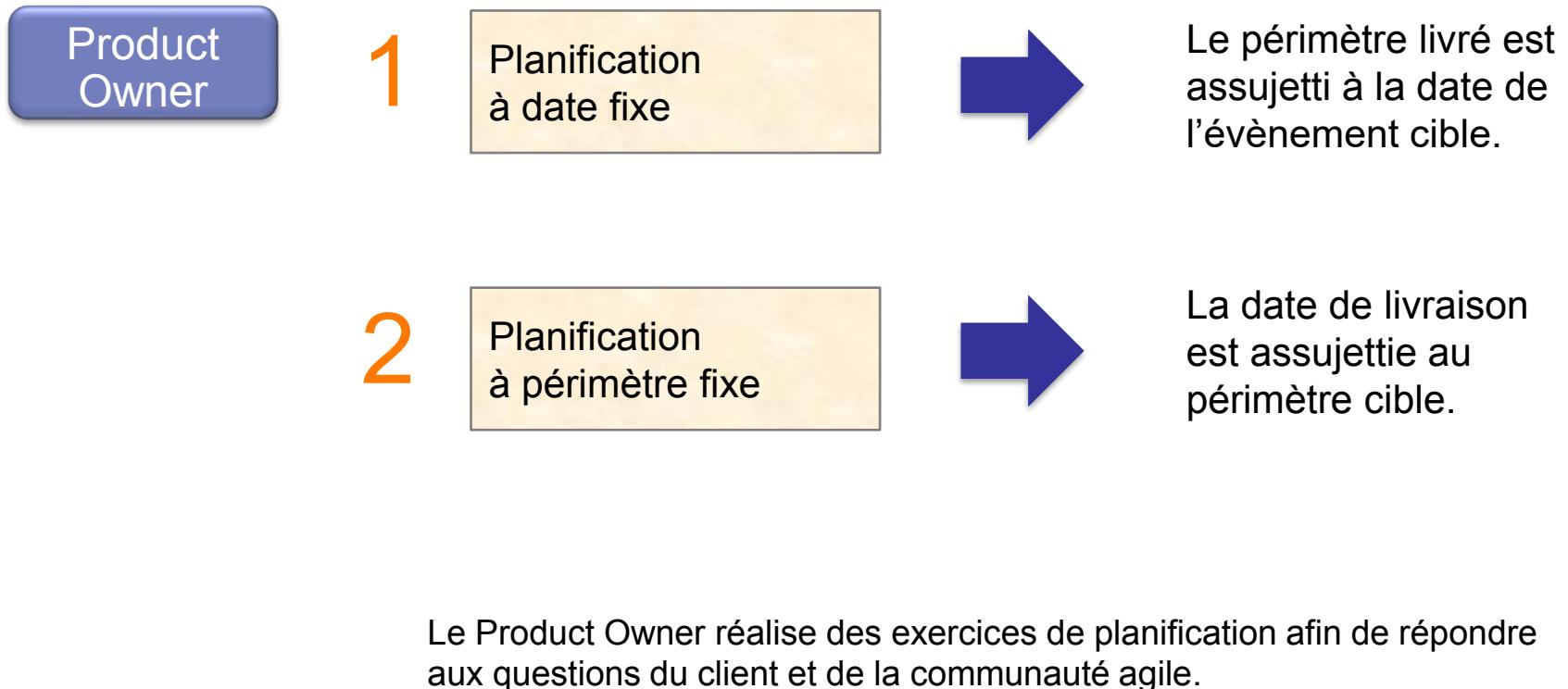


- aborder les sujets à partir du point de vue de l'utilisateur
- aborder les sujets à partir du point de vue du client



# Le rôle du Product Owner

---



# Contribution du PO aux cérémonies agiles

---

Product Owner	L	C	A	N
POKER PLANNING	X			
SPRINT PLANNING				X
DAILY STAND-UP			X	
SPRINT REVIEW		X		
RETROSPECTIVE		X		

## Poker Planning

Le Product Owner est leader lors de la présentation des stories proposées pour la prochaine itération.

Il répond aux questions des développeurs afin de leur permettre d'évaluer la complexité des stories présentées en réunion.

Le Product Owner sélectionne les stories de la prochaine itération en respectant la vélocité de l'équipe de réalisation.

## Légende



Leader



Auditeur



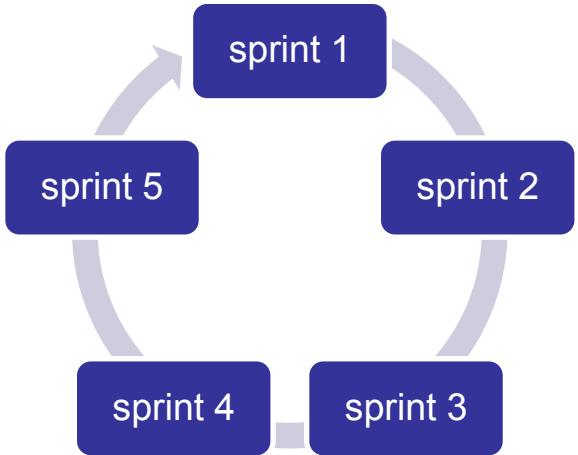
Contributeur



Ne participe pas

# Les méthodes agiles

Merci



# Méthodes agiles

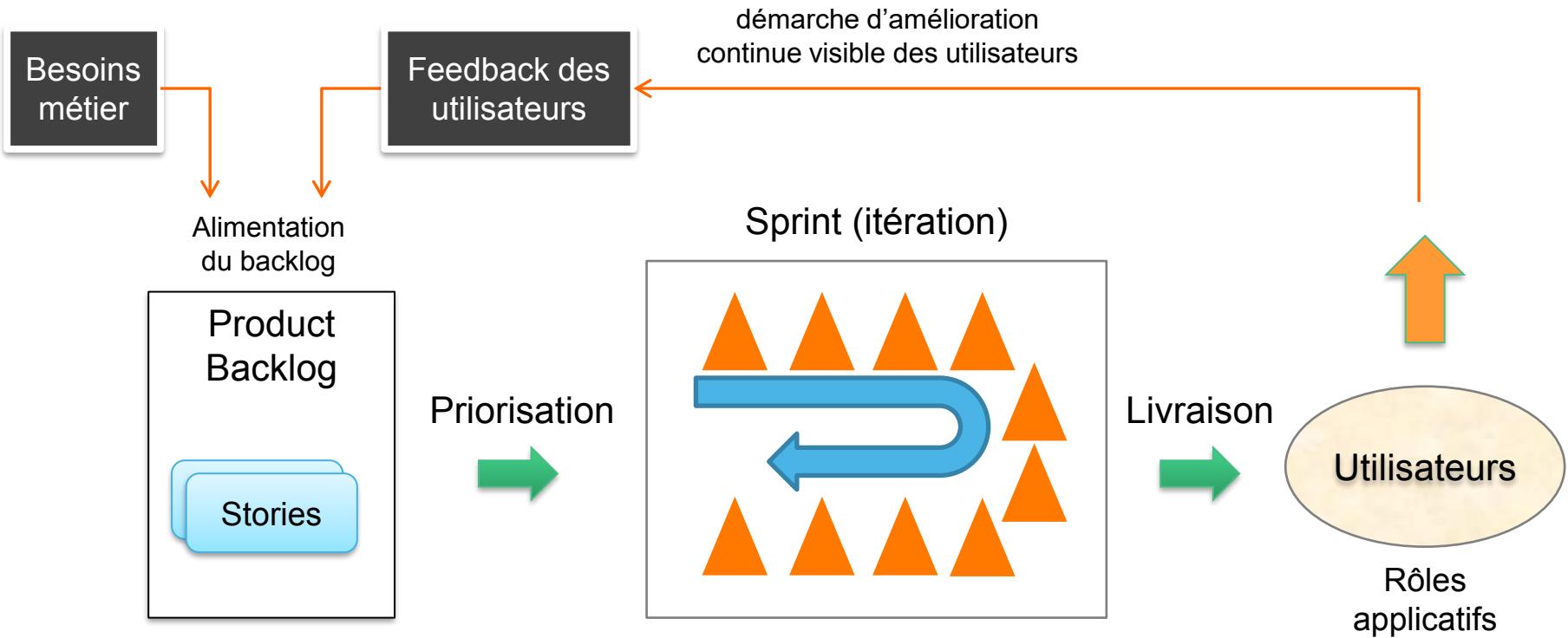
Méthode  
SCRUM



**Conduite de projet Agile**  
Le Product Backlog



# Méthode SCRUM



# Product Backlog

---

Le Product Backlog est le référentiel des travaux qui restent à faire.

Le Product Backlog accueille sous la forme de stories **tous les travaux** qui seront confiés aux développeurs.

**Tous les travaux** doivent être enregistrés dans le Product Backlog car leur existence doit être déclarée afin de permettre leur évaluation et leur priorisation avant de les engager dans une itération à venir.

Story

Rédacteur

User Story

Product Owner

Technical Story

Développeur

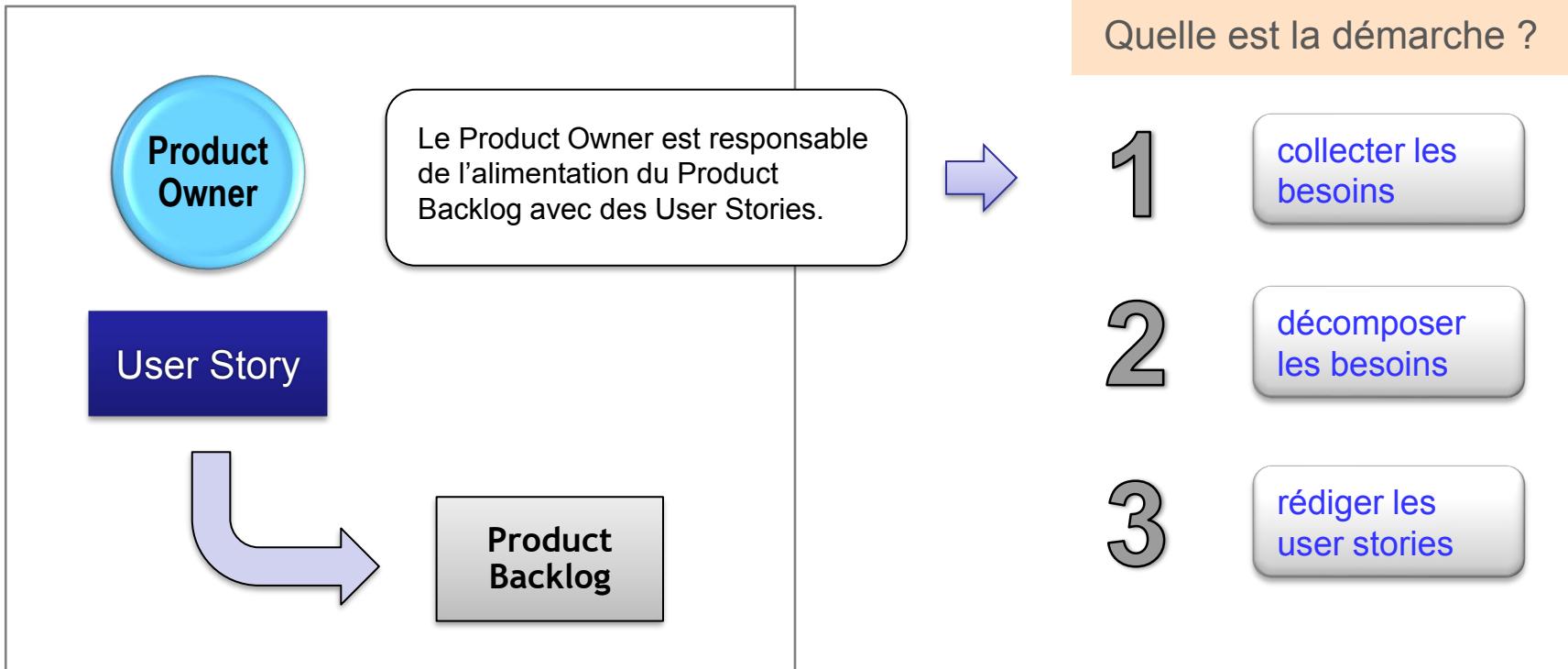
Anomalie

Utilisateur

Etude

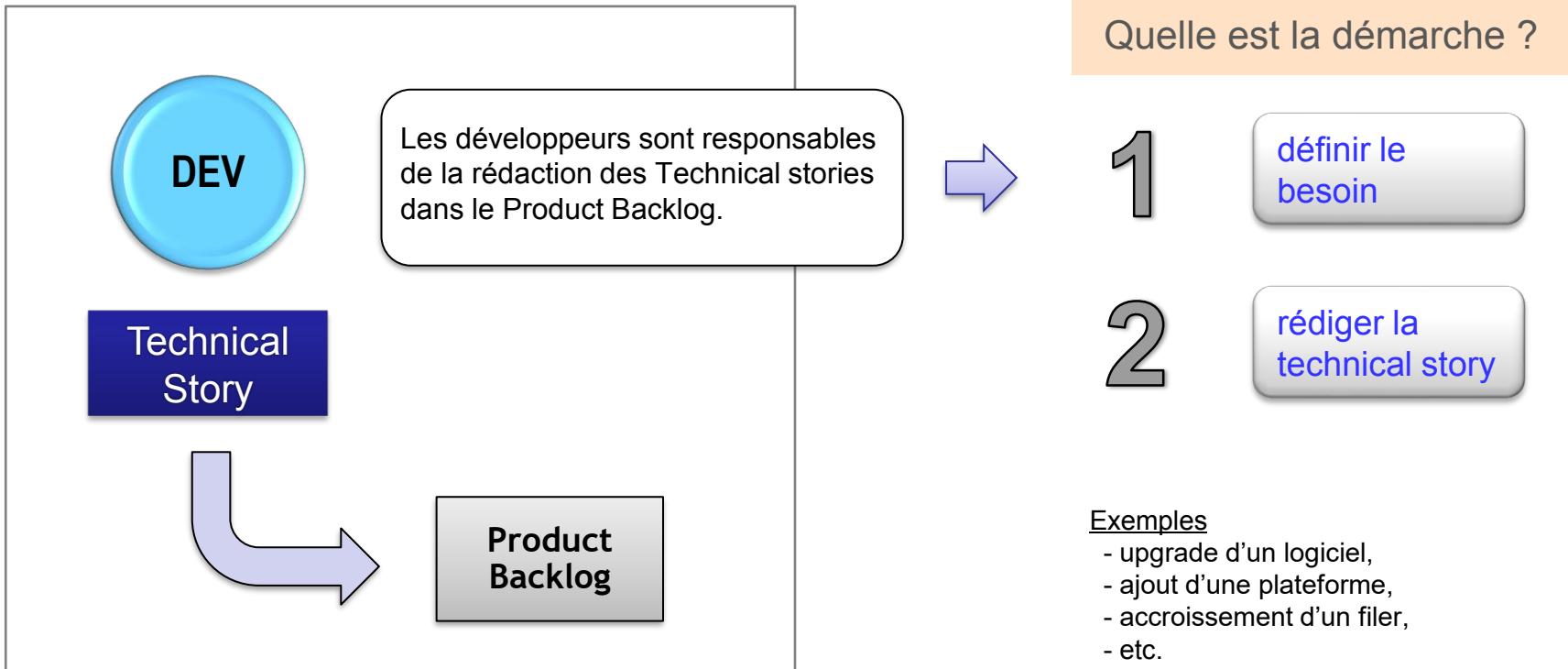
Product Owner

# User stories

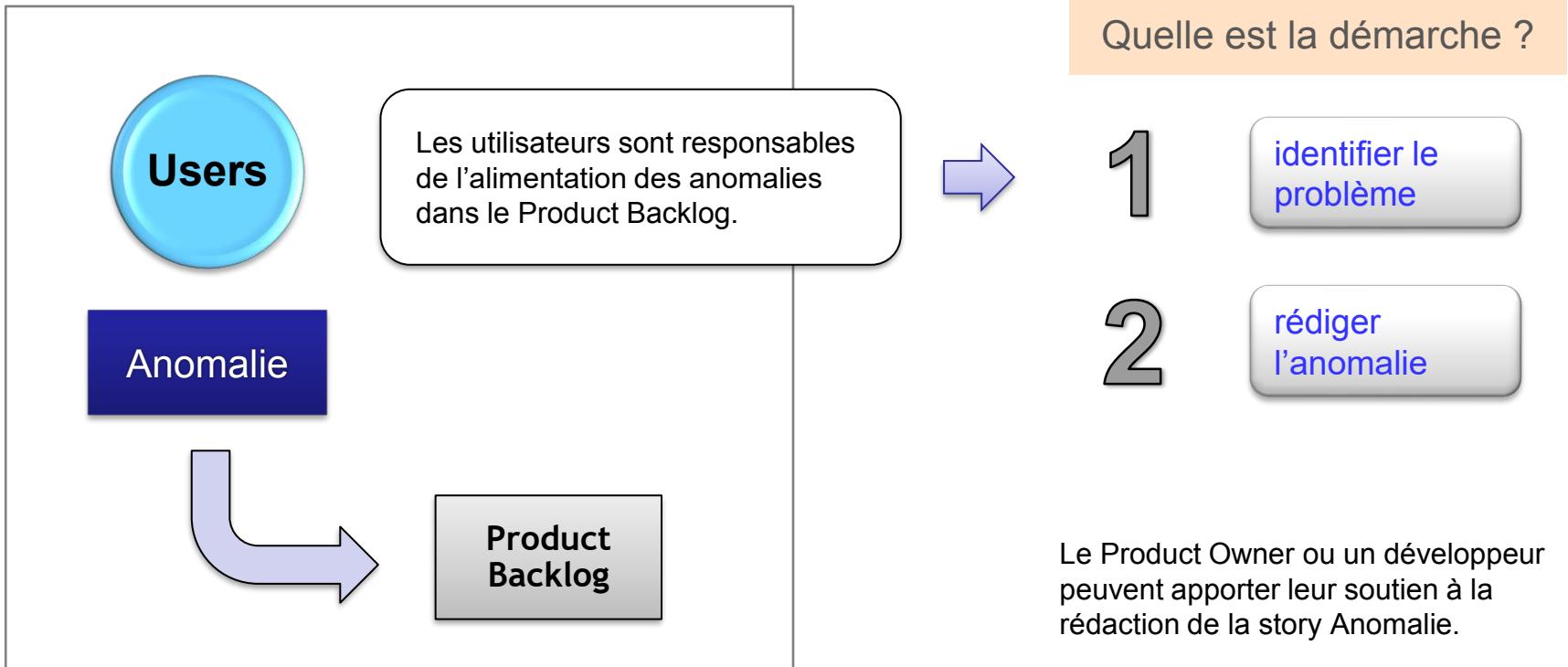


Les User Stories du Product Backlog sont les stories les plus nombreuses.

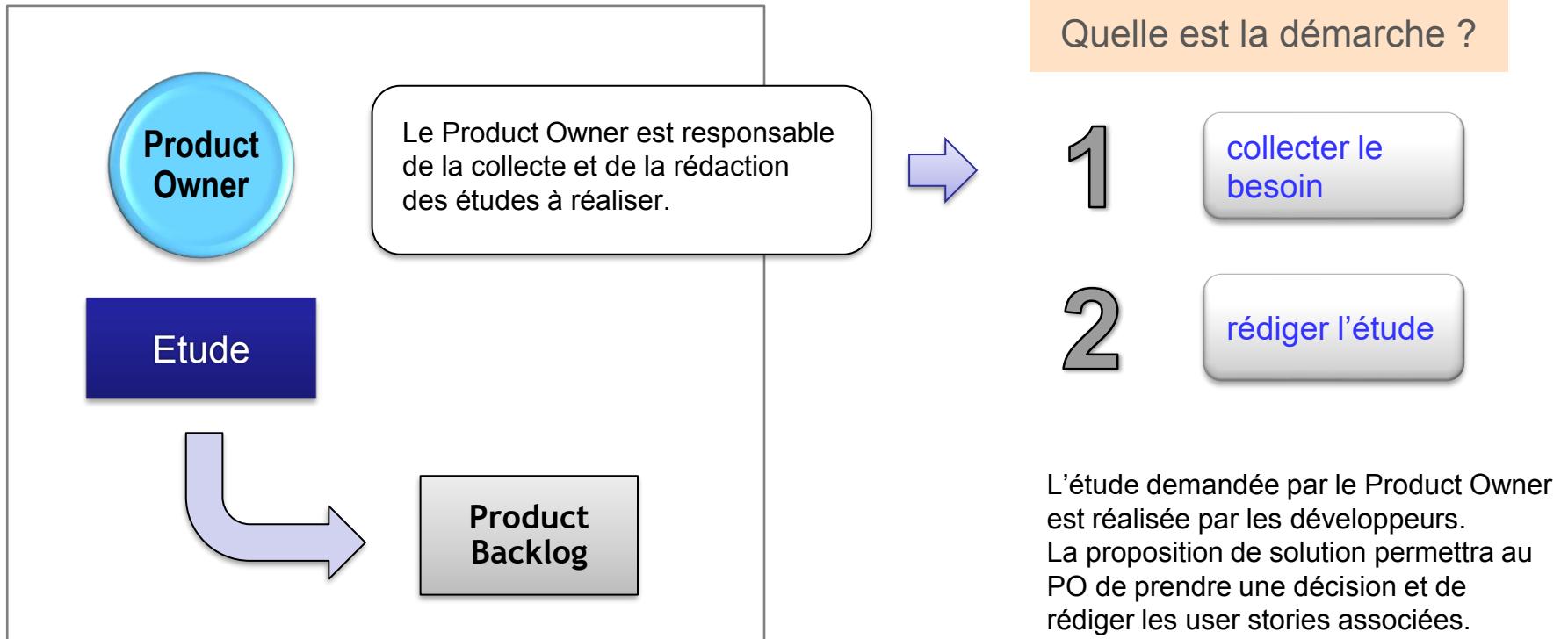
# Technical stories



# Anomalies - Defects



# Etudes



# Ce qu'il faut retenir

---

1

User stories

Les **User Stories** définissent les incrémentations fonctionnelles du produit. Elles suivent les règles agiles (décomposition simple, mémo INVEST, story READY) et incluent les règles de gestion validées par les métiers. Les User Stories sont priorisées par le Product Owner et évaluées par les développeurs.

2

Technical stories

Les **Technical Stories** sont soumises par l'équipe de réalisation. Elles représentent des travaux utiles aux environnements et/ou aux plateformes du projet. Elles permettent également de réaliser l'installation des nouveaux logiciels et des outils nécessaires au projet.

3

Anomalies

Les **Anomalies** sont soumises par les utilisateurs afin de corriger les défauts observés sur l'application en production. Les anomalies sont priorisées en fonction de leur sévérité (Critique, Majeure, Mineure) et du niveau d'effort associé.

# Les études

---

1

Le démarrage d'un projet agile doit s'accompagner de l'identification et de la mise en œuvre des études visant à apporter des réponses et des propositions de solution aux problématiques du Produit.

2

Les études sont enregistrées dans le Product Backlog par le Product Owner et représentent un niveau d'effort pour l'équipe de développeurs au même titre que les autres stories.

3

En vie de solution, l'introduction d'une nouvelle interface, d'un nouveau rôle utilisateur, ou le traitement d'un temps de réponse dégradé requièrent la rédaction d'une story de type Etude.

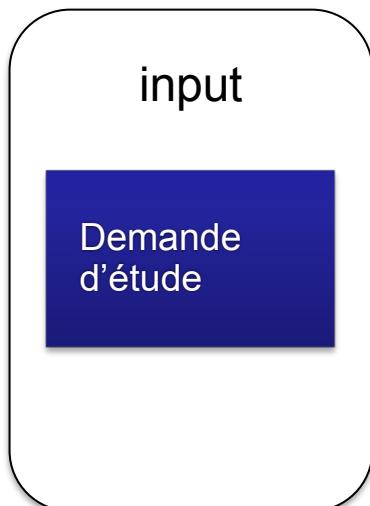
# Les études

## Comment ça marche ?

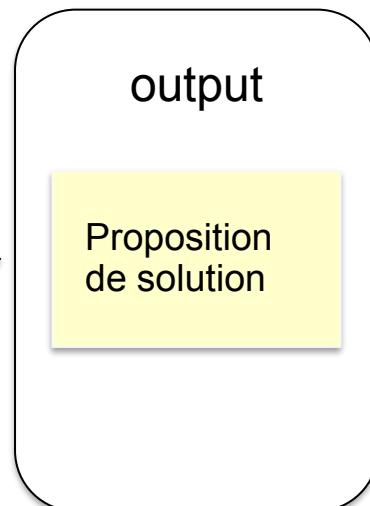
Catégorie

- Evolution
- Nouveau rôle
- Interface
- Architecture
- Performance
- Sécurité

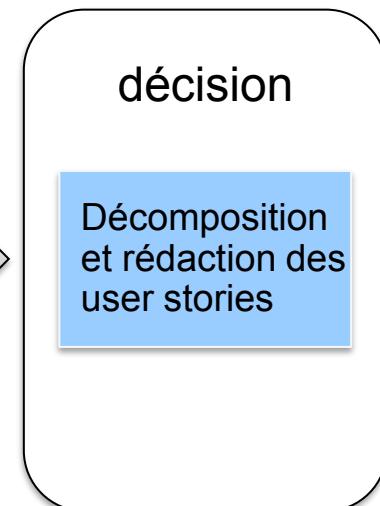
1



2



3

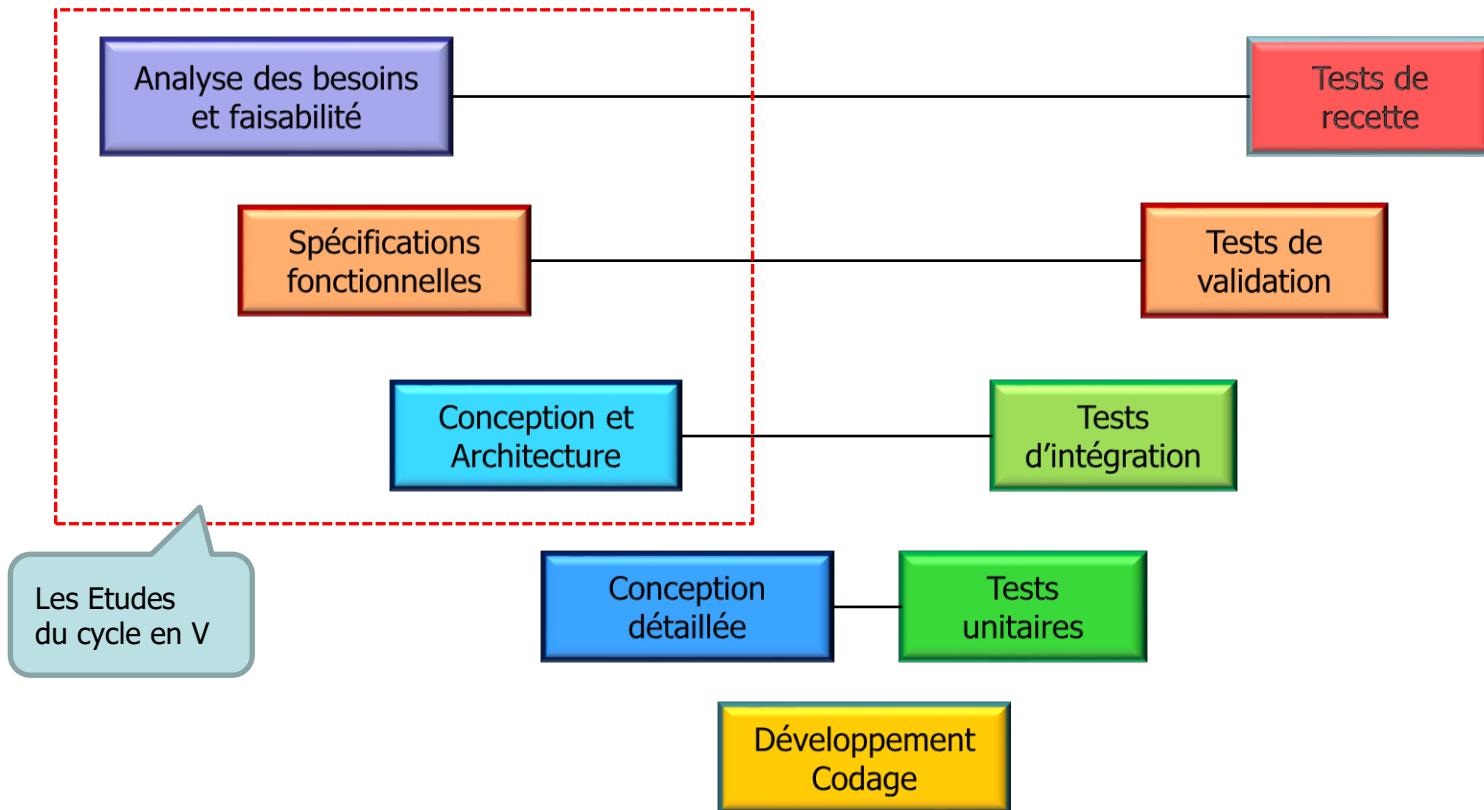


Product Owner

Développeurs

Product Owner

# Les études - Projet classique



# Les études - Projet classique

---

Les études correspondent à un investissement important en termes de budget (k€).

Les études correspondent à un investissement significatif en termes de valeur métier et de connaissances.

Les études sont le socle qui porte l'engagement de résultat du prestataire.

L'introduction de changements imprévus remet en cause les études initiales et donc l'obligation de résultat du prestataire.

Le calendrier et le budget du projet classique peuvent être significativement impactés par l'introduction de changements.

# Les études - Projet agile

---

Les études doivent être identifiées à partir de la Vision du Produit définie par le Client.

Elles peuvent être découvertes à partir des problématiques embarquées avec le Produit conformément à la Vision du Produit validée par le client.

Les études doivent prendre en compte les besoins transverses (NFR) rédigés dans les Critères du Done des User stories du Product Backlog.

Les principales catégories d'études sont les suivantes :

Evolution

Architecture

Nouveau rôle

Performance

Interface

Sécurité

NFR : Non-Functional Requirements  
Il s'agit des contraintes transverses qui s'applique au produit dans son ensemble.

# Les études - Projet agile

---



La vie de solution correspond à la période du projet qui intervient après la livraison de la première version du Produit.

La première version du Produit livré en production aux utilisateurs correspond au MVP (Minimum Viable Product) du Produit. Cette version inclut le noyau critique des fonctionnalités et des besoins transverses nécessaires aux utilisateurs du Produit.

En vie de solution, les retards d'identification des études à lancer sont autant de risques de ne pas accomplir la Vision du Produit dans le cadre du calendrier attendu par le client.

# Ce qu'il faut retenir

---

- Le Product Backlog est créé et alimenté par le Product Owner.
- Les stories du Product Backlog sont classées par ordre de priorité.
- Le Product Backlog s'enrichit de nouvelles stories avec l'avancement de la découverte et de l'exploration fonctionnelle du Produit.
  
- La priorisation du Product Backlog est à faire en amont de la préparation de la prochaine itération (avant le Sprint Planning).
- Le volume des stories engagées avec la prochaine itération (Sprint Backlog) est assujetti à la vélocité de l'équipe de réalisation.

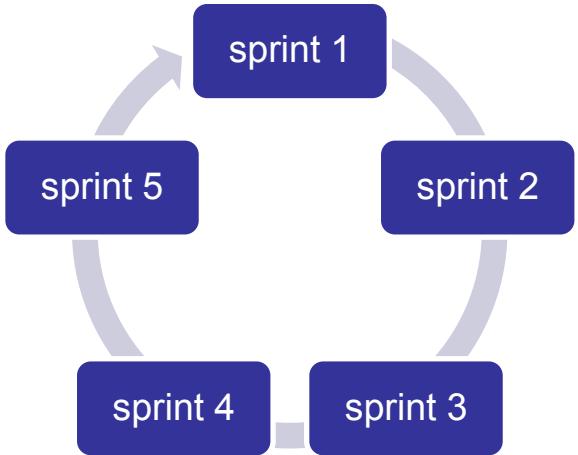
# Ce qu'il faut retenir

---

- L'alimentation continue du Product Backlog pendant toute la durée du projet agile conduit le Product Owner à réaliser de nouvelles actions de priorisation à valider avec le client.
- Ces actions de priorisation conduisent à revisiter le plan de livraison des versions et des releases afin que le nouveau calendrier soit en accord avec la trajectoire fonctionnelle validée par le client avant d'être partagée avec toute la communauté agile du projet.
- Conclusion : la trajectoire prévisionnelle d'un projet agile n'est pas respectée. Il s'agit du fonctionnement normal d'un projet agile qui intègre par définition les changements qui modifient sa trajectoire fonctionnelle.

# Méthodes agiles

Merci

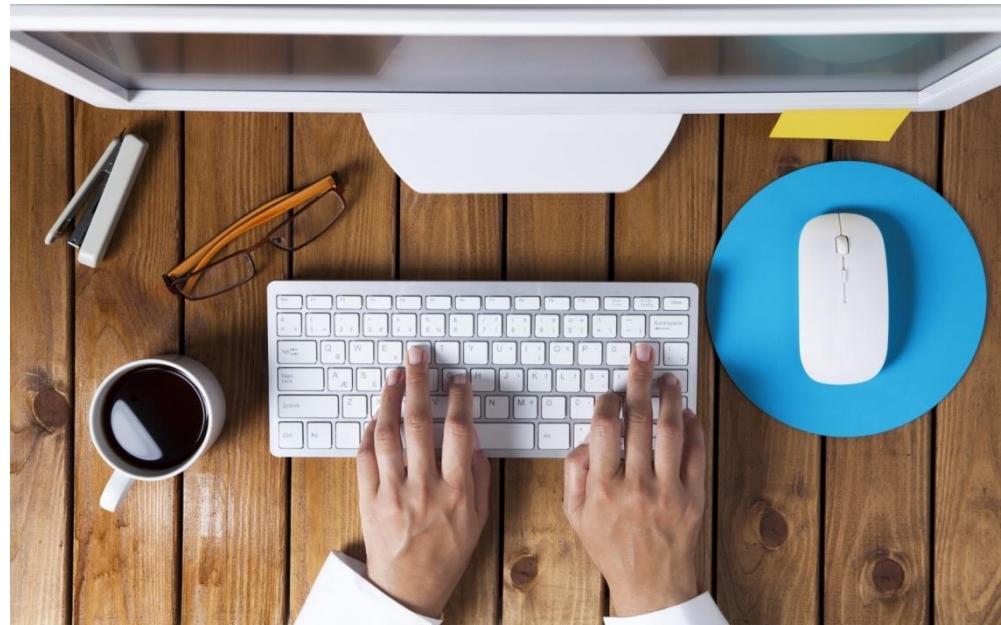


# Méthodes agiles

Méthode  
SCRUM



**Conduite de projet Agile**  
Priorisation



# Priorisation du Product Backlog

---

La priorisation du Product Backlog est une action réalisée par le Product Owner. C'est une action importante car elle a pour objectif d'identifier et d'ordonner les stories qui sont prêtes à être engagées dans une prochaine itération.

La priorisation du Product Backlog est une action Qualité car elle a pour objectif d'identifier les stories qui ne sont pas (encore) prêtes.

Réaliser un exercice de priorisation a un sens si et seulement si le Product Backlog est alimenté. Un Product Backlog presque vide ou peu alimenté donne peu de valeur à un exercice de priorisation.

La priorisation du Product Backlog est un prérequis à la planification du projet agile. Un Product Backlog presque vide ou peu alimenté ne donnera aucune visibilité au client.

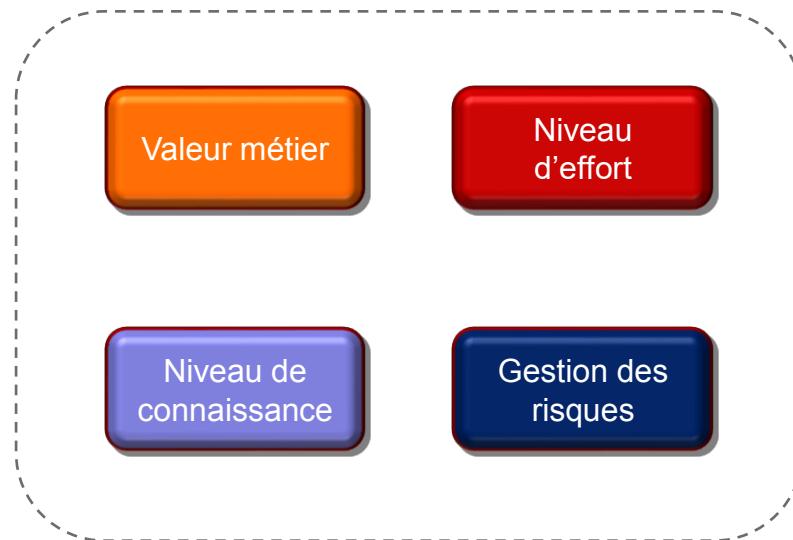
L'alimentation du Product Backlog est un signe de bonne santé car il témoigne non seulement de la qualité de la phase de collecte des besoins mais aussi de la capacité du projet agile à être priorisé et planifié.

# Priorisation du Product Backlog

---

Les Stories doivent être priorisées afin que la partie haute du Product Backlog soit prête pour alimenter les prochaines itérations.

Dans ce but, la priorisation du Product Backlog s'appuie sur 4 critères.



# Priorisation du Product Backlog

---

Valeur métier



La priorisation des stories doit maximiser leur valeur métier.  
Il s'agit d'atteindre le plus rapidement possible la Vision métier demandée par le client.

Niveau de connaissance



Une collecte de besoins insuffisante ou une exploration inachevée des règles de gestion ont un impact sur la qualité des user stories.  
Plus simplement, les user stories qui ne sont pas encore prêtées doivent être dépriorisées.

# Priorisation du Product Backlog

---

Niveau d'effort



La priorisation prend en compte la vélocité et donc la capacité à faire de l'équipe. Un niveau d'effort trop important conduit à déprioriser des stories du Sprint Backlog afin de respecter la vélocité de l'équipe de réalisation.

Pour rappel, la somme de la complexité des stories d'une itération à venir doit être inférieure ou égale à la vélocité de l'équipe de réalisation.

Gestion des risques



Les risques juridiques visent à identifier les stories qui ne sont pas en conformité avec les lois et les règlements.

Les risques techniques visent à identifier les stories dont les Critères du Done incluent des contraintes transverses (NFR) non satisfaites.

Ces risques sont autant de réserves à lever.

Le retard à lancer les études correspondantes constitue également un risque.

Les users stories qui portent des risques non résolus ne sont pas prêtes et doivent donc être dépriorisées.

# Priorisation MoSCoW

---



La méthode MoSCoW est une méthode de priorisation qui peut significativement aider le Product Owner.

MoSCoW est une méthode qui permet d'affecter les stories du Product Backlog à une catégorie parmi 4.

## 1. Must do (M) – ce que je DOIS faire

Il s'agit des stories correspondant aux travaux qui doivent impérativement être réalisés afin de pouvoir utiliser le Produit. Si vous ne réalisez pas ces travaux, le projet est un échec, le Produit est inutile, la valeur métier n'est pas libérée.

## 2. Should do (S) – ce que je DEVRAIS faire

Il s'agit des stories correspondant aux travaux qui apportent une grosse valeur ajoutée au projet car elles sont associées à des fonctionnalités importantes mais moins prioritaires que celles du Must.

## 3. Could do (C) – ce que je POURRAIS faire

Il s'agit des stories correspondant à des travaux de confort que vous pourriez faire s'il vous reste des ressources après avoir terminé les périmètres fonctionnels des catégories **Must** et **Should**.

## 4. Won't do (W) – ce que je NE FERAIS PAS

L'idée est ici de rassembler les travaux que l'on décide d'exclure temporairement du projet car les stories ne sont pas prêtes. Gardez cependant ces travaux dans votre liste d'idées d'amélioration du Produit.

# Priorisation MoSCoW

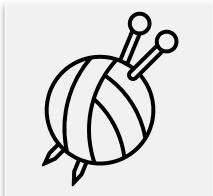
---



Nous venons de définir que la priorisation du Product Backlog peut s'appuyer sur 4 paramètres (valeur métier, niveau de connaissance, niveau d'effort, niveau de gestion des risques).

Cette démarche permet d'identifier les travaux dont la réalisation peut être engagée mais aussi les travaux dont la réalisation ne doit pas être engagée car les stories ne sont pas prêtes et/ou ne sont pas au niveau de qualité ou au niveau de connaissance attendue.

A noter que le niveau de connaissance ou la valeur métier de ces stories pourront évoluer dans le temps afin que les concepts et les règles métier qu'elles utilisent puissent être maîtrisés. En conséquence, les stories associées à ces concepts et à ces règles métier pourront être mieux rédigées et ainsi devenir éligibles afin d'être priorisées avec une prochaine itération du projet.

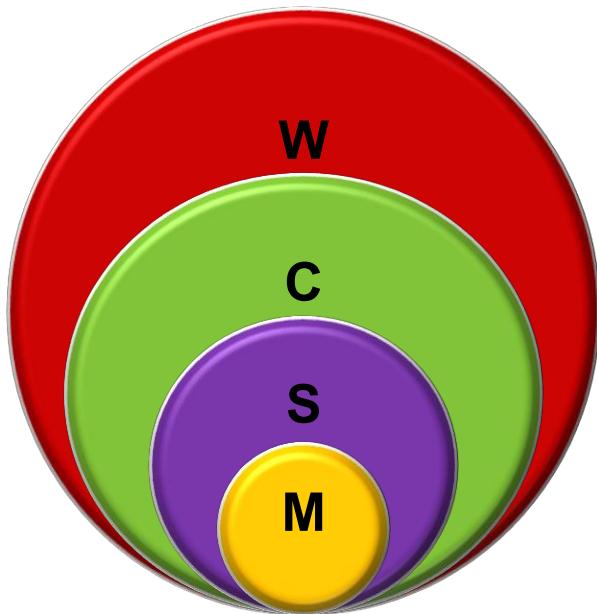


La méthode MoSCoW permet d'aider la priorisation en examinant les stories dont la réalisation peut être engagée.

Le client et le Product Owner s'appuient sur les catégories Must, Should et Could afin de visualiser la séquence des fonctionnalités à livrer.

Au début du projet, le client et le Product Owner peuvent convenir que la catégorie Must représente le périmètre fonctionnel du MVP; c'est-à-dire le périmètre de la première version du Produit.

# Priorisation MoSCoW



Vital

**MUST**

Il s'agit d'identifier ce qui est incontournable et doit être impérativement livré avec la prochaine version / release.

Essentiel

**SHOULD**

Il s'agit de fonctions essentielles qui devraient aussi être livrées mais sont cependant moins prioritaires.

Confort

**COULD**

Il s'agit des autres fonctions qui pourront être livrés plus tard avec une autre version / release.

Exclusion  
temporaire

**WON'T**

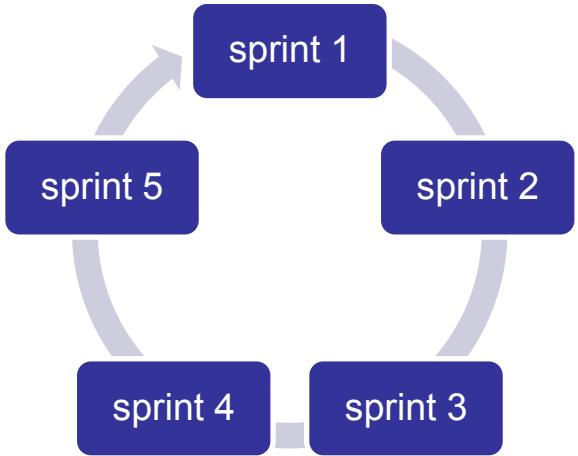
Il s'agit ici d'identifier les stories qui bien que rédigées et insérés dans le Product Backlog ne sont pas prêtes et ne sont pas à traiter dans l'immédiat.

# Priorisation MoSCoW

	Version (i) / Release (n)
1	gestion des comptes utilisateurs
2	administration du back-office
3	enregistrement d'un témoignage météo
4	interrogation de la base des témoignages
5	restitution des résultats
6	sécurisation des données
7	sécurité des accès
8	enregistrement d'un témoignage météo via SMS
9	gestion des profils utilisateurs
10	gestion des publicités
11	restitution des publicités
12	gestion des alertes
13	statistiques utilisateurs
14	envoi des alertes via SMS
15	restitution des résultats graphiques
16	envoi des alertes à un tiers

# Méthodes agiles

Merci



# Méthodes agiles

Méthode  
SCRUM

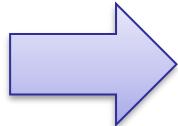


**Conduite de projet Agile**  
Planification



# Planification du projet agile

## Le Produit et sa planification

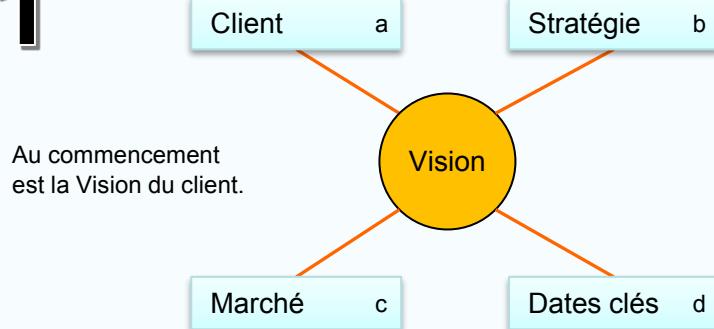


### Bénéfices attendus

La planification permet au Client de respecter ses engagements vis-à-vis du marché et de ses utilisateurs.

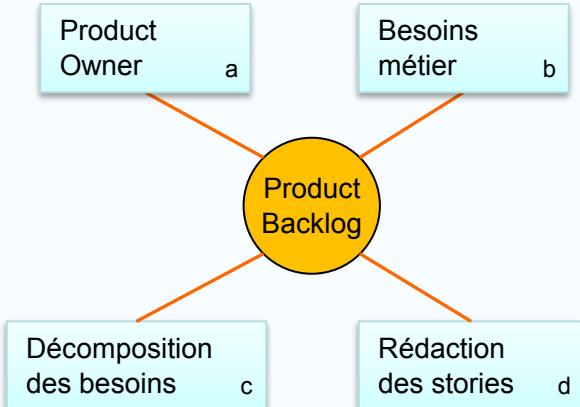
La planification permet au Client de maîtriser son projet agile en intégrant l'évolution de ses besoins et d'exprimer la trajectoire fonctionnelle souhaitée par le Client.

# 1



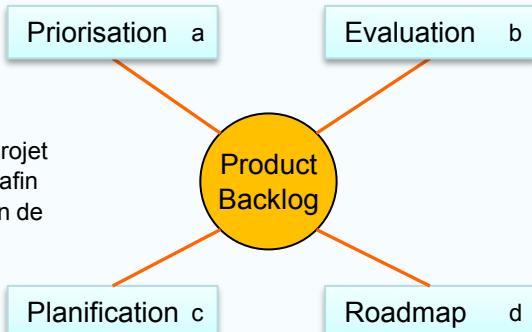
# 2

Puis vient le temps des décisions en s'appuyant sur la phase de collecte des besoins et la rédaction des stories.



# 3

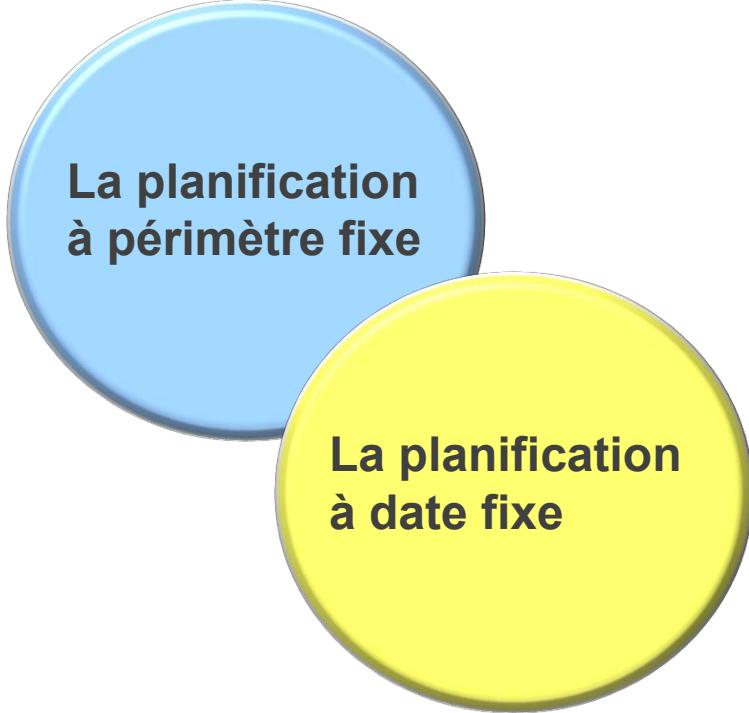
Les étapes du projet prennent forme afin de nourrir le plan de livraison.



# Planification

---

La méthode Scrum utilise deux types de planification.



**La planification  
à périmètre fixe**

**La planification  
à date fixe**

# La planification à périmètre fixe

---

Le Produit  
et sa planification  
à périmètre fixe

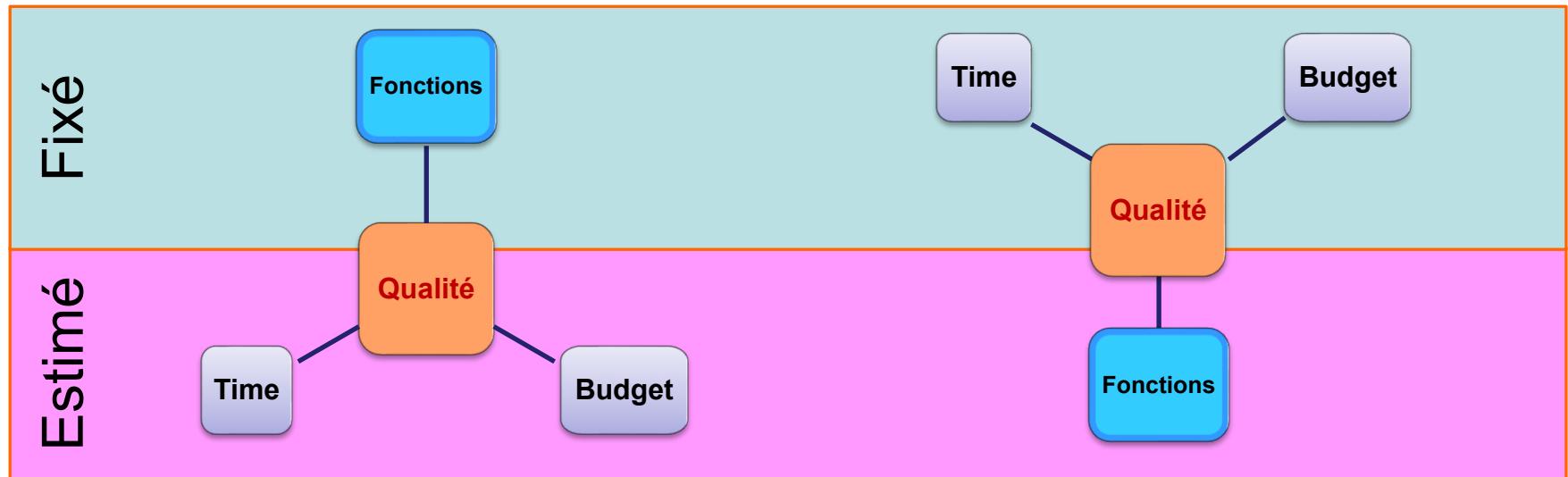


## Bénéfice attendu

Il s'agit de savoir répondre à la question suivante : quelle est la date de livraison pour un périmètre fonctionnel choisi par le Client ?

Réponse : la date de livraison est assujettie au périmètre fonctionnel à livrer.

# La planification à périmètre fixe



Obligation de résultat

Approche classique

Obligation de moyens

Approche Agile

# La planification à périmètre fixe

1 date	13 pts
2 liste d'amis	8 pts
3 invitations	20 pts
4 animations	5 pts
5 clown	8 pts
6 ballons	13 pts
7 jouets	13 pts
8 gateaux	20 pts
9 cadeaux	20 pts

41 pts

39 pts

40 pts

Release "Alpha"

Sprint 1

Sprint 2

Sprint 3

## Actions

- Alimenter le Product Backlog
- Prioriser les stories
- Evaluer les stories en points
- Identifier les sprints
- Construire les versions /releases

10 Avec Agile	13 pts
11 Je sais faire	8 pts
12 Des choses	3 pts
13 Que je ne pensais	5 pts
14 Pas pouvoir	8 pts
15 Faire avant	20 pts
16 Avec ma méthode	13 pts
17 En V	8 pts

37 pts

41 pts

Release "Omega"

Sprint 4

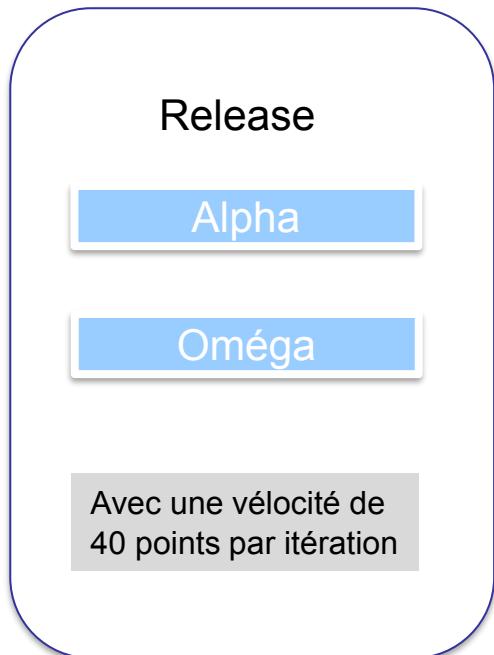
Sprint 5

## Prérequis

- Connaitre la vitesse de l'équipe de réalisation afin de pouvoir identifier les sprints nécessaires aux travaux du périmètre choisi
- Exemple : vitesse = 40 points

# La planification à périmètre fixe

---



- Alpha sera disponible dans 3 itérations
- Oméga sera disponible 2 itérations plus tard

Le périmètre fonctionnel choisi par le client est entièrement planifié :

- il sera réalisé en 5 itérations
- il sera livré avec 2 releases

# Planification

---



**La planification  
à date fixe**

**La planification à  
périmètre fixe**

# La planification à date fixe

---

Le Produit  
et sa planification  
à date fixe

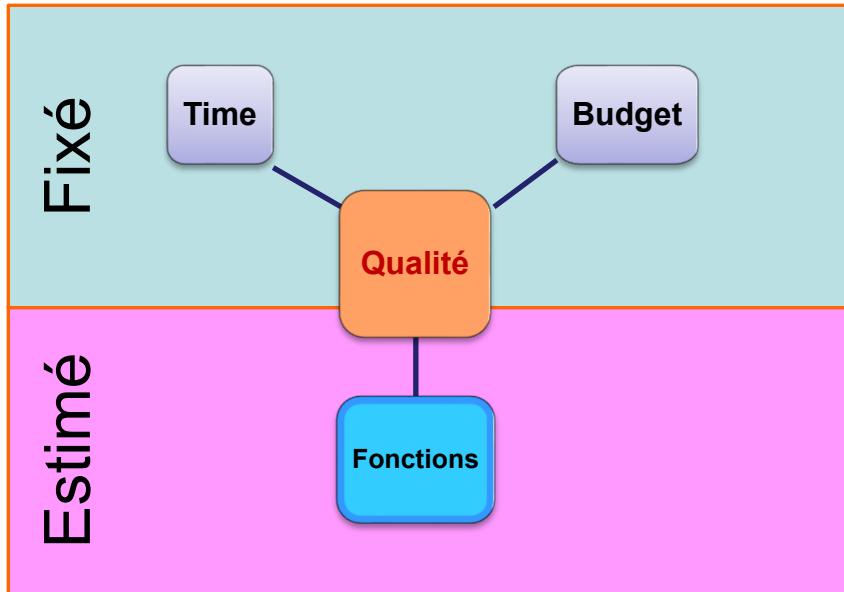


## Bénéfice attendu

Il s'agit de savoir répondre à la question suivante : quel est le périmètre fonctionnel qui sera disponible à une date choisie par le Client ?

Réponse : le périmètre fonctionnel est assujetti à la date de livraison.

# La planification à date fixe



La planification à date fixe correspond plus à une approche Agile :

- le marché guide le projet afin de livrer une nouvelle version
- la population des utilisateurs ciblés permet de collecter le feedback à intégrer dans le backlog
- le budget nécessaire à cette nouvelle version est disponible
- le client choisit une date clé afin de répondre aux attentes du marché et des utilisateurs

Approche Agile

# La planification à date fixe

Le Product Backlog du projet contient 200 points.

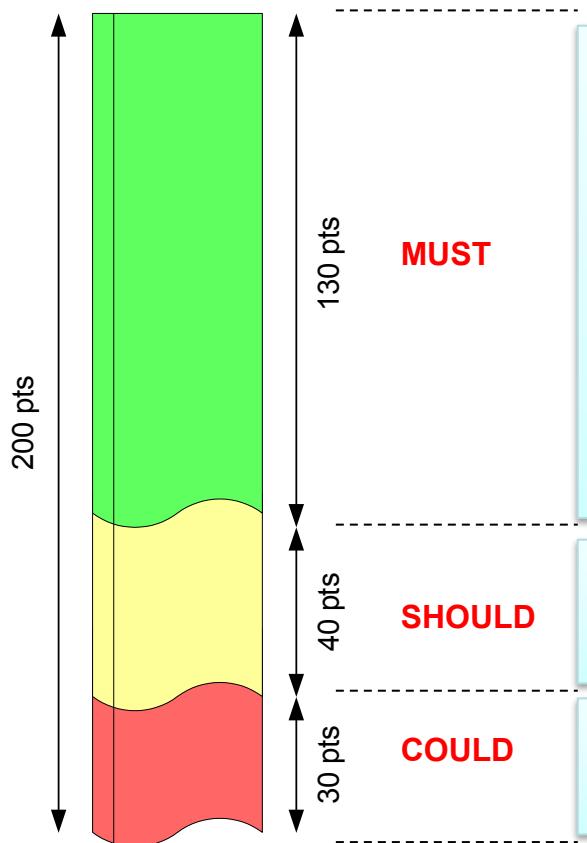
La vélocité de l'équipe est de 25 points (itération de 2 semaines)

Le client demande au Product Owner de prévoir une livraison dans 3 mois maximum afin d'être au rendez-vous du marché.

Le client demande au Product Owner quel sera le périmètre fonctionnel disponible à cette date.

## Planification à date fixe

- Délai fixé par le client : 3 mois
- 3 mois correspondent à 6 itérations de 2 semaines
- 6 itérations de 25 points correspondent à 150 points.



# La planification à date fixe

---

Périmètre

MUST

SHOULD

COULD

Quel périmètre le Product Owner peut-il annoncer au Client avec une vélocité de 25 points par itération ?



Le périmètre MUST (130 points) sera livré.



Le périmètre SHOULD (40 points) sera peut-être livré.



Le périmètre COULD (30 points supplémentaires) ne sera pas livré.

# La planification à date fixe

---

Périmètre

MUST

SHOULD

COULD

Quel périmètre le Product Owner peut-il annoncer au Client avec une vélocité de 25 points par itération ?



Le périmètre SHOULD sera *peut-être* livré si les 20 points  $= (130 + 40 - 150)$  peuvent être traités lors des 6 itérations compte tenu de l'accroissement de la vélocité (\*).

(\*) Un accroissement de 3,5 points de la vélocité correspond à une augmentation de 14% de la capacité à faire de l'équipe.

# Ce qu'il faut retenir

---

Le client pose toujours sous une forme variable les 2 mêmes questions :

- 1** A quelle date ce périmètre fonctionnel peut-il être livré aux utilisateurs ?
  
- 2** Quel est le périmètre fonctionnel qui peut être livré en production à la date dd/mm/yyyy ?



## Réponse

Faire une planification à périmètre fixe

Faire une planification à date fixe

# Méthodes agiles

Merci

