

Cours 3

Développement agile avec SCRUM

**Taskboard,
Revue de sprint et Rétrospective**

V. Deslandres © – IUT de LYON



Sommaire

Management visuel, cérémonies SCRUM

- TaskBoard
- Daily Meeting – 11
- Fin de sprint (revue + rétrospective) – 15

Management visuel

Le task board

Comment rapidement voir
l'avancement

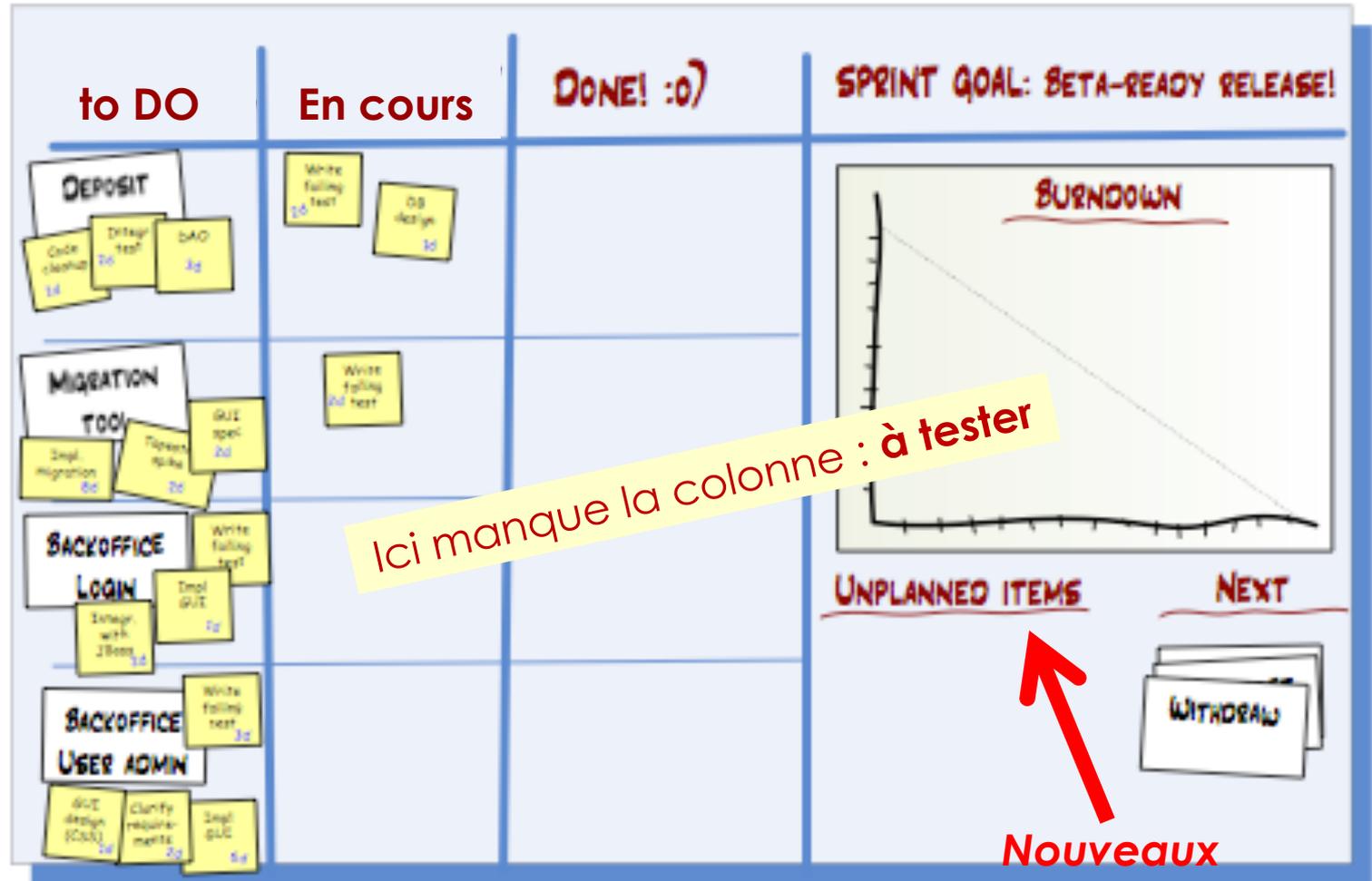
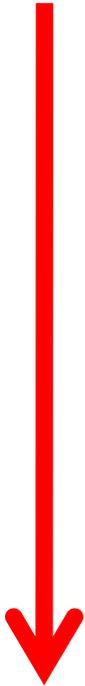
Task Board SCRUM

- Tableau de suivi de l'avancement
- On y place les US (post-it) dans les **colonnes KANBAN** :
 - **A FAIRE**
 - **EN COURS** on peut y ajouter des US supplémentaires
(un tas « US non planifiées »)
 - **À TESTER**
 - **FINIES**

Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Test the... 8	Code the... DC 4	Test the... SC 6	Code the... DC 8 Test the... SC 8 Test the... SC 6 Test the... SC 6
As a user, I... 5 points	Code the... 8 Test the... 8	Code the... DC 8		Test the... SC 8 Test the... SC 6 Test the... SC 6

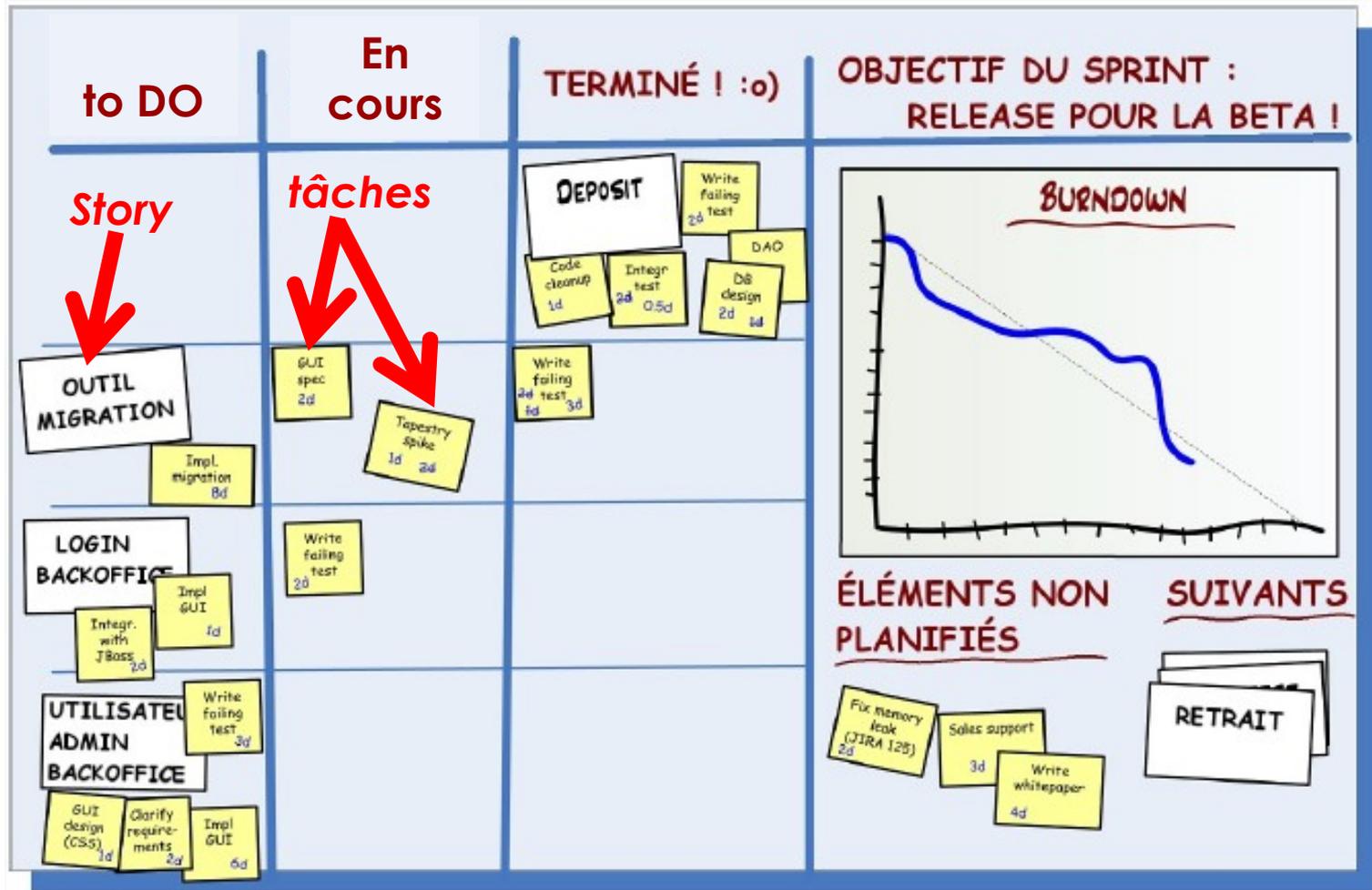
Un ex. de TaskBoard – Jour 1

Priorité
décroissante



D'après Erik Kniberg, Scrum & XP from the Trenches

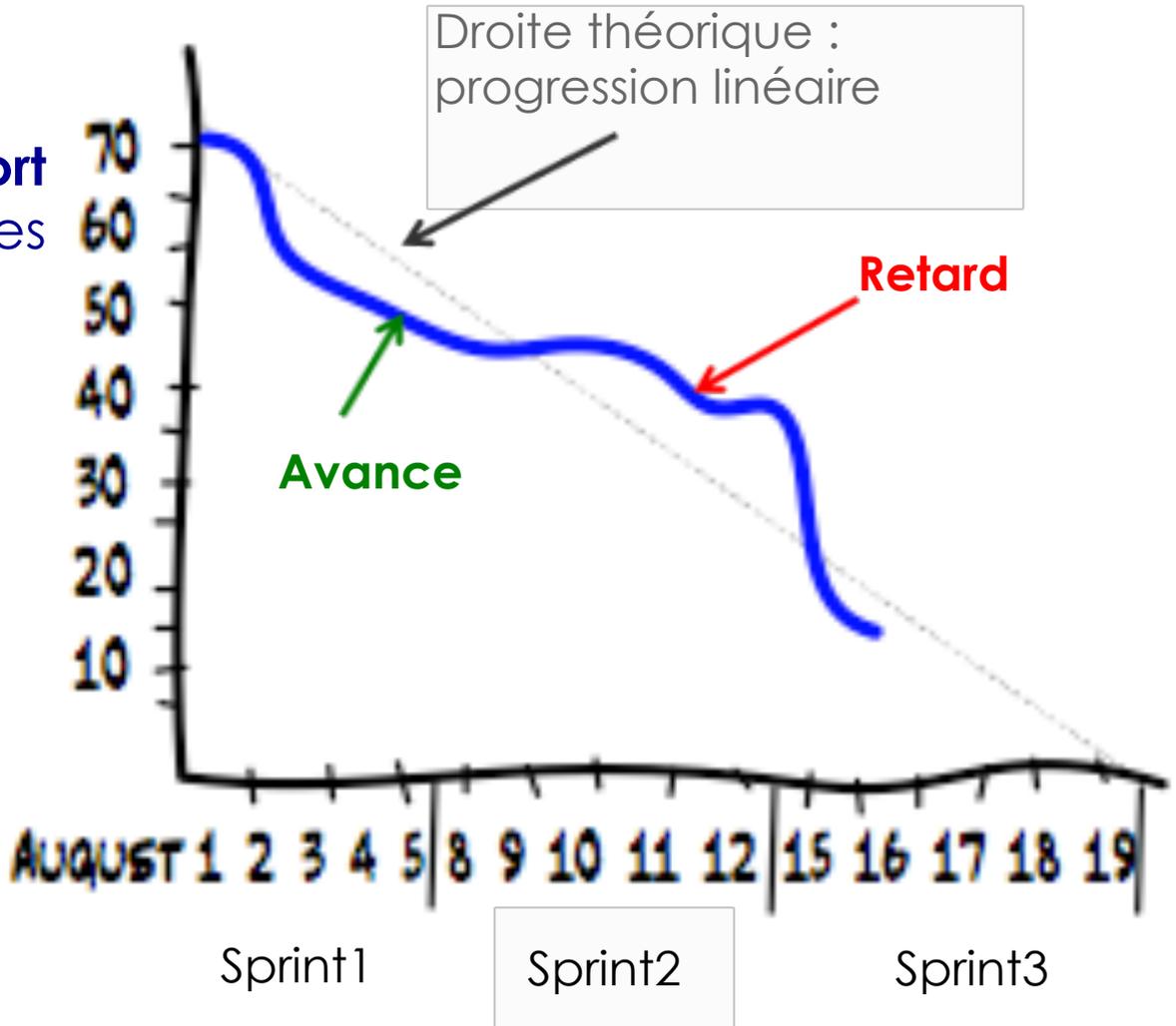
Task Board – Jour *n*



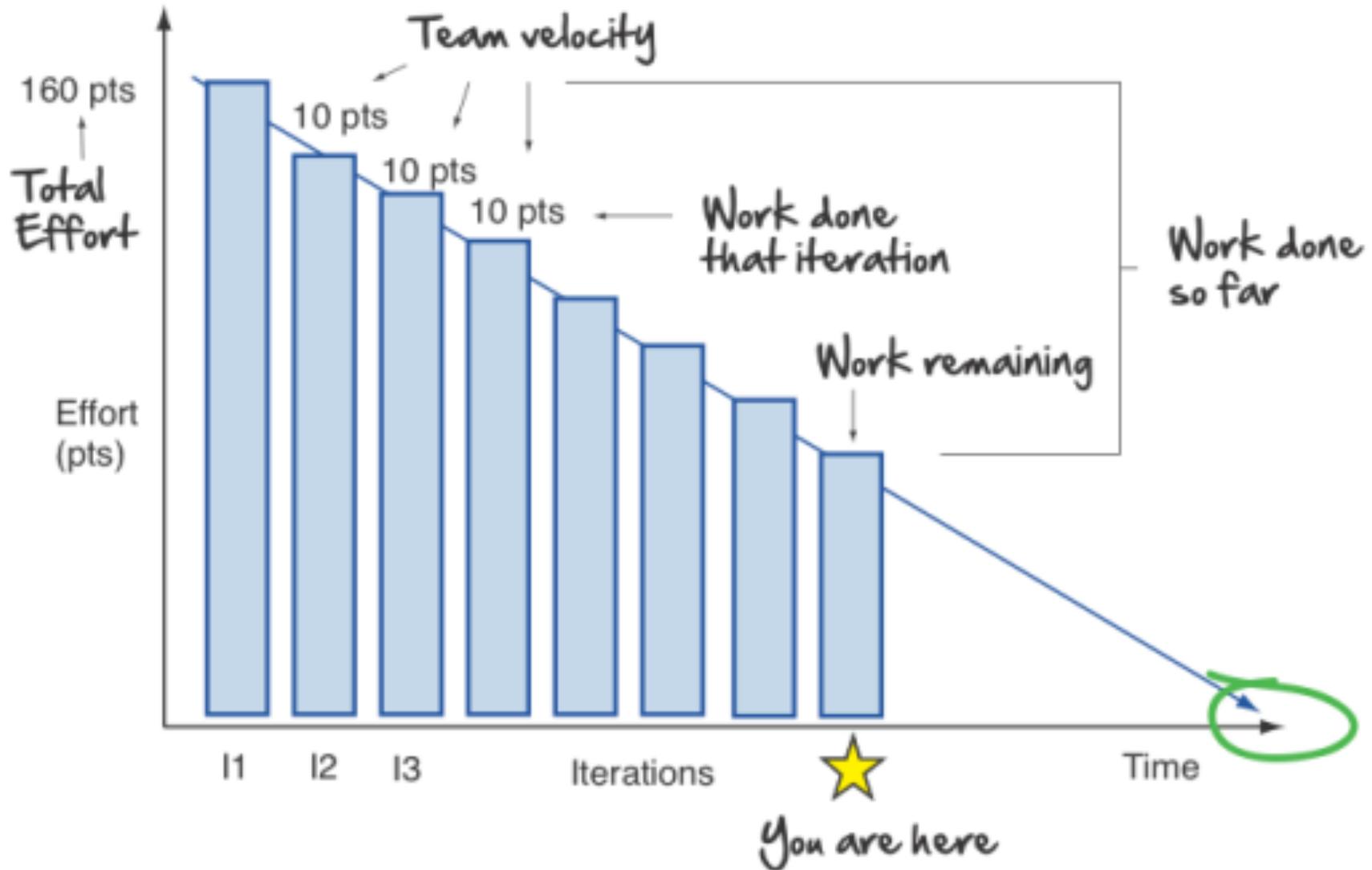
D'après Erik Kniberg, *Scrum & XP from the Trenches*

Burndown Chart (graphique d'avancement)

Points d'effort nécessaires



Dates de fin d'itération



Informations fournies par le burndown chart :

9

- La **vélocité** de l'équipe (= le travail **effectué** à chaque itération, nb de points d'effort validés)
- Le travail **restant**
- La quantité de **travail déjà livré**
- Quand on peut s'attendre à avoir **x% fini**

Le tout sur un graphique 😊

Vidéo (3') : <http://www.agilenutshell.com/episodes/6-burndown-charts>

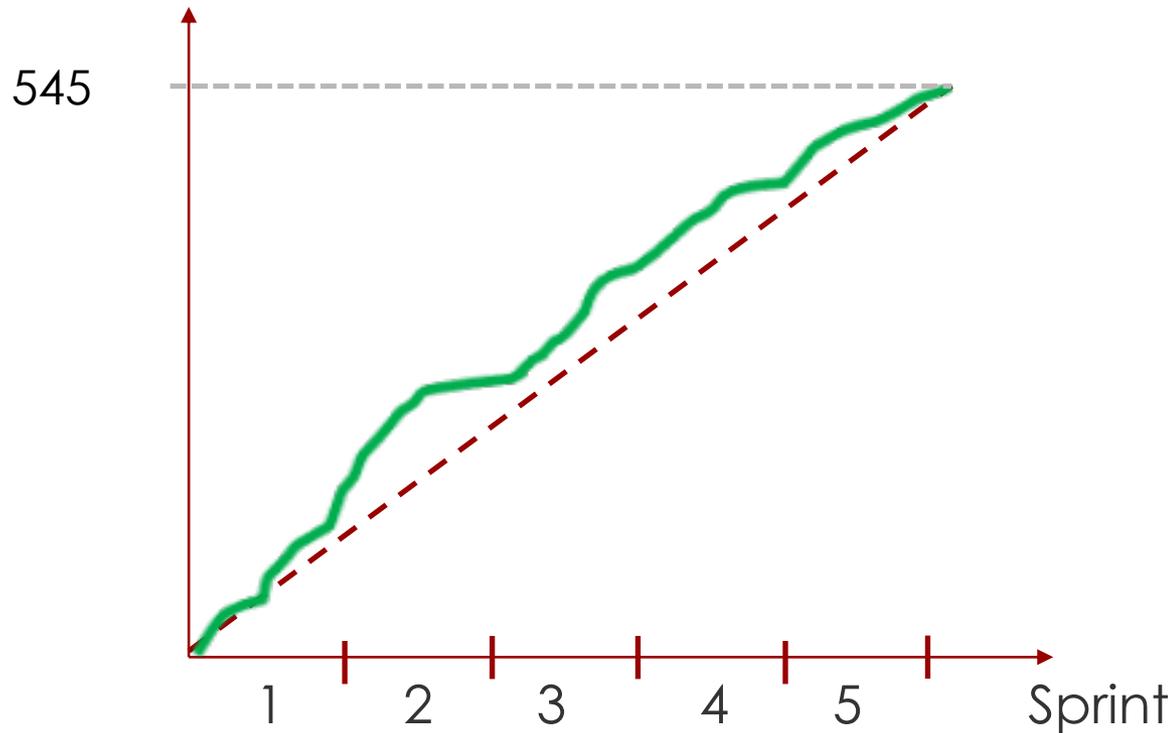
Avec Excel : <https://www.youtube.com/watch?v=gjU5vA8m1Jo>

NOTA : il existe le burnUP chart, qui affiche les points efforts effectués plutôt que les points d'effort restant

Un BurnUp Chart

10

Nb points d'efforts réalisés



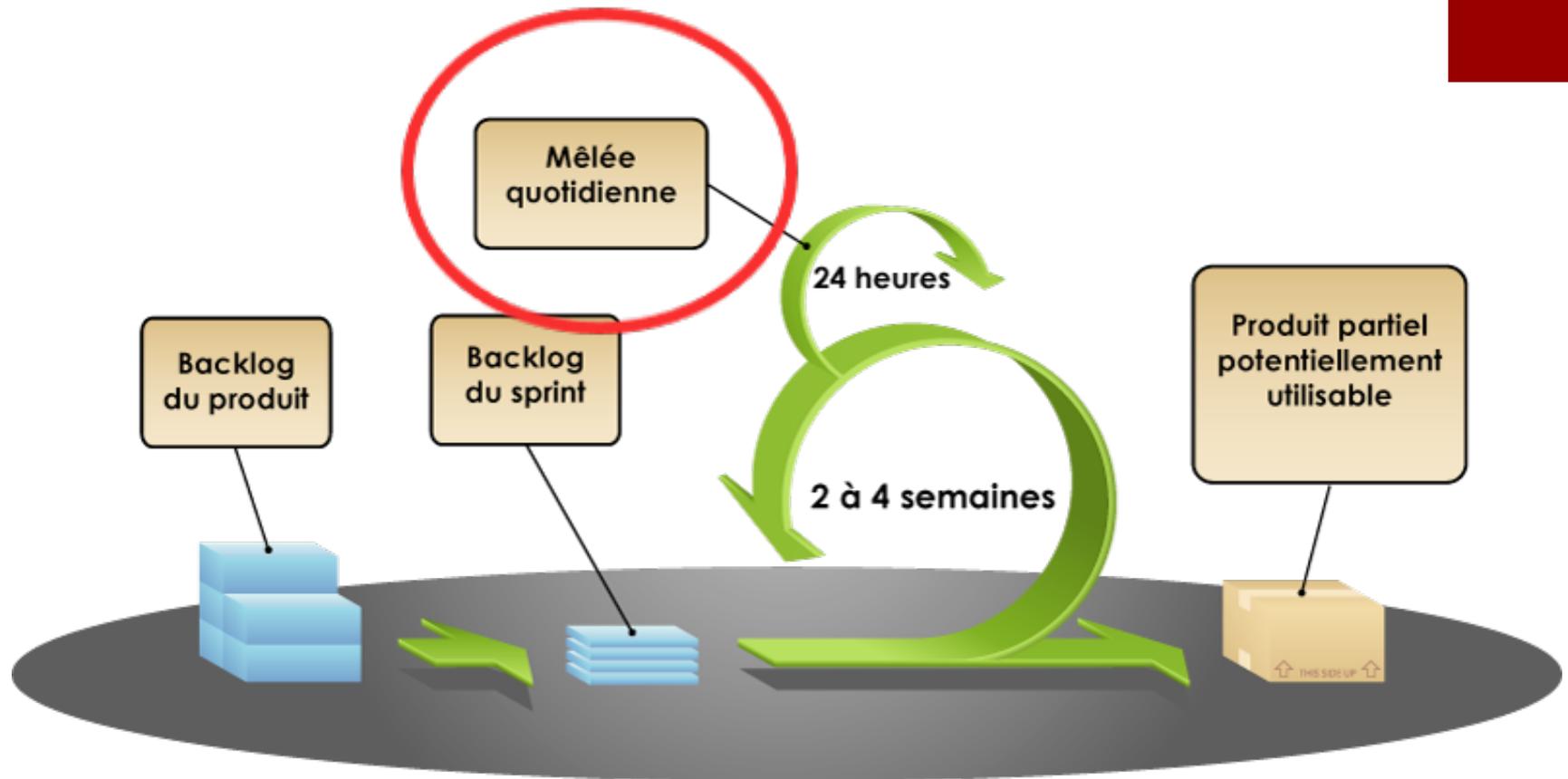


Daily meeting

Chacun fait le point

Chaque jour de l'itération, faire le point sur son avancement

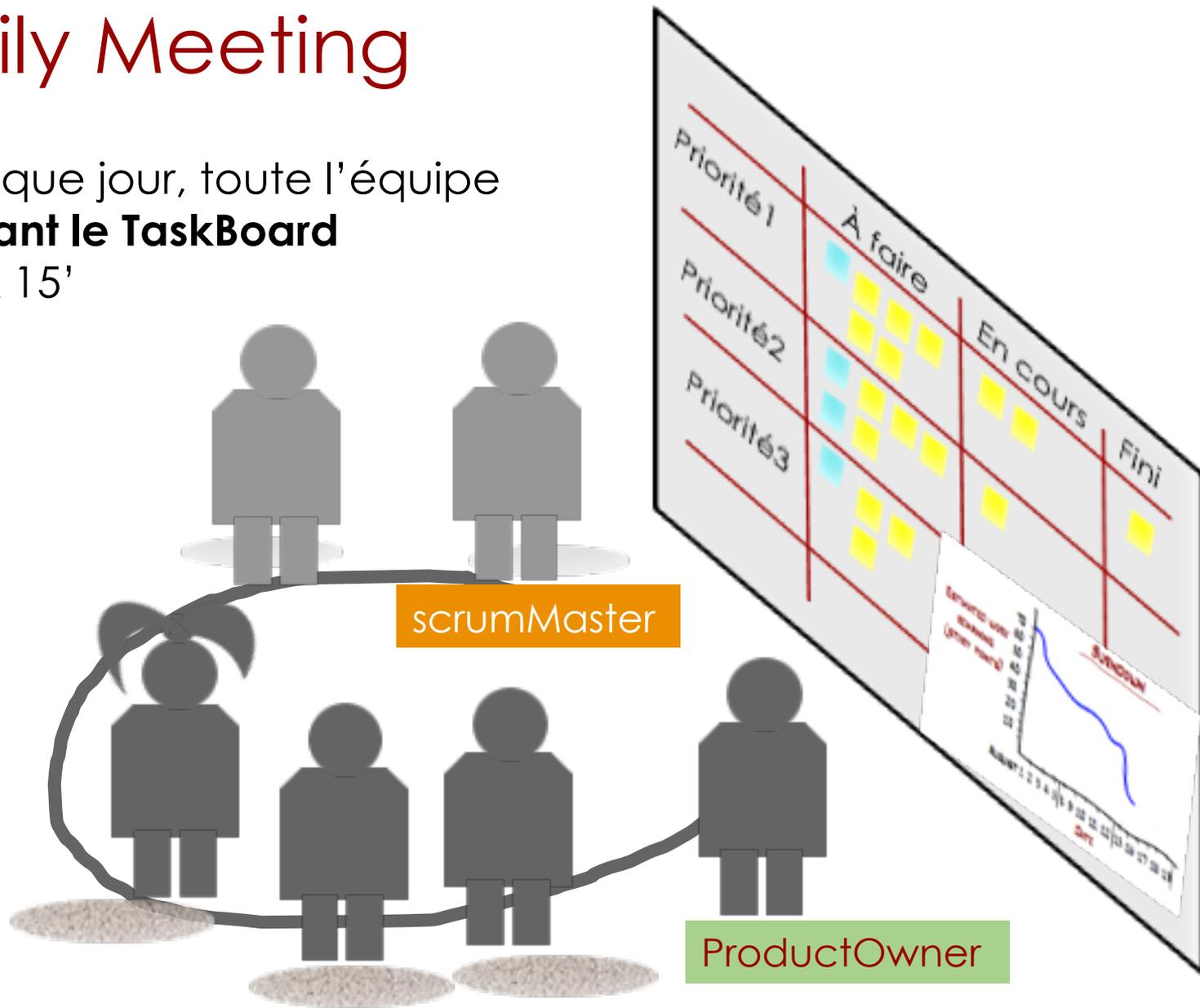
12



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Daily Meeting

Chaque jour, toute l'équipe
devant le TaskBoard
max 15'



Daily Meeting : 3 questions

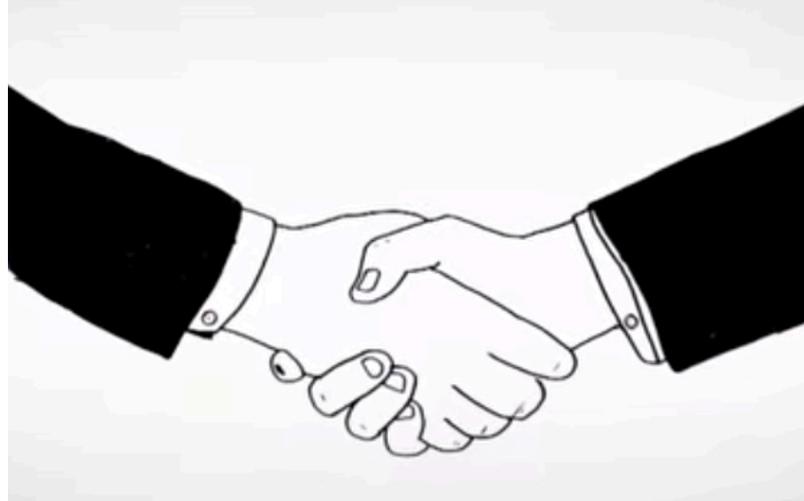
- Chaque développeur s'exprime (max 2') :

Qu'est-ce que **j'ai fait** hier ?

Qu'est-ce que **compte faire** aujourd'hui ?

Quelles sont mes **difficultés** / points bloquants ?

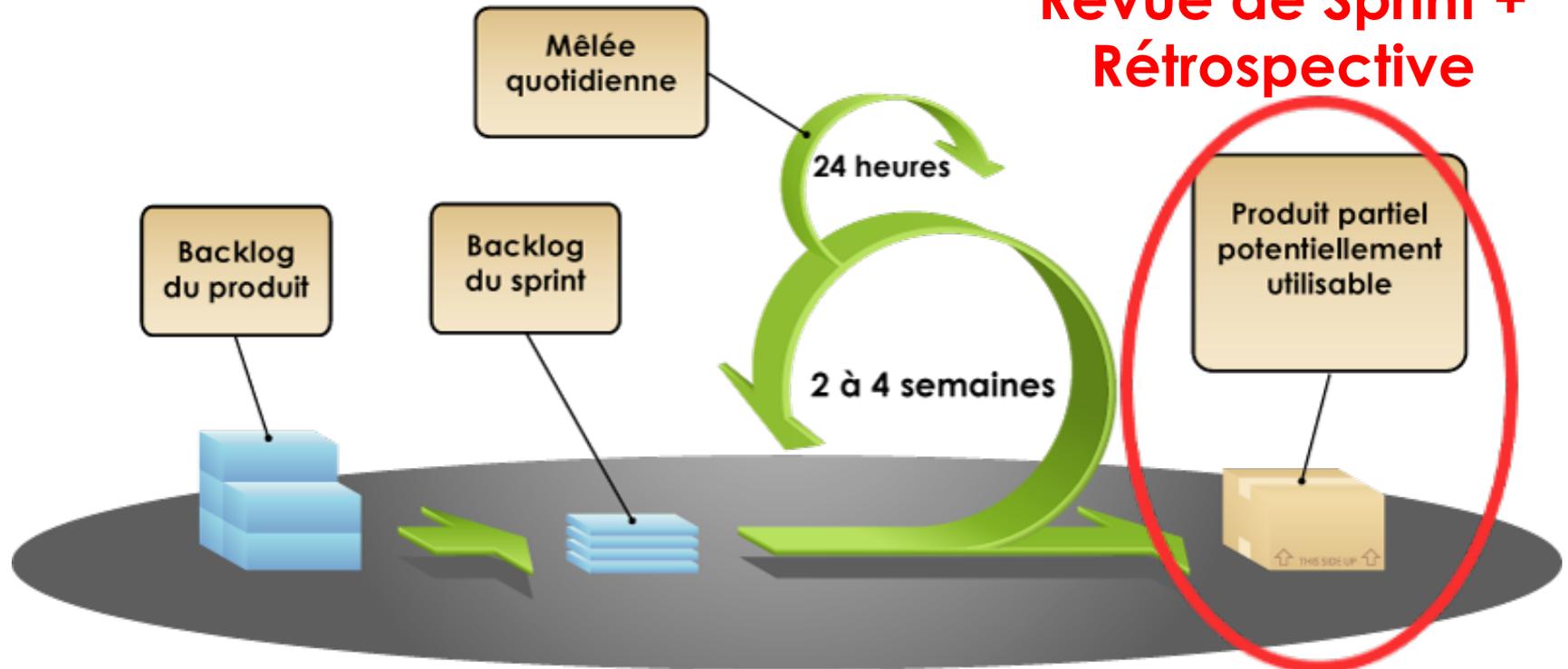
En fin de sprint...



Les cérémonies Scrum

Fin de sprint : 2 cérémonies

Revue de Sprint + Rétrospective



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE



La revue de sprint

Démo,
Feedback,
Actualisation du Backlog

Revue de Sprint

- Dernier jour du sprint : **Démonstration des livrables**
- On invite **toutes les personnes** concernées (parties prenantes)
 - Des **décisions** peuvent être prises sur l'avenir du produit
- Pas de slides, max 2h pour un sprint de 2 sem.
- **Déroulement** :
 - Rappel des objectifs du sprint, les US prévues et celles montrées
 - Démonstration, si possible en suivant un **scénario utilisateur**
 - Collecter le **feedback**, discuter
 - **enrichissement du backlog produit**
 - Calculer la **vélocité** (seules les US validées sont comptées)
 - **Affiner** le backlog

Revue Sprint : bonnes pratiques

- C'est un **résumé du travail accompli**
 - Comme un CR de match de rugby : on ne parle que des essais réussis
 - Ne montrer **que ce qui est fini**
- Pour la démo, laisser **la parole à un binôme** composé du SM et un développeur, pas plus
- Jeux de données, tests : la démo doit demander un **minimum de préparation** (max 1h)
- Pas de CR en général

+ discussions, (re)**définition du « fini »**



La rétrospective

Amélioration continue

Rétrospective : bilan du fonctionnement de l'équipe

- **Après** la revue de sprint ; dure en général 1h
- Agile = développement itératif et incrémental. Le **feedback interne** est essentiel pour l'amélioration continue
- Ne concerne que **l'équipe (dév + SM + PO)**
 - (Pas les managers, les commerciaux, etc.)
- Qu'est-ce qui a **bien marché** ? → Capitaliser
- Ce qui était **moins bien** ? → Ne pas reproduire des erreurs

- A chaque fin d'itération, il s'agit de trouver au minimum **UNE action d'amélioration**
 - Action **réalisable** et **mesurable**
 - Par ex. *faire une revue de code systématique* avant de passer au statut de 'Terminé'
 - Passer en *pair programming*
 - Cohérente avec son périmètre de contrôle
 - **Ex. mauvaise action** : « imposer aux designers d'accélérer leurs livraisons »
- On **analysera l'effet de l'action** sur le sprint suivant :

Amélioration continue

Rétrospective : les Méthodes

MAD, SAD, GLAD

- Chaque membre de l'équipe se remémore et écrit des **évènements** qui ont eu lieu et qui les ont marqués (post-it, 15')
- Chacun les place ensuite au tableau dans les colonnes :
 - **Mad**: grave, ça rend fou (avis <0) ex. « Description des tâches JIRA pas claires »
 - **Sad** : désolant, par ex.: « Tjrs en attente de validation d'accès partagé au répertoire XXX »
 - **Glad** : fierté, bonheur, ex.: « Optimisation du rendu effectué plus tôt que prévu »

Méthodes pour la Rétrospective (2) : les « 5 pourquoi »

- Brainstorming pour rechercher les causes indirectes d'un problème
- Principe : remonter l'arbre des causes sur 5 niveaux

Un exemple des 5P

Problème = « Fuite de liquide de coupe au pied d'une machine outil »

- **Why ? le liquide au sol provient-il bien d'une fuite de la machine ?**
 - Oui, la fuite provient d'un joint endommagé
- **Why ? Le joint a été endommagé par usure ou par accident ?**
 - Par accident, on découvre que des copeaux présents dans le bac de liquide de coupe ont sans doute endommagé le joint
- **Why ? Pourquoi ces copeaux sont-ils présents dans le bac ?**
 - La grille sensée retenir les copeaux est cassée
- **Why ? Pourquoi cette grille est cassée ?**
 - Les opérateurs reconnaissent que de temps à autre des outils tombent sur la grille, qui n'est pas protégée
- **Why ? Pourquoi n'est-elle pas protégée ?**
 - Ça a été conçu comme ça



Si on s'était arrêté ici en changeant le joint, le problème serait survenu à nouveau !

Rétrospective (3) : vote anonyme

- Pour certains points, il est délicat pour les membres de **s'exprimer ouvertement**
 - Ex. dénoncer les faiblesses techniques de certains membres, au contraire féliciter qq'un pour son comportement/travail
- On peut alors définir une échelle d'évaluation :
 - 1 = très mauvais / 4 = très bon
 - (jamais un nb impair !)
- Pour un sujet donné (par ex. la qualité de code de l'équipe), demander le vote anonyme sur post-it, récolté dans un chapeau
- Faire la moyenne : on obtient le ressenti de l'équipe sur ce point délicat, **qui est nécessaire pour progresser**



Vidéos Rétrospective

- La vidéo des Grenoblois (FR, rigolos, 4'30) :
<https://www.youtube.com/watch?v=Qz00T8FiEyM>

Quizz Je retiens...

28

- Le **taskboard SCRUM** comporte les colonnes KANBAN et une courbe
 - Noms des colonnes ?
 - Nom de la courbe ? Que représente-t-elle ? Son objectif ?
- Le **Daily Meeting**
 - Fréquence, durée ?
 - Les 3 questions posées
- Les **2 cérémonies de fin de sprint** : leur nom ?
 - Durée ? Audience ? Objectif ?
 - Citer 2 méthodes pour la 2^{ème} cérémonie

Application au mini-projet

29

- Faites **les 2 cérémonies** de fin de sprint
 - Se chronométrer !
 - Rétro : définir une **action** pour le Sprint 2
- Faites un affinage du Backlog
- **Tout noter dans votre taskBoard pour le Sprint#1**
- **Démarrer le Sprint#2**
- Rappel durées des étapes de l'itération ici :

1. Planification de Sprint et découpage en tâches (15')
2. **Réalisation des tâches / Codage (1h15)**
3. Revue de sprint (15')
4. Rétrospective (10')
5. Affinage du backlog (10')