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		
organisations		
Rôles		
Réunions		
Artefacts		
Environnement		
Conclusion		

Backlog

TODO	DOING	DONE
organisations Rôles Réunions Artefacts Environnement Conclusion	valeurs de Scrum	

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

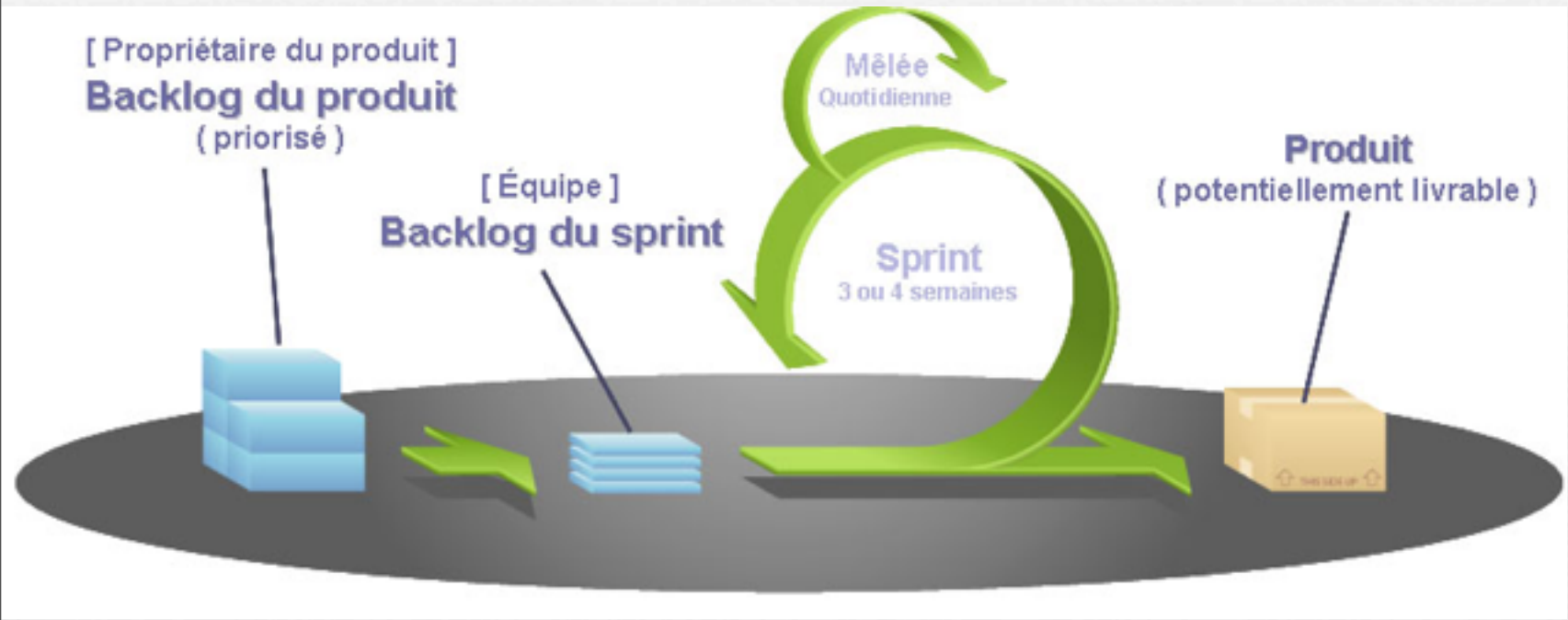
Scrum en 100 mots

- 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.

Backlog

TODO	DOING	DONE
Rôles	organisations	valeurs de SCRUM
Réunions		
Artefacts		
Environnement		
Conclusion		

Scrum : processus général



http://www.computure.net/fr/methodes/111-lessentiel-de-scrum-suite

Backlog

TODO	DOING	DONE
Réunions	Rôles	valeurs de SCRUM
Artefacts		organisations
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

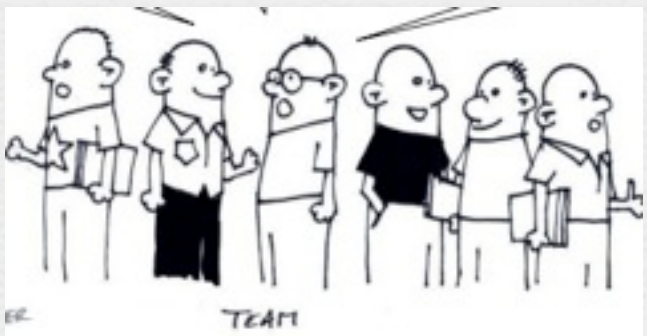
Les Rôles : le Scrum Master

- 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

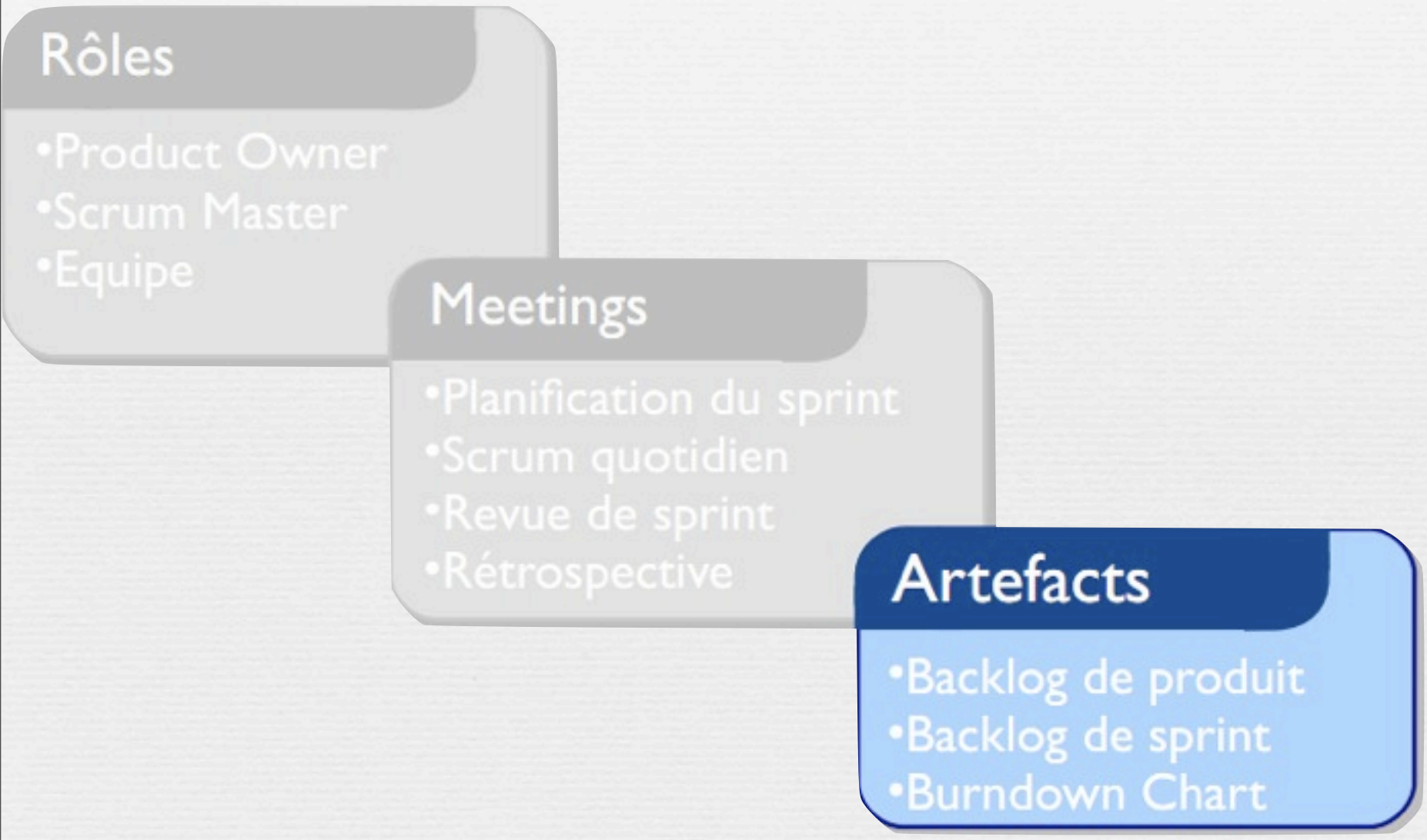
Martin (2003): "The team is in it for the long term. They work hard, at a pace that can be sustained indefinitely. They conserve their energy, treating the project as a **marathon** rather that a **sprint**."

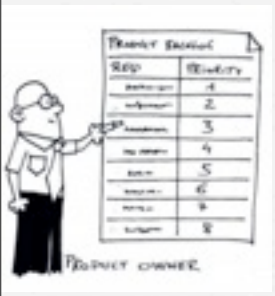


Backlog

TODO	DOING	DONE
Réunions	Artefacts	valeurs de scrum
Environnement		organisations
Conclusion		Rôles

Caractéristiques de Scrum





Artefacts : le Product Backlog ou PDL (Plan de Développement Logiciel)

- Recueil des exigences et des “User Stories”

  « En tant que <rôle>, je veux pouvoir <besoin>, afin de <bénéfice, plus-value>. »
- 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
- Les plus importantes sont les plus détaillées.

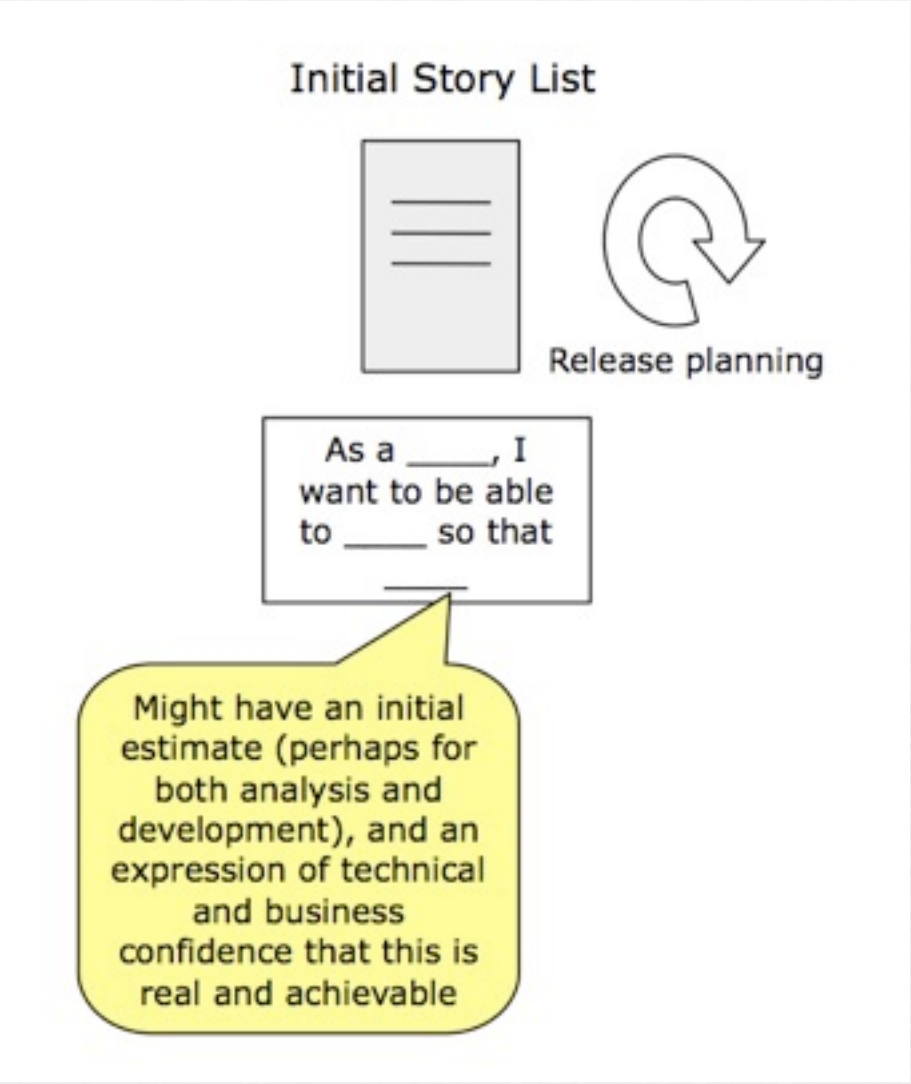
User stories / (Récits ou histoires d'utilisateur)

- 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

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

User stories

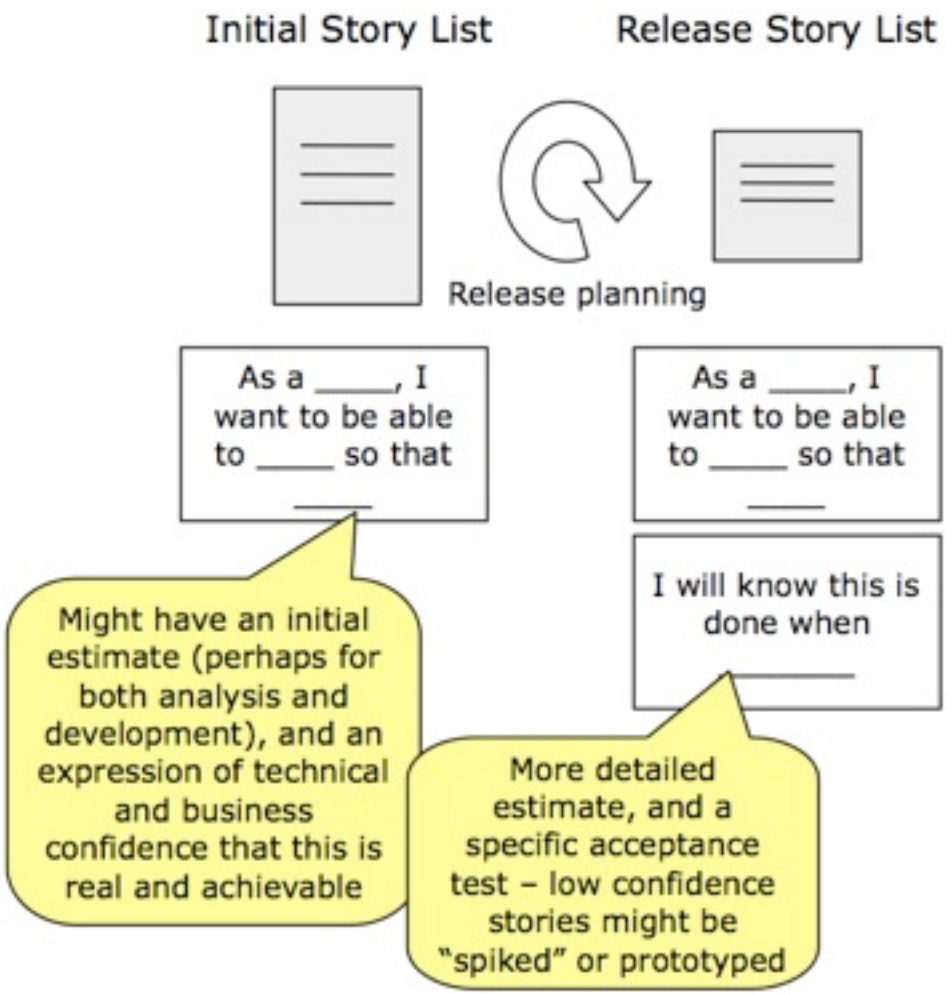


Au Recto :

- Fonctionnalité exigée sous la forme d'un «récit utilisateur»,
- Sa priorité attribuée par l'utilisateur et
- Le temps estimé par l'équipe pour sa réalisation.

(voir planning pocker)

User stories

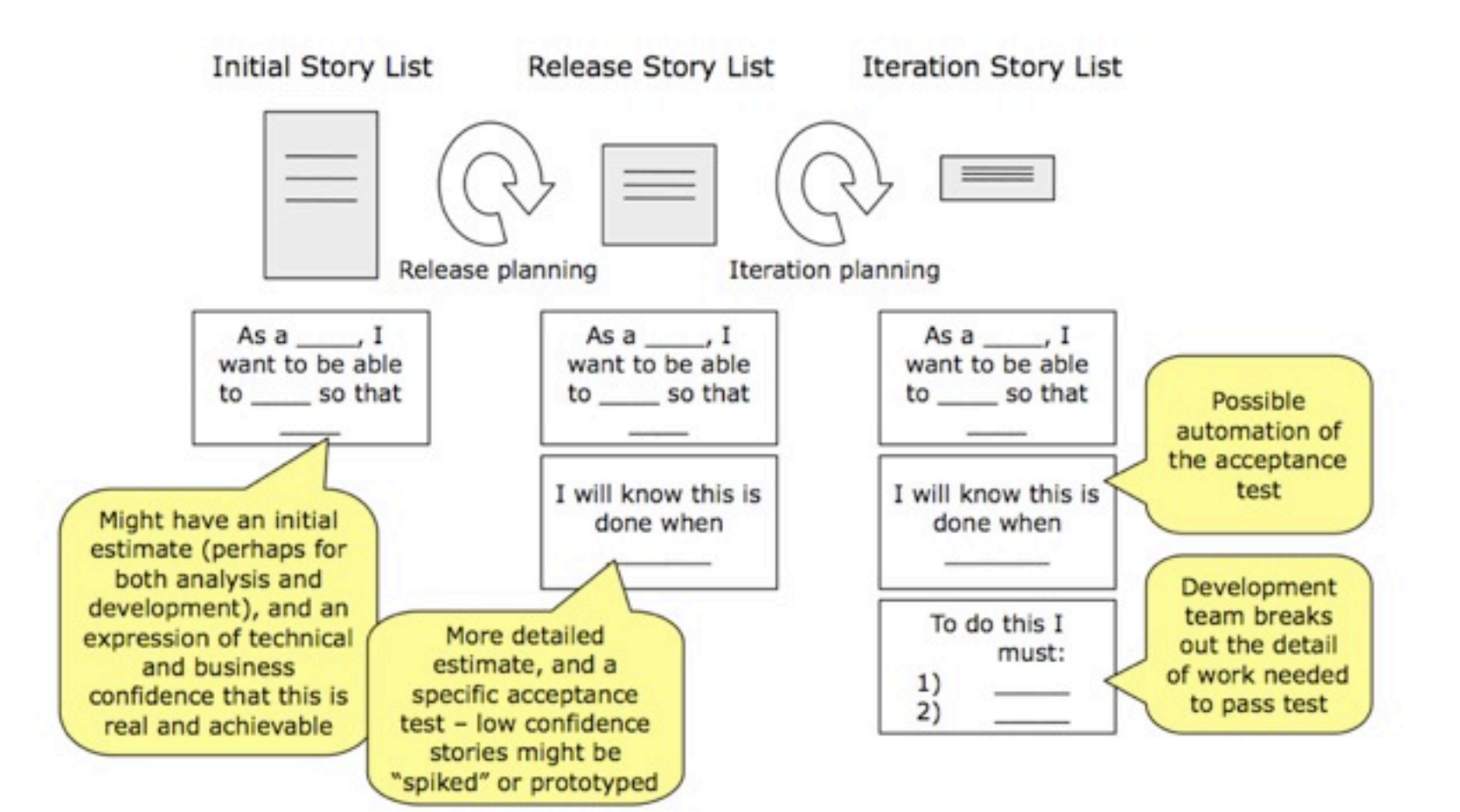


Au Verso :

- conditions d'acceptation définies par l'utilisateur
- niveau de risque (optionnel)
- modifications de la demande (optionnel)
-...

Estimation plus détaillée, et un test d'acceptation spécifique- des histoires imprécises peuvent être «enrichies» ou prototypées

User stories



L'équipe de développement détaille les travaux nécessaires pour passer le test

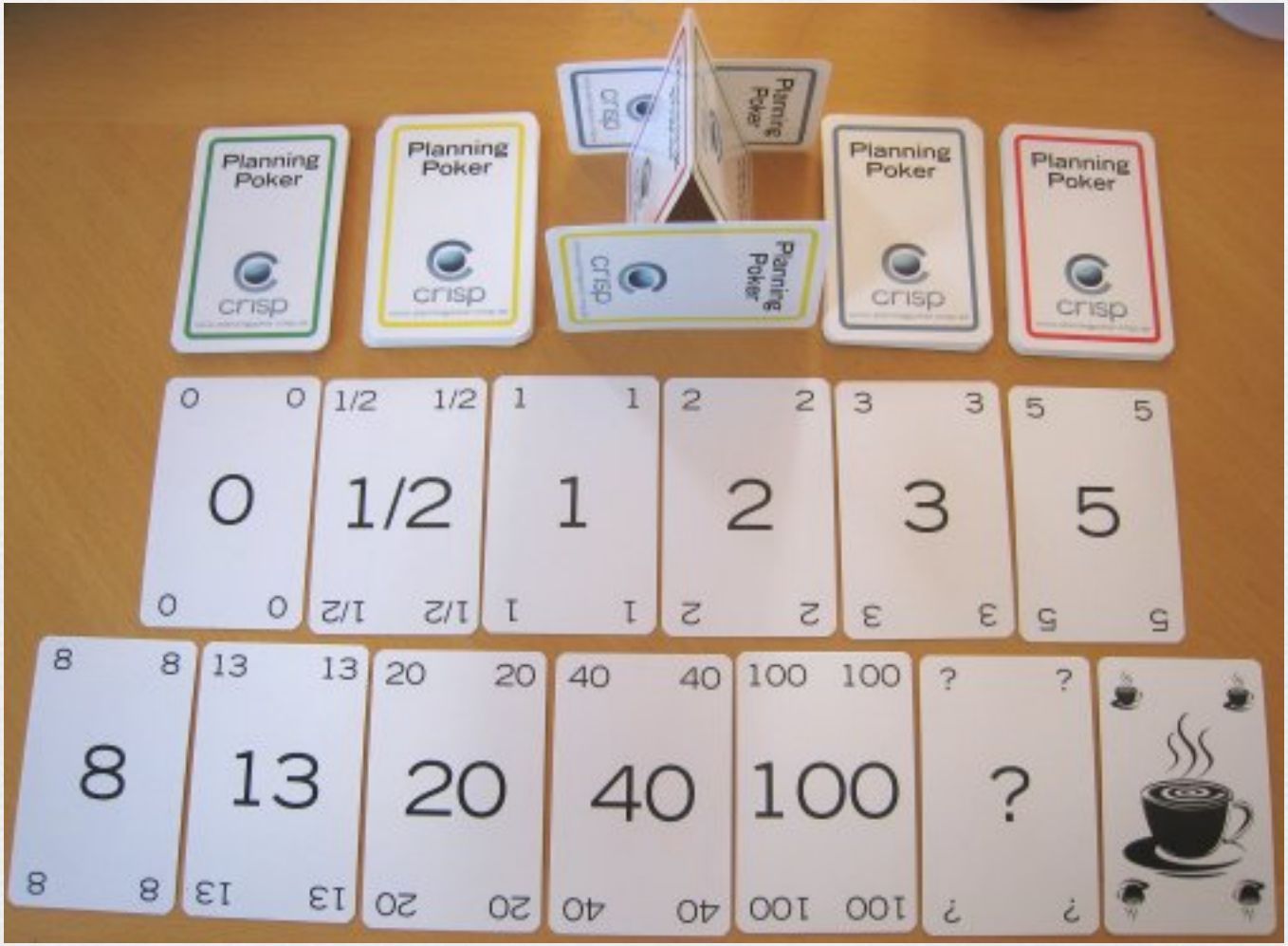
User Stories versus Use cases

- Une "User story" : un **but**, pas une **séquence d'actions**.
- Une "User story" souvent seulement **un des scénarios** du "Use case".
- Une "User story" émerge plus **rapidement** => moins d'analyse
- Les "User stories" plus **faciles à lire** (Mouais...)
- Les "User stories" sont rédigées (en principe) sur des cartes: leur durée de vie est limitée et elles n'ont pas pour vocation à être conservées (contrairement aux "Use cases"). Or, ce besoin de traçabilité peut être réel.
- Les "User stories" reposent sur un **mode oral, collaboratif**, de proximité : elles sont discutées (Customer / Developer) et tout n'est pas rédigé (contrairement aux "Use cases") => Pbme dans le cas de gros projets, équipe importante, offshore…
- Une "User story" doit être implémentée et testée en une itération. Un "Use case" peut être traité sur plusieurs itérations (scénario nominal sur une, scénarios alternatifs sur une autre) en fonction des risques à lever.
- ....

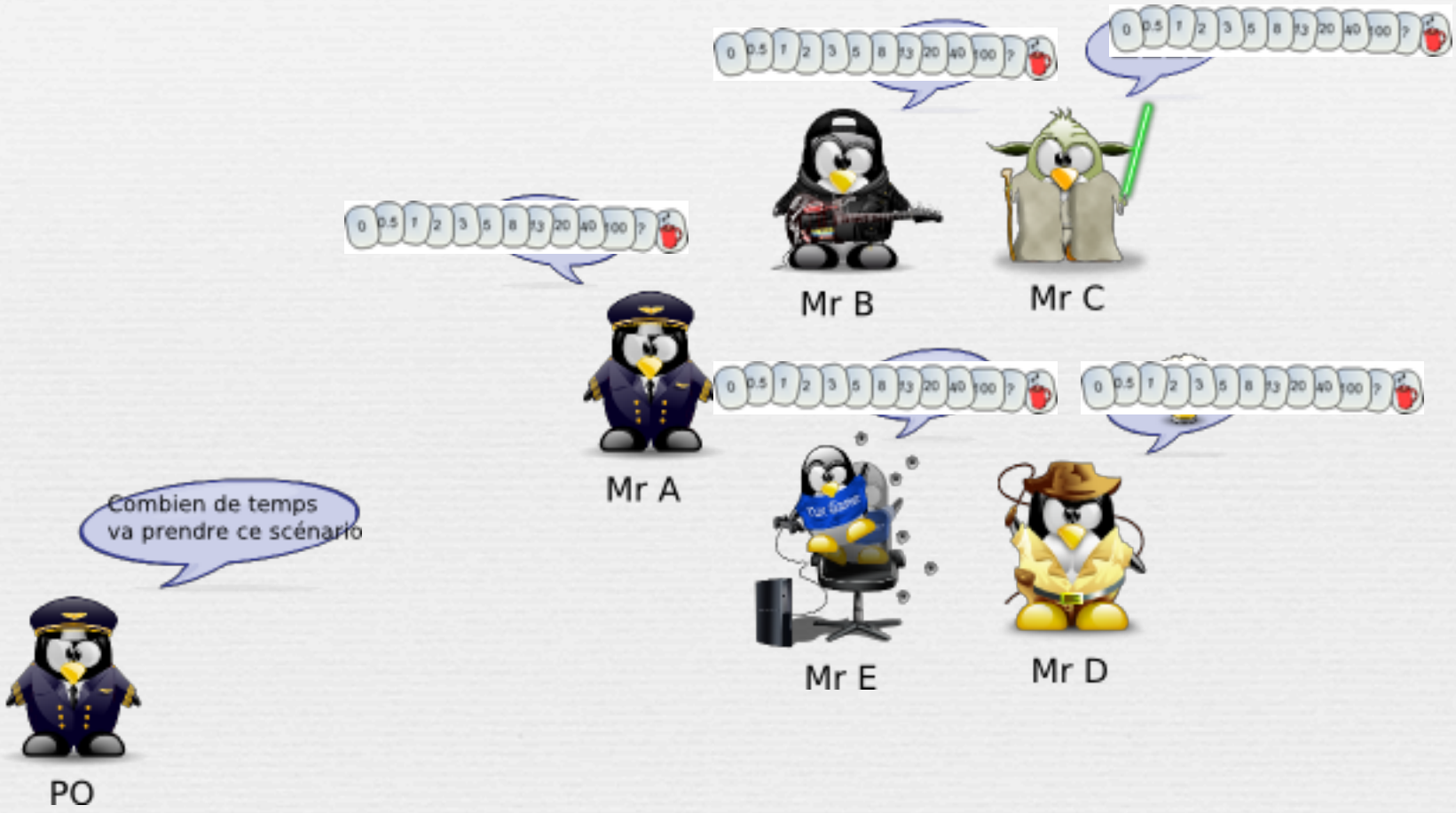
Estimation d'une US par les développeurs

Le planning Pocker

Planning Pocker

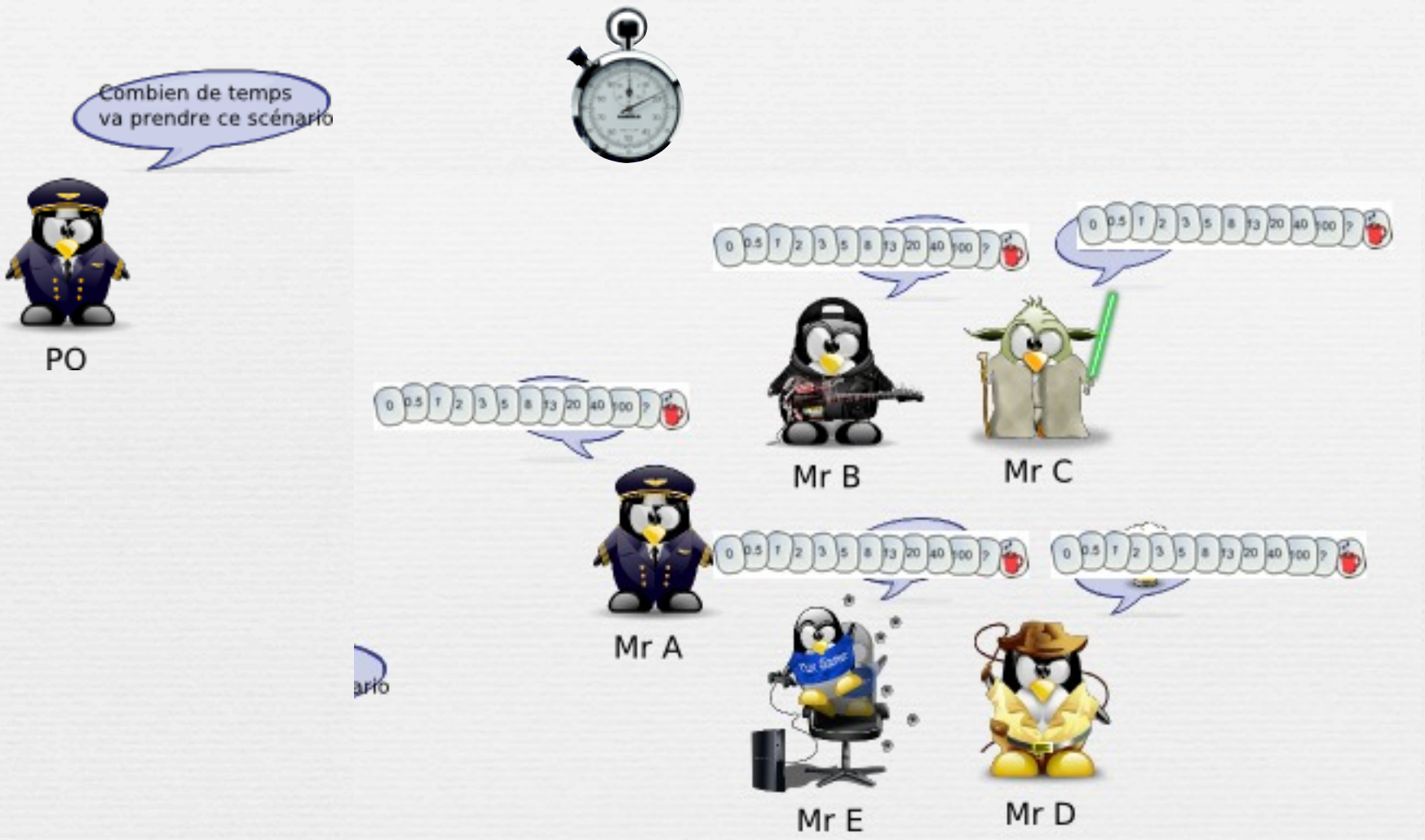


Planning Pocker

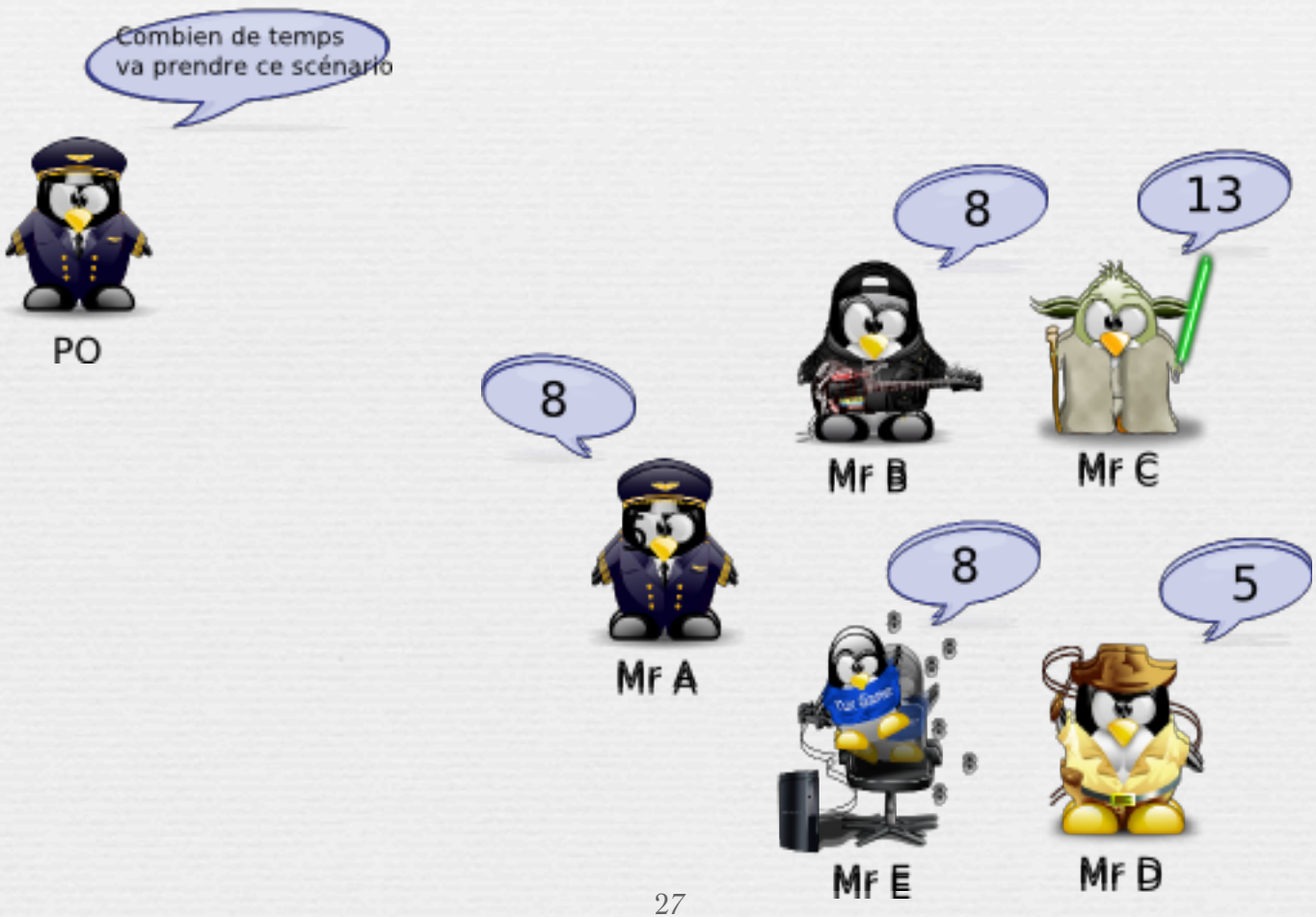


http://www.openagile.net/?p=155

Planning Pocker

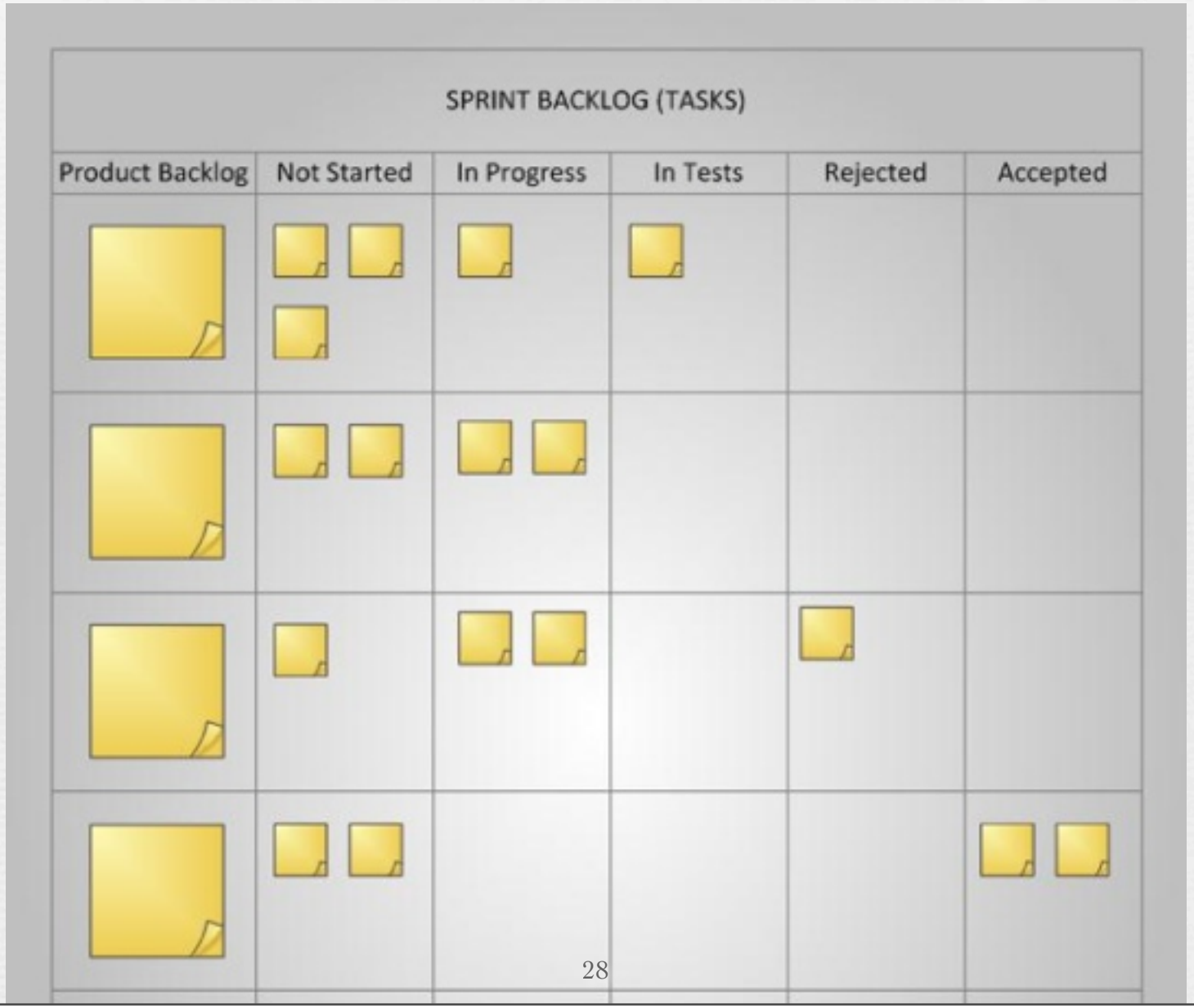


Planning Pocker



http://blog.valkir.net/2011/02/24/core-ingredients-of-scrum/

«Product Backlog»



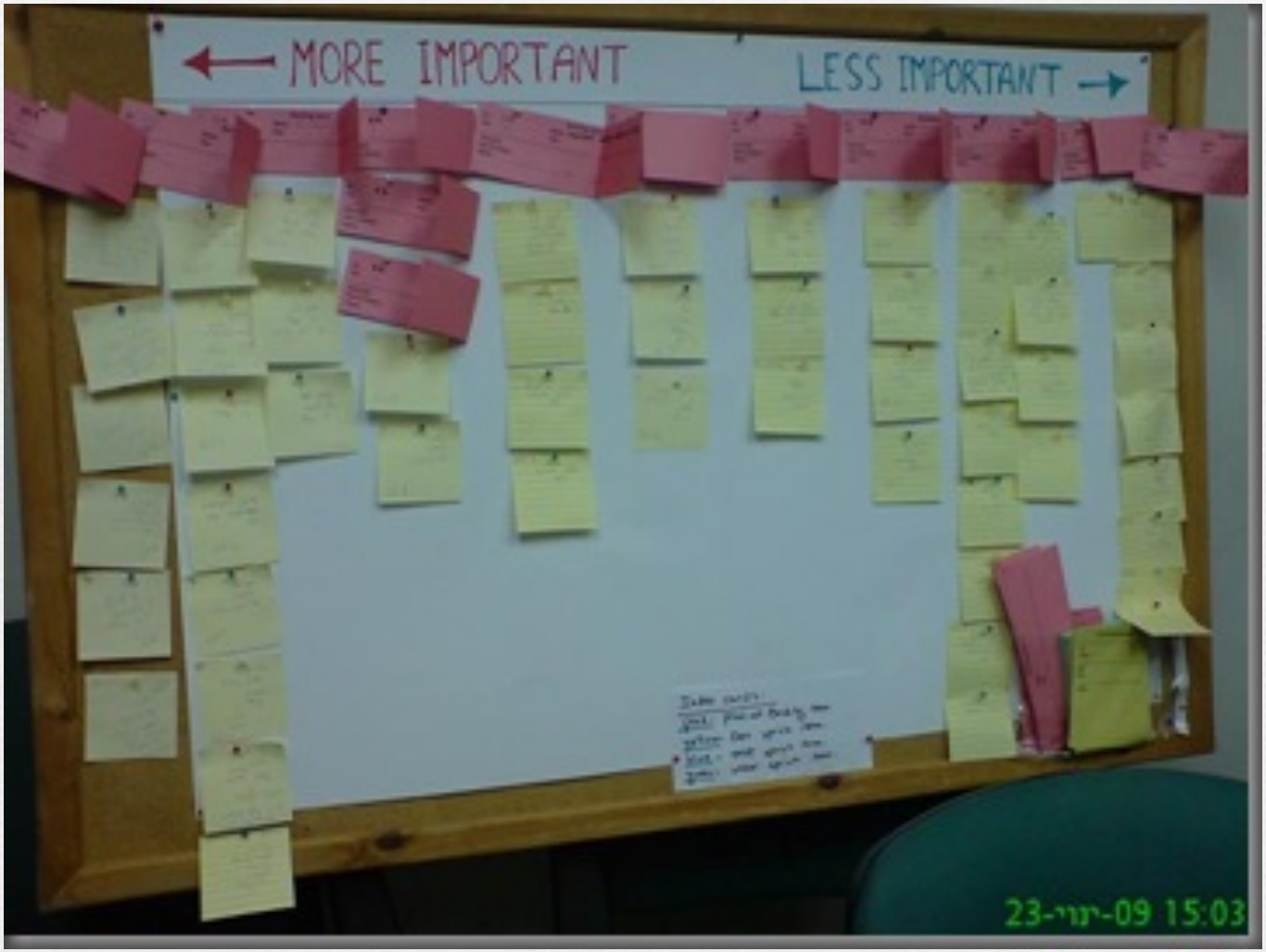
Exemple de Product Backlog

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.

Base du diagramme de suivi des tâches

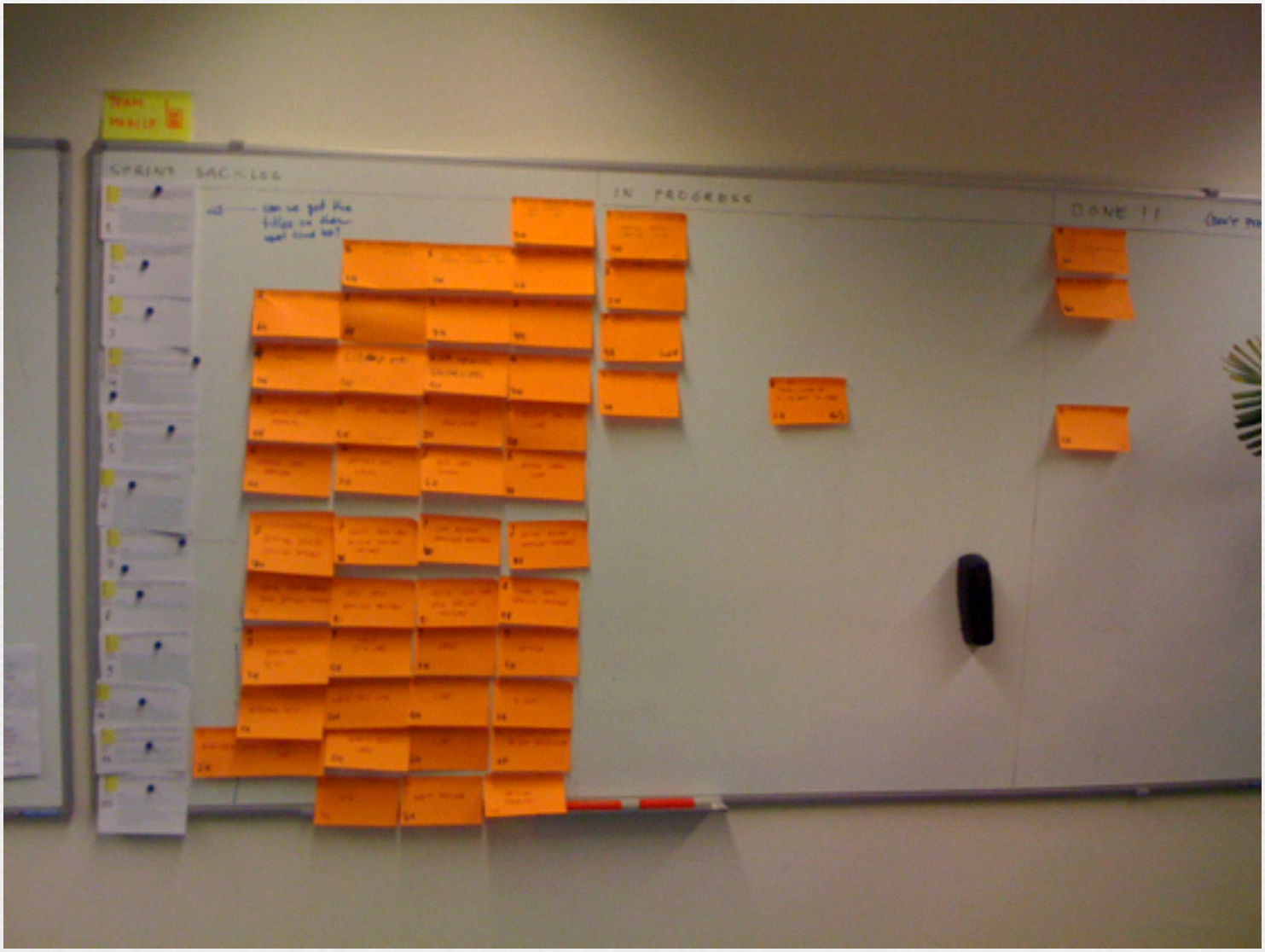
http://blogs.microsoft.co.il/blogs/kolbis/archive/2009/06/24/what-gives-scrum-at-nana10.aspx

Constuire le «Product Backlog»

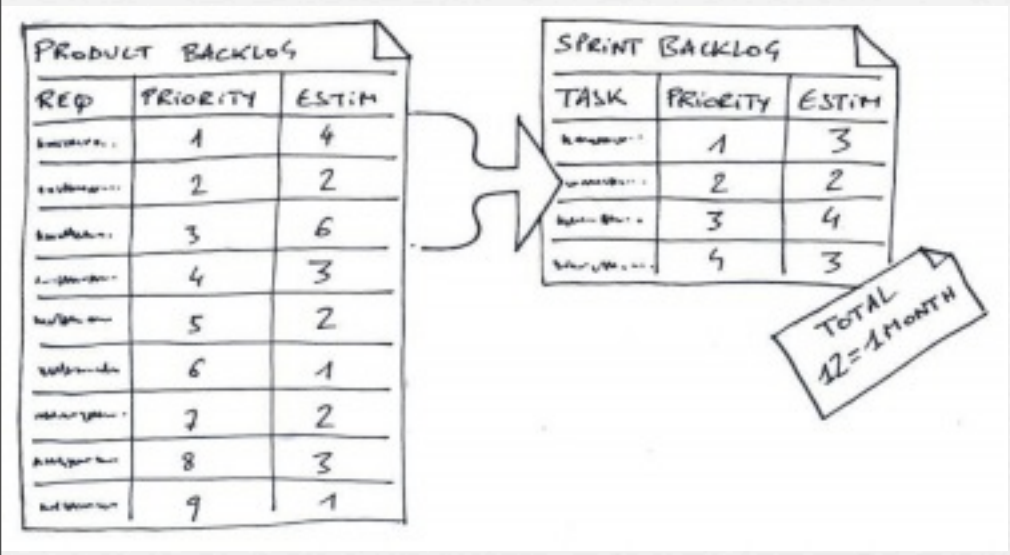


http://blogs.microsoft.co.il/blogs/kolbis/archive/2009/06/24/what-gives-scrum-at-nana10.aspx

«Product Backlog»



Artefacts : le sprint Backlog

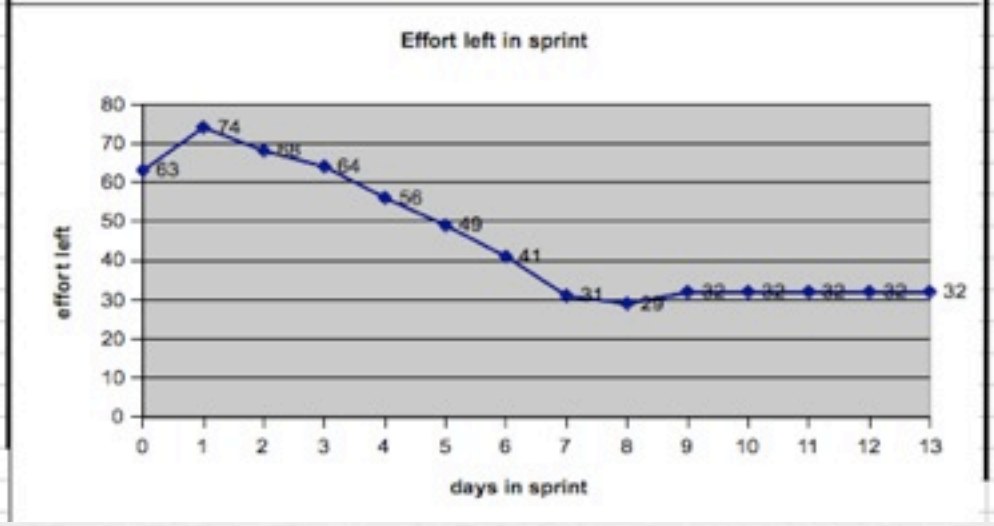


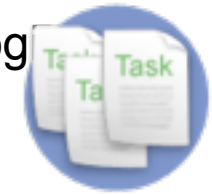
- Recueil des différentes tâches, extraites du Product Backlog, que l'équipe s'engage à réaliser lors du Sprint.
  - Le travail n'est jamais assigné par un autre
- L'estimation du reste à faire est ajustée chaque jour
- Si une tâche n'est pas claire, ou trop volumineuse, la décomposer en tâches plus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

http://agilesoftwaredevelopment.com/system/files/SimpleProductBacklog.xls

Sprint Backlog (autre version)

Story ID	Story/task	0	1	2	3	4	5	6	7	8	9	10	11	12	13
		63	74	68	64	56	49	41	31	29	32	32	32	32	32
10	**Fetch one day temperature data from the weather provider system**														
	Make our server connect and authenticate to the provider system	4	16	12	8	3	3	3	3	3	3	3	3	3	3
	Read provider's data directory	8	7	7	7	4	0	0	0	0	0	0	0	0	0
	Parse the current temperature out of the data	6	6	4	4	4	1	1	1	1	1	1	1	1	1
	Push the temperature data to the client	16	16	16	16	16	16	8	2	0	0	0	0	0	0
11	**Fetch rain, snow, etc details from the provider**														
	Parse snow/rain data from the provider's data	4	4	4	4	4	4	4	0	0	0	0	0	0	0
	Push the snow/rain data to the client	4	4	4	4	4	4	4	4	4	0	0	0	0	0
	Redesign client screen a bit										3	3	3	3	3
	Refactor the server code										4	4	4	4	4
12	**Fetch several days data from the provider**														
	Parse the weather data in day packs	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	Push several days data to the client	3	3	3	3	3	3	3	3	3	3	3	3	3	3
13	**Auto-refresh feature**														
	Make the client ping server once per 4 hours	6	6	6	6	6	6	6	6	6	6	6	6	6	6
	Make the server update the client	2	2	2	2	2	2	2	2	2	2	2	2	2	2





Monitor the task board

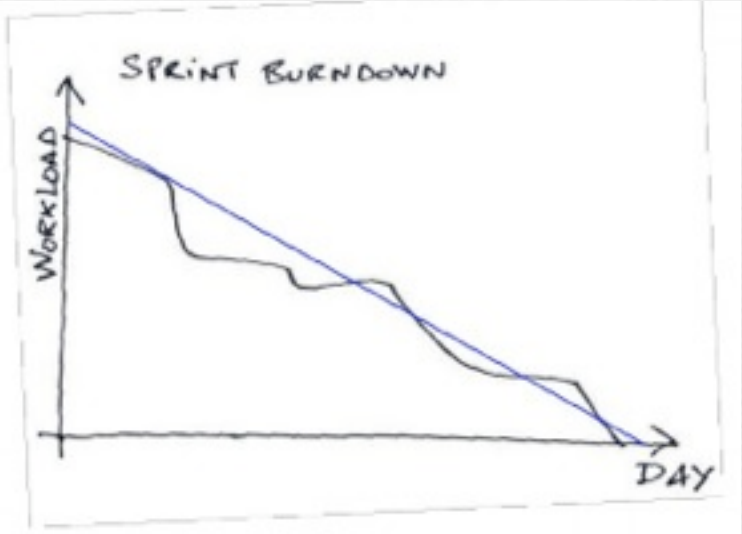
Story	To Do	Tests Ready	In Process	To Verify	Hours
A user can… 5	Code the … 8 Code the … 5 Test the … 6	X	Code the … SC 6 Code the … DC 4	Code the … LC 2	31
A user can… 2	Code the … 8 Code the … 5				13
A user can… 3	Code the … 3 Code the … 6	X	Code the … MC 4		13

http://blogs.microsoft.co.il/blogs/kolbis/archive/2009/06/24/what-gives-scrum-at-nana10.aspx

Sprint Backlog (autre version)

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.

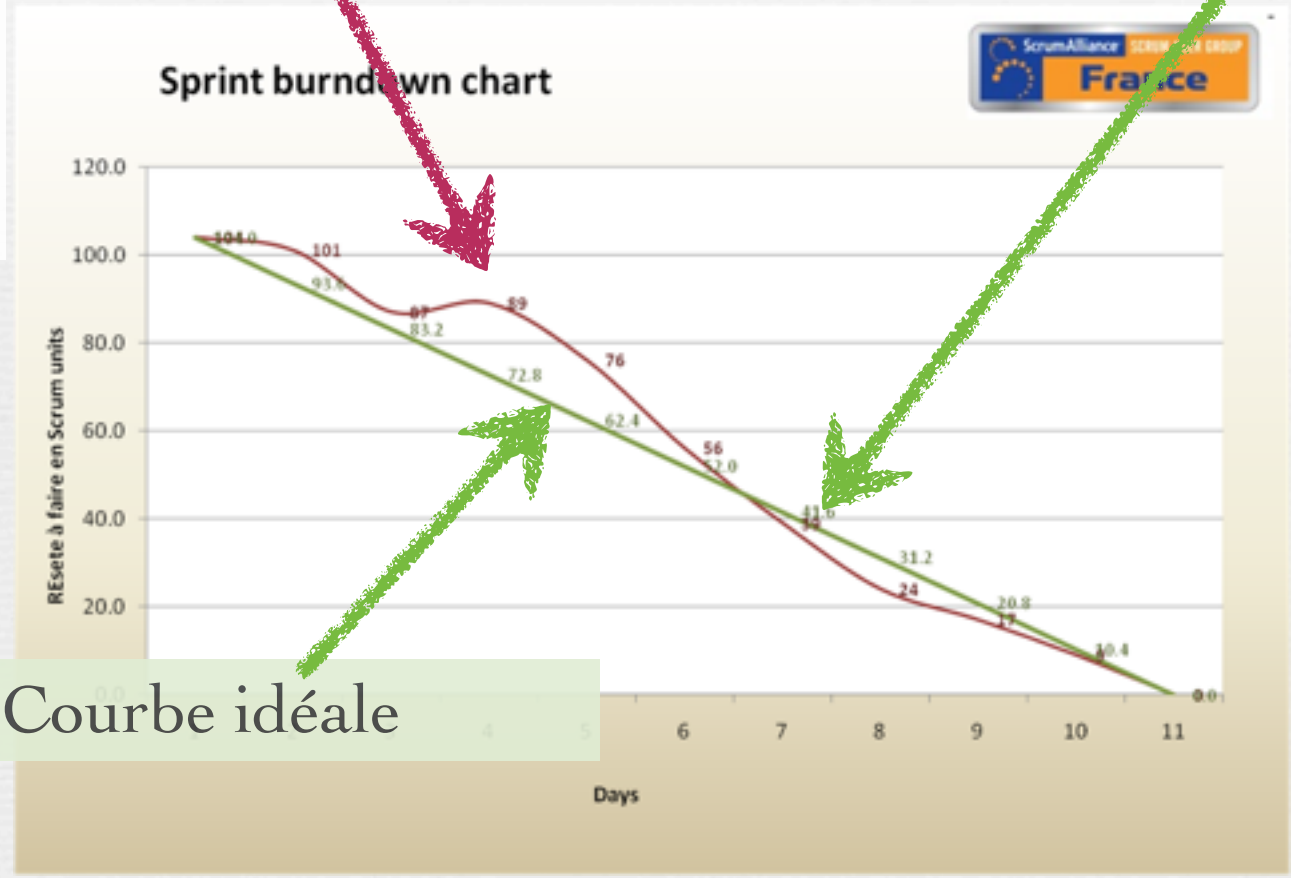
La ligne courbe (en noir) représente la "Vélocité" véritable de l'équipe.

Après chaque Daily Scrum Meeting, en fonction des travaux de la veille de chacun, le Burndown Chart est mis à jour

Un pbme est détecté au 4è jour; la charge est réévaluée

Accélération

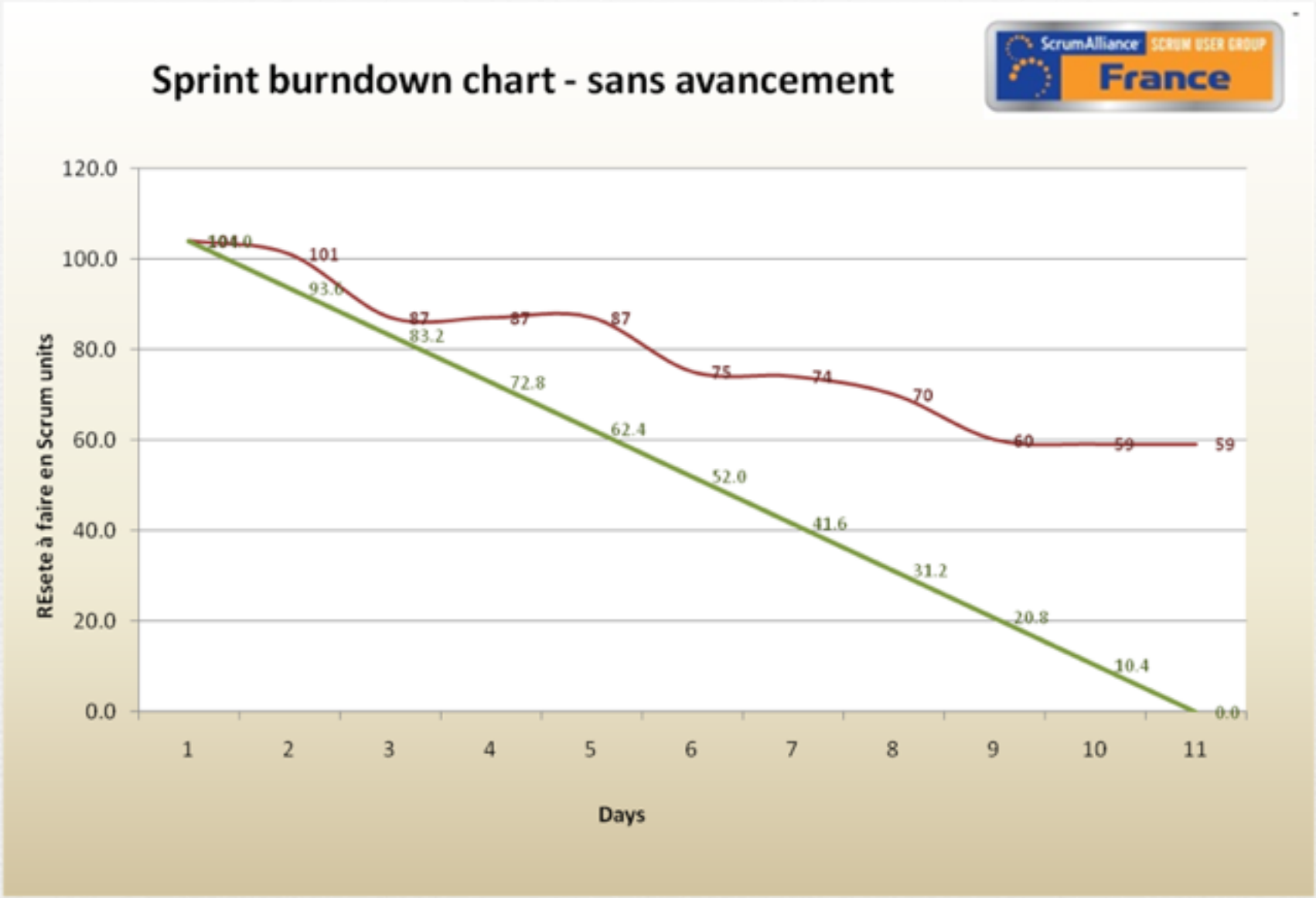
104 points scrum



Courbe idéale

Durée du sprint fixée à 10 jours

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



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

Backlog

TODO	DOING	DONE
Environnement	Réunions	valeurs de scrum
Conclusion		organisations
		Rôles
		Artefacts

Caractéristiques de Scrum

Rôles

- Product Owner
- Scrum Master
- Equipe

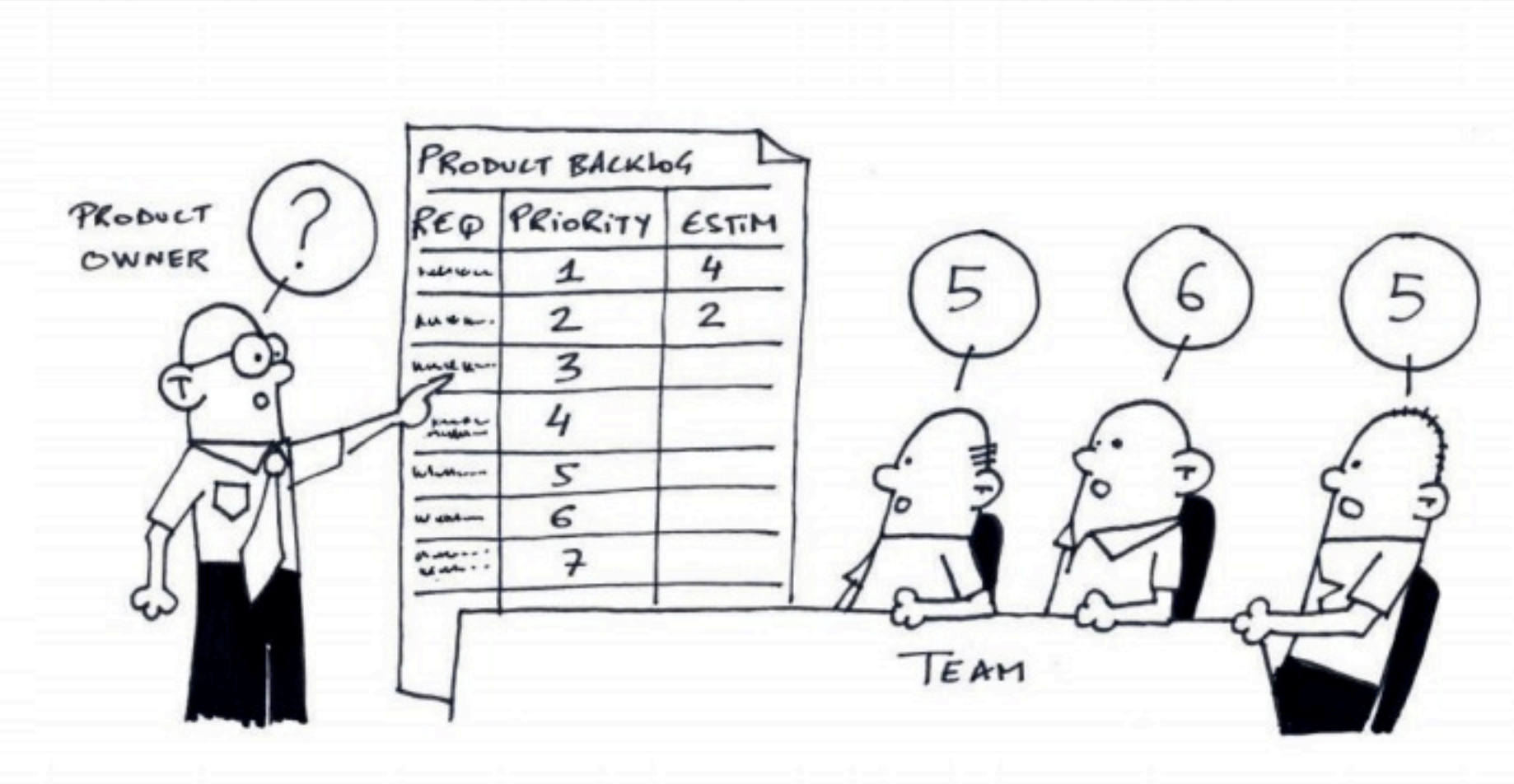
Meetings

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

Artefacts

- Backlog de produit
- Backlog de sprint
- Burndown Chart

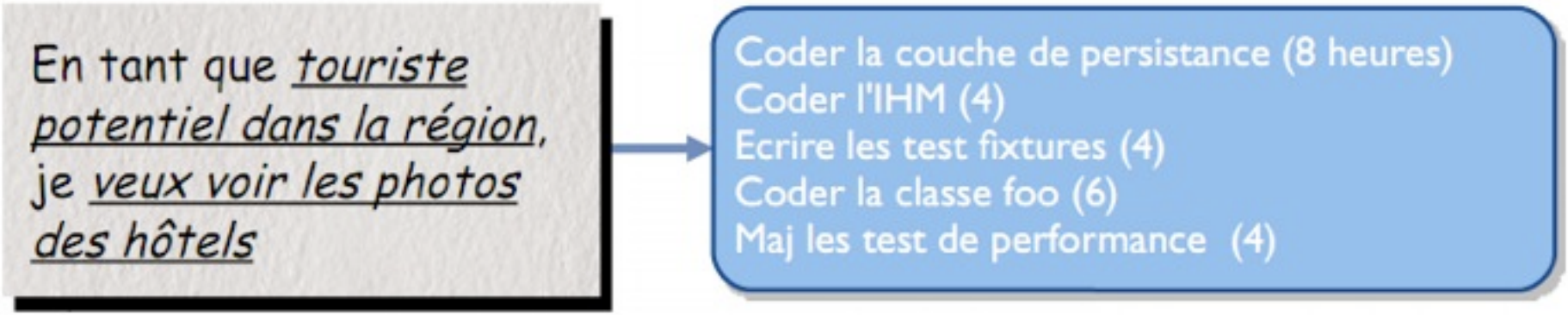
Meetings : Planification du sprint



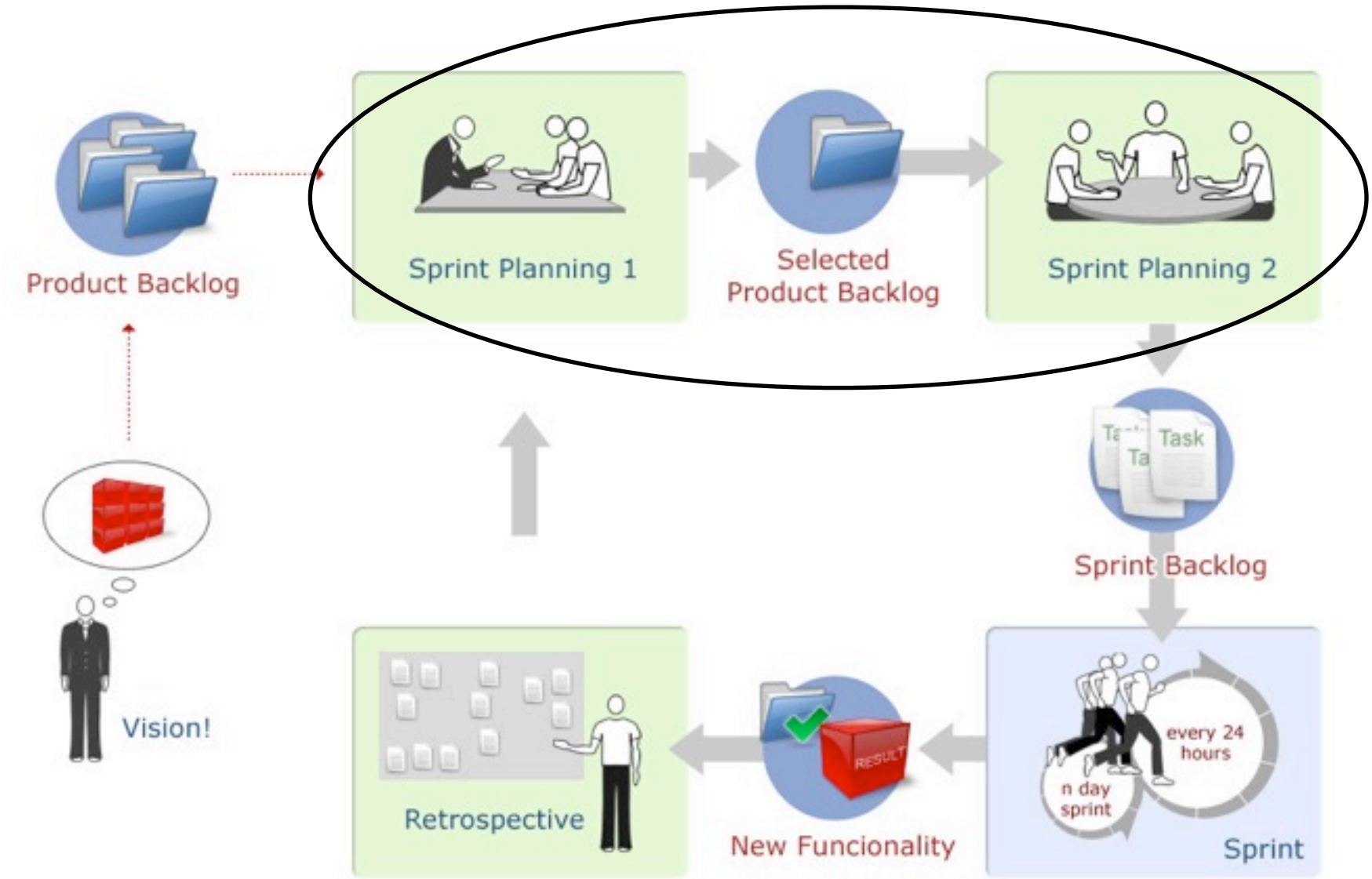
Meetings : Planification du sprint

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



ScrumFlow


Product Backlog
Sprint Planning 1
Selected
Product Backlog
Sprint Planning 2
Task
Sprint Backlog
every 24
hours
n day
sprint
Sprint
New Funcionality
RESULT
Retrospective
Vision!

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).

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

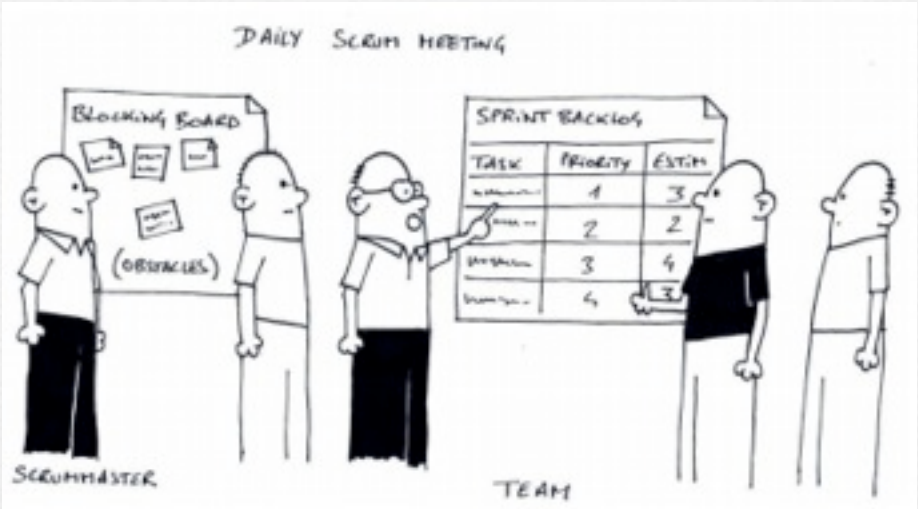
Sprint Planning 2 : une approche

- 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

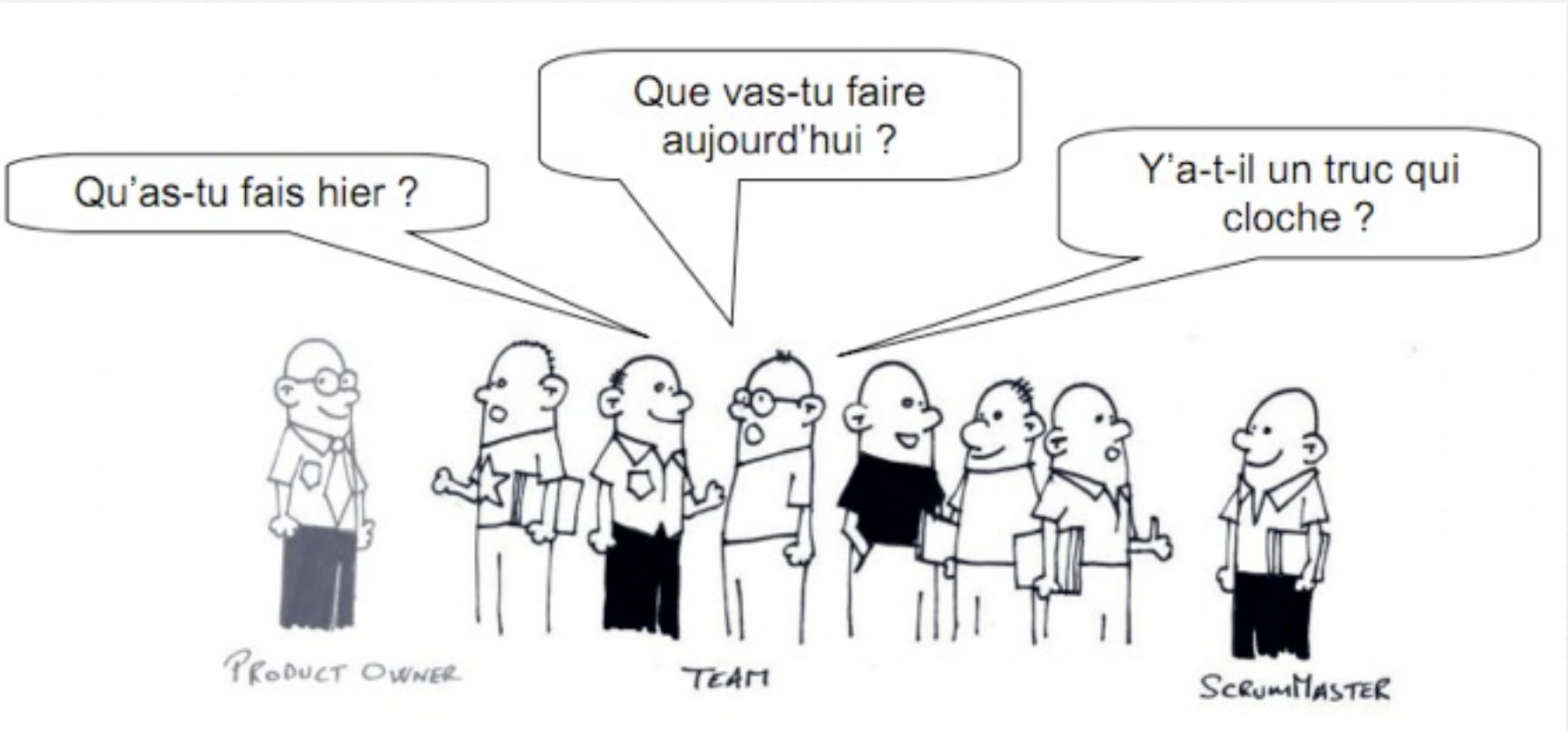
Meetings : Scrum quotidien

- 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 réunion

Meetings : Scrum quotidien



Il ne s'agit pas de compte-rendus au Scrum Master.

=> Ce sont des engagements devant les pairs

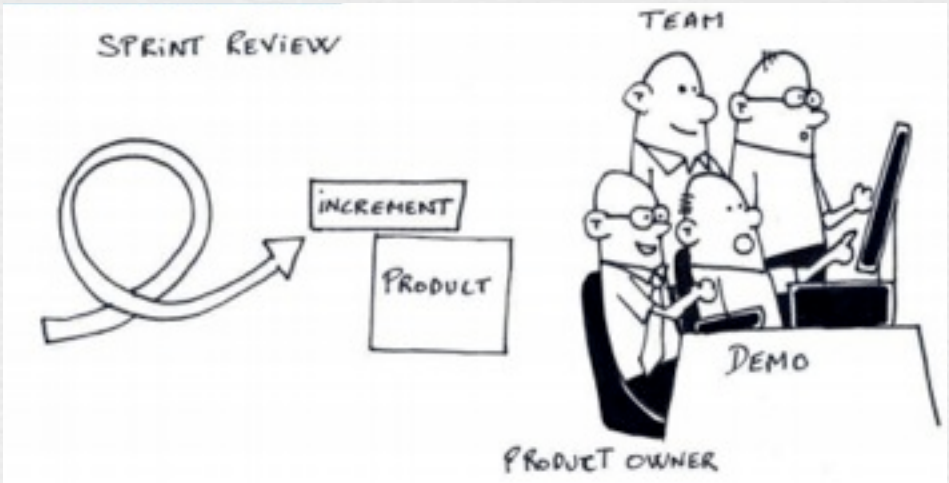
Scrum Teams

Rules of Etiquette:

- Never use the word "you" because the other person may feel on the spot and defensive.
- Never refer to history (e.g., "three months ago, you said...!").
- Be on time for meetings; if you are late, apologize and pay a late "penalty."
- If everyone is talking at once, use a pen to determine who talks. Whoever is holding the pen talks, everyone else listens.
- Everyone's opinion is important and needs to be understood and taken into account.
- No name calling.

Meetings : Revue de Sprint

- L'équipe présente ce qu'elle a fait pendant le Sprint.
- L'équipe effectue une démo des nouvelles fonctionnalités inclues 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.



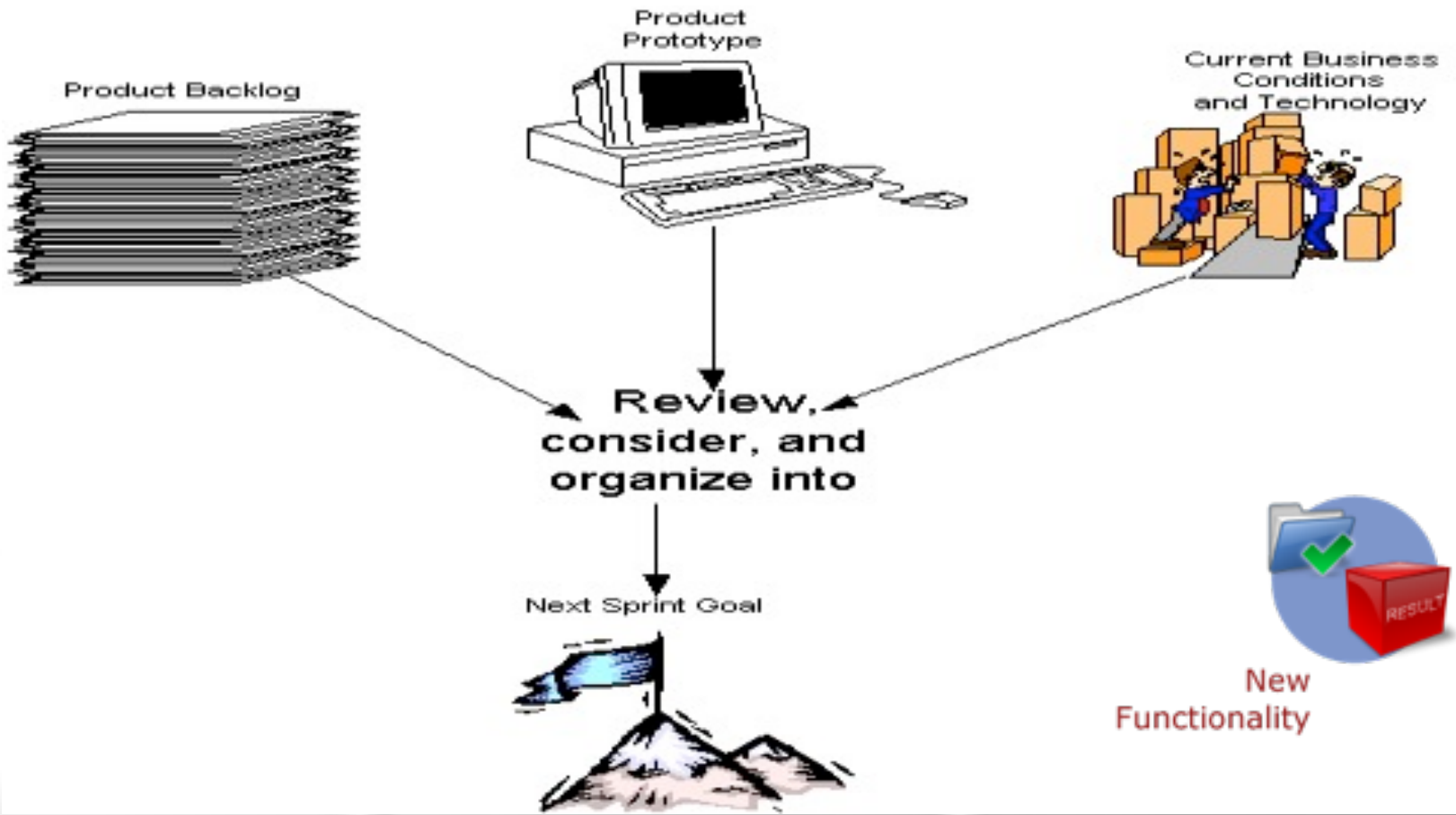
Mais que signifie TERMINE ? DONE?

- 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

Sprint Review Meeting



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

Evaluation Consequences

1. Restoring unfinished functionality to the Product Backlog and prioritizing it.
2. Removing functionality from the Product Backlog that the team unexpectedly completed.
3. Working with the ScrumMaster to reformulate the team.
4. Reprioritizing the Product Backlog to take advantage of opportunities that the demonstrated functionality presents.
5. Ask for a release Sprint to implement the demonstrated functionality, alone or with increments from previous Sprints.
6. Choosing not to proceed further with the project and not authorizing another Sprint.
7. Requesting that the project progress be sped up by authorizing additional teams to work on the Product Backlog.

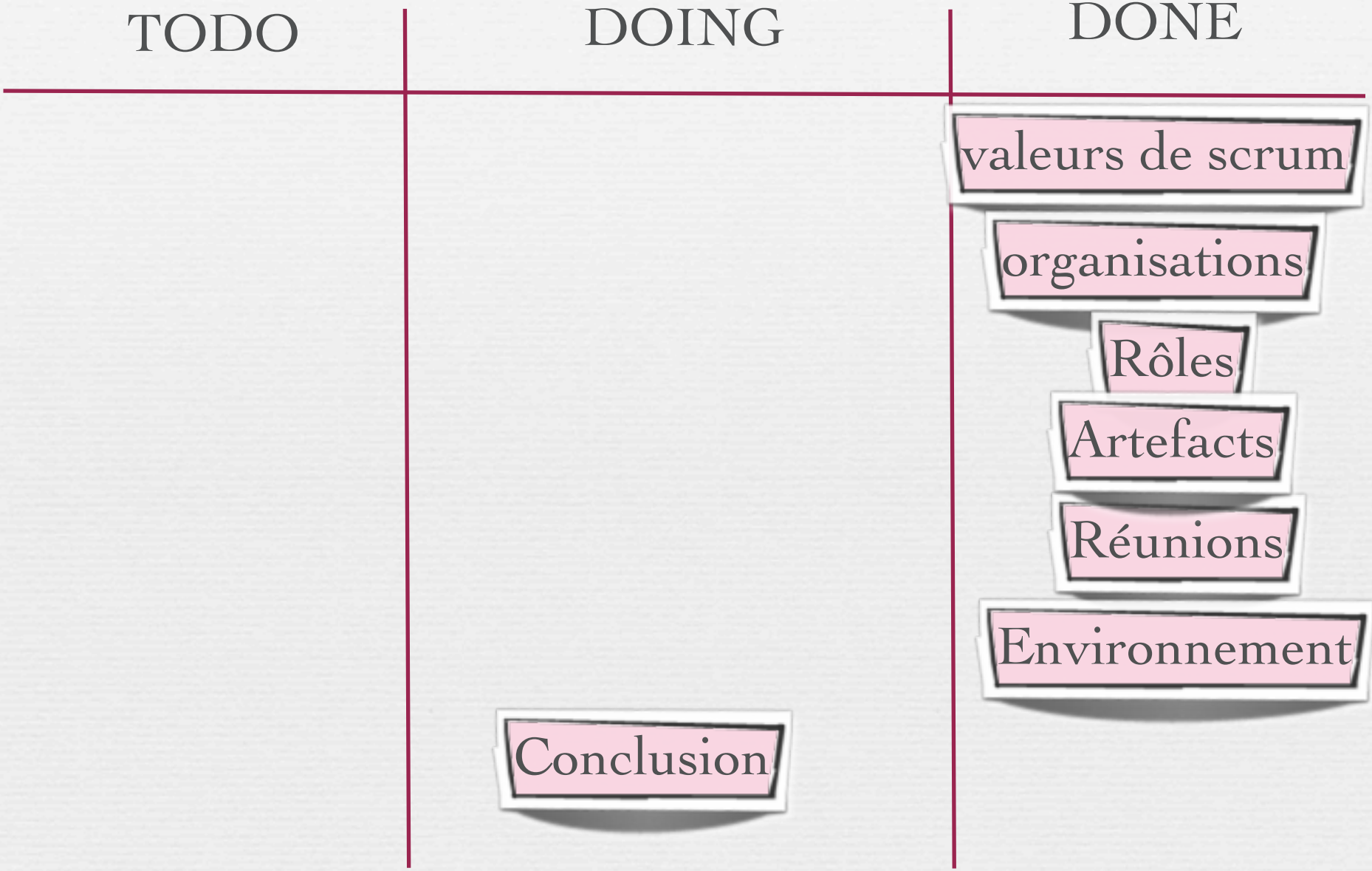
Backlog

TODO	DOING	DONE
Conclusion	Environnement	valeurs de scrum organisations Rôles Artefacts Réunions

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)
- Conditions propices nécessaires
- Expérimentations prometteuses
- Principal bénéfice : des équipes motivées

Perspectives

- Défauts à palier
  - Absence de dépendance entre les tâches
  - Polyvalence des programmeurs
  - Productivité équivalente supposée

  ⇒ Grande **maturité** nécessaire
- Contrats à adapter
- Stratégie d'introduction de Scrum en entreprise

Agilité et Systèmes d'information

LeMondeInformatique.fr
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)

Backlog

TODO	DOING	DONE
	Bonus	valeurs de scrum organisations Rôles Artefacts Réunions Environnement Conclusion

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