

# Mise en oeuvre d'une méthode Agile - Scrum



<http://www.sebastienchabal.fr/wp-content/uploads/2010/08/meleeBlack-Bleu.jpg>

(Ce cours suppose les principes d'XP connus)

Merci à tous ceux qui ont rendu leurs cours et exposés disponibles sur le web & dans les livres, voir Biblio. & refs dans les slides

M. Blay-Fornarino

# Bibliographie

- ❧ Scrum et l'agilité des équipes de développement, NormandyJUG, Par Dimitri Baeli & Nicolas Giard
- ❧ The Zen of Scrum, Boris Gloger, David Koontz
- ❧ Scrum, état de l'art, François Potentier.
- ❧ eXtreme Programming & Scrum Practices, Embrace Change, Naresh Jain

# Contrôler le Chaos ...

## SCRUM

- ☛ Met l'accent sur la gestion de projet
- ☛ Met l'accent sur les caractéristiques livrées et l'ajustement selon les résultats
- ☛ L'objectif est de trouver un équilibre entre permettre au métier de changer d'approche et à l'équipe de développement de faire un travail de qualité à sa portée.

# Backlog

TODO

DOING

DONE

Valeurs de  
SCRUM

Organisation

Rôles

Réunions

Artefacts

Environnement

Conclusion

# Backlog

TODO

DOING

DONE

Valeurs de Scrum

Organisation

Rôles

Réunions

Artefacts

Environnement

Conclusion

# Scrum en 100 mots

- ❧ Scrum est un processus agile qui vise à 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 répondre aux 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.

# Les valeurs de SCRUM

- ❧ **Engagement.** Soyez prêt à vous engager sur un objectif. Scrum assure aux développeurs l'autorité dont ils ont besoin pour remplir leurs engagements.
- ❧ **Focus.** Faites votre travail. Concentrer tous vos efforts et vos compétences à faire le travail que vous vous êtes engagé à faire. Ne vous inquiétez pas d'autre chose
- ❧ **Transparence.** Scrum laisse tous les éléments d'un projet visibles à tous.
- ❧ **Respect.** Les individus sont façonnés par leurs antécédents et leur expérience. Il est important de respecter les différentes personnes qui composent une équipe.
- ❧ **Courage.** Ayez le courage de vous engager, d'agir, d'être ouvert et d'attendre du respect

# Backlog

TODO

DOING

DONE

Valeurs de  
SCRUM

Organisation

Rôles

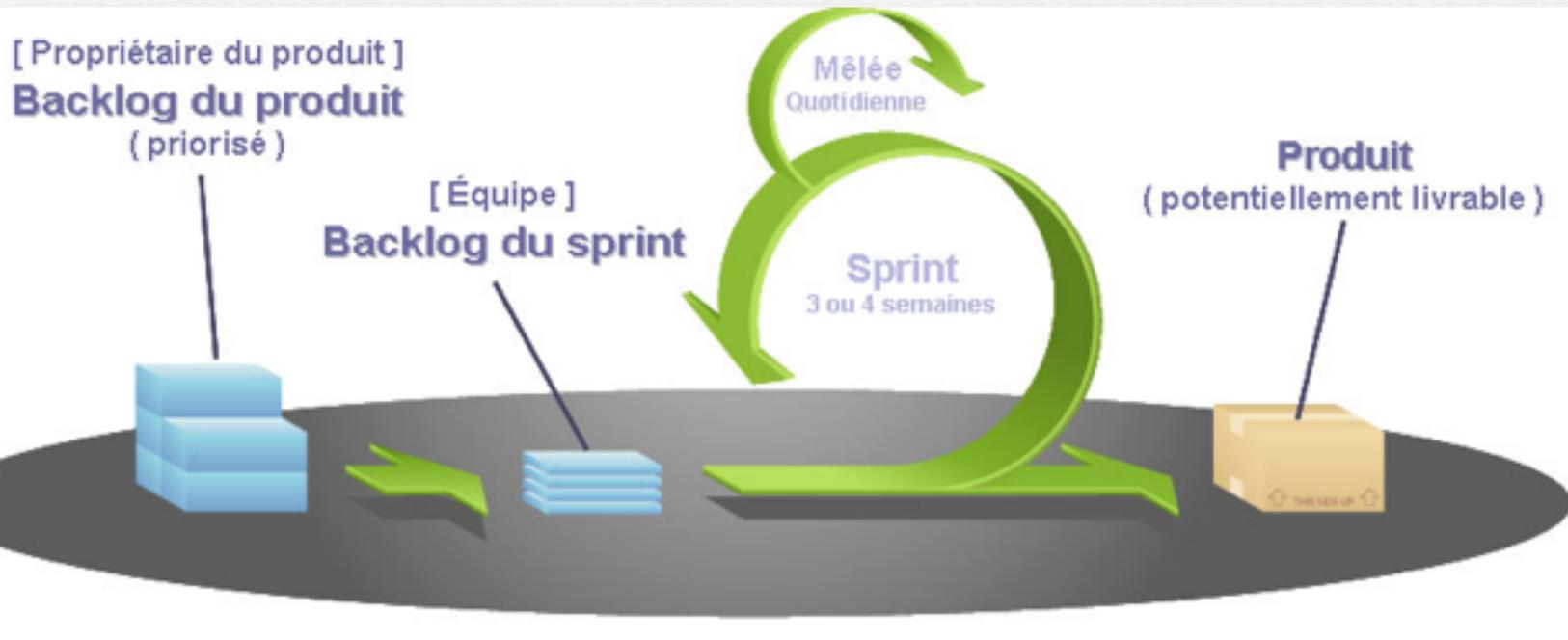
Réunions

Artefacts

Environnement

Conclusion

# Scrum : processus général



<http://www.computure.net/fr/methodes/111-essentiel-de-scrum-suite>

# Backlog

TODO

DOING

DONE

Valeurs de  
SCRUM

Organisation

Rôles

Réunions

Artefacts

Environnement

Conclusion

# Caractéristiques de 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 livraison
- ❖ 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

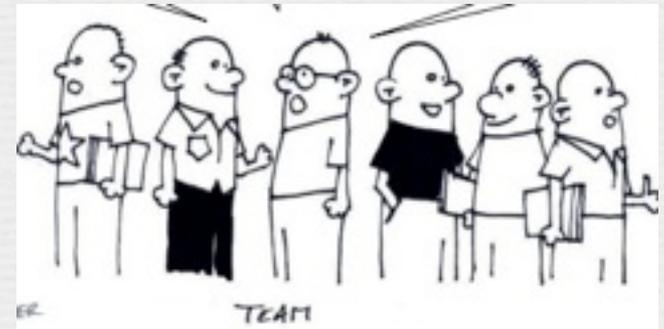
- ❧ Représente le management du projet
- ❧ Responsable de faire appliquer par l'équipe les valeurs et les pratiques de Scrum
- ❧ Résout des problèmes
- ❧ S'assure que l'équipe est complètement fonctionnelle et productive
- ❧ Facilite une coopération poussée entre tous les rôles et fonctions
- ❧ Protège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



# Les Rôles : l'équipe

- 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

Martin (2003): "The team is in it for the long term. They work hard, at a pace that can be sustained indefinitely. They conserve their energy, treating the project as a **marathon** rather than a **sprint**."



# Backlog

TODO

DOING

DONE

Réunions

Artefacts

Valeurs de scrum

Organisation

Rôles

Environnement

Conclusion

# Caractéristiques de Scrum

## Rôles

- Product Owner
- Scrum Master
- Equipe

## Meetings

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

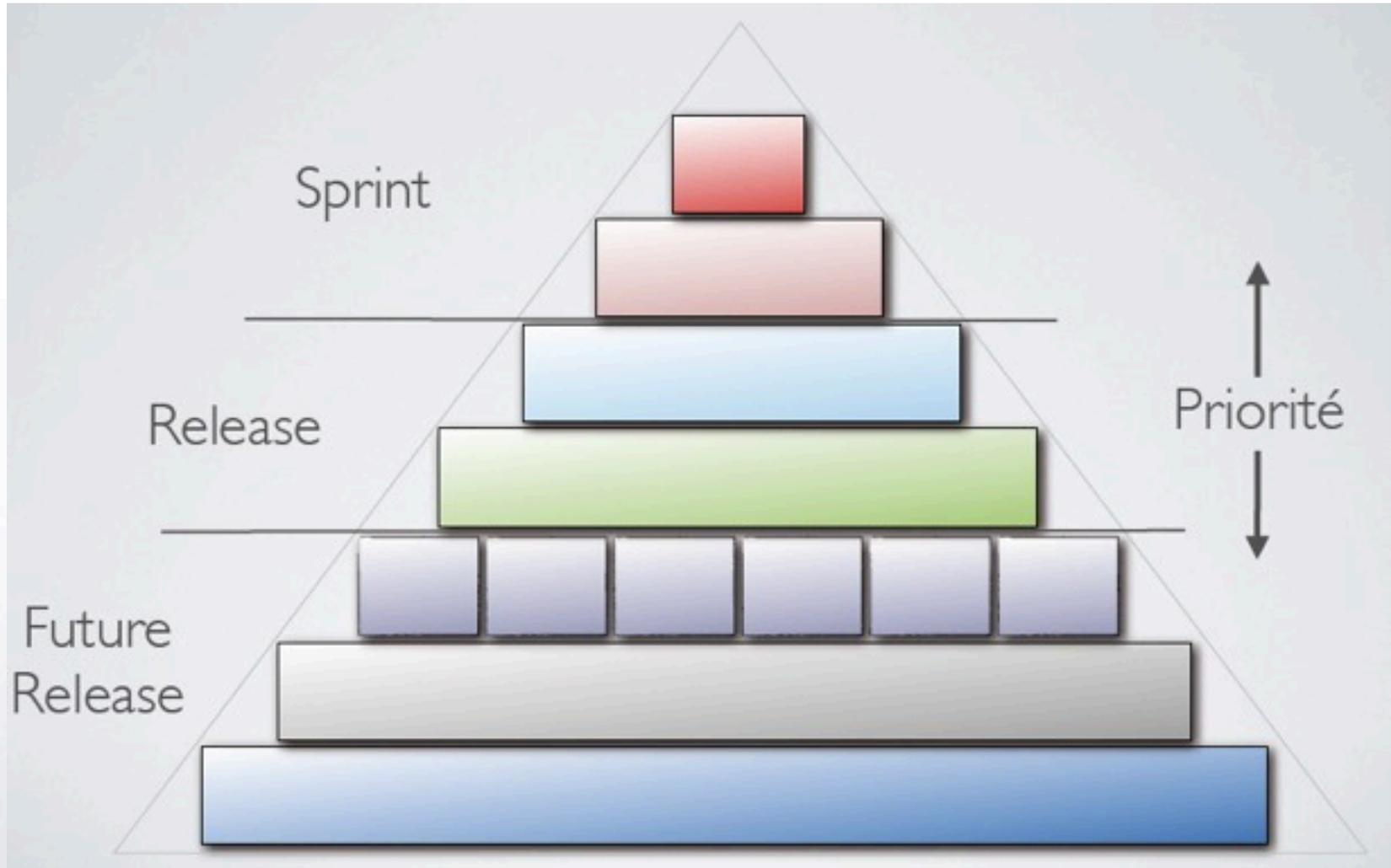
## Artefacts

- Backlog de produit
- Backlog de sprint
- Burndown Chart

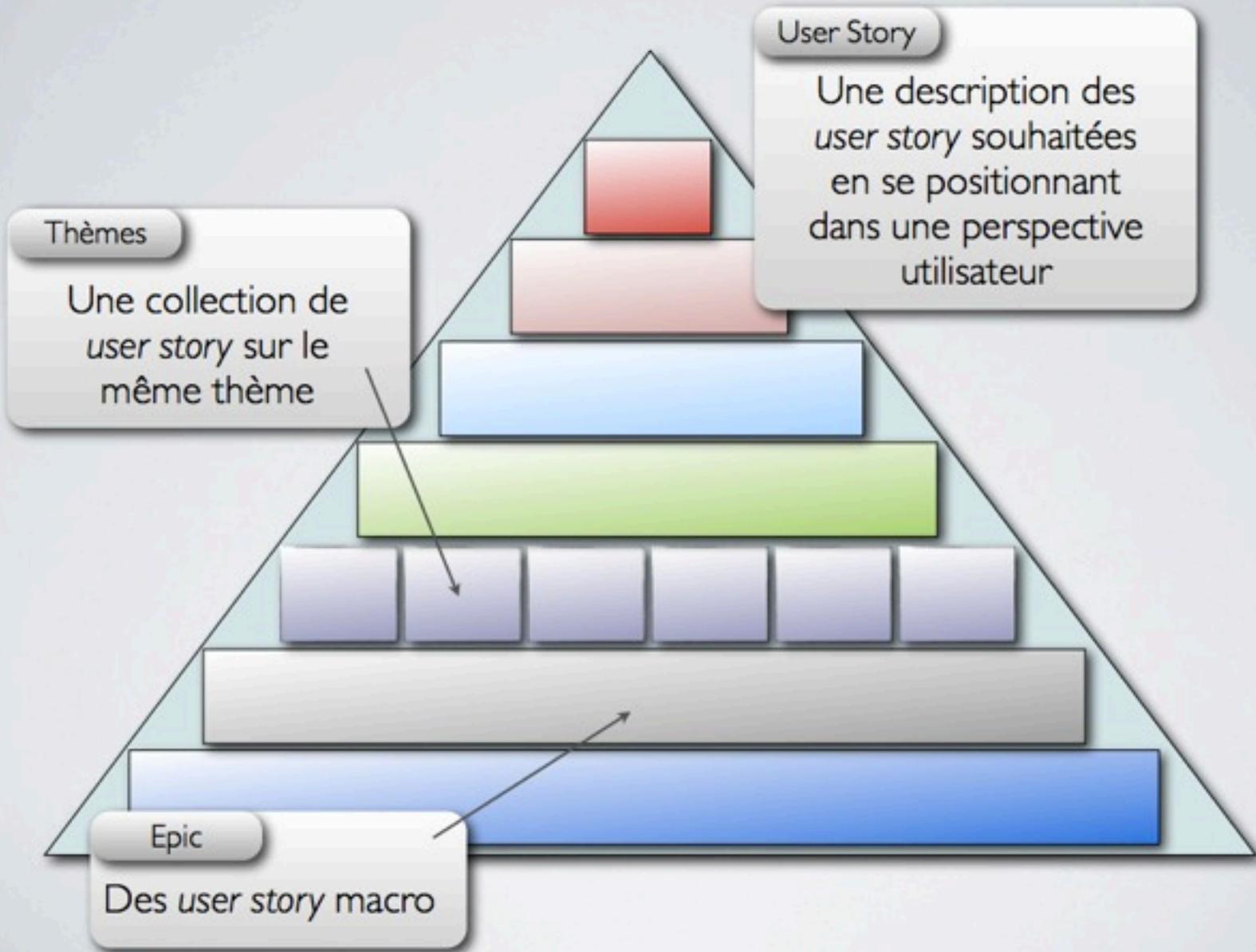
# BACKLOG DE PRODUIT

- Les exigences du produit
- Toutes (Idées, fonctionnalités, Epic, Thème, etc.)
- Exprimées en User Stories
- Le PO le maintient organisé
- Toujours estimé et avec les priorités

# Iceberg du product backlog



Yannick Quenec'hdu  
[yquenechdu@gmail.com](mailto:yquenechdu@gmail.com)



**Thèmes**  
Une collection de *user story* sur le même thème

**User Story**  
Une description des *user story* souhaitées en se positionnant dans une perspective utilisateur

**Epic**  
Des *user story* macro

# User stories / (Récits ou histoires d'utilisateur)

- ❧ Une ou deux phrases résument ce que veut l'utilisateur
- ❧ Décrit comment le système est sensé travailler
- ❧ Ecrite sur une carte
- ❧ Contient suffisamment de détails pour pouvoir être estimée

Voir cours sur ce sujet

Martin (2003): “As part of selecting each desired feature, the customers define automated **acceptance tests** to show that the feature is working.”

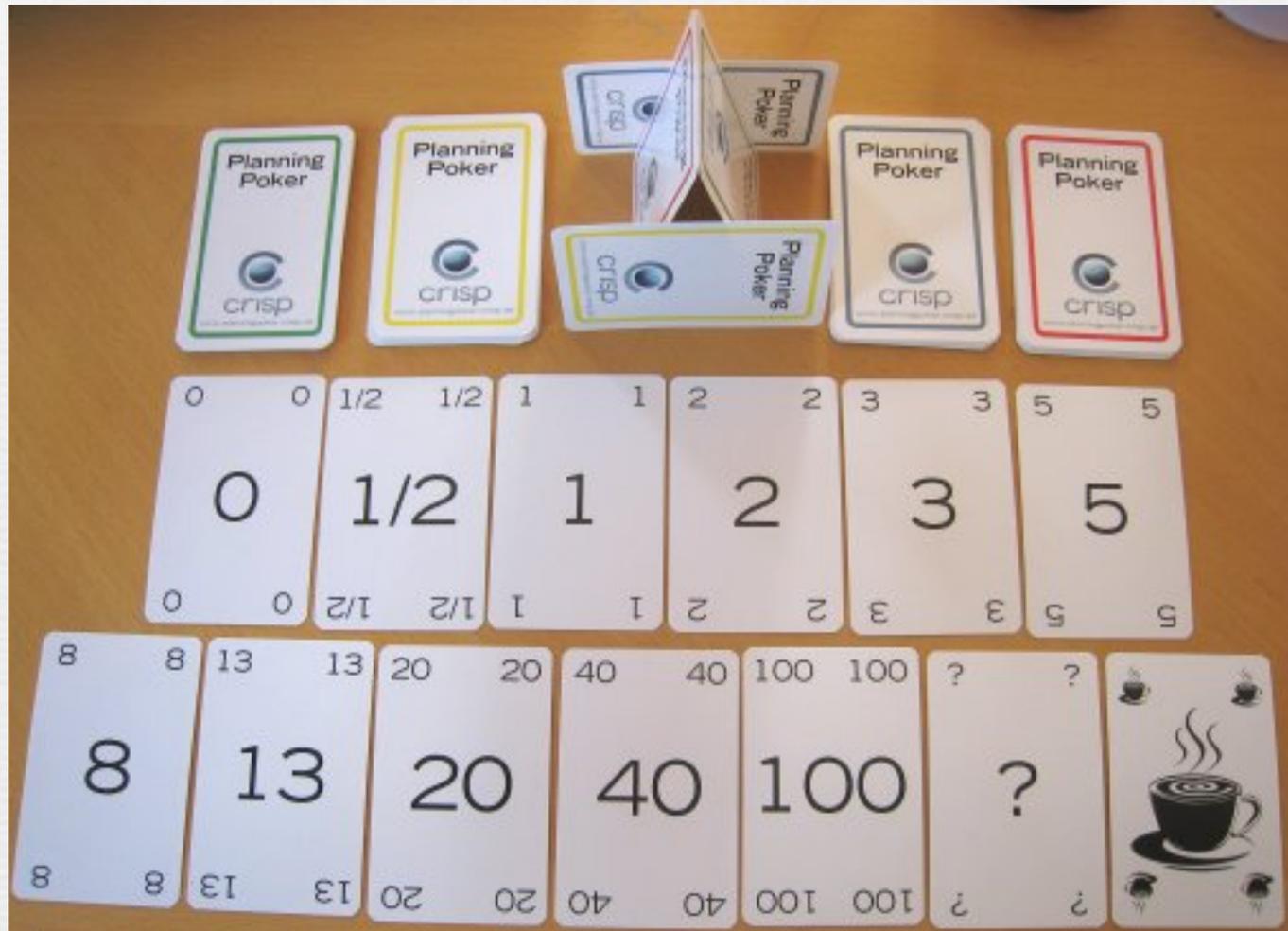
Lorsqu'une expression du besoin existe en UML, elle peut être utilisée

# Estimation d'une US par les développeurs

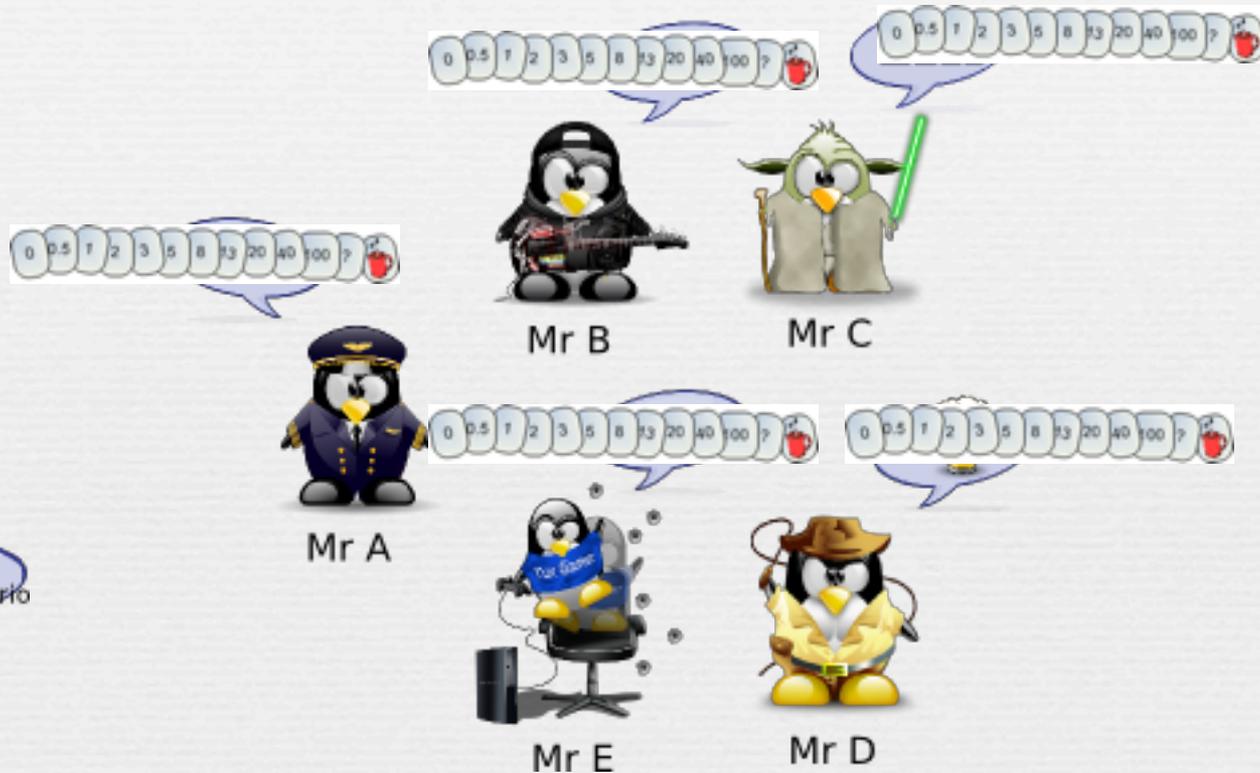
## Le planning Pocker



# Planning Poker



# Planning Pocker



<http://www.openagile.net/?p=155>

# Planning Pocker



Combien de temps va prendre ce scénario



PO



Mr B



Mr C



Mr A



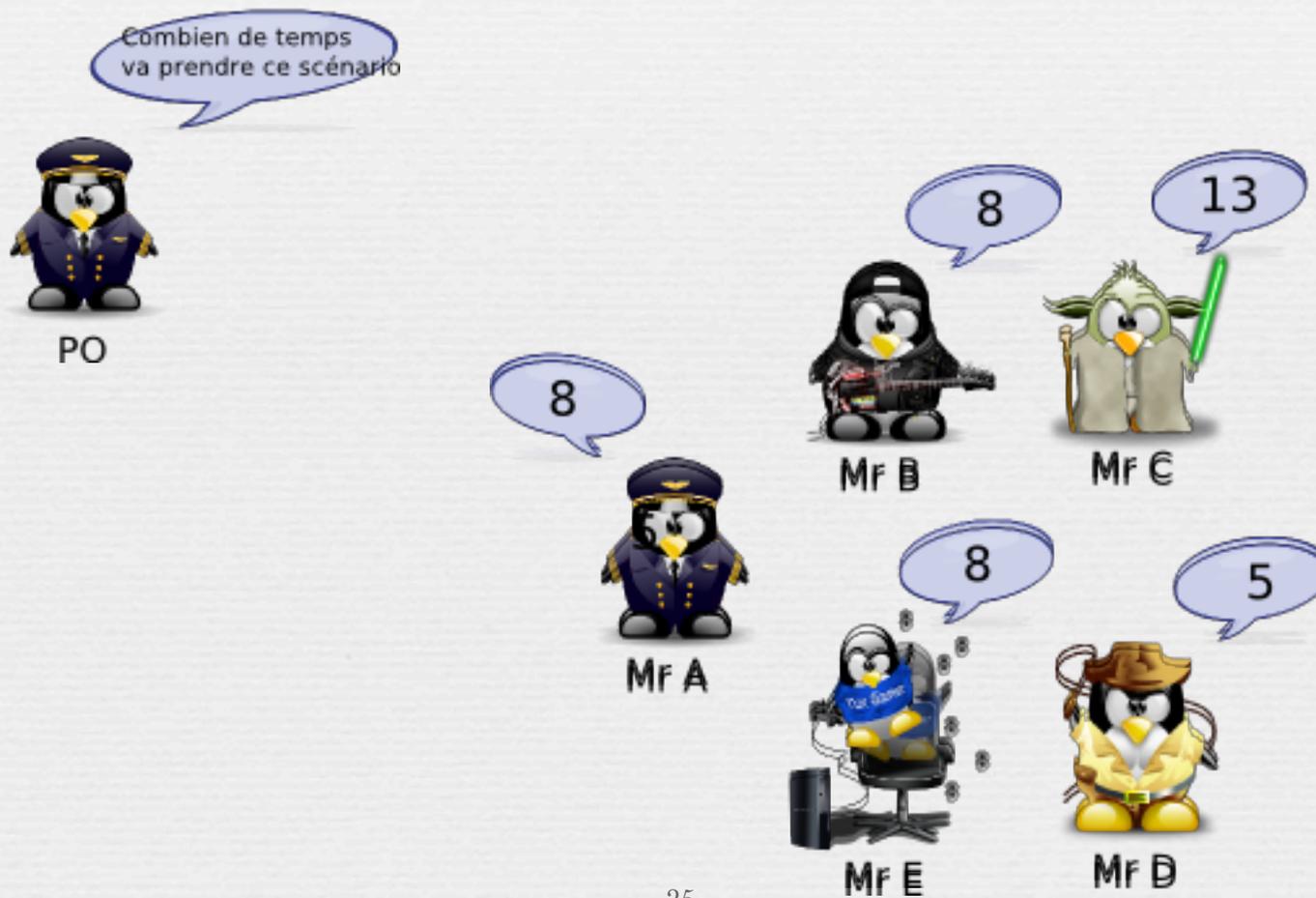
Mr E



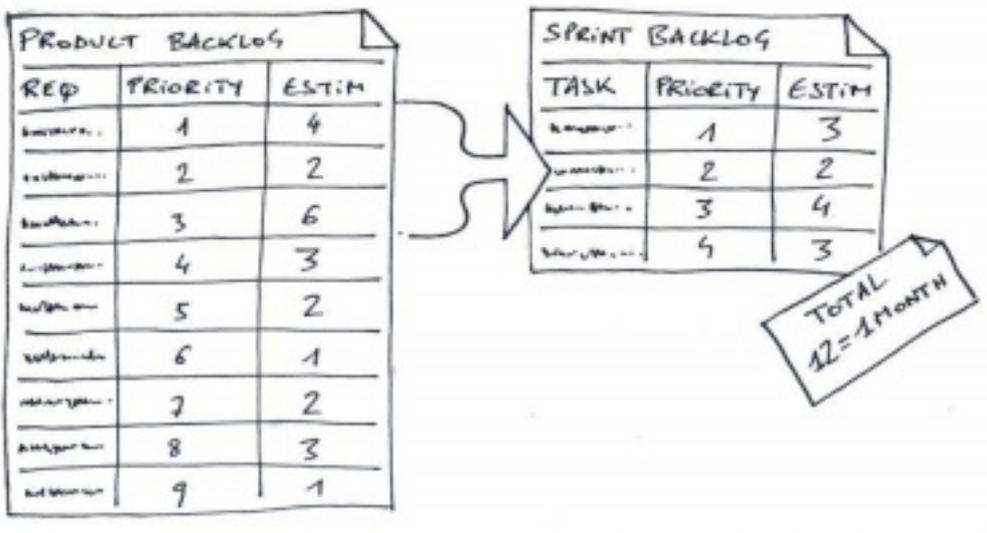
Mr D



# Planning Pocker



# Artefacts : le sprint Backlog



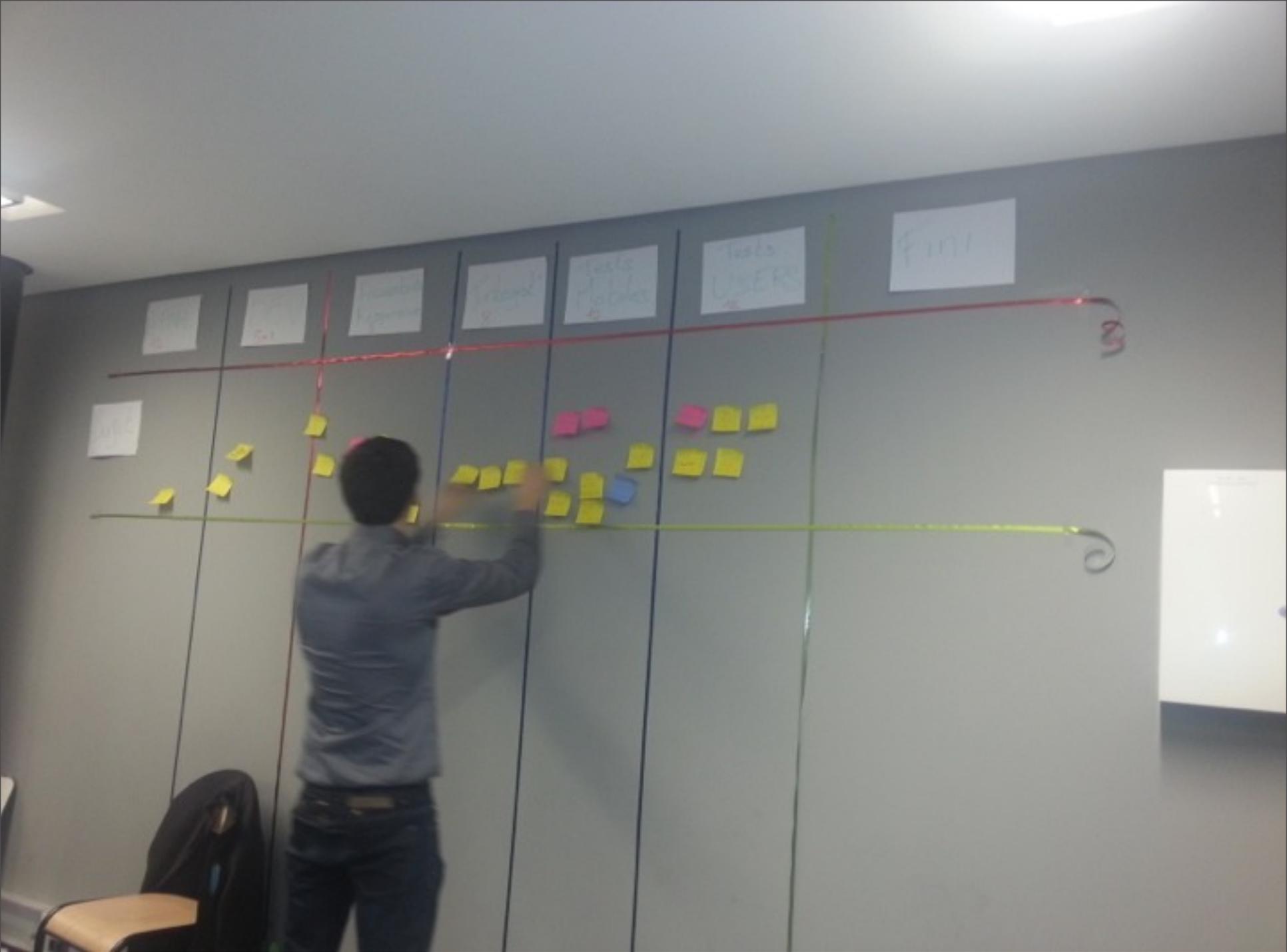
- ❧ Recueil des différentes tâches, extraites du Product Backlog, que l'équipe s'engage à réaliser lors du Sprint.
  - ❧ Le travail n'est jamais assigné par un autre
- ❧ L'estimation du reste à faire est ajustée chaque jour
- ❧ Si une tâche n'est pas claire, ou trop volumineuse, la décomposer en tâches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

# «Sprint Backlog»

SPRINT BACKLOG (TASKS)					
Product Backlog	Not Started	In Progress	In Tests	Rejected	Accepted
	  				
	 	 			
		 			
	 				 

# Version adaptée par les étudiants IUT 2013





# Version électronique : JIRA

**JIRA** Dashboards ▾ Projects ▾ Issues ▾ Agile ▾ Tempo ▾ **Create Issue**  ? ⚙️ 👤

**LP IDSE** Plan Work Report Board ▾

QUICK FILTERS: [Only My Issues](#) [Recently Updated](#) ⌵

**19 To Do** **2 In Progress** **4 Done** [Release...](#)

▼ Expedite 1 Issue

- 📄 **LPDEUX-2**   
🛑 Création de tâche blocante

▼ Everything Else 24 Issues

To Do	In Progress	Done
<ul style="list-style-type: none"><li>📄 <b>LPQUATREB-7</b>  ⬆️ Diagramme de classe et de scénario</li></ul>	<ul style="list-style-type: none"><li>➕ <b>LPTROIS-6</b>  ⬆️ Conception Visual Paradigm</li></ul>	<ul style="list-style-type: none"><li>➕ <b>LPQUATREB-5</b>  ⬆️ Feature test</li></ul>
<ul style="list-style-type: none"><li>📄 <b>LPDEUX-7</b>  ⬆️ Tuto - Mock</li></ul>	<ul style="list-style-type: none"><li>📄 <b>LPIDSEGRUN-1</b>  ⬆️ Test</li></ul>	<ul style="list-style-type: none"><li>📄 <b>LPIDSEGRUN-3</b>  ⬆️ Test</li></ul>
<ul style="list-style-type: none"><li>📄 <b>LPIDSEGRUN-5</b>  ⬆️ Bug Graphique[test Tempo]</li></ul>		<ul style="list-style-type: none"><li>➕ <b>LPIDSEGRUN-2</b>  ⬆️ Ajout d'une nouvelle fonctionnalité</li></ul>
<ul style="list-style-type: none"><li>➕ <b>LPTROIS-5</b>  ⬆️ Permettre le filtrage des informations par type</li></ul>		<ul style="list-style-type: none"><li>📄 <b>LPQUATREB-2</b>  ⬆️ Test de creation de tache</li></ul>

# Version électronique : Redmine

Aperçu Activité Demandes Nouvelle demande Gantt Agile Calendrier Annonces Documents Wiki Fichiers Configura

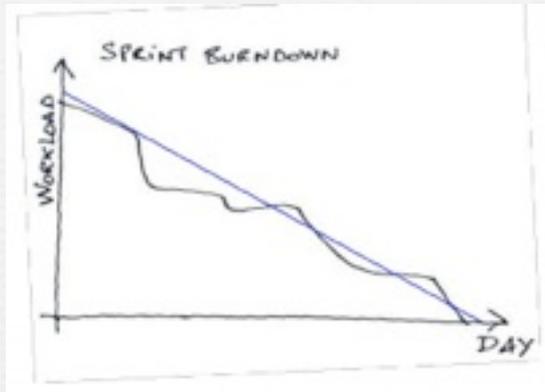
## Agile board

▼ Filtres  
☑ Statut tous  
Ajouter le filtre  
► Options

✓ Appliquer Effacer

Nouveau (57)	Assigné (23)	Résolu (6)	Commentaire (2)	Fermé (3)	Rejeté
<b>Livrable #1659</b> Glossaire	<b>Tâche initiale #1625</b> Séance 1 : Création de l'environnement Nasreddine dib	<b>Tâche initiale #1690</b> Séance 1 Anthony giroud	<b>Anomalie #1715</b> Retard sur le suivi de projet	<b>Evolution #1635</b> Construction de L'environnement Thomas Valverde	
<b>Livrable #1619</b> Document Spécification	<b>Tâche initiale #1643</b> 1.3. Création d'un repertoire de projet Vincent boucher- thouveny	<b>Tâche initiale #1695</b> création de la phase 3	<b>Anomalie #1717</b> Suivi de projet	<b>Tâche initiale #1639</b> Cas d'utilisation Mathilde dibo	
<b>Livrable #1629</b> Phase 1: Kanban	<b>Livrable #1628</b> Phase 1: creation du document (glossaire ,diagrammes,...)	<b>Tâche initiale #1697</b> Configuration dossier SVN		<b>Tâche initiale #1681</b> Mise en place du KANBAN Christophe malepart	
<b>Livrable #1621</b> Kanban	<b>Livrable #1658</b> Communication : Phase 1	<b>Tâche initiale #1698</b> Découverte du sujet			
<b>Livrable #1622</b> Répertoire de projet		<b>Tests #1682</b> Test du SVN par les deux membres			
<b>Livrable #1624</b> Document explicatif d'integration aux autres groupes					
<b>Tâche initiale #1689</b>					

# 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

Un problème est détecté  
au 4<sup>e</sup> jour; la charge est  
réévaluée

Accélération

104  
points  
scrum

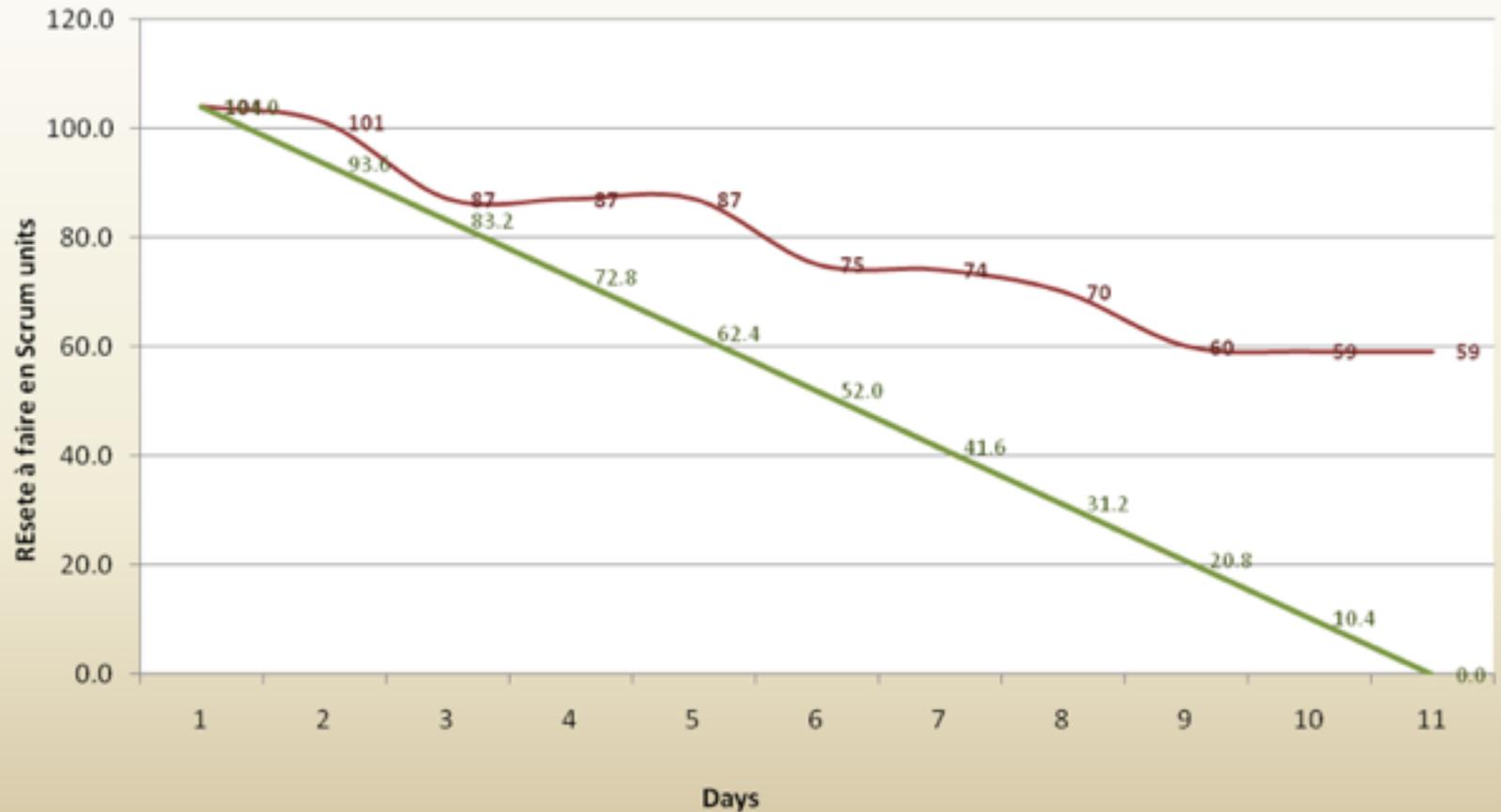


Courbe idéale

Durée du  
sprint fixée à  
10 jours

<http://www.frenchsug.org/display/FRSUG/Burndown+charts>

## Sprint burndown chart - sans avancement



<http://www.frenchsug.org/display/FRSUG/Burndown+charts>

# Agile charts

▼ Filtrer

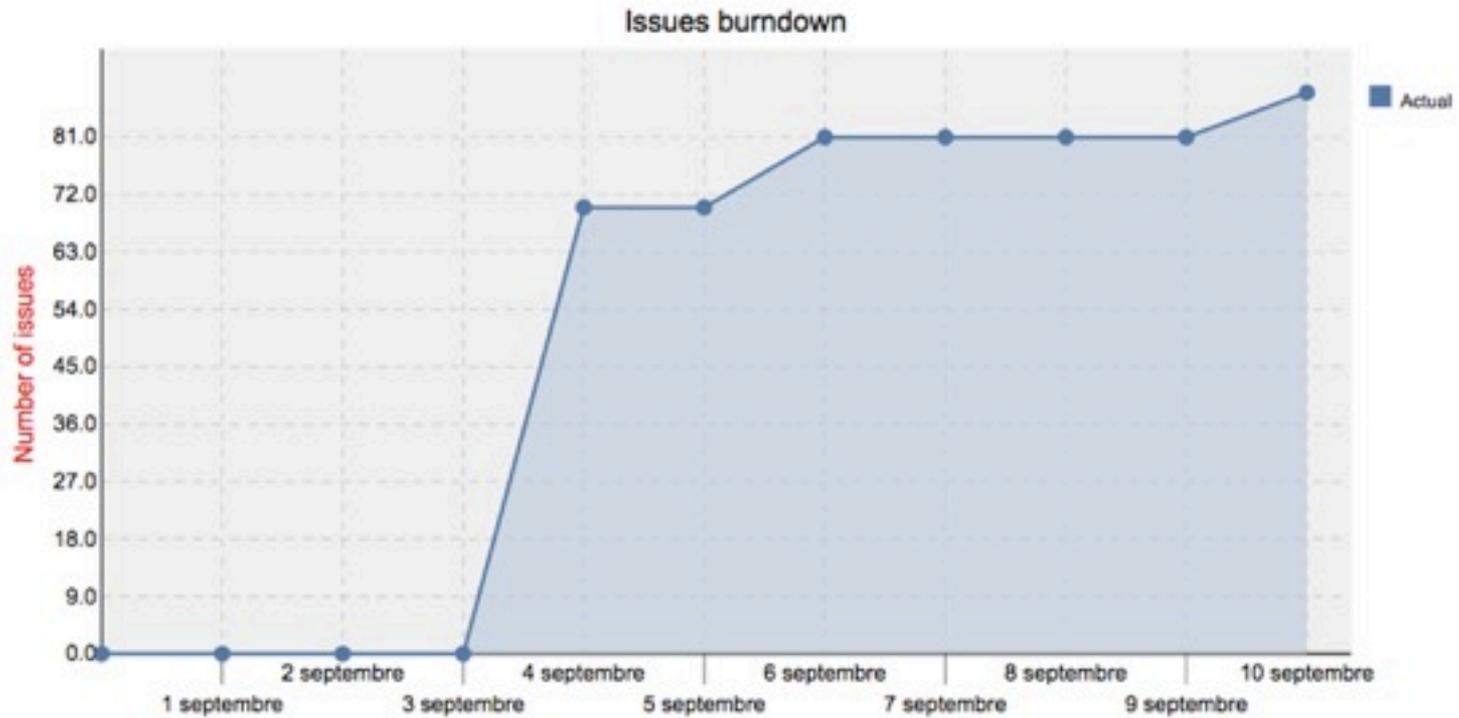
► Options

Chart: Issues burndown

From:  To: 2014-09-1

✓ Appliquer ✕ Effacer

Ajouter le filtre



# Backlog

TODO

DOING

DONE

Valeurs de scrum

Organisation

Rôles

Artefacts

Réunions

Environnement

Conclusion

# Caractéristiques de Scrum

## Rôles

- Product Owner
- Scrum Master
- Equipe

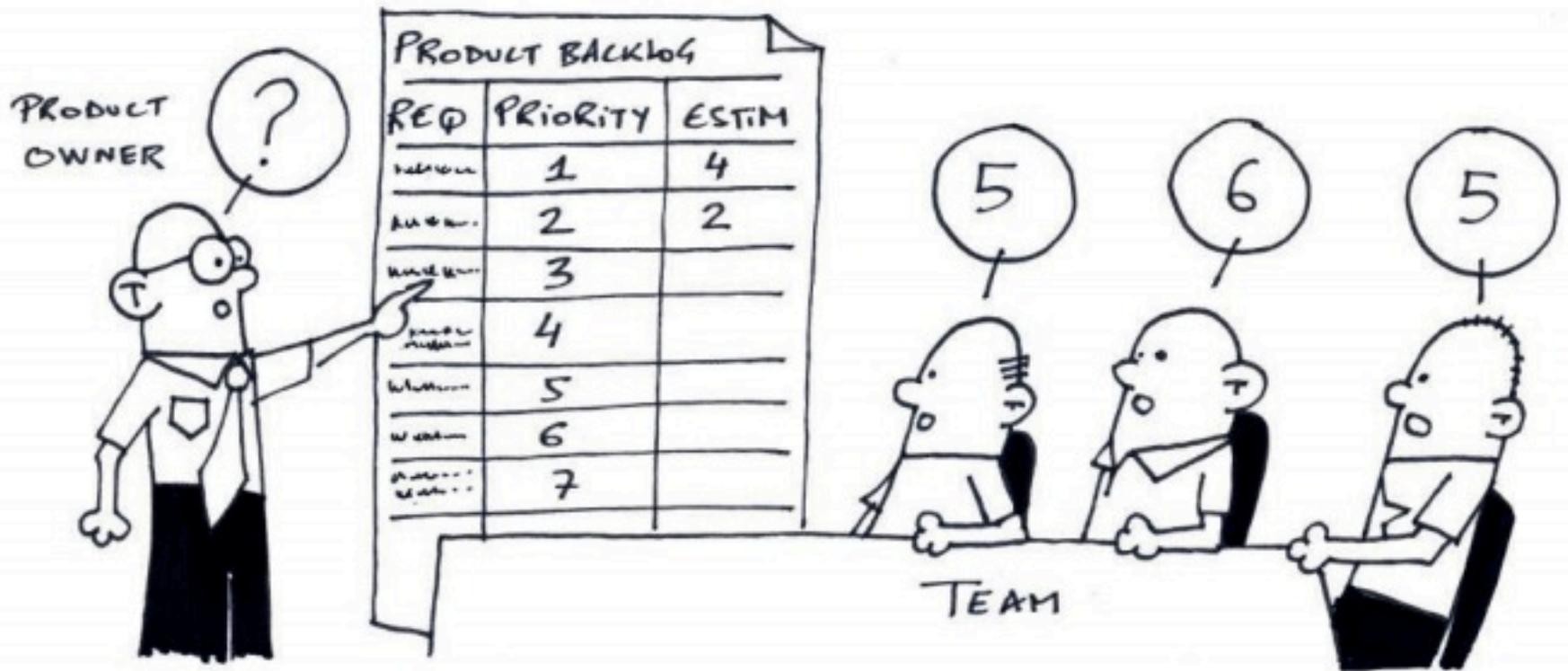
## Meetings

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

## Artefacts

- Backlog de produit
- Backlog de sprint
- Burndown Chart

# Meetings : Planification du sprint



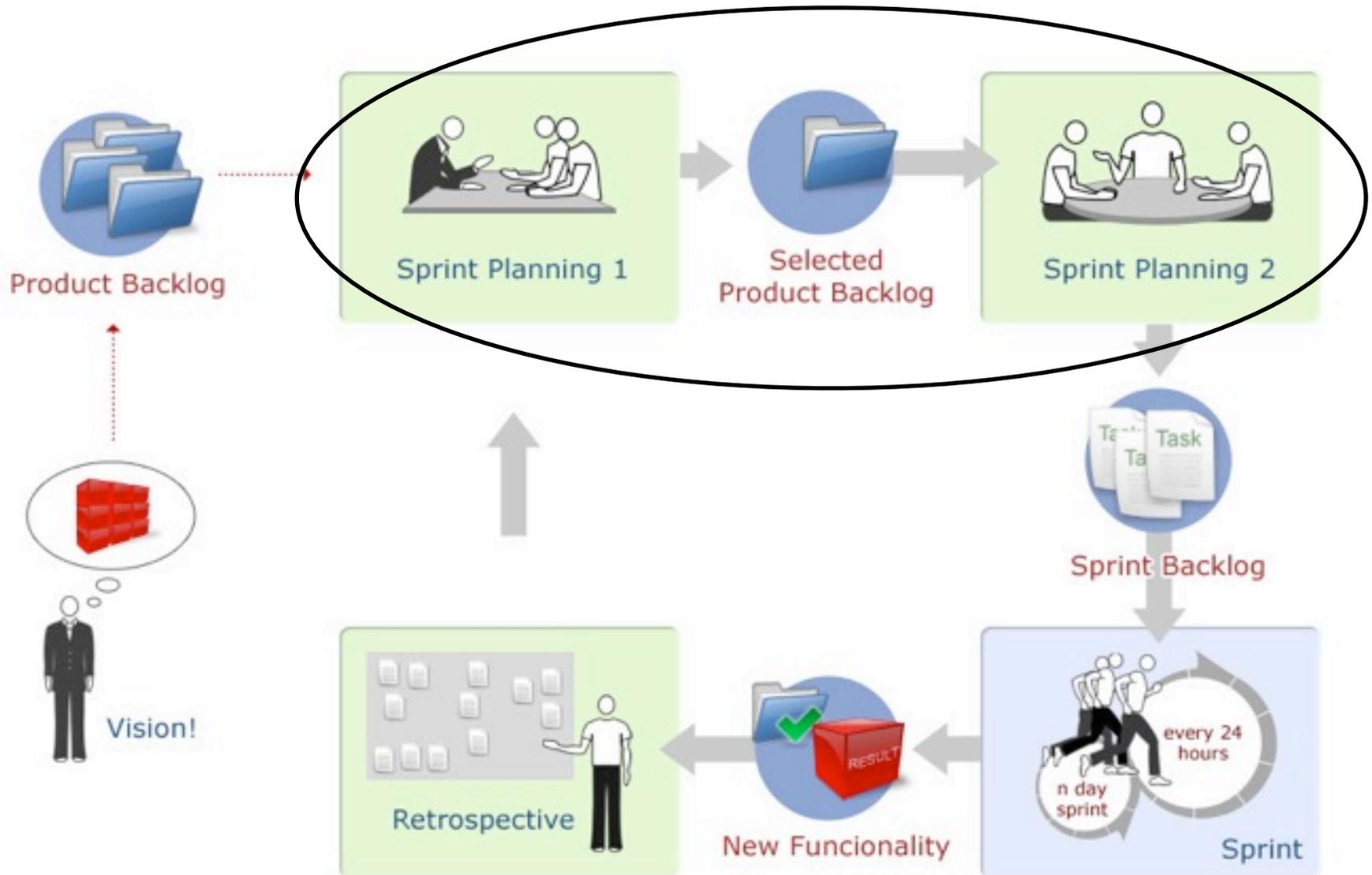
# Meetings : Planification du sprint

L'équipe choisit, à partir du backlog de produits, 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)



# Sprint Planning 1

- ❧ In Sprint Planning 1, the Implementation Team and the Product Owner negotiated **which "stories" would be implemented in the coming sprint.**
- ❧ The team made sure it understood the stories, in particular the acceptance criteria (I recommend agreeing on 'How to Demo').

<http://www.scrum-breakfast.com/2011/02/how-we-do-sprint-planning-2.html>

# Sprint Planning 2 (1)

- ❧ During Sprint Planning 2, the Implementation Team must figure out how to solve the problem it took on in Sprint Planning 1. This consists of two parts:
  - ❧ **A solution concept - a design, architecture** or whatever, which explains how the problem is to be solved/feature is to be realized.
  - ❧ **A list of tasks - what steps must the team do to get each selected backlog item to the state 'done'.**
- ❧ The goal is not an absolutely perfect design or task planning. It's about getting a clear enough concept that the team can start work.

<http://www.scrum-breakfast.com/2011/02/how-we-do-sprint-planning-2.html>

# Sprint Planning 2 : une approche

- Sprint Planning 2 est ici limité à 2 hours (6 histoires) ...
- Agenda for Sprint Planning 2
  - 14.00 - 14.05 Formation des paires et distribution des histoires
  - 14.05 - 14.35 **Concept** - Réflexion sur les aspects techniques -> Production de documents à présenter à l'équipe
  - 14.35 - 15.05 **Présentation** des paires limitée à 5 Minutes par histoire (Q/R comprises)
  - 15.05 - 15.35 **Tâches** - Chaque paire découpe les histoires en ensemble de tâches d'au plus un jour.
  - 15.35 - 16.00 Présentation des tâches en 4 Minutes maximum (25 Minutes / 6 Stories ).

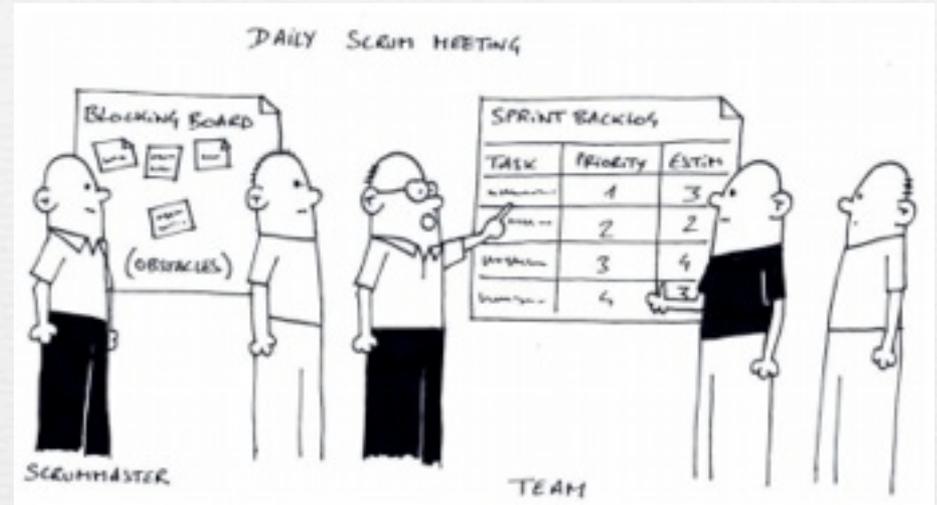
# Sprint Planning 2 : une approche

- ➔ Au moins 2 personnes ont travaillé chaque tâche
- ➔ La réunion est structurée.
- ➔ Les solutions sont présentées et discutées avec l'ensemble de l'équipe qui peut alors s'entraider.

<http://www.scrum-breakfast.com/2011/02/how-we-do-sprint-planning-2.html>

# 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

# Meetings : Scrum quotidien



Il ne s'agit pas de compte-rendus au Scrum Master.  
=> Ce sont des engagements devant les pairs

# 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, une démo si possible.
- ❧ Toute l'équipe participe.
- ❧ Tout le monde est invité.



# Mais que signifie TERMINE ? DONE?

- Il est interdit de livrer un item inachevé, même avec l'intention de « le terminer plus tard ».
  - ➡ Maintenir la confiance avec le client de ne pas « cacher » le travail non « terminé ».
- Functionality has been code reviewed, functionality has been integrated and built, acceptance tests have been run, and documentation has been created.
- Code adheres to standards, is clean, has been refactored, has been unit tested, has been checked in, has been built, and has had a suite of unit tests applied to it

# Meetings : Rétrospective

- A la fin de chaque sprint
- 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



# Backlog

TODO

DOING

DONE

valeurs de scrum

organisations

Rôles

Artefacts

Réunions

Environnement

Conclusion

# Environnement de collaboration

❧ Organisation en «war room»



# Backlog

TODO

DOING

DONE

valeurs de scrum

organisations

Rôles

Artefacts

Réunions

Environnement

Conclusion

# Conclusion (1)

- ❧ Méthode de gestion de projet – développement logiciel
- ❧ A compléter avec des techniques d'ingénierie logicielle (XP est un support intéressant, les Design patterns indispensables, ...)
- ❧ Conditions propices nécessaires
- ❧ Principal bénéfice : des équipes motivées

# Perspectives

## ❧ Défauts à palier

- ❧ Absence de dépendance entre les tâches

- ❧ Polyvalence des programmeurs

⇒ Grande **maturité** nécessaire

## ❧ Contrats à adapter

## ❧ Stratégie d'introduction de Scrum en entreprise

# Agilité et Systèmes d'information

LeMondelInformatique  
Toute l'info et les tendances du monde IT

19/9/11 : Agilité des SI, la révolution de l'adaptabilité permanente

- ❧ **Face aux évolutions permanentes de leur business, les entreprises misent sur l'agilité. ....**
- ❧ L'adoption des méthodes agiles itératives reste plus que jamais d'actualité. En s'inspirant de Scrum, XP ou Unified Process, on accélère la délivrance de chaque projet en accord avec les priorités des métiers. Parallèlement, une démarche de Lean Management implique plus fortement les développeurs vis-à-vis de la qualité au meilleur coût.

Article à lire (beaucoup d'autres informations)

# Développements Agile

LeMondInformatique

Toute l'info et les tendances du monde IT

27/2/2013

## Une meilleure gestion des priorités

En ce qui concerne les avantages d'agile, 90 % des personnes interrogées ont déclaré que la méthode leur avait permis d'améliorer leur capacité à gérer les changements de priorités. La visibilité du projet a également été citée comme aspect positif d'agile. Enfin, les principales raisons ayant poussé les développeurs à adopter agile sont une mise sur le marché plus rapide, une gestion plus facile des changements de priorités, et une meilleure cohérence entre l'aspect commercial et l'IT.

Les entreprises ne limitent pas l'usage d'agile à de petites équipes et à des projets uniques. Par rapport à l'enquête réalisée il y a un an par VisionOne, le recours à la méthode a augmenté de 15 % et les personnes interrogées ont déclaré qu'elles menaient de front au moins cinq projets agile. Et celles qui prévoient de recourir à agile dans des projets de développement sont passées de 59 % à 83 % en un an. Enfin, plus de 84 % de leurs projets en cours sont agile, contre 80 % en 2011.

<http://www.lemondeinformatique.fr/actualites/lire-developpement-agile-pourquoi-ca-coince-encore-52655.html>

# Développements Agile

LeMondInformatique

Toute l'info et les tendances du monde IT

27/08/2014

Le 27 Aout 2014

## DevOps et méthodes agiles, piliers de l'efficacité IT de Voyages-SNCF.com

En termes de management, la réflexion sur l'arrivée des méthodes agiles au sein de VSCT a abouti à la nécessité de mettre en place d'équipes multicompetences et pluridisciplinaires permettant de mettre ensemble architectes, développeurs, scrum master afin de lui octroyer une certaine autonomie multi technologique et métier. Avec pour objectif que chacun puisse être en mesure de former ceux qui n'ont pas l'expertise ad hoc.

Méthodes agiles et DevOps ont été mis en place pour optimiser les développements sans faire l'impasse sur la qualité de service, inscrite en tant que préoccupation numéro 1 de l'ensemble des équipes de Voyages-SNCF.com et de VSCT et ce, dans un contexte où les volumétries à gérer sont conséquentes avec 66 millions de connexions par mois et jusqu'à 22 billets achetés par seconde en pic.

<http://www.lemondeinformatique.fr/actualites/lire-devops-et-methodes-agiles-piliers-de-l-efficacite-it-de-voyages-sncfcom-58435.html>

# Backlog

TODO

DOING

DONE

valeurs de scrum

organisations

Rôles

Artefacts

Réunions

Environnement

Conclusion

Bonus



mercredi 10 septembre 14

# Retour sur la méthodologie agile (1)

Ce stage a été l'occasion de pratiquer la méthodologie agile. Cette méthodologie comporte de très nombreux avantages. Elle **limite les risques pour les clients** de ne pas être satisfaits parfaitement par l'application, elle fait des miracles pour la **compréhension client-développeurs** en permettant de décrire les besoins du client en se basant sur une application existante développée lors de l'itération précédente. Elle limite la nécessité d'avoir un client expérimenté dans le développement. Elle fait **gagner beaucoup de temps** lors de la capture des exigences et **améliore la qualité** des exigences exprimées.

Par  
Pierre Neu,  
Stage Ingénieur  
2011, Amadeus

# Retour sur la méthodologie agile (2)

Le développement agile ne fonctionne que si les développeurs sont **capables de modifier et de refactoriser** l'application existante. Cela doit rester une préoccupation majeure des

développeurs et il ne faut pas se laisser déborder par les exigences du client, en sachant préserver du temps pour réfléchir à l'amélioration du code existant. Cela peut parfois nécessiter d'avoir une capacité du côté des développeurs à faire face et à savoir convaincre les clients de faire passer cette refactorisation avant le développement de nouvelles fonctionnalités, car le client ne perçoit pas les bénéfices de cette étape, malgré qu'elle soit cruciale pour la maintenabilité et la qualité du produit sur le long terme en général.

Par  
Pierre Neu,  
Stage Ingénieur  
sept. 2011,  
Amadeus

# Retour sur la méthodologie agile (3)

Cette méthodologie ne doit cependant pas faire disparaître la documentation et les spécifications. Car ces documents permettent de garder un cap et des objectifs pour l'application. Ils constituent malgré tout un guide primordial pour les développeurs. Les interactions entre le client et les développeurs sont très fréquentes et nécessaires. Il ne faut cependant pas perdre de vue les objectifs de ces réunions qui doivent rester courtes. A mon avis il était très positif dans mon cas qu'elles soient organisées par quelqu'un d'extérieur, c'est-à-dire ni un client ni un développeur, pour garder une vision plus neutre.

De plus ces échanges fréquents ont parfois comme résultat un trop grand nombre de demandes de modifications du client, ou de besoins très volatils. L'expérience des développeurs est alors très importante pour évaluer ces besoins et échanger avec le client sur leur réelle nécessité, et prioriser ensemble ces nouveaux besoins, pour garder un cap dans le développement.

Les développeurs doivent aussi être capables de fournir une estimation du surcoût induit par un changement de besoin du client.

# Retour sur la méthodologie agile (4)

Les limites de la méthodologie agile existent. Les projets de grande ampleur ne sont par exemple pas des candidats idéaux pour la méthodologie agile. Je pense cependant qu'il est possible d'adapter la méthodologie agile pour conserver certains de ses atouts dans des projets de grande taille.

Cette méthodologie ne doit également pas être un va-tout lorsque les clients ont du mal à décider de leurs besoins et n'ont pas une vision claire de l'application voulue. Même si l'Agile demande moins de documentation au départ, elle requiert un réel et continu investissement de la part des clients pour fonctionner. Les autres limites sont liées justement à l'absence de contrat. Comment arbitrer un conflit entre les clients et les développeurs ? De plus la planification du développement et l'estimation des coûts sont très difficiles, surtout lorsque les clients modifient les besoins.

Par  
Pierre Neu, Stage Ingénieur sept. 2011,  
Amadeus