



**NormandyJUG**

# **Scrum et l'agilité des équipes de développement**

**Par Dimitri Baeli & Nicolas Giard**

**23 Février 2010**



# Présentation des intervenants

**Dimitri Baeli**

<http://twitter.com/dbaeli>

VP Quality

Enterprise Scrum Master  
chez eXo platform

<http://www.exoplatform.com>

Addicted golfer :p



**Nicolas Giard**

<http://www.noocodecommit.com>

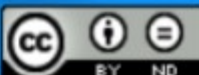
<http://twitter.com/nicogiard>

Consultant NTIC chez Team Partners

Co-fondateur du NormandyJUG

<http://www.normandyjug.org>

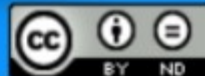
Spécialiste [Apache Wicket](#)





# Sommaire

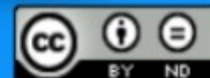
- **L'esprit Agile**
- Scrum en 5 min
- Revue pratique & outils
- Retours d'expérience
- Débat : comment commencer ?





# L'esprit Agile

- Avant l'agile :
  - Trop d'échecs en développement classique
  - Besoin d'améliorer le développement logiciel
  - Sa propre méthode (pas BTP, pas Industrie)





# L'esprit Agile

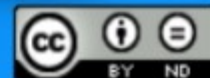
## Développement logiciel

- Spécifications, Développement, Tests, Livraison
- MOA, MOE, Equipe de développement, Sysadmins
- Comité de pilotage, responsable projet, ...

## Méthodes de développements :

- Classiques : Cascade, En V, En Spirale
- Formelles : RUP, Merise, CMM
- Agiles : XP, Scrum, Cristal, Lean

Sur deux axes : Formalisme, Durée d'un cycle





# L'esprit Agile

## Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

<http://agilemanifesto.org>



# L'esprit Agile

## Les 4 valeurs

**L'interaction avec les personnes** plutôt que les processus et les outils.

**Un produit opérationnel** plutôt qu'une documentation pléthorique.

**La collaboration avec le client** plutôt que la négociation de contrat.

**La réactivité face au changement** plutôt que le suivi d'un plan.

## Les 12 Principes

1/ Notre première priorité est de **satisfaire le client** en livrant tôt et régulièrement des logiciels utiles.

2/ **Le changement est accepté**, même tardivement dans le développement. Les processus agiles exploitent le changement comme avantage compétitif pour le client.

3/ **Livrer fréquemment** une application fonctionnelle, toutes les deux semaines à deux mois, avec une tendance pour la période la plus courte.

4/ Les experts métier et les développeurs doivent **collaborer quotidiennement** au projet.

5/ Bâissez le projet autour de **personnes motivées**. Donnez leur l'environnement et le soutien dont elles ont besoin, et croyez en leur capacité à faire le travail.

6/ La méthode la plus efficace de transmettre l'information est une **conversation en face à face**.

7/ **Un logiciel fonctionnel** est la meilleure **unité de mesure** de la progression du projet.

8/ Les processus agiles promeuvent un **rythme de développement soutenable**. Commanditaires, développeurs et utilisateurs devraient pouvoir maintenir le rythme indéfiniment.

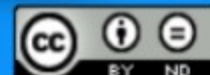
9/ Une attention continue à l'**excellence technique** et à la qualité de la conception améliore l'agilité.

10/ La **simplicité** - l'art de maximiser la quantité de travail à ne pas faire - est essentielle.

11/ Les meilleures architectures, spécifications et conceptions sont issues d'**équipes qui s'auto-organisent**.

12/ À intervalle régulier, l'équipe **réf échit aux moyens de devenir plus efficace**, puis accorde et ajuste son comportement dans ce sens.

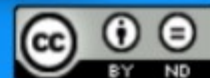
Source [http://fr.wikipedia.org/wiki/Manifeste\\_agile](http://fr.wikipedia.org/wiki/Manifeste_agile)





# Sommaire

- L'esprit Agile
- **Scrum en 5 min**
- Revue pratique & outils
- Retours d'expérience
- Débat : comment commencer ?





# Scrum en 100 mots

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

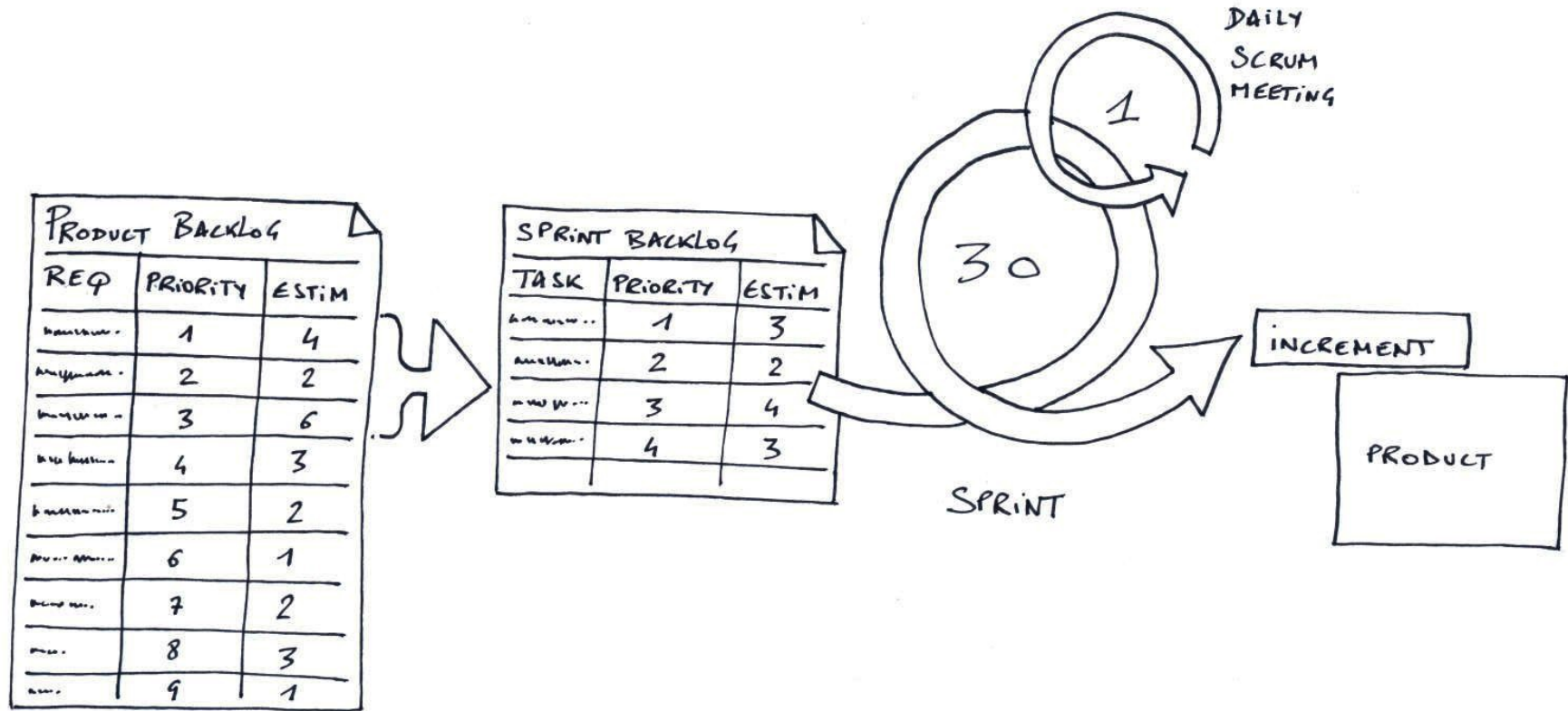


# Caractéristiques de Scrum

- Equipe responsable, en auto-organisation
- Avancement du produit par une série de « sprints » d'un mois ou moins
- Exigences définies comme des éléments d'une liste appelée « Backlog du produit »
- Pas de prescription de pratiques d'ingénierie
- Utilisation de règles génériques permettant de créer un environnement agile pour un projet
- Un des « processus agiles »



# Scrum : Le Process





# Le « Framework » Scrum

## Rôles

- Product Owner
- Scrum Master
- Equipe

## Meetings

- Planification du sprint
- Scrum quotidien
- Revue de Sprint
- Rétrospective

## Artefacts

- Backlog de produit
- Backlog de sprint
- Burndown Chart





# Les Rôles : le Product Owner

- Définit les fonctionnalités du produit
- Choisit la date et le contenu de la release
- Responsable du retour sur investissement
- Définit les priorités dans le backlog en fonction de la valeur « métier »
- Ajuste les fonctionnalités et les priorités à chaque sprint si nécessaire
- Accepte ou rejette les résultats



PRODUCT OWNER





# Les Rôles : le Scrum Master

- Représente le management du projet
- Responsable de faire appliquer par l'équipe les valeurs et les pratiques de Scrum
- Résout des problèmes
- S'assure que l'équipe est complètement fonctionnelle et productive
- Facilite une coopération poussée entre tous les rôles et fonctions
- Protège l'équipe des interférences extérieures



SCRUMMASTER

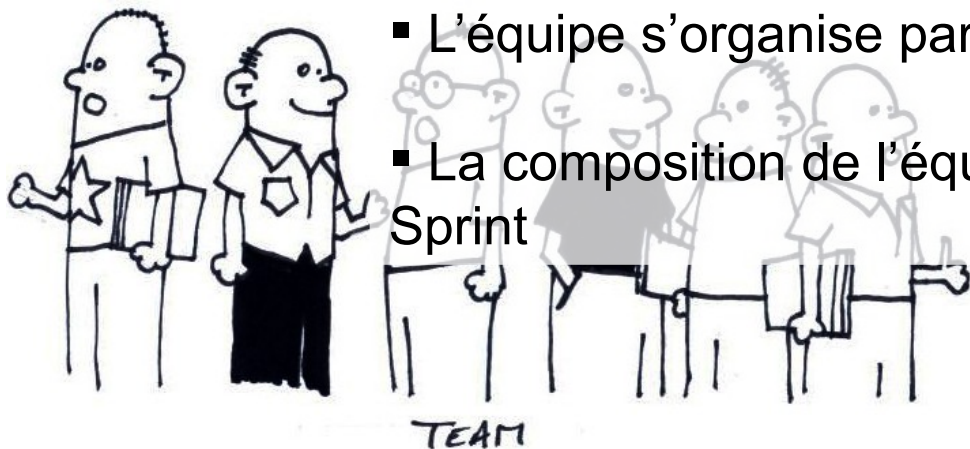




# Les Rôles : l'Equipe

- De 5 à 10 personnes
- Regroupant tous les rôles
  - Architecte, concepteur, développeur, spécialiste IHM, testeur, etc.
- A plein temps sur le projet, de préférence
  - Exceptions possibles (administrateur, ...)

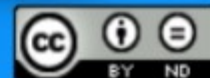
- L'équipe s'organise par elle-même
- La composition de l'équipe ne doit pas changer pendant un Sprint





# Sommaire

- Scrum en 5min
- **Revue pratique : review, retro, planning, daily, impediment**
- Les outils : du papier à l'iphone
- L'esprit derrière tout ça [Agile Manifesto, Lean, Scrum, ...]
- Retours d'expérience (petites équipes, grandes équipes)
- Débat : comment commencer ?





# Le « Framework » Scrum

## Rôles

- Product Owner
- Scrum Master
- Equipe

## Meetings

- Planification du sprint
- Scrum quotidien
- Revue de sprint
- Rétrospective

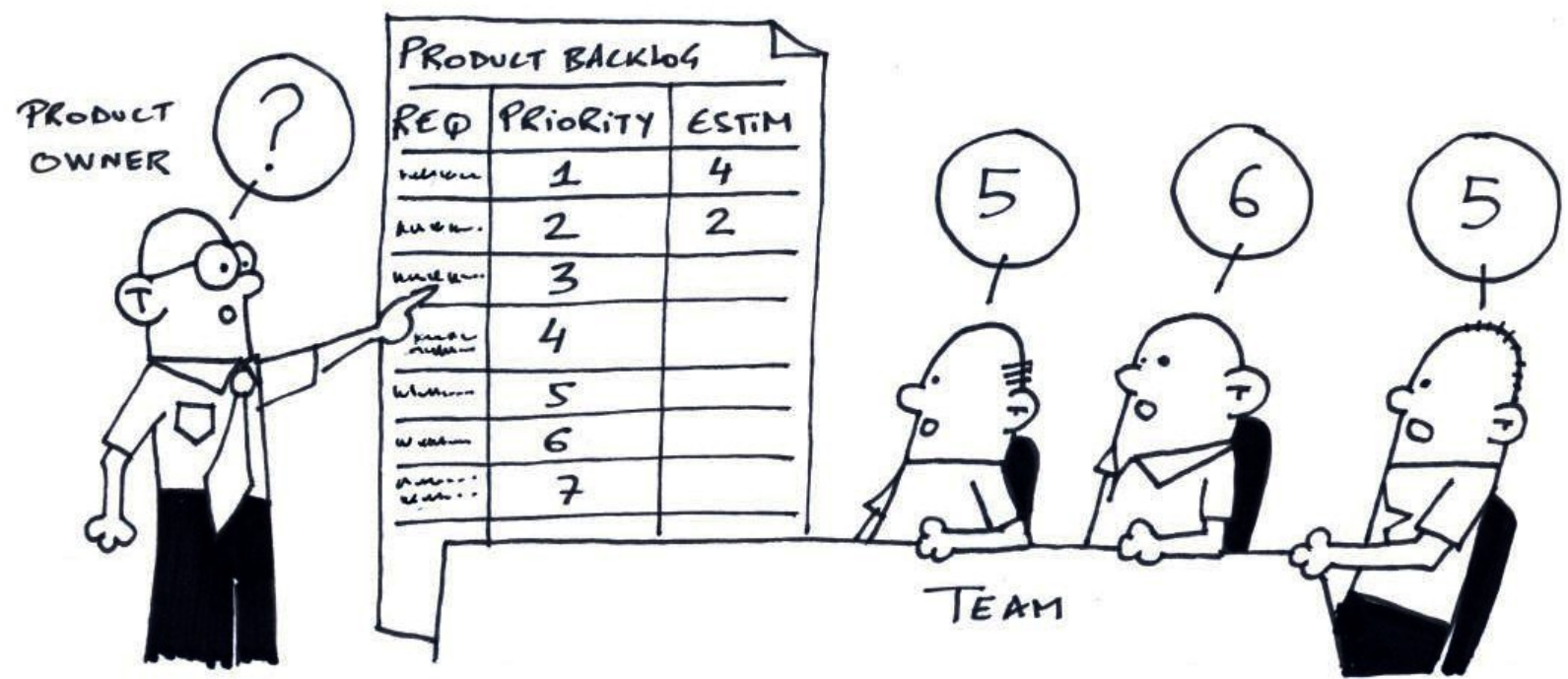
## Artefacts

- Backlog de produit
- Backlog de sprint
- Burndown Chart





# Les Meetings : Planification de Sprint





# Les Meetings : Planification de Sprint

- L'équipe choisit, à partir du backlog de produit, les éléments qu'elle s'engage à finir.
- Le backlog de sprint est créé.
  - Les tâches sont identifiées et estimées (1-16 heures)
  - Collectivement, pas seulement par le ScrumMaster
- La conception de haut niveau est abordée.

En tant que touriste  
potentiel dans la région,  
je veux voir les photos  
des hôtels

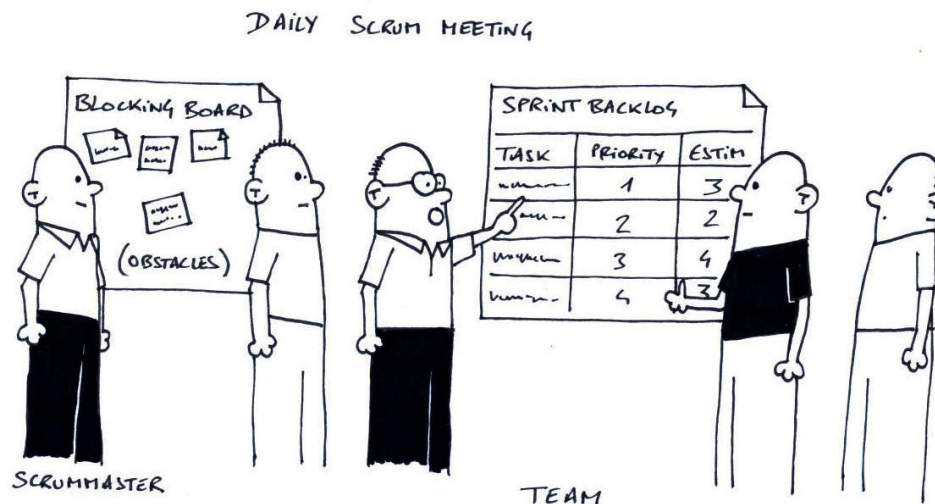
Coder la couche de persistance (8 heures)  
Coder l'IHM (4)  
Ecrire les test fixtures (4)  
Coder la classe foo (6)  
Maj les test de performance (4)





# Les Meetings : Scrum quotidien

- Tous les Jours
- 15 minutes (time boxed)
- Debout

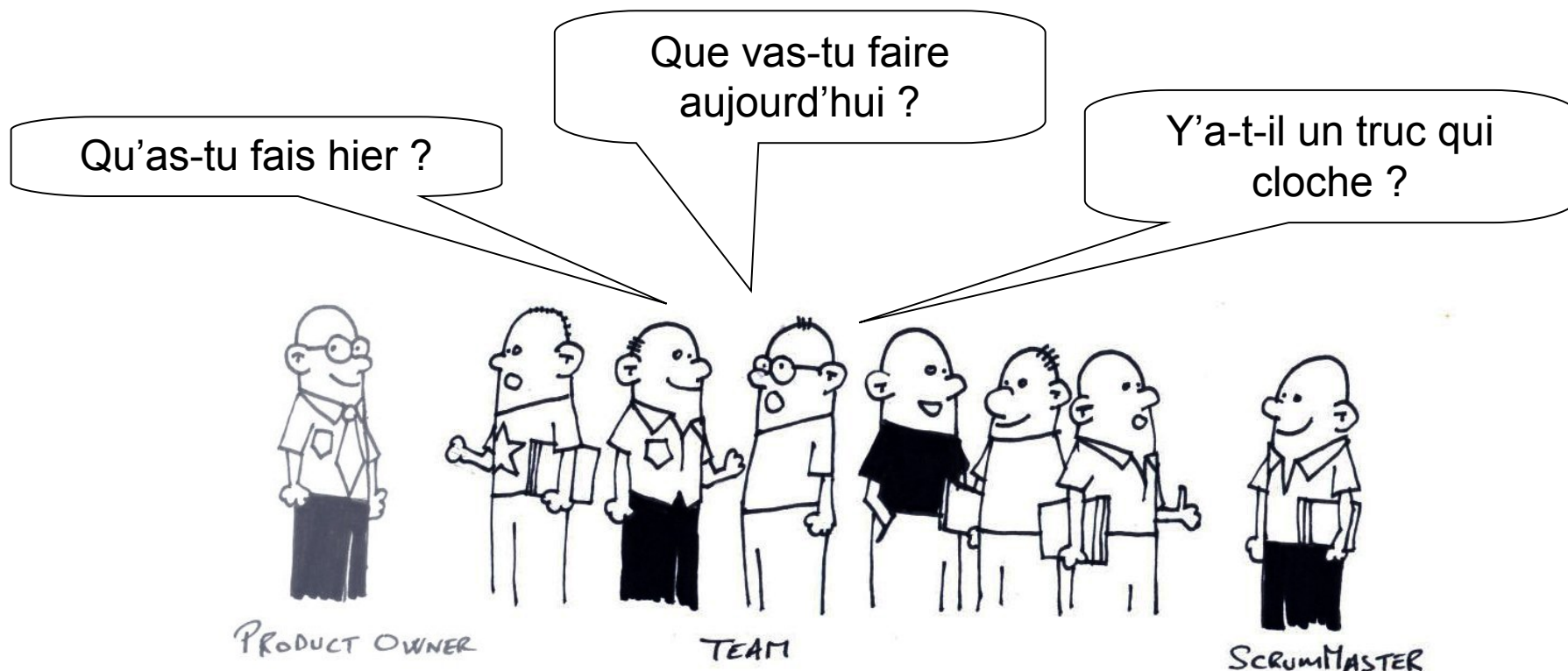


- Pas fait pour résoudre les problèmes
  - Tout le monde est invité
  - Seuls les membres de l'équipe peuvent parler
- Permet d'éviter l'organisation d'autres réunions.





# Les Meetings : Scrum quotidien



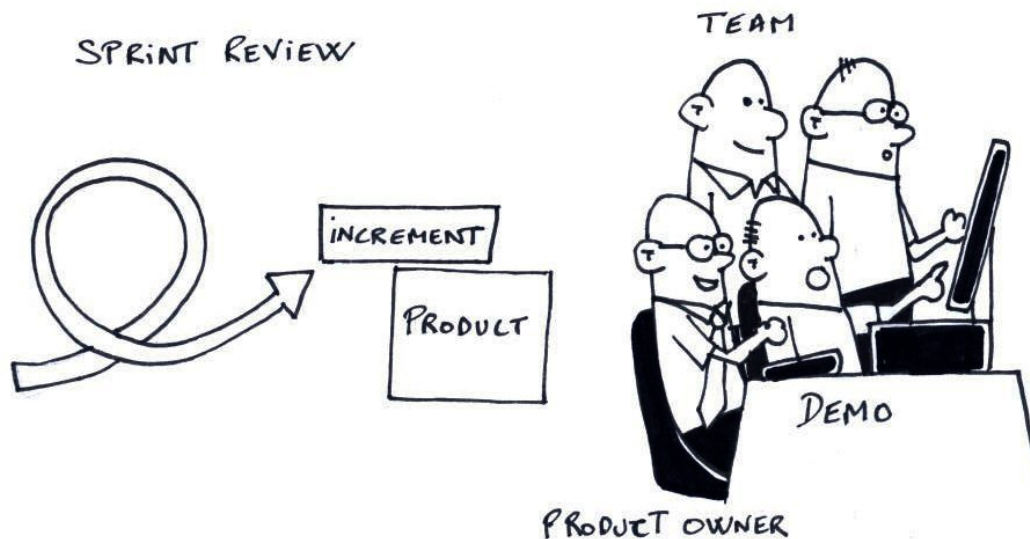
- Il ne s'agit pas de compte-rendus au Scrum Master.
- Se sont des engagements devant ses pairs.





# Les Meetings : Revue de Sprint

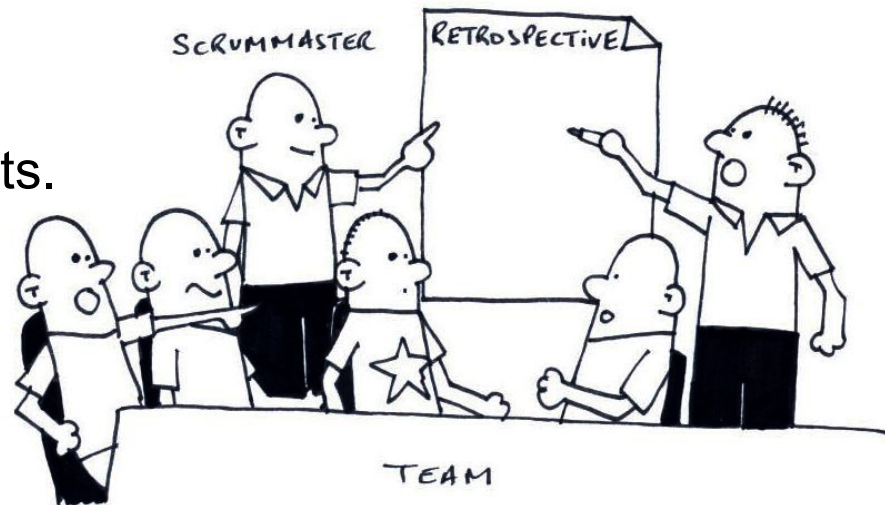
- L'équipe présente ce qu'elle a fait pendant le Sprint.
- L'équipe effectue une démo des nouvelles fonctionnalités incluses dans le livrable de ce Sprint.
- La revue de Sprint est "Informel".
  - Le temps de préparation doit être minimisé.
  - Pas de Slides.
- Toute l'équipe participe.
- Tout le monde est invité.





# Les Meetings : Rétrospective

- Permet de réfléchir régulièrement à ce qui marche et ce qui ne marche pas.
- Dure en général de 15 à 30 minutes.
- Fait à la fin de chaque Sprint.
- Toute l'équipe participe.
  - Scrum Master
  - Product Owner
  - Equipe
  - Eventuellement d'autres intervenants.





# Le « Framework » Scrum

## Rôles

- Product Owner
- Scrum Master
- Equipe

## Meetings

- Planification du sprint
- Scrum quotidien
- Revue de sprint
- Rétrospective

## Artefacts

- Backlog de produit
- Backlog de sprint
- Burndown Chart





# Les Artefacts : le Product Backlog

- Recueil des exigences et des “User Stories”
- Etabli et maintenu par le Product Owner
- Chaque élément doit apporter une valeur ajoutée au produit final (pas de superflu)
- Les “importances” sont définies par le Product Owner
- Les “importances” sont revues à chaque Sprint

PRODUCT BACKLOG

REQ	PRIORITY
fonctionnalité	1
amélioration	2
correction	3
nouveau	4
bug	5
nouveau	6
amélioration	7
bug	8

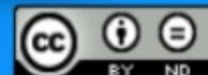


PRODUCT OWNER



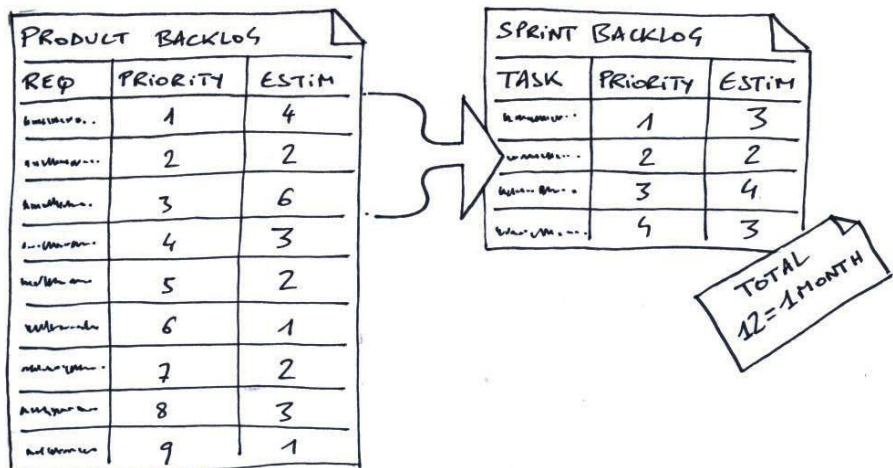
# Exemple de Product Backlog

ID	Nom	Importance	Estimation	Démo	Notes
1	Dépôt	30	5	Authentification, ouvrir la page de dépôt, déposer 10€, aller sur la page du solde et vérifier que ça a bien augmenté de 10€.	Nécessite un diagramme de séquences UML. Ne pas se soucier du cryptage pour l'instant.
2	Voir l'historique de ses transactions	10	8	Authentification, cliquez sur « transactions ». Faire un dépôt. Revenir aux transactions, vérifier que le nouveau dépôt apparaît.	Utiliser la pagination pour éviter des requêtes volumineuses. Conception semblable à la page des utilisateurs.





# Les Artefacts : le Sprint Backlog



- Recueil des différentes tâches, extraites du Product Backlog, que l'équipe s'engage à réaliser lors du Sprint.
  - Le travail n'est jamais assigné par un autre
  
- L'estimation du reste à faire est ajustée chaque jour
  
- Si une tâche n'est pas claire, ou trop volumineuse, la décomposer en tâches plus petites.



# Les Artefacts : Le Burndown Chart

- Graphique permettant de voir le reste à faire sur un Sprint
  - En abscisse : le nombre de jours du Sprint
  - En ordonnée : la quantité de travail à réaliser
- La ligne droite (en bleu) représente la “Vélocité” idéale de l’équipe.
- La ligne courbe (en noir) représente la “Vélocité” véritable de l’équipe.
- Après chaque Daily Scrum Meeting, en fonction des travaux de la veille de chacun, le Burndown Chart est mis à jour.





# Sommaire

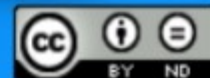
- L'esprit Agile
- Scrum en 5 min
- **Revue pratique & outils**
- Retours d'expérience
- Débat : comment commencer ?





# Sommaire

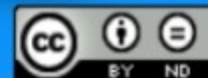
- L'esprit Agile
- Scrum en 5 min
- Revue pratique & outils
- **Retours d'expérience**
- Débat : comment commencer ?





# Sommaire

- L'esprit Agile
- Scrum en 5 min
- Revue pratique & outils
- Retours d'expérience
- **Débat : comment commencer ?**

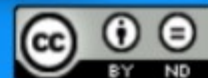




# Crédits

Cette présentation contient des extraits de :

- “Introduction à Scrum” par Mike Cohn, traduction par Claude Aubry.
- “Scrum Checklists” par Sprint-it (<http://www.sprint-it.com>).
- “Scrum and XP from the trenches” par Henrik Kniberg.
- [Emmanuel Chenu](#) pour les images d’illustrations





# Titre

