

# Cours 2

## Développement agile

### La méthode SCRUM

**Vision, cycle, backlog Produit**

V. Deslandres ©



# Sommaire de ce cours

- **Scrum** : présentation rapide
- Le cycle - 6
- La vision produit - 12
- Le backlog Produit - 20

3

# SCRUM : les rôles

L'équipe  
SCRUM : max 8  
membres; auto  
organisée

**PRODUCT OWNER**

Porte la vision Produit,  
définit le backlog, donne  
les retours des users



**SCRUMMASTER**

- Animateur
- Permet à l'équipe d'avancer
- Pas un chef, peut être SM tournant



**TESTEUR**



**DEVELOPPEUR**

Responsable du  
développement



**ARCHITECTE**

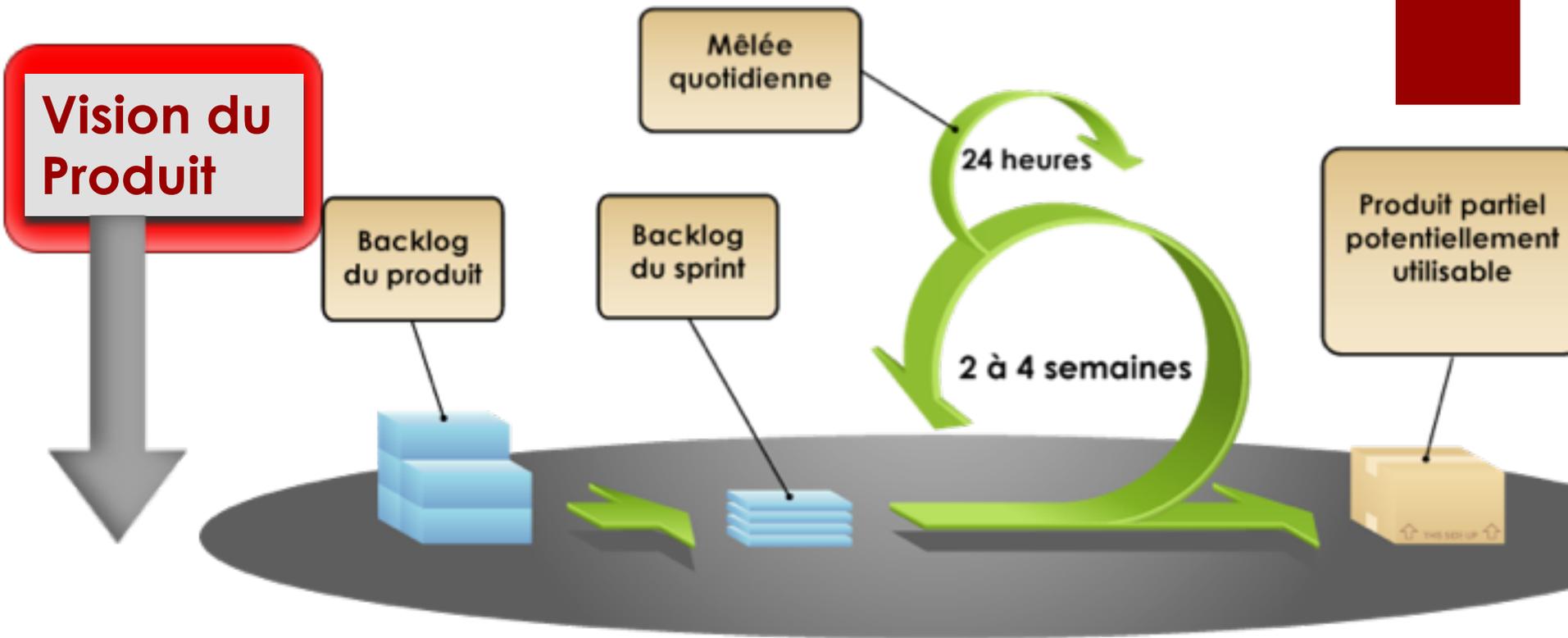


**DESIGNER**

**EQUIPE**

# Scrum – Cycle 1/6

4

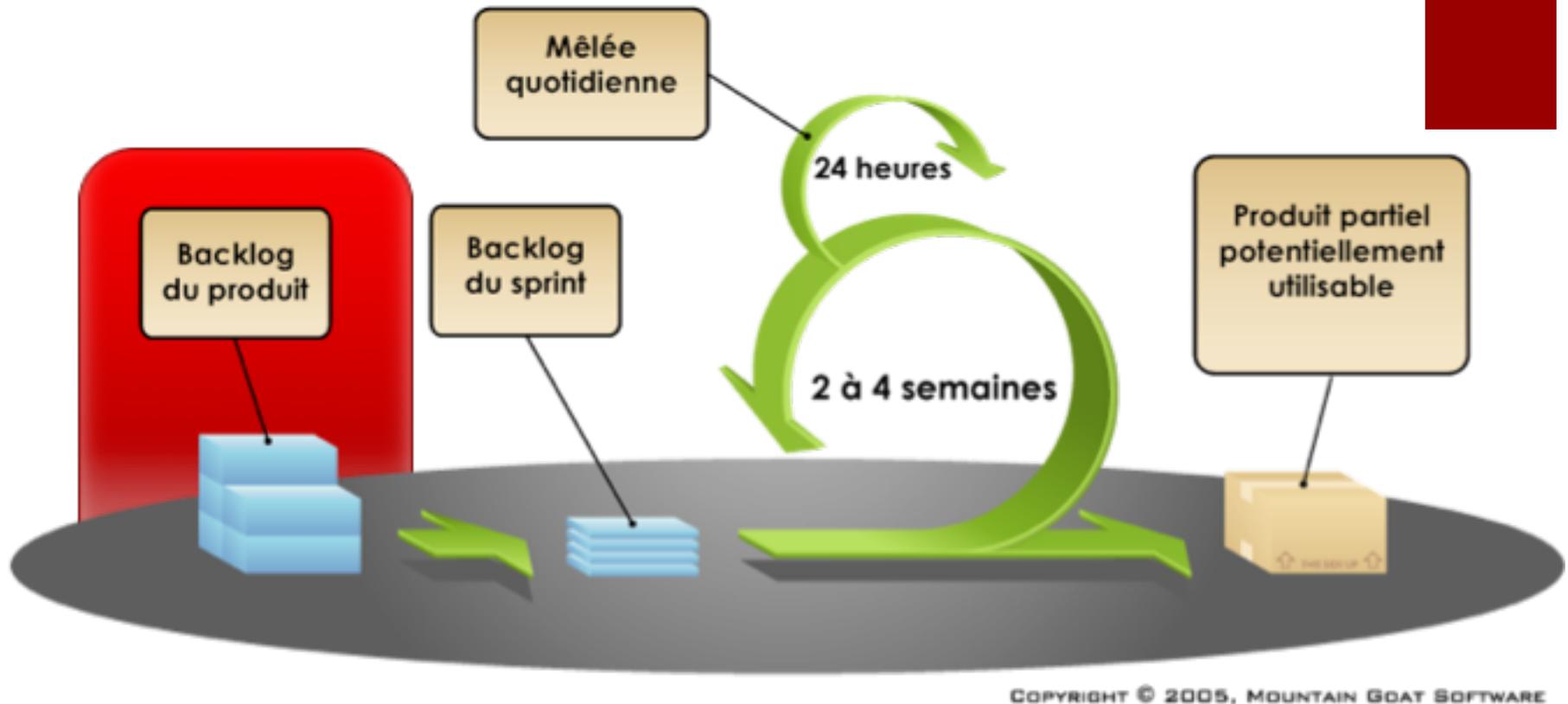


## 1. Définition de la vision du produit

- Le cap
- Une ou deux phrases

# Scrum – Cycle 2/6

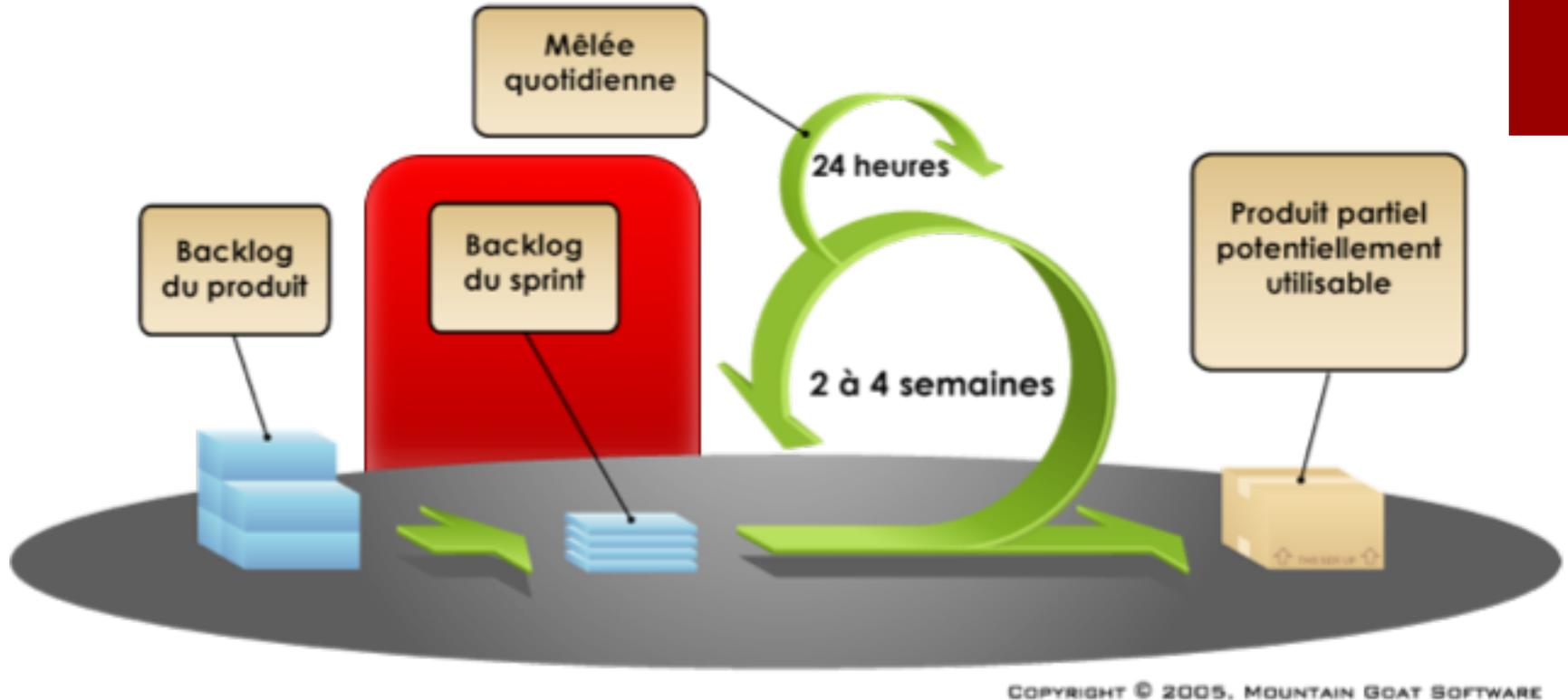
5



## 2. Backlog produit (ou carnet de produit, catalogue des besoins)

- Besoins priorisés par le product owner
- Besoins estimés par l'équipe, qui évoluent et sont affinés

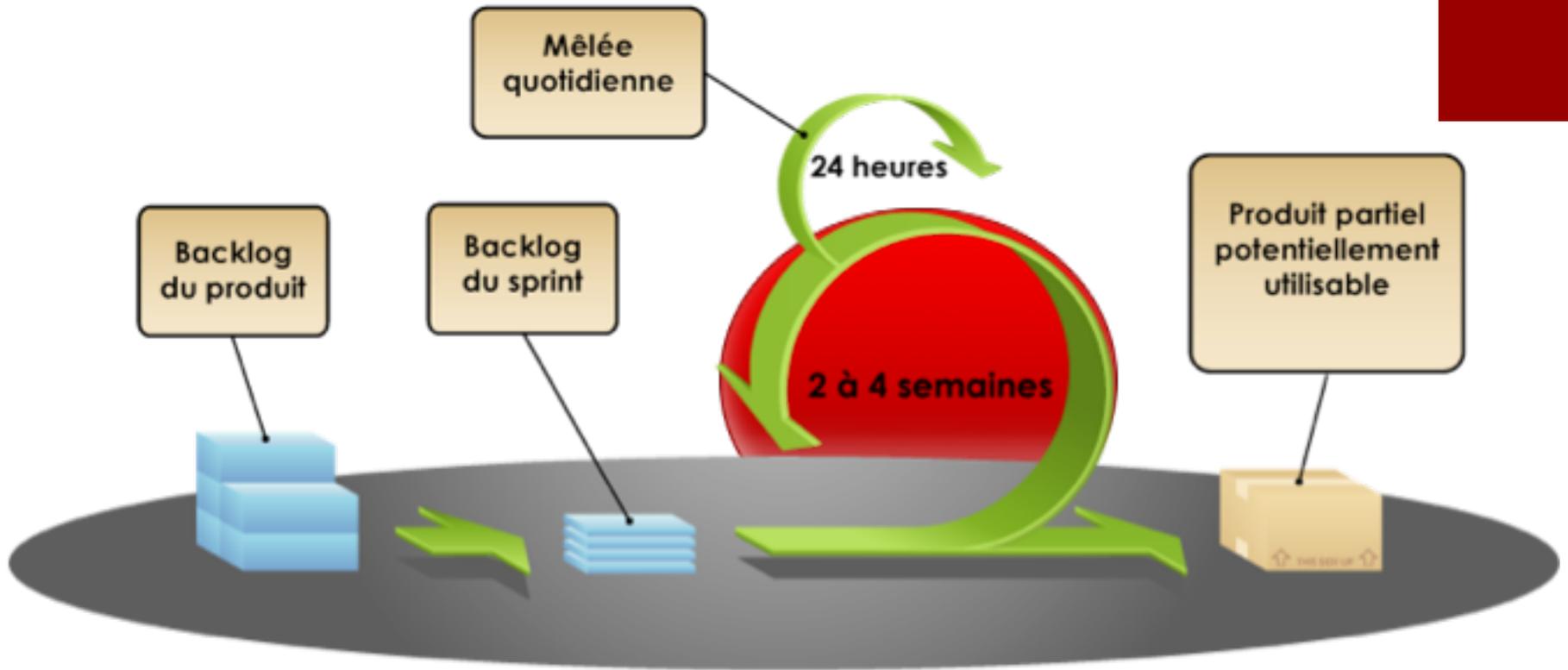
# Scrum – Cycle 3/6



## 3. Backlog de *sprint*

- **Sélection** des besoins à réaliser sur le sprint, extrait du backlog produit
- **Découpage** en tâches, répartition de l'effort, planification

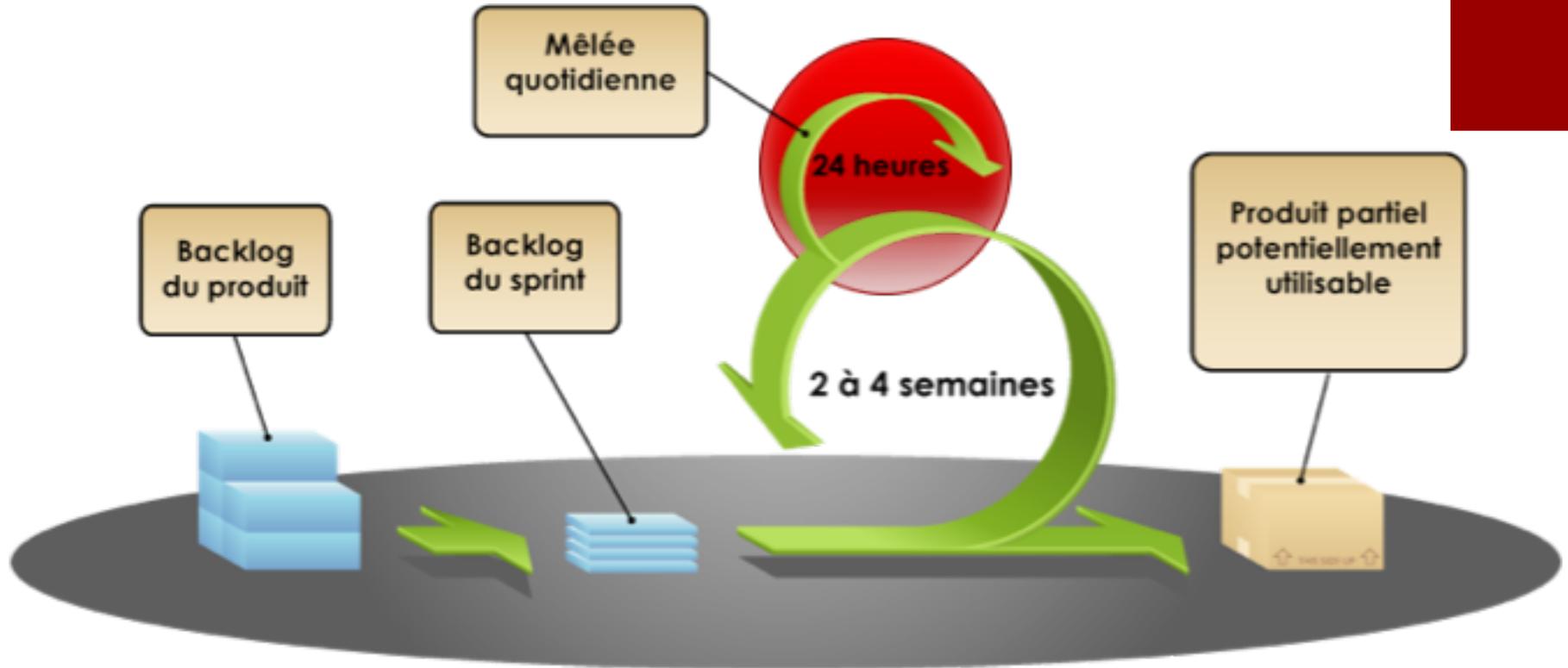
# Scrum – Cycle 4/6



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

## 4. Sprint

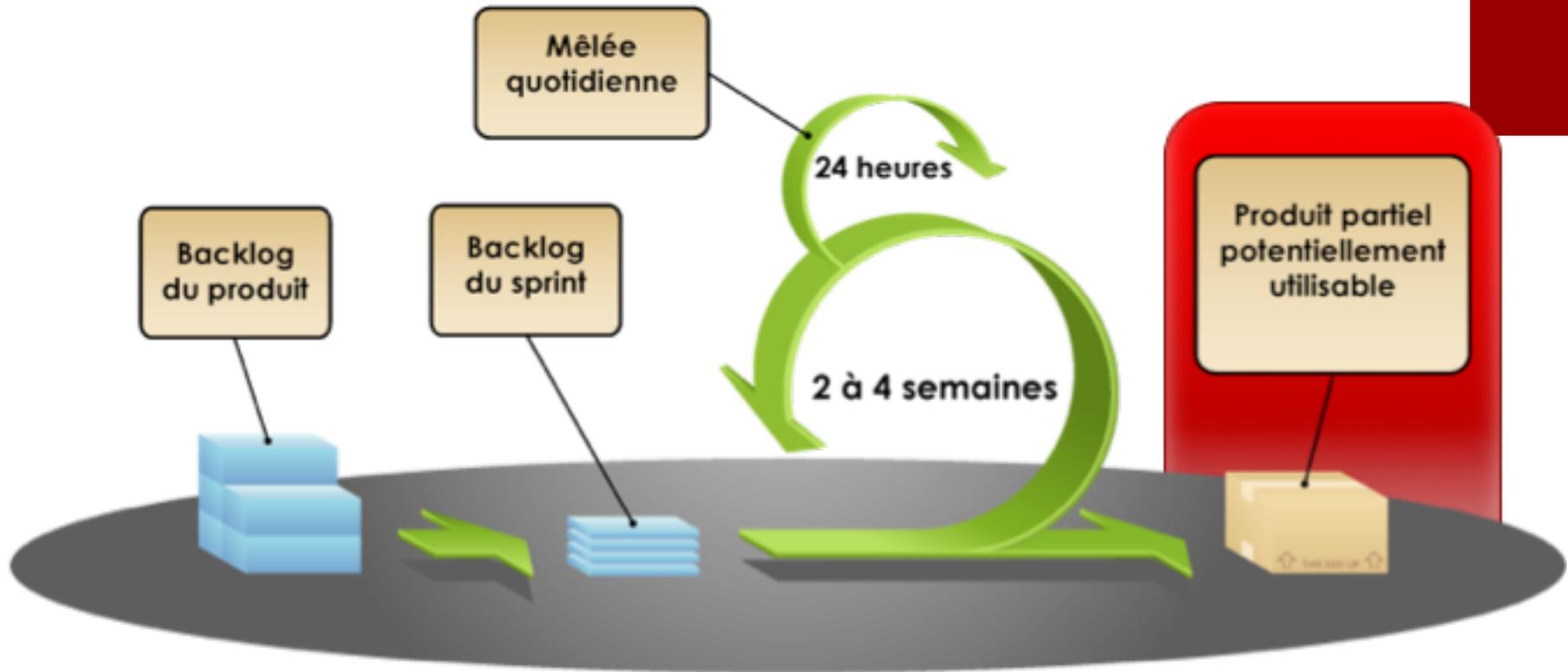
- Développement des fonctionnalités du backlog de sprint
- Pas modification du backlog de sprint possible
- Affinage du backlog Produit : une fois par semaine



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

## 5. Mêlée quotidienne

- Point de contrôle quotidien de l'équipe
- Interventions régulées – 2 min. par personne



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

**6. Incrément logiciel** : livré au Product Owner à la fin du sprint.



# Caractéristiques Méthodes Agiles

10

- Les **itérations** sont bornées dans le temps
- Les **réunions** sont bornées dans le temps
- Tout est borné !
- Et donc le **coût est prévisible** et a moins de chance d'être dépassé



# ÉTAPE 0 : LA VISION

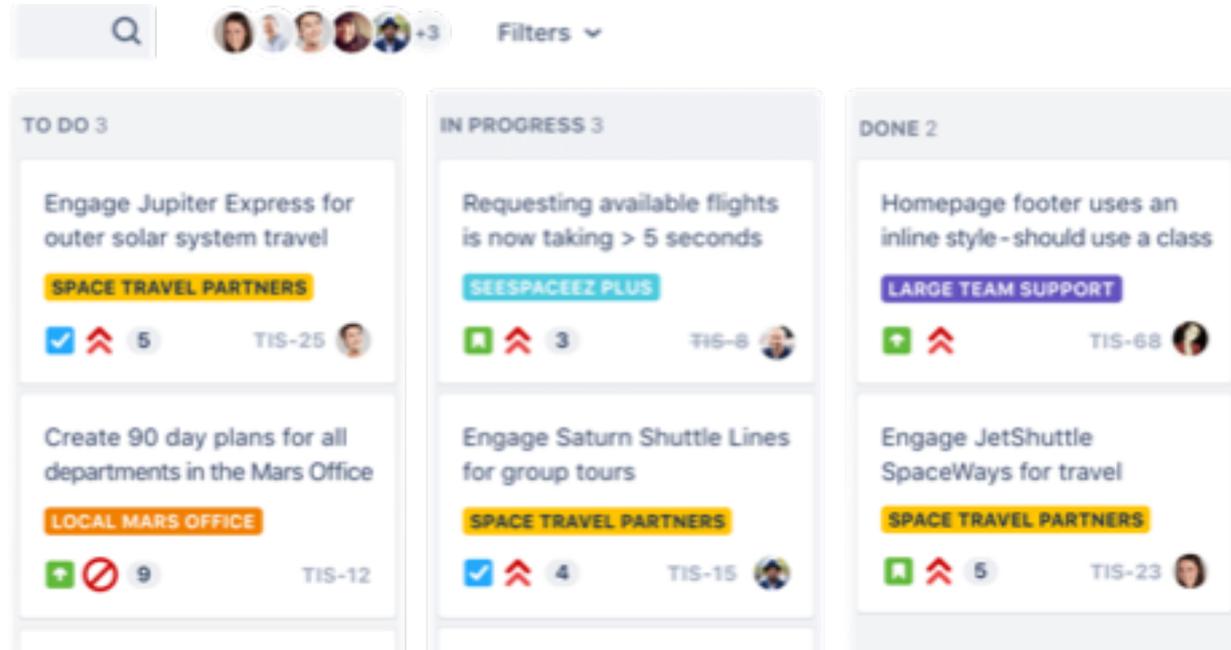


- Le cap : si vous ne savez pas où vous voulez aller, où irez-vous?
- Sans vision, les retards sont pris **en tout début** de projet ; l'équipe est alors hésitante, refusant de prendre **le risque de se tromper** de direction...
- On peut **adapter** le cap en fonction du feedback
- **Objectifs :**
  - situer le produit dans l'organisation
  - lui donner un sens (raison d'être du produit)

# Exemple de vision

Dans un contexte de SAV, pour un système de **gestion de tickets** :

Ici, JIRA



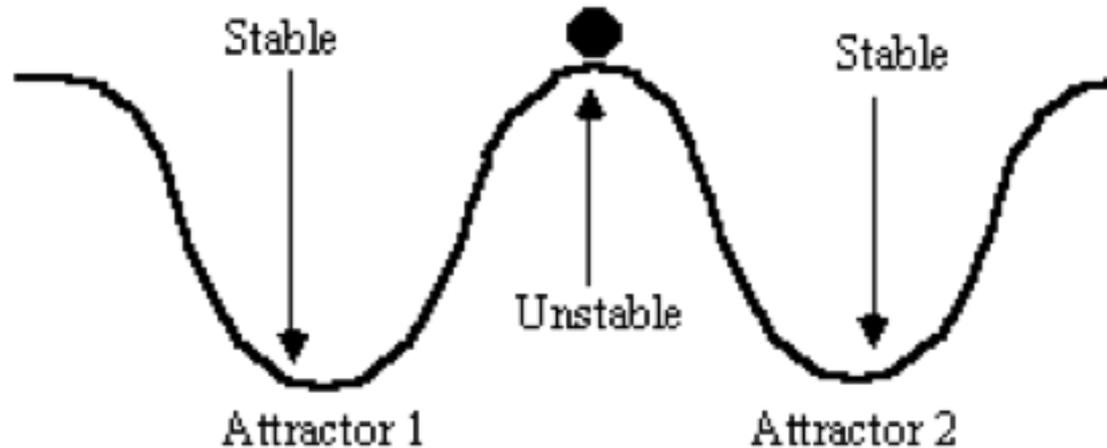
« Le système permettra de voir d'un seul coup d'oeil les tâches à faire, avec l'urgence, le demandeur (client), qui le prend en charge, la date du ticket, l'état d'avancement du ticket »

# Vision : qui fait quoi ?

Définir la vision : le RACI

	<b>Product Owner</b>	<b>Utilisateurs, Manager</b>	<b>Développeurs</b>
<b>Autorité</b>	X		
<b>Rédige</b>	X	X	
<b>Conseille</b>			X
<b>Est Informé</b>		X	X

# Vision = un attracteur



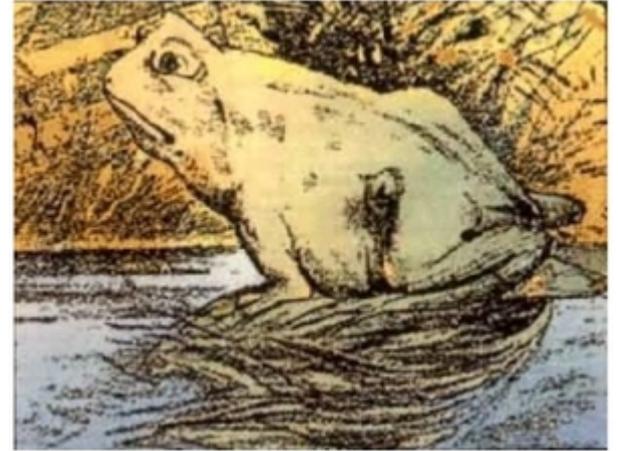
- Selon la théorie de l'auto organisation, la vision est un **attracteur** de tous les éléments épars qui participent à réaliser une activité
- Elle réunit tous les points de vue vers un **objectif commun** connu de tous
  - Même de façon inconsciente

# Ex. d'attracteur visuel

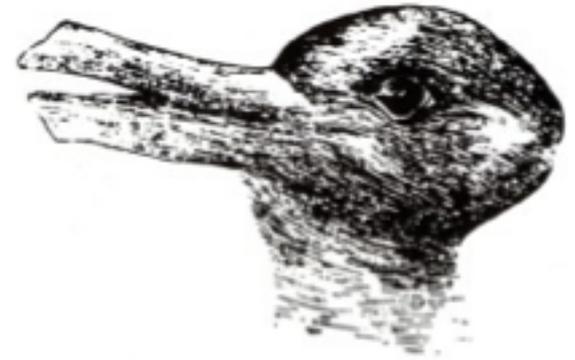
- Que voyez-vous sur cette image ?



- Et sur celle-ci ?



- Et celle-là ?



# Un attracteur visuel : la mémoire



- Une fois qu'un objet a été identifié par le cerveau, les stimuli visuels activent la mémoire
  - l'ensemble des stimuli s'organise alors par rapport à cet attracteur
- Un effort est nécessaire pour tenter de voir l'autre image
- La **vision Produit** crée cet attracteur au sein de l'équipe, pour toutes les phases :
  - Planification
  - Choix des Solutions techniques
  - Codage
  - Améliorations

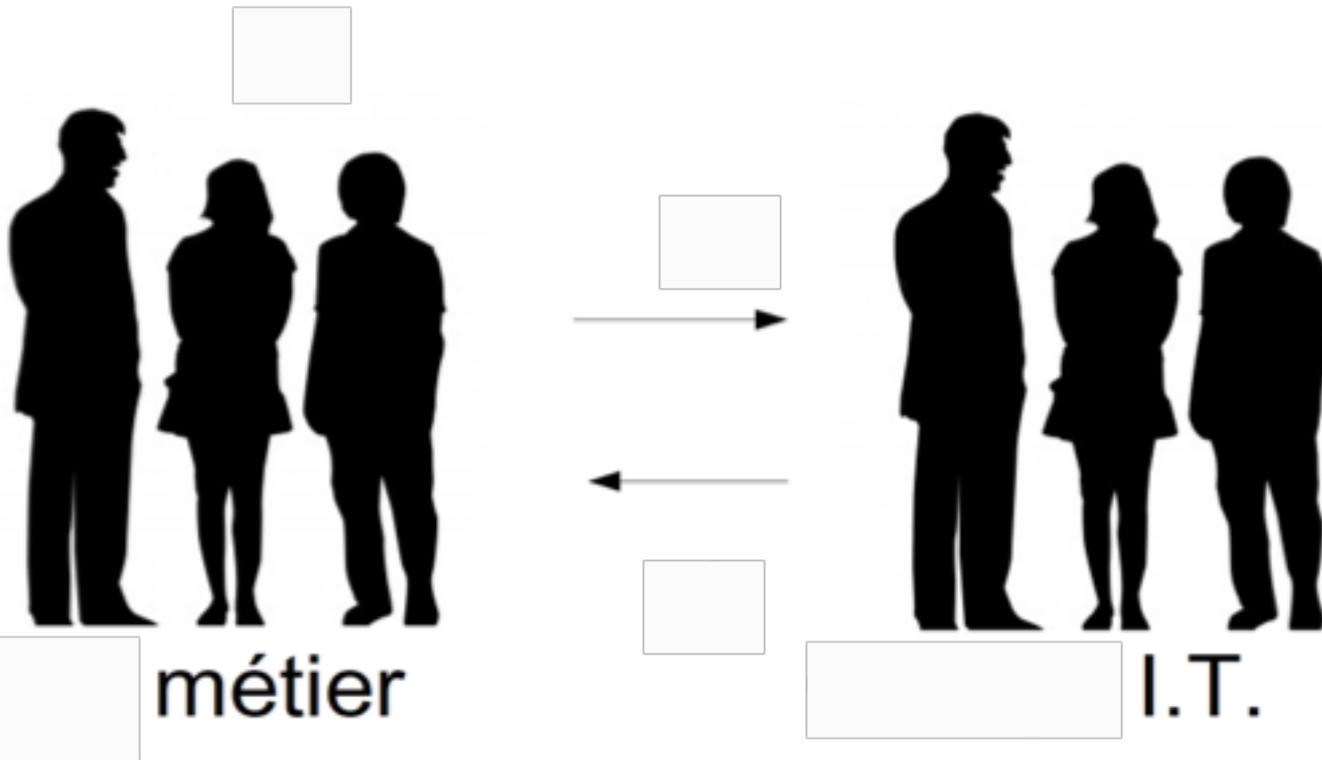
# EN SAVOIR PLUS sur la vision ...

Créer une vision pour un nouveau système d'écoute  
musicale en ligne (UK, 9')

<https://www.youtube.com/watch?v=6h3avYCgh9Q>

# Backlog Produit

## la CAPTURE des EXIGENCES



# Backlog Produit (Carnet de Produit)

- Contient des **User Stories (Scrum)**
- **DEF User Story**
  - Une demande **fonctionnelle** d'un utilisateur
  - Apporte de la **valeur business** au produit
  - Ecrite en langage **naturel**
- Exemple : « **En tant** que client, **je souhaite** pouvoir ajouter un produit dans mon panier **afin de** pouvoir l'acheter »

# Combien d'User Stories ?

- Au **max 20** US au début, pour respecter le vœux agile « peu de stock »
- Principe : « **être précis à court terme, grossier à long terme** »
- Attention : imprécision pour le **long terme** ne signifie pas “inutile”
  - Cela donne “le cap” de la *release*
  - On ne les oubliera pas plus tard
  - Permet d'estimer le « reste à faire »
- Une US = un **résumé formaté**, pour avoir une vision rapide de la demande
- Elle sera **détaillée** ensuite avec l'équipe

# Granularité des US ?

- Une user story doit pouvoir être implémentée **en une itération**
- Une itération doit comporter **au moins 4 User Stories**



# Comment obtenir les US ?

- C'est le problème du Product Owner
- Méthodes possibles : **Brainstorming, définition de personnae, identification des scénarios utilisateur,...**
- Par exemple, pour un **logiciel de gestion de carnet d'adresses**, un scénario est :
  - *"Je rentre un nom ou prénom, et le logiciel affiche la liste de toutes les personnes qui possèdent ce nom ou ce prénom"*
  - *"Je peux choisir d'exporter mon carnet d'adresses au format HTML ou XML"*

# Description des US

## Notion de Fini

- Une user story pourra contenir : une description, des critères d'acceptations, des **tests d'acceptation**, un résumé simple, des règles de gestion,...
- Les tests d'acceptation permettent de **définir le DONE**, càd. expliciter *quand* une US est considérée comme **terminée** :
  - « ça marche à peu près » : ???
  - « c'est OK pour 80% des data »
  - « c'est tout bon, mais la version Anglaise reste à faire »

# Format des User Stories

En tant que... (rôle)  
Je peux .... (tâche)  
Afin de .... (but)

**En tant que** Pilote,

**je peux** régler le commutateur en mode « horizontal »

**afin de** maintenir les ailes à l'horizontal pour que l'avion reste sur sa trajectoire

- Les user stories proposées par le PO sont **discutées avec l'équipe** pour lever toute ambiguïté de compréhension

# Exemples de User Story

**ETQ** Assureur, **je peux** récupérer des contrats d'assurance sur le site web **afin de** vérifier leur précision et leur légalité

**ETQ** Assureur, **je peux** chercher les infos du permis de conduire du candidat chez la préfecture concernée **afin de** vérifier leur exactitude

*Source : chaîne YouTube:  
BA-Experts  
(Business Analysis)*

# Règles pour écrire des US correctes

- **REGLE 1** : rester simple (principe KISS)
- **REGLE 2** : parler du **QUOI** (pas du COMMENT)
- **REGLE 3** : rester dans le **périmètre** du projet (vision!), et dans le **champ de responsabilités** de l'organisation / du service
- **REGLE 4** : lever l'**ambiguïté** des termes
- **REGLE 5** : pratiquer si possible la **réécriture** des règles



# Rester simple : principe KISS (Keep It Simple & Stupid)

- Voici une user story définie pour un logiciel d'assurance auto :

**En tant** qu'internaute,  
**je peux** naviguer sur le site, saisir mes informations personnelles et celles du véhicule, et soumettre une demande en ligne,  
**Afin** d'obtenir une couverture d'assurance automobile.

- Qu'en pensez-vous ?



# KISS sur un exemple

**En tant** qu'internaute,  
**je peux** naviguer sur le site,  
saisir mes informations  
personnelles et celles du  
véhicule, et soumettre une  
demande en ligne,  
**Afin** d'obtenir une  
couverture d'assurance  
automobile.

- Trop de 'choses à faire' mentionnées  
→ pas **simple**
- **REGLE : pas de User Story composée**
  - Éviter les **ET** (et **OU**)
    - pour 'je peux'
    - sur les données : données personnelles,  
données véhicules
  - Éviter les '**à moins que**', '**excepté pour**'...
- **Les scinder**



# KISS Assurance - réponse



- Ici on découpe en 3 User Stories :

- « **En tant** qu'internaute, **je peux** naviguer sur le site, **Afin** de choisir la couverture d'assurance automobile qui me convient »
- « **En tant** qu'internaute, **je peux** saisir mes informations personnelles et celles du véhicule, **Afin** de comparer les offres d'assurance automobile »
- « **En tant** qu'internaute, **je peux** soumettre une demande d'assurance automobile en ligne, **Afin** d'obtenir un contrat »



# Lever l'ambiguïté (Règle 4)

- Ex., un Responsable du Stock écrit la US :

**En tant** que Resp. du Stock,  
**je peux** commander la bonne  
quantité de produits que nous  
allons vendre  
**Afin** d'éviter d'avoir des coûts de  
stock trop élevés

- Qu'est-ce qui est ambigu ici ?
- « la **bonne** quantité » : quelle valeur ? Quelle unité ? (produit unitaire, palette, chargement camion,...)
- « coûts **trop** élevés » : idem, subjectif --> valeur seuil
- « nous **allons** vendre » : quand ? demain, la semaine prochaine ?



# Pratiquer la réécriture des US (Règle 5)

- Par un collègue, si possible très différent de soi
  - **Consigne : ne reprendre aucun nom / verbe identique**
  - **On discute des différences**
- Ex.: pour une agence de voyage, Tim écrit :



**En tant** qu'opérateur du Centre d'Appel,  
**je peux** saisir au moins 12 réservations par  
heure en période de forte activité  
**Afin** de réduire le temps d'attente des clients



Et Cynthia propose la réécriture suivante :

**En tant** qu'agent de voyage,  
**je suis capable** d'effectuer un minimum de  
12 demandes de voyages en 60 min durant  
la partie de l'année la plus chargée  
**Afin** de minimiser les abandons  
téléphoniques

**Votre avis ?**

# Réécriture – analyse sur l'ex.(1)

**En tant** qu'opérateur du Centre d'Appel,  
**je peux** saisir au moins 12 réservations par  
heure en période de forte activité  
**Afin** de réduire le temps d'attente des clients

**En tant** qu'agent de voyage,  
**je suis capable** d'effectuer un minimum de  
12 demandes de voyages en 60 min durant  
la partie de l'année la plus chargée  
**Afin** de minimiser les abandons  
téléphoniques

- « opérateur du Centre d'Appel » ou « agent de voyage » ?  
→ *préférer le rôle (agent de voyage) plutôt que la fonction*
- « saisir » ou « effectuer » ? → *'effectuer' est trop vague, à éviter !*
- « 12 réservations » ou « 12 demandes » ? → *préférer les termes métiers, réservation*

# Réécriture – analyse sur l'ex. (2)

**En tant** qu'opérateur du Centre d'Appel,  
**je peux** saisir au moins 12 réservations par  
heure en période de forte activité  
**Afin** de réduire le temps d'attente des clients

**En tant** qu'agent de voyage,  
**je suis capable** d'effectuer un minimum de  
12 demandes de voyages en 60 min durant  
la partie de l'année la plus chargée  
**Afin** de minimiser les abandons  
téléphoniques

- « en période de forte activité » signifiait pour TIM, 'dans la journée'
- Mais Cynthia introduit la **notion de saisonnalité** qui est intéressante : en période de vacances, familles → réservations plus longues ! Ajouter une précision

# Réécriture – analyse sur l'ex. (3)

**En tant** qu'opérateur du Centre d'Appel,  
**je peux** saisir au moins 12 réservations par  
heure en période de forte activité  
**Afin** de *réduire le temps d'attente des clients*

**En tant** qu'agent de voyage,  
**je suis capable** d'effectuer un minimum de  
12 demandes de voyages en 60 min durant  
la partie de l'année la plus chargée  
**Afin** de *minimiser les abandons  
téléphoniques*

« réduire le temps d'attente »

ou « minimiser les abandons tél. » ?

→ préciser les attentes Métier

Ici : minimiser les abandons

## Réécriture – fin (4)

- Nouvelle proposition pour cette US :



**En tant** qu'agent de voyage,  
**je peux** saisir au moins 12 réservations par  
heure pendant le pic d'appels de la journée  
(hors vacances)

**Afin** de réduire les abandons téléphoniques  
dus à une attente trop longue

# Granularité du besoin

## ■ Vocabulaire SCRUM

- « **Feature** » : grosse fonctionnalité Produit
- « **Epic** » : grosse story
- « **User Story** » : besoin élémentaire d'un Utilisateur
- « **Task** » : tâche technique pour faire la US

# BACKLOG de PRODUIT: Je retiens....



- Les **features** (fonctionnalités) sont déclinées en **epic**, elles-mêmes en **user stories**, et les **US** en **tâches**
- Les *features* sont réparties sur les différentes **versions** du produit

## RAPPELEZ-VOUS

- Une user story doit être implémentée **sur une seule itération**
- Un sprint doit comporter **au moins 4 US**
- J'applique les principes **KISS** pour écrire les US
- La méthode de **Réécriture** permet de **valider** mes US

# EN SAVOIR PLUS sur le découpage agile

Comment disposer de plusieurs user stories alors que le client n'a exprimé qu'un seul besoin ?

<http://coach-agile.com/2014/09/coach-agile-decouper-ses-user-stories/>