

La méthode SCRUM

La méthode SCRUM

Fondamentaux et pratiques

Plan

- **Introduction à Scrum**
- **Rappel sur les méthodes agiles**
- **Scrum**
 - **Principes clés**
 - **Les rôles**
 - **Planifier un projet**
 - **Organisation**
 - **Indicateurs de projet**
 - **Techniques d'ingénierie logicielle**
 - **Scrum pour de grandes équipes**
- **Perspectives**
- **Conclusion**

Plan

Introduction

Introduction

- **Contexte et origines**
- **La gestion de projet classique**
- **Rappel sur les méthodes agiles**
- **Le manifeste Agile**

Contexte et origines

- **Qu'est-ce qu'un projet ?**

Contexte et origines

- **Particularités des projets de développement de logiciels ?**

La gestion de projet classique

- **Particularités ?**
- **Déroulement ?**
- **Avantages et inconvénients ?**

Le manifeste Agile

- **Voir document**

Rappel sur les méthodes agiles

- Manifeste de l'agilité publié en 2001
- 4 valeurs :
 - Les personnes et les interactions plutôt que les outils et les processus
 - Le logiciel fonctionnel plutôt que de la documentation exhaustive
 - La collaboration avec le client plutôt que la négociation de contrat
 - L'adaptation au changement plutôt que le respect d'un plan pré-établi

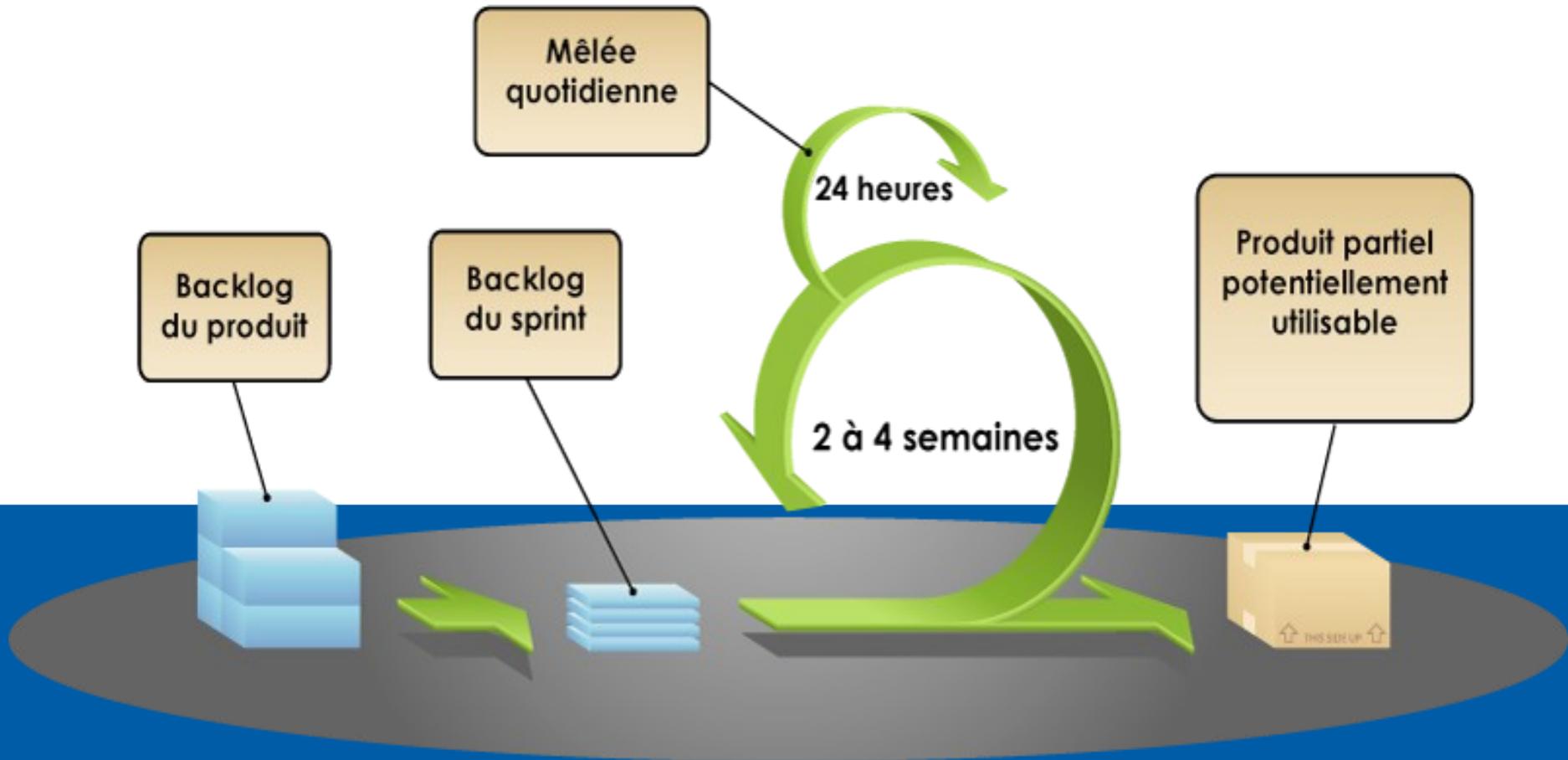
Plan

Les idées clés

Les idées clés

- **Tout scrum**
- **Le ScrumBok**
- **Les rôles**
- **Les backlogs**
- **L'équipe**
- **La planification**
- **Les burndowns charts**
- **Le feedback**

Tout Scrum



Introduction à Scrum



- **Scrum = mêlée en rugby**
- **Objectifs :**
 - **Satisfaire au mieux les besoins du client**
 - **Maximiser les chances de réussite du projet**
- **Méthode itérative et incrémentale**
- **Equipes de 8-9 personnes, mécanismes d'extension possibles**
- **Méthode agile la plus utilisée avec eXtreme Programming**

Scrum – La théorie

- **3 piliers :**
 - **La transparence**
 - **L'inspection**
 - **L'adaptation**

Scrum – Contenu

- **Le cadre Scrum :**
 - **Des équipes Scrum**
 - **Des blocs de temps (time-boxes)**
 - **Des artéfacts**
 - **Des règles**

Scrum – Contenu

- **Equipe Scrum**
 - **Product owner**
 - **ScrumMaster**
 - **Team members**

Scrum – Contenu

- **Des blocs de temps :**
 - **Réunion de planification de livraison**
 - **Réunion de planification de sprint**
 - **Sprint**
 - **Mêlée quotidienne**
 - **Revue de sprint**
 - **Rétrospective du sprint**

Scrum – Contenu

- **Des artéfacts :**
 - **Carnet de produit (product backlog)**
 - **Carnet de sprint (sprint backlog)**
 - **Graphique de progression (burndown chart)**
 - **Graphique de progression de livraison (release burndown chart)**
 - **Graphique de progression de sprint (sprint burndown chart)**

Scrum – Contenu

- **Des règles :**
 - **Elles lient les blocs de temps, les rôles et les artéfacts**
 - **Exemple : seuls les membres de l'équipe peuvent parler pendant la mêlée quotidienne**

Scrum – Principes clés

- Conforme au manifeste de l'agilité
- Met l'accent sur :
 - Auto-organisation de l'équipe
 - Pouvoir de décision donné à l'équipe
 - Délais fixes
 - Sprint en isolement
 - Réunions quotidiennes
 - Livrer un logiciel fonctionnel - démonstration du résultat du sprint
 - Planning adaptatif

Sprints

- **Les projets Scrum progressent par une série de sprints**
 - **Equivalents aux itérations XP**
- **La durée d'un sprint est de 2 à 4 semaines**
- **Une durée constante apporte un meilleur rythme**
- **Le produit (partiel) est conçu, codé et testé pendant le sprint**

La séquence est différente

Exigences

Conception

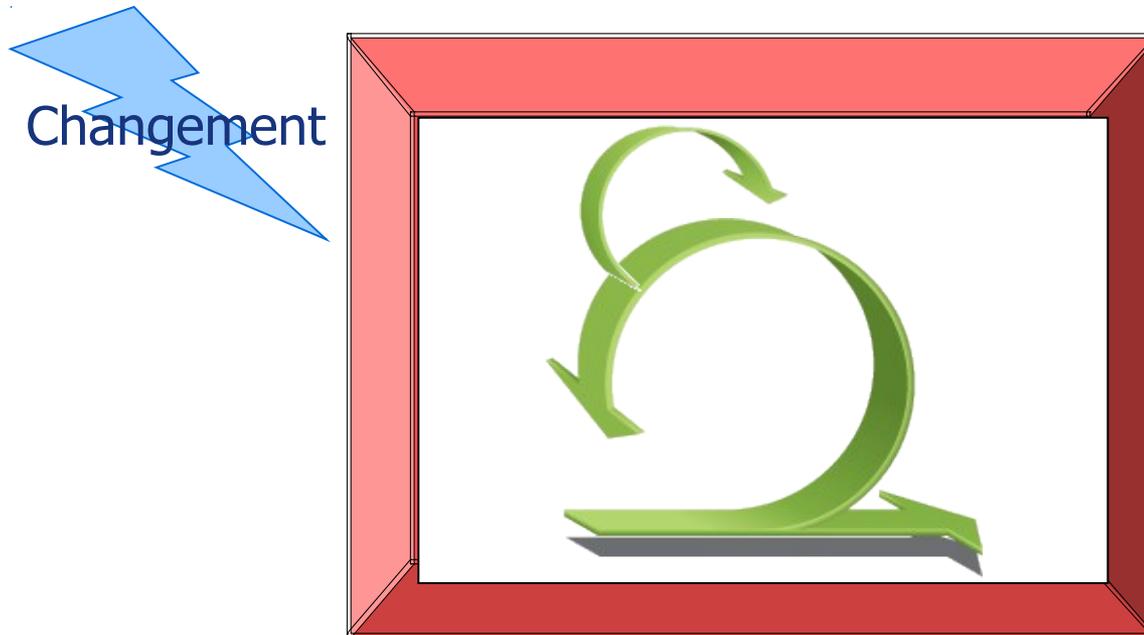
Code

Test

Plutôt que de faire toute une discipline d'un coup...

...Les équipes Scrum font un peu de tout tout le temps

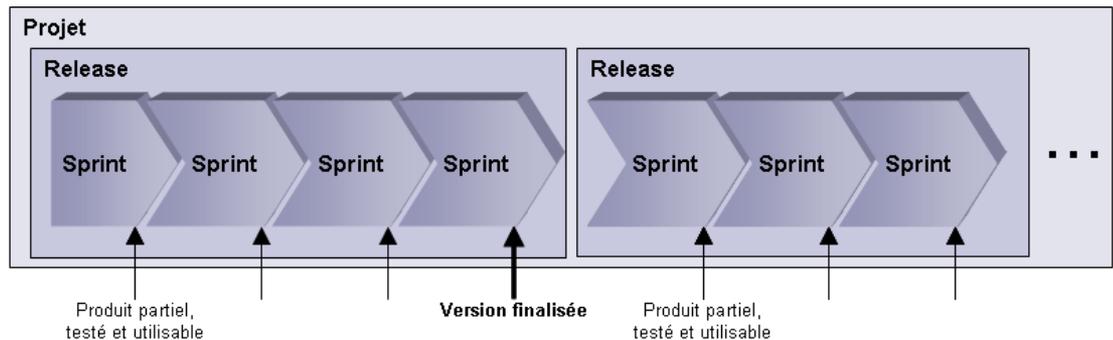
Stabilité pendant le sprint



- Planifier la durée pour permettre de différer la prise en compte d'un changement jusqu'au prochain sprint

Scrum – Planifier un projet

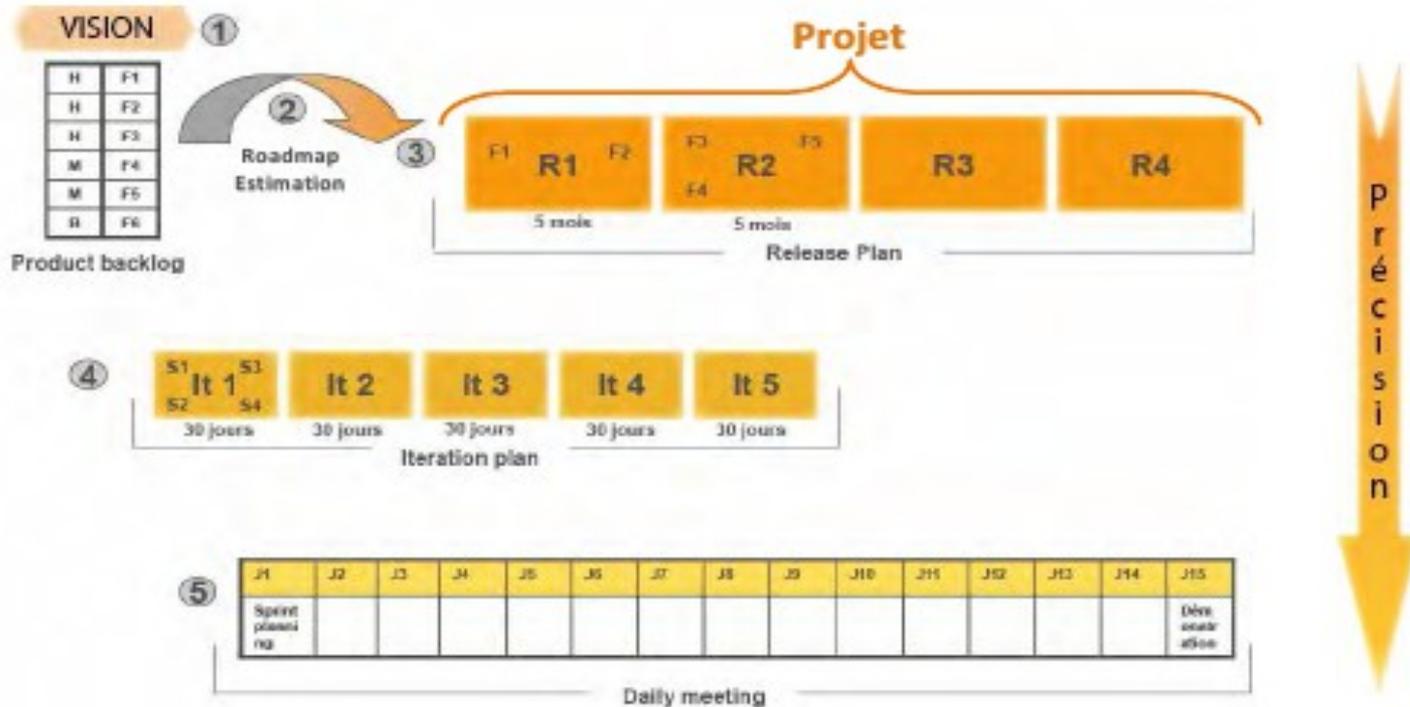
Backlog produit - Site marchand XY					
ID_Item	Titre	Importance	Estimation	Démonstration de la fonctionnalité	Commentaires
SPRINT 1					
1	Besoin 1	130	12	xxxxxxxx	xxxxxxxx
2	Besoin 2	120	9	xxxxxxxx	xxxxxxxx
3	Besoin 3	115	20	xxxxxxxx	xxxxxxxx
SPRINT 2					
4	Besoin 4	110	8	xxxxxxxx	xxxxxxxx
5	Besoin 5	100	20	xxxxxxxx	xxxxxxxx
6	Besoin 6	95	12	xxxxxxxx	xxxxxxxx
SPRINT 3					
7	Besoin 7	80	10	xxxxxxxx	xxxxxxxx
8	Besoin 8	70	8	xxxxxxxx	xxxxxxxx
9	Besoin 9	60	10	xxxxxxxx	xxxxxxxx
10	Besoin 10	40	14	xxxxxxxx	xxxxxxxx
SPRINT 4					
11	Besoin 11	35	4	xxxxxxxx	xxxxxxxx
12	Besoin 12	25	6	xxxxxxxx	xxxxxxxx
13	Besoin 13	10	7	xxxxxxxx	xxxxxxxx
14	Besoin 14	10	11	xxxxxxxx	xxxxxxxx
15	Besoin 15	10	3	xxxxxxxx	xxxxxxxx



Source : <http://fr.wikipedia.org>

- Constitution du backlog produit par le product owner.
- Répartition en sprints et en releases.

Planification Agile



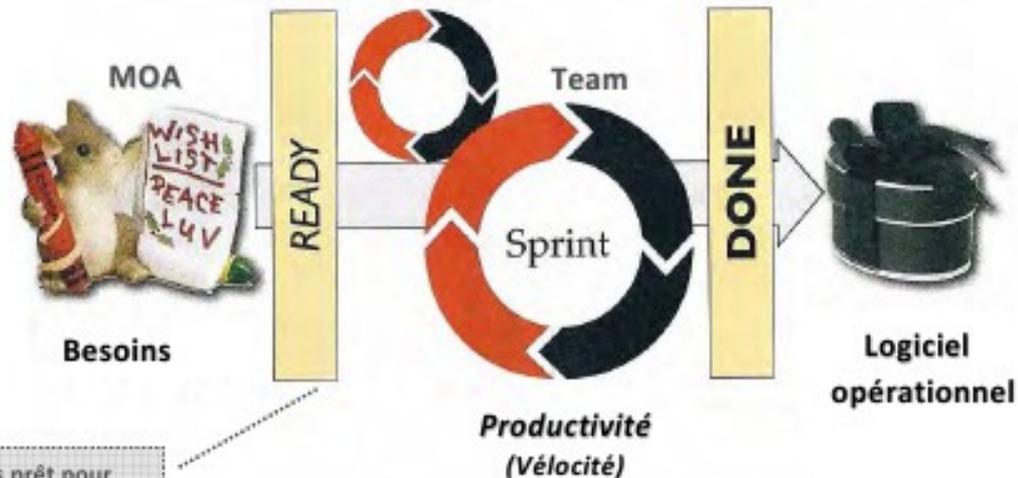
Une planification initiale (Iteration Plan) affinée à chaque itération

- En fonction de l'avancement et des impondérables
- Des changements de priorités et des nouveautés
- De l'expérience

→ Planning réaliste et visible (indicateurs)

Création de valeur

Un flux de fonctionnalités et de briques techniques



Être toujours prêt pour alimenter l'équipe avec la plus forte valeur et avec un stock réaliste

Le projet n'est plus une boîte noire pour la MOA.

La structure étendue cherche en permanence à optimiser le flux de valeur en fonction du planning, des livraisons prévues, de la vélocité de l'équipe.

→ Arriver le plus vite possible à avoir un logiciel opérationnel et de qualité qui répond aux attentes des utilisateurs.

Le cadre Scrum

Rôles

- Product owner
- ScrumMaster
- Equipe

Réunions

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

Artefacts

- Backlog de produit
- Backlog de sprint
- Beurdones

Plan

Rôles et responsabilités

Les responsabilités

Rôles

- Product owner
- ScrumMaster
- Equipe

Exercice

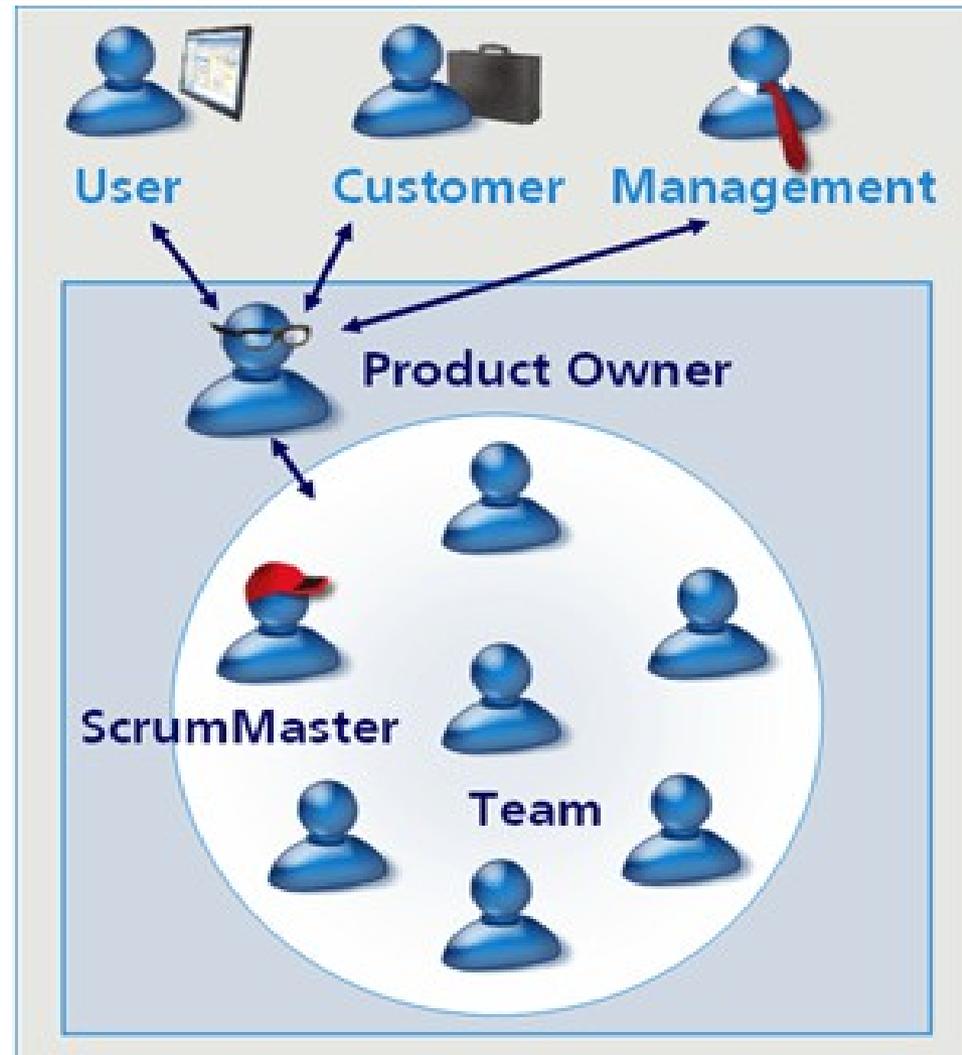
- Quelle est la différence entre implication et engagement ?

Scrum – Les rôles

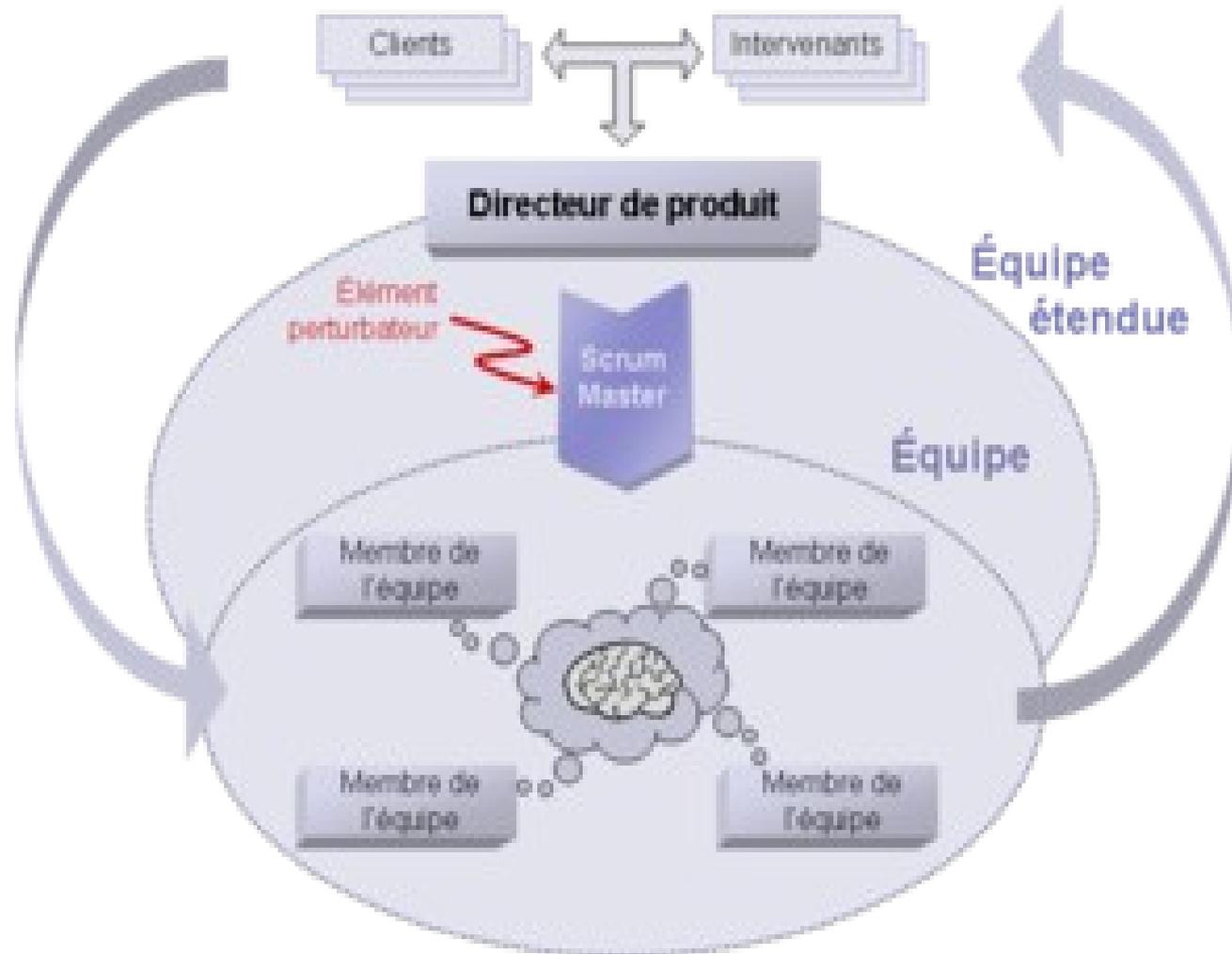
- **Les poules et les cochons**
 - **Implication et engagement**
- **Les cochons :**
 - **Le product owner**
 - **Le scrum master**
 - **L'équipe**
- **Les poules :**
 - **Tous ceux qui ont un intérêt dans le projet**

Acteurs et rôles

- Product owner
- Scrum master
- Equipe de développement



Rôles



Product owner / Directeur produit



- 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 / Directeur produit

Le Propriétaire de produit est responsable de gérer le carnet du produit et de s'assurer de la valeur du travail de l'Équipe. Cette personne maintient le carnet du produit et s'assure qu'il soit bien visible à tous. Comme tout le monde sait quels sont les éléments ayant la plus haute priorité, tous savent ce sur quoi l'Équipe devra travailler.

Le Propriétaire du produit est une seule et unique personne, et non pas un comité. Les comités peuvent exister pour conseiller et influencer la personne exerçant ce rôle, mais quiconque veut changer la priorité des éléments du Carnet du produit devra en convaincre le Propriétaire du produit. Les entreprises qui adoptent Scrum réaliseront que cela influence fortement leur façon d'établir les besoins et les priorités dans le temps.

Pour qu'un Propriétaire du produit réussisse, tous les membres de l'organisation devront respecter ses décisions. Personne ne doit être en mesure de dire à l'Équipe de travailler sur des priorités différentes que celles établies par le Propriétaire du produit. Les équipes ne doivent pas écouter qui que ce soit qui dirait le contraire. Les décisions prises par le Propriétaire du produit sont représentées dans la sélection et la priorisation des éléments du Carnet du produit. C'est cette extrême visibilité qui rend la tâche du Propriétaire de produit à la fois exigeante, mais aussi très satisfaisante.

SUGGESTION

Dans le cadre d'un développement commercial, le Propriétaire du produit peut être le gestionnaire de produit. Dans un développement maison, le Propriétaire du produit pourrait être le gestionnaire de la fonction d'affaires qui doit être automatisée.

SUGGESTION

Le Propriétaire de produit peut aussi être un membre de l'Équipe, faisant lui aussi du développement logiciel. Cette responsabilité supplémentaire peut toutefois affecter ses capacités à bien communiquer avec les autres parties prenantes. En aucun cas le Propriétaire de produit ne devrait être aussi ScrumMaster.

ScrumMaster



- 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



Le ScrumMaster s'assure que l'Équipe Scrum adhère aux valeurs, pratiques et règles de Scrum. Il aide l'Équipe et l'organisation à adapter Scrum. Le ScrumMaster forme l'Équipe Scrum en l'encadrant et en la guidant pour la rendre plus productive et pour qu'elle produise des produits de meilleure qualité. Le ScrumMaster aide l'Équipe Scrum à comprendre et à utiliser l'autogestion ainsi que ses capacités transversales (cross-functionality). Le ScrumMaster aide également l'Équipe Scrum à faire du mieux qu'elle peut à l'intérieur d'un environnement organisationnel qui peut ne pas être encore adapté au développement de produits complexes. Dans ce rôle, le ScrumMaster élimine les obstacles (impediments) pour permettre à l'équipe de mieux performer.

Toutefois, **le ScrumMaster ne gère pas l'Équipe Scrum; l'Équipe Scrum se gère elle-même.**

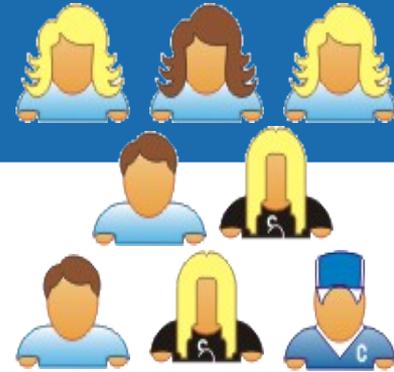
SUGGESTION

Le ScrumMaster travaille avec les clients et la direction pour identifier un Propriétaire de produit. Le ScrumMaster peut être amené à éduquer le Propriétaire et à lui apprendre comment faire le travail de façon convenable. Les Propriétaires de produit doivent être en mesure de maximiser la valeur de ce que l'Équipe produit en utilisant Scrum. S'ils n'y parviennent pas, on attribue la responsabilité au ScrumMaster.

SUGGESTION

Le ScrumMaster peut être un membre de l'Équipe ; par exemple, un développeur qui a des tâches à effectuer durant le Sprint. Toutefois, cela amène souvent des conflits lorsque le ScrumMaster doit choisir entre enlever des obstacles au profit de l'Équipe Scrum dans son entier et réaliser ses propres tâches. Le ScrumMaster ne devrait jamais être un Propriétaire de produit.

Equipe Scrum



- 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

Equipe Scrum

La responsabilité de l'Équipe de développement est de transformer le contenu du Carnet du produit en un sous-ensemble de fonctionnalités potentiellement livrables à la fin de chaque Sprint. Comme l'Équipe est multidisciplinaire, elle possède toutes les compétences nécessaires pour produire un sous-ensemble du produit livrable. Les membres de l'Équipe possèdent souvent des compétences spécialisées tel que la programmation, le contrôle de qualité, l'analyse d'affaire, l'architecture, la conception d'interface usager, ou bien la conception de base de données. Toutefois, la compétence qu'ils partagent, soit d'être en mesure de transformer un besoin en un produit utilisable, a plus d'importance que les autres compétences. Un individu qui refuse de programmer parce qu'il est un architecte ou un concepteur ne serait pas utile dans l'Équipe. Tous contribuent, même s'ils doivent pour se faire apprendre une nouvelle compétence ou se remémorer une lointaine compétence. Il n'y a pas de titres dans l'Équipe et aucune exception à cette règle n'est tolérée. L'Équipe n'est pas non plus divisée en sous-groupes dédiés à un domaine de compétence comme les tests ou bien l'analyse d'affaires.

Equipe Scrum

L'Équipe se gère elle-même. Personne, pas même le ScrumMaster, ne peut dire à l'Équipe comment transformer le contenu du Carnet du produit en sous-ensemble de fonctionnalités livrables. L'Équipe s'organise elle-même pour arriver à cette fin. L'Équipe utilise son expérience et son expertise pour résoudre les problèmes auxquels elle fait face. La synergie résultante améliore l'efficacité et le rendement de l'Équipe au complet.

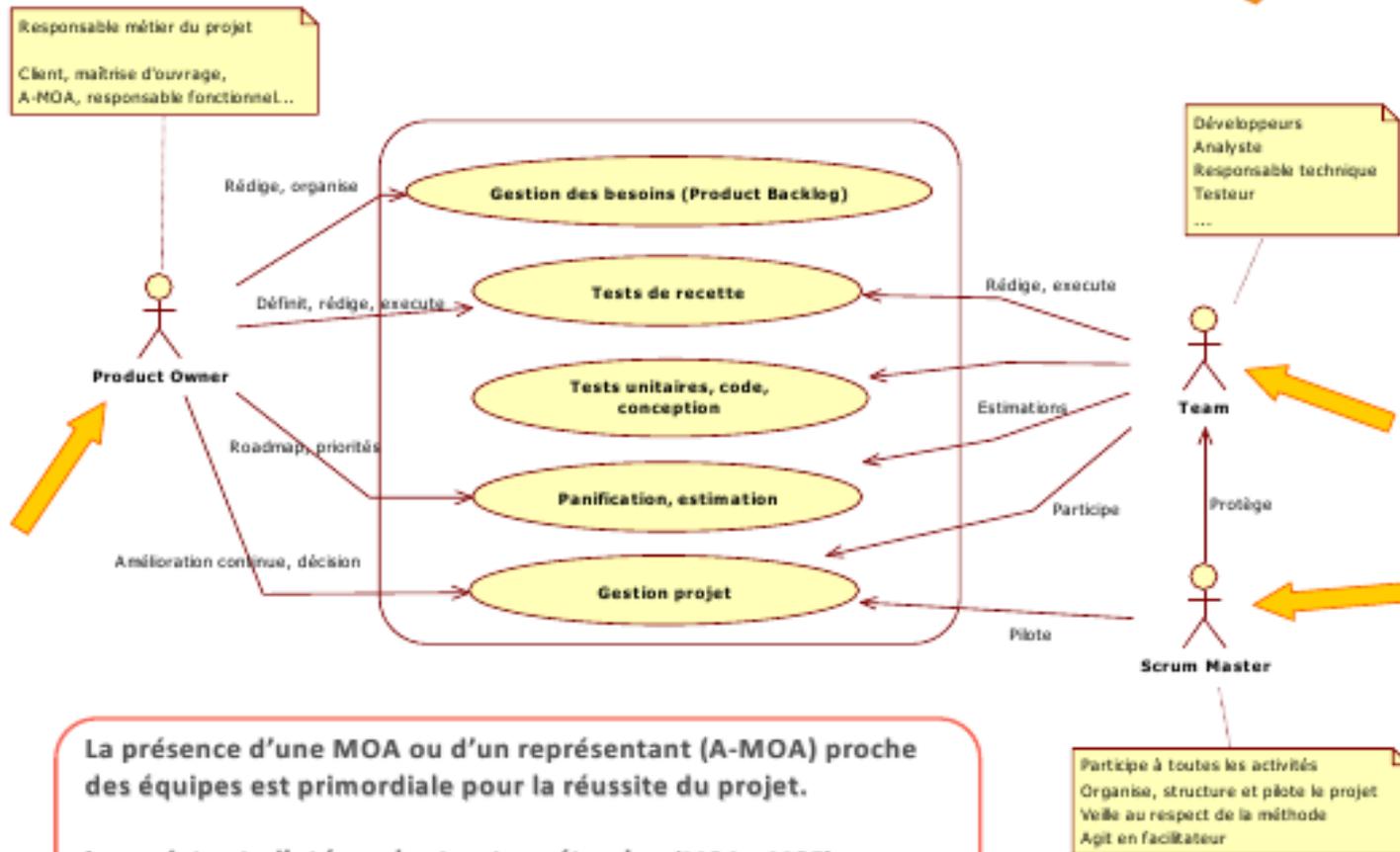
Equipe Scrum

La taille optimale d'une Équipe est de sept personnes, plus ou moins deux personnes. Quand l'Équipe est plus petite, il y a moins d'interactions entre les membres ce qui entraîne une réduction du gain de productivité. De plus, l'Équipe peut rencontrer des contraintes au niveau des compétences requises, résultant en une incapacité de livrer un sous-ensemble du produit à la fin du Sprint. D'un autre côté, s'il y a plus de neuf personnes, le besoin de coordination nécessaire devient trop important. Une grande Équipe génère trop de complexité pour être gérée de façon empirique. Toutefois, nous avons rencontré des Équipes qui ont réussi malgré qu'elles n'étaient pas de taille optimale et ce, qu'elles soient plus grandes ou plus petites. La taille de l'Équipe n'inclut pas les rôles du ScrumMaster et du Propriétaire du produit à moins que ces derniers ne soient également des « cochons ».

Equipe Scrum

La composition de l'Équipe peut changer à la fin d'un Sprint. Toutefois, chaque changement à la composition de l'Équipe engendre une réduction du gain de productivité provenant de l'autogestion. Il faut donc porter particulièrement attention à tout changement au niveau de la composition de l'Équipe.

Responsabilités



La présence d'une MOA ou d'un représentant (A-MOA) proche des équipes est primordiale pour la réussite du projet.

Le projet est piloté par la structure étendue (MOA + MOE)

- Casser l'effet tunnel
- Changement, prise de décision → Optimiser la valeur

Plan

Les besoins

Les besoins

- Avant la 1ère itération
- Le sprint 0
- La formation de l'équipe
- L'environnement de test
- Le product backlog
- La 1ère release planning

Le cadre Scrum

Réunions

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

Planification du sprint

Périmètre

- Analyser et évaluer le backlog de produit
- Définir le but du sprint

Plan

- Décider comment s'y prendre (conception)
- Créer la liste des tâches à partir des éléments du backlog de produit
- Estimer les tâches en heures

Capacité de l'équipe

Backlog de produit

Conditions métier

Produit actuel

Technos

But du sprint

Back-log du sprint

Plan

Les artefacts : backlogs et burndown charts

Les artéfacts

- Les différents types de backlog
- Les user stories
- Les estimations
- L'utilisation du backlog produit
- Les outils
- Représentation du backlog
- Les diagrammes

Le cadre Scrum

Artefacts

- Backlog de produit
- Backlog de sprint
- Beurdones

Backlog de produit



Ceci est le
backlog de produit

- Les exigences
- Une liste de tout ce qui va entraîner du travail au projet
- Chaque élément doit apporter de la valeur aux utilisateurs ou clients du produit
- Les priorités sont définies par le directeur produit
- Les priorités sont revues à chaque sprint

Backlog de produit

C'est la liste des exigences produit, exprimées si possible sous la forme de User Stories :

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

Backlog de produit

La complexité est estimée par l'équipe SCRUM, par rapport à un item de référence (complexité 1), et sur une échelle issue de la suite de Fibonacci.

La Business Value et la complexité sont utilisés pour calculer la priorité de chaque item du Product Backlog.

Un backlog de produit

Élément de backlog	Estimation (en points)
Un invité peut faire une réservation	3
En tant qu'invité, j'annule une réservation	5
En tant qu'invité, je change les dates d'une réservation.	3
En tant qu'employé de l'hôtel, je produis les rapports de revenu par chambre	8
Améliorer la gestion des exceptions	8
...	30
...	50

Un backlog de produit

N°	Date	Orig	Thème	Type	Intitulé	Description demande	Prio	Point	Heures	Statut	Sprint
1	29/09/2009	Interne	Client	E	En tant qu'utilisateur je veux pouvoir gérer une adresse sur mes clients afin de pouvoir les contacter	Une adresse est définie par 3 lignes alphanumériques de saisie, un code postal et une ville	1				1
2	01/10/2009	Interne	Commande	E	En tant qu'utilisateur je veux pouvoir saisir une commande sur un client afin d'enregistrer les produits vendus	Une commande est définie par un numéro, un montant total et plusieurs lignes de commande. Chaque ligne dispose d'un libellé, d'un montant total calculé comme le produit du montant unitaire par la quantité. Le montant total de la commande correspond à la somme des montants de chacune des lignes.	1				1
3	03/10/2009	Client	Commande	E	En tant qu'utilisateur je veux pouvoir facturer une commande pour transmettre la facture au client	Une facture est un document papier qui peut être envoyé par mail, elle reflète les éléments de la commande	2				1

User story

[titre] L'utilisateur peut rechercher un emploi
~~Le programme sera écrit en Java (techno !)~~

[description, 1 ou 2 phrases support à la conversation]

L'utilisateur peut voir les informations de chaque emploi qui correspond à sa recherche.

[détails, questions. Les décisions prises deviennent des tests !]

- Marco suggère d'afficher la description, le salaire et le lieu.
- Les cartes de paiement Discover sont-elles acceptées ?
- ~~On accepte les cartes Visa, Mastercard et American Express!~~ (c'est 1 test : voir au verso)

[risque technique]

[coût, en story points]

2

4

[expectations, acceptance tests. Si tous sont ok la story est achevée]

- Tester avec une description de job vide.
- Tester avec une description de job très longue.
- Tester sans indiquer de salaire.
- Tester avec un salaire à 6 chiffres.

- Tester avec Visa, MasterCard et American Express → succès !
- Tester avec Diner's Club → échec !

[une bonne story doit être INVEST (Bill Wake, 2003)]

- **Independent** (entre elles, revoir le découpage sinon)
- **Negotiable** (se n'est pas un contrat écrit !)
- **Valuable** (par l'utilisateur, l'acheteur, le client : pour le planning)
- **Estimatable** (par le développeur)
- **Small** (idéalement pas plus de qq jours, excepté les epics)
- **Testable** (si la story est non testable → comment savoir si finie ?)

Le but du sprint

- Un bref énoncé de sur quoi le travail va essentiellement porter pendant le sprint

Database Application

Faire tourner l'application sur une base MySQL en plus d'Oracle.

Sciences de la vie

Offrir les fonctions pour les études génétiques.

Services financiers

Offrir plus d'indicateurs que le produit ABC sur les données de streaming .

Vie du backlog de sprint

- Chacun s'engage sur du travail qu'il choisit
 - Le travail n'est jamais assigné par un autre
- L'estimation du reste à faire est ajustée tous les jours
- N'importe qui peut ajouter, supprimer ou changer le backlog de sprint
- Le travail du sprint émerge progressivement
- Si un travail n'est pas clair, définir une tâche avec plus de temps et la décomposer après
- MAJ du travail restant quand il devient connu

Backlog de sprint

Le Sprint Backlog est organisé en trois colonnes : « To do » (à faire), « In progress » (en cours) et « Done » (terminé).

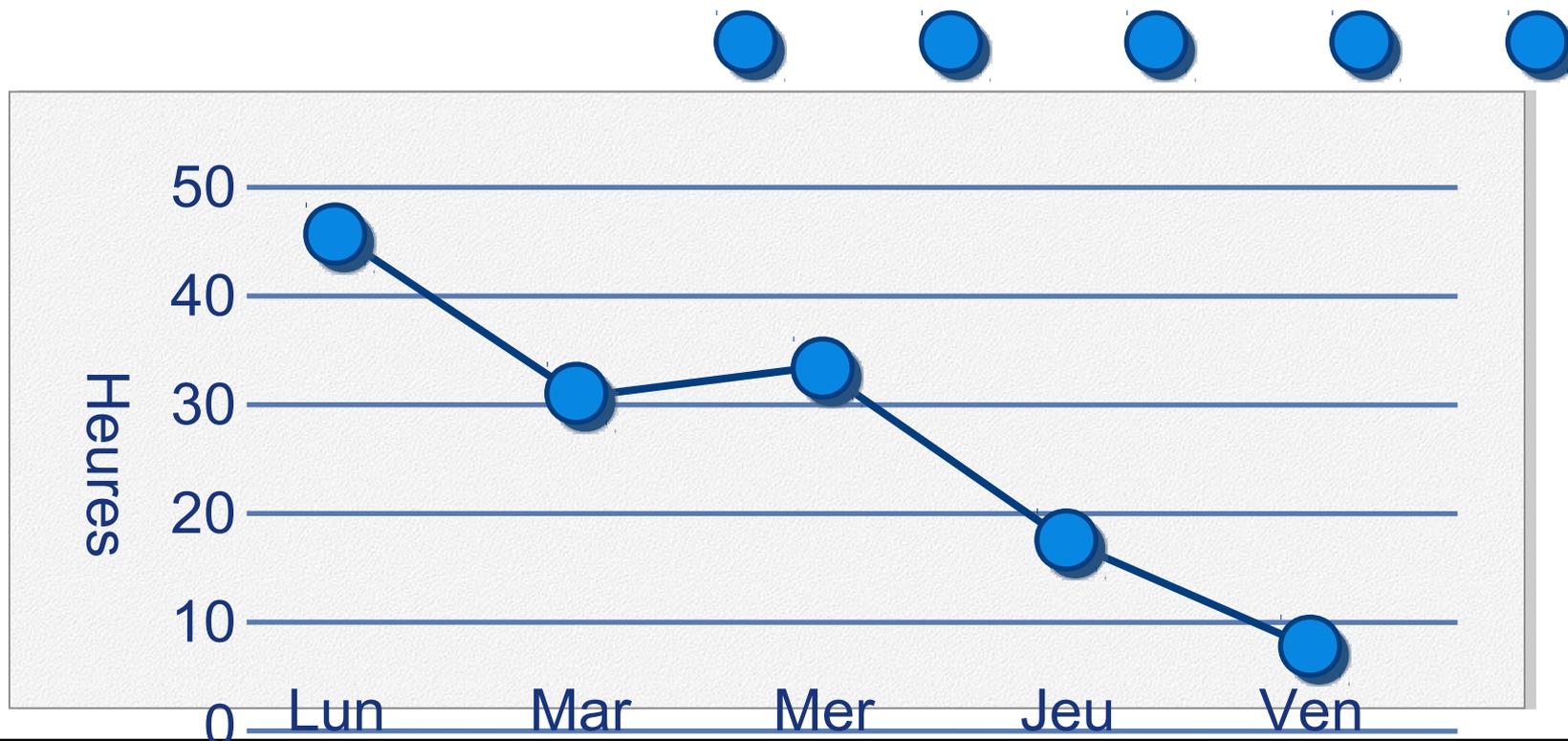
Chacune des colonnes reçoit au cours du sprint les tâches du Sprint Backlog qui sont en cours de traitement puis terminées.

A la fin du sprint, toutes les tâches doivent être dans la colonne « Done ».

Un backlog de sprint

Tâches	Lun	Mar	Mer	Jeu	Ven
Coder l'IHM	8	4	8		
Coder couche métier	16	12	10	4	
Tester l'intégration	8	16	16	11	8
Ecrire l'aide en ligne	12				
Ecrire la classe foo	8	8	8	8	8
Tracer les erreurs			8	4	

Tâches	Lun	Mar	Mer	Jeu	Ven
Coder l'IHM	8	4	8		
Coder la persistance	16	12	10	7	
Tester l'intégration BD	8	16	16	11	8
Ecrire l'aide en ligne	12				



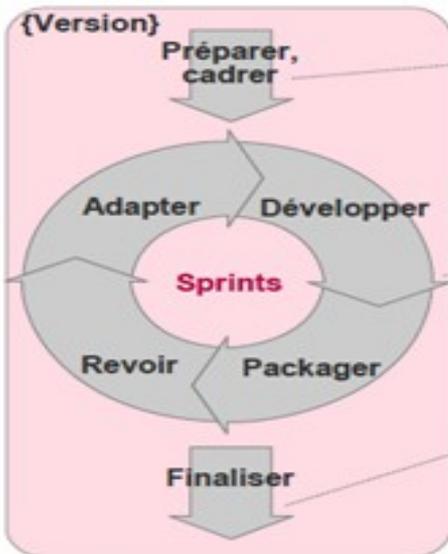
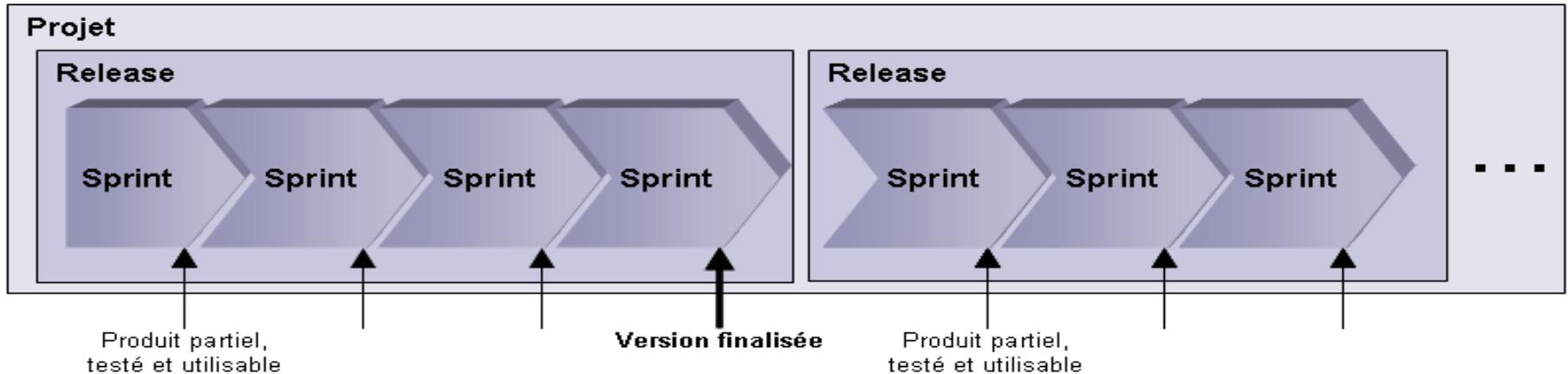
Plan

La release

La release

- **Définition**
- **La release planning**
- **Définition des users stories**
- **L'engagement**
- **Les estimations**
- **Le Release Burndown Chart**

Planification



Pré-jeu

- Planning

- Définition du backlog, de l'équipe
- Identification des facteurs de risques
- Choix des outils, infrastructure+ logistique
- Budget, chiffrage et estimation des coûts

- Architecture générale et conception générale

Jeu

- Sprints
- Développement (analyse, conception, codage -> Livraison -> révision -> adaptation)

Post-jeu

- Packaging global
- Intégration et jeu de tests

Estimation, planification, avancement

■ Estimations :

- par analogie de préférence (utilisation du réalisé comme référence), ou par point de fonction
- Utilisation de méthodes agiles ("planning poker")

■ Planification

- Le planning doit permettre à l'équipe de "souffler" entre 2 sprints
 - Exemple : démos & rétrospective le vendredi matin pour le sprint n, planification du sprint n+1 le lundi matin ► vendredi AM libre pour de la veille, ...

Estimation, planification, avancement

- Les estimations et la planification se basent sur le calcul de la vélocité

Vélocité estimée = (nb de jour/homme disponibles) x (coefficient d'effort)

réel, après application
congés, temps partiel, ...

- Le coefficient d'effort est une estimation sur la concentration de l'équipe. Il peut être évalué à partir des sprint précédents. Par exemple :

LAST SPRINT'S FOCUS FACTOR:

$$40\% = \frac{18 \text{ STORY POINTS}}{45 \text{ MAN-DAYS}}$$

THIS SPRINT'S ESTIMATED VELOCITY:

$$50 \text{ MAN-DAYS} \times 40\% = 20 \text{ STORY POINTS}$$

La correction d'anomalie peuvent être intégrées dans ce coefficient, les difficultés techniques peuvent être vues comme des "tech stories"

© SCRUM and XP from the Trenches, Henrik Kniberg

Aspects contractuels

- Problématique : comment gérer un projet Scrum sur le plan contractuel, sachant que les concepts Agiles préfèrent la collaboration à la négociation de contrat?
 - Contrats pour les forfaits "classiques"
 - prix fixe, périmètre fixe
 - 4 axes caractérisés et figés : Périmètre, budget, délai, qualité

Aspects contractuels

- Dans un contrat plus agile, on peut imaginer
 - De la part du soumissionnaire, un engagement sur les moyens (compétences, méthodes, pilotage, ...) pour assurer la qualité
 - Estimation du rapport qualité/prix plutôt que prix le plus bas
 - Exemples de critères : niveau de qualité (méthodes, ...), fréquence des feedbacks, expériences des ressources, capacité à s'adapter, communication, maturité de la relation client
 - Un budget et des dates de livraisons (durée du sprint) fixés par le client
 - Un périmètre initial tracé de façon collaborative (backlog produit initialisé conjointement)

Aspects contractuels

▪ Différentes approches possibles pour l'agilité

- Contrat à engagement de moyens (~régie) : périmètre variable et coût fixe
 - Risques principaux : retard de livraison, dé-responsabilisation du soumissionnaire

• Co-sourcing

Préconisé
par Valtech

- Equipe mixte, sur le site du client
- Visibilité, transfert de compétences intrinsèque, risques partagés (des ressources des deux parties sont impliquées)

• Forfait collaboratif

Préconisé
par Aubry

- Le coût et le délai sont fixés par le client ► le périmètre est la variable d'ajustement
- Le soumissionnaire se positionne sur sa capacité et son savoir-faire (engagement sur des critères de qualité)
- Le contrat a donc bien un prix et une date fixés. La différence est dans la fréquence de livraison et dans l'ajustement du périmètre.
- Forte implication des parties

• Forfait avec engagement mesurable

Préconisé par
timwi

- Définition de points de mesures concrets et représentatifs (par exemple : Définition d'unités d'œuvre, engagement sur un niveau de qualité, ...)

Aspects contractuels

Le client a besoin de connaître à l'avance le coût du projet

C'est lui qui donne le coût et le délai

Le client a besoin de choisir entre plusieurs fournisseurs

Le fournisseur se positionne sur sa capacité et son savoir-faire

Aspects contractuels

Bénéfices du forfait Agile

▶ Client

- ▶ Le coût est borné
- ▶ Il peut changer d'avis pendant le projet
- ▶ Il définit les priorités
- ▶ Il a un produit partiel à la fin de chaque itération

▶ Fournisseur

- ▶ Il est choisi sur sa compétence

Aspects contractuels

Bénéfices mutuels

Une plus grande
confiance dans les
relations

Aspects contractuels

Appel d'offres Agile

- ▶ Le client donne en plus de d'habitude :
 - ▶ sa Vision (plus ou moins fait)
 - ▶ le coût et le délai (d'habitude il ne les dit pas bien qu'il en ait une idée voire plus)
- ▶ Les fournisseurs répondent
- ▶ Le client sélectionne les réponses en fonction de la qualité de la réponse :
 - ▶ Le processus proposé (Agile !)
 - ▶ Les compétences de l'équipe
 - ▶ Pas le prix !

Aspects contractuels

Principes du contrat Agile

- ▶ Le fournisseur s'engage sur des critères de qualité
- ▶ Les développements peuvent s'arrêter à tout moment à la fin d'une itération

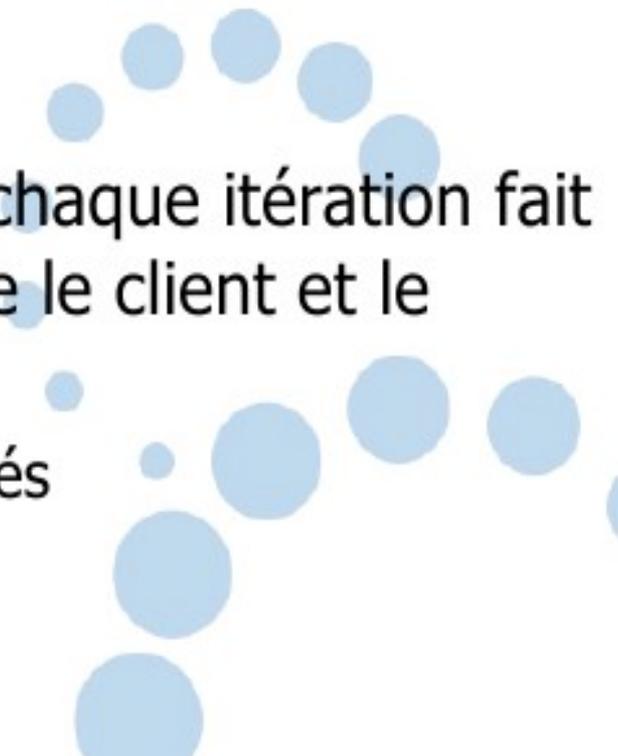
Aspects contractuels

Mise en oeuvre

- ▶ Le client donne le prix P et le délai D
- ▶ Le fournisseur et le client définissent ensemble la durée d'une itération I
 - ▶ Le tarif par itération est PxI/D
- ▶ A la fin de chaque itération le client voit ce qui fonctionne et peut décider de :
 - ▶ Changer de fournisseur, le client est payé : $nxPxI/D$ (plus un pourcentage lié à l'investissement)
 - ▶ Changer de priorité dans les exigences

Aspects contractuels

Visibilité

- ▶ Le backlog actualisé à chaque itération fait l'objet d'un accord entre le client et le fournisseur
 - ▶ Les éléments sont estimés
 - ▶ Vélocité de l'équipe
 - ▶ Burndown charts
- 

Aspects contractuels

- ▶ Après une première phase qui sert à obtenir la vélocité et à constituer un backlog avec chaque élément estimé
- ▶ L'accord (engagement) du fournisseur peut porter sur un engagement à réaliser un certain nombre de points
 - ▶ Pas un contenu précis
 - ▶ Mais une taille précise

Forfait Agile en points

Avantages (1/3)

Cadre de production, méthodologies

- Augmente productivité et qualité
- Plus de souplesse et de créativité
- Résultat conforme aux attentes
- Pilotage au quotidien
- Règles définies clairement



Avantages(2/3)

Copilotage avec le client

- Propice à une confiance réciproque
- Transparence sur l'avancement
- Adaptabilité



Avantages(3/3)

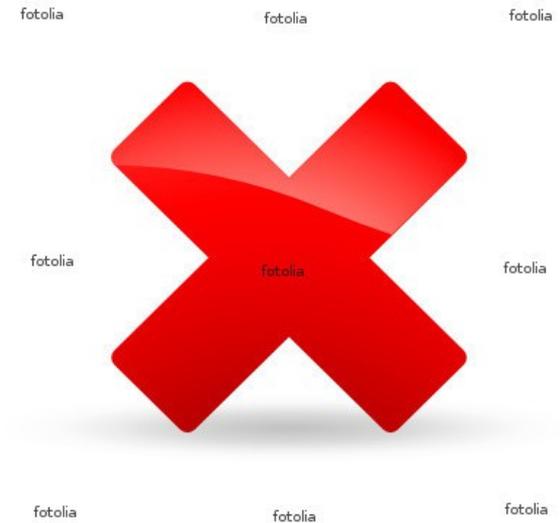
Equipe & responsabilité collective

- Pas d'individualisme ->Améliorer la communication
- Amélioration permanente
- Chaque équipe a son lot de responsabilité



Inconvénients

- Peu, voire pas, de documentation écrite
- Grande disponibilité de tous les acteurs
- Coûts (susceptibles de dériver)



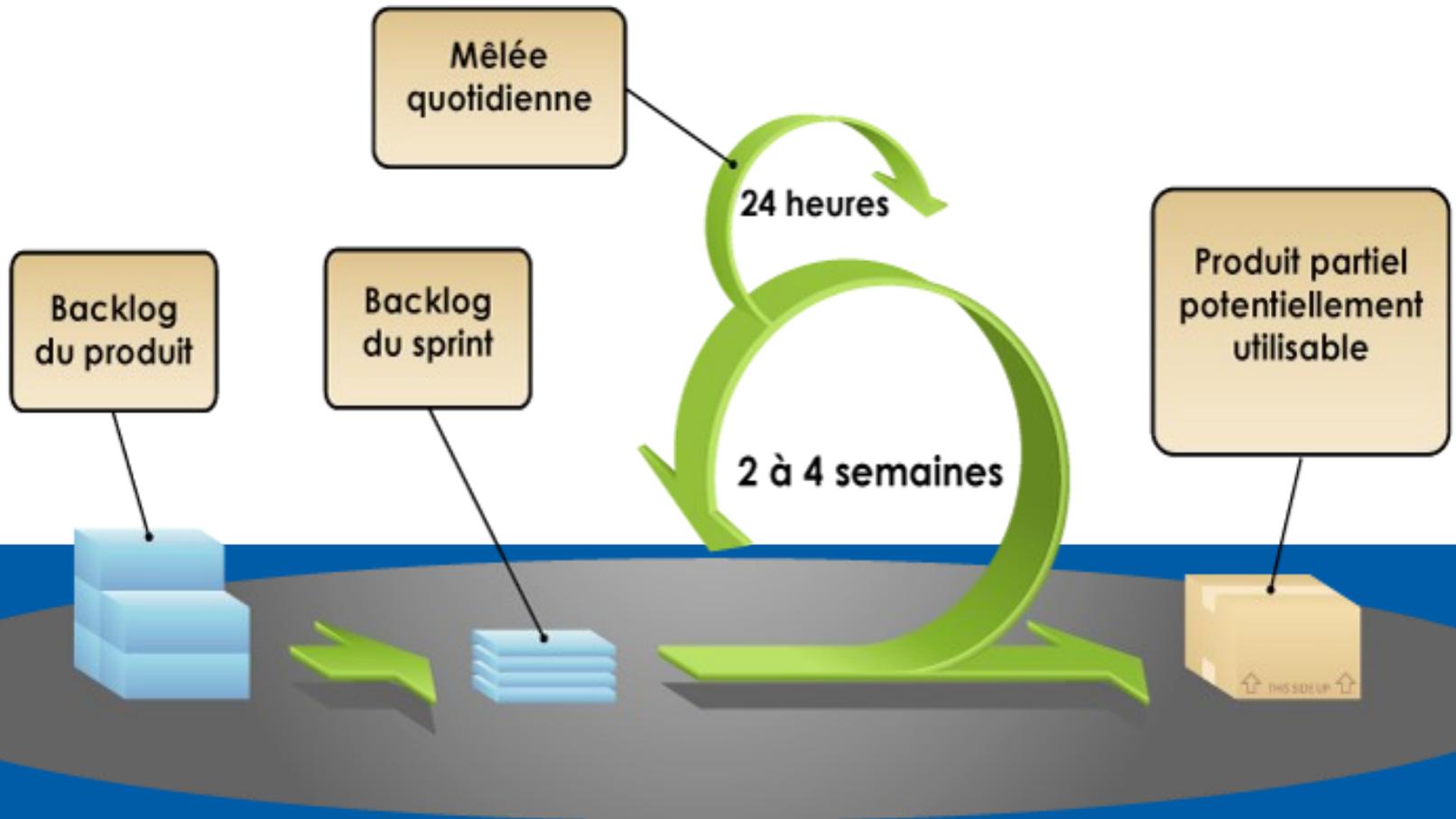
Plan

Le sprint

Le sprint

- **Le déroulement d'un sprint**
- **Le sprint planning**
- **Les outils**
- **Les différents sprints**

Déroulement d'un sprint



Sprint planning

1ère partie :

- Engagement de l'équipe
- Définition des enjeux du sprint
- Participants : Product Owner, SCRUM Master, l'équipe

2ème partie :

- Item => Tâche
- Tâche => Temps
- Participants: Scrum Master, l'équipe



Planification du 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)

Scrum quotidien

- **Paramètres**
 - Tous les jours
 - 15 minutes
 - 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**



Chacun répond à 3 questions

1

Qu'as tu fait hier ?

2

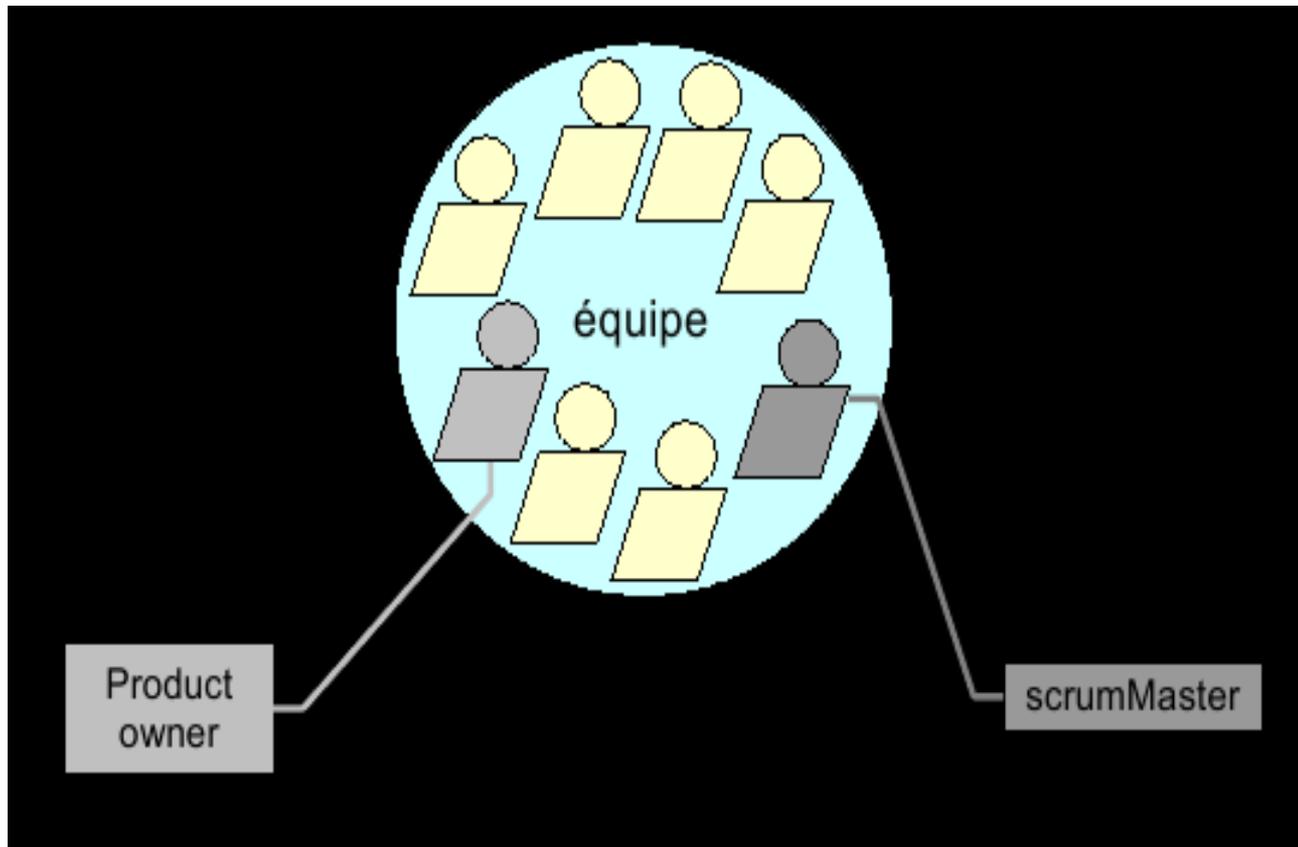
Que vas-tu faire aujourd'hui ?

3

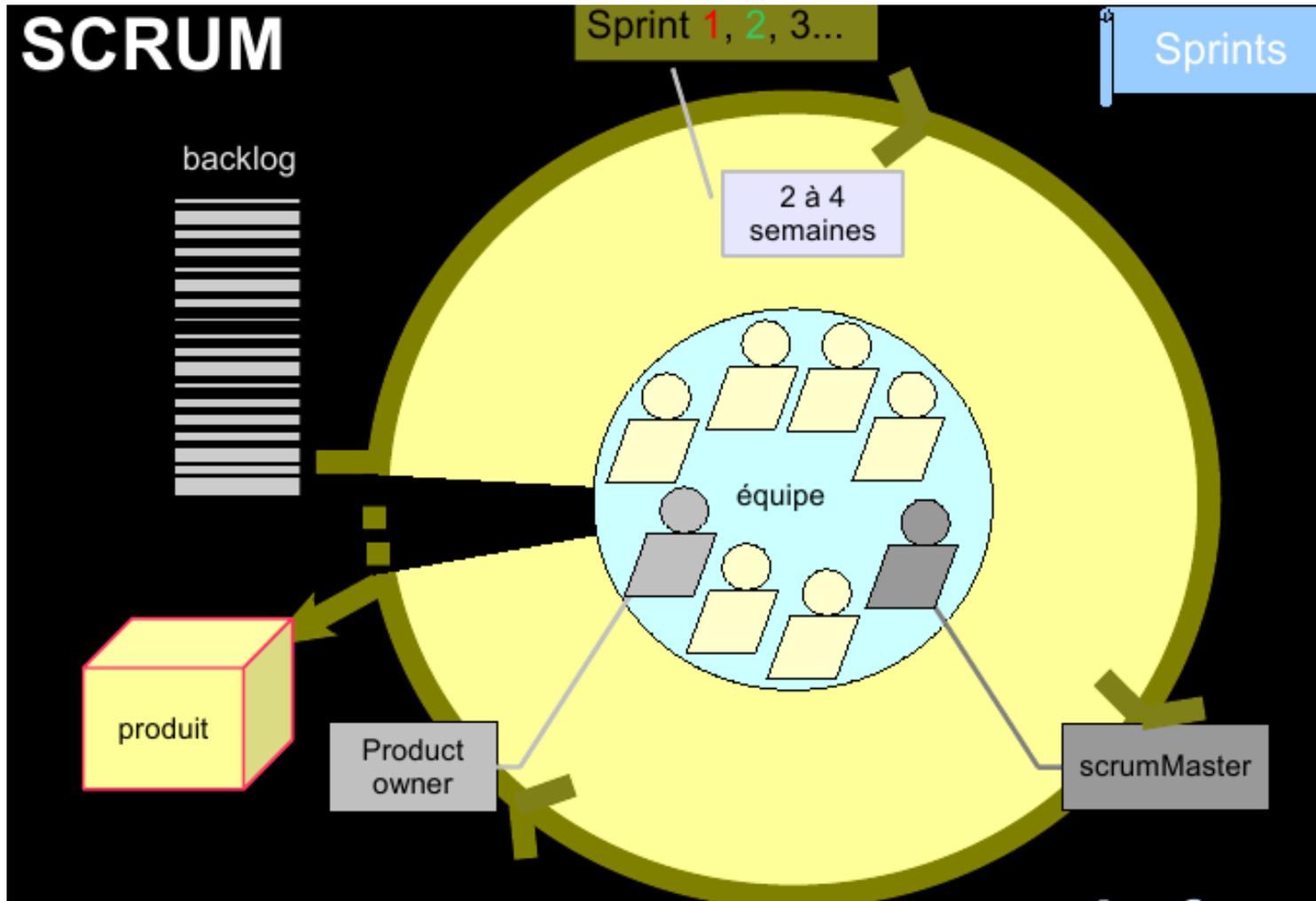
Y a t-il un truc qui cloche ?

- Il ne s'agit **pas** de compte-rendus au ScrumMaster
- Ce sont des engagements devant des pairs

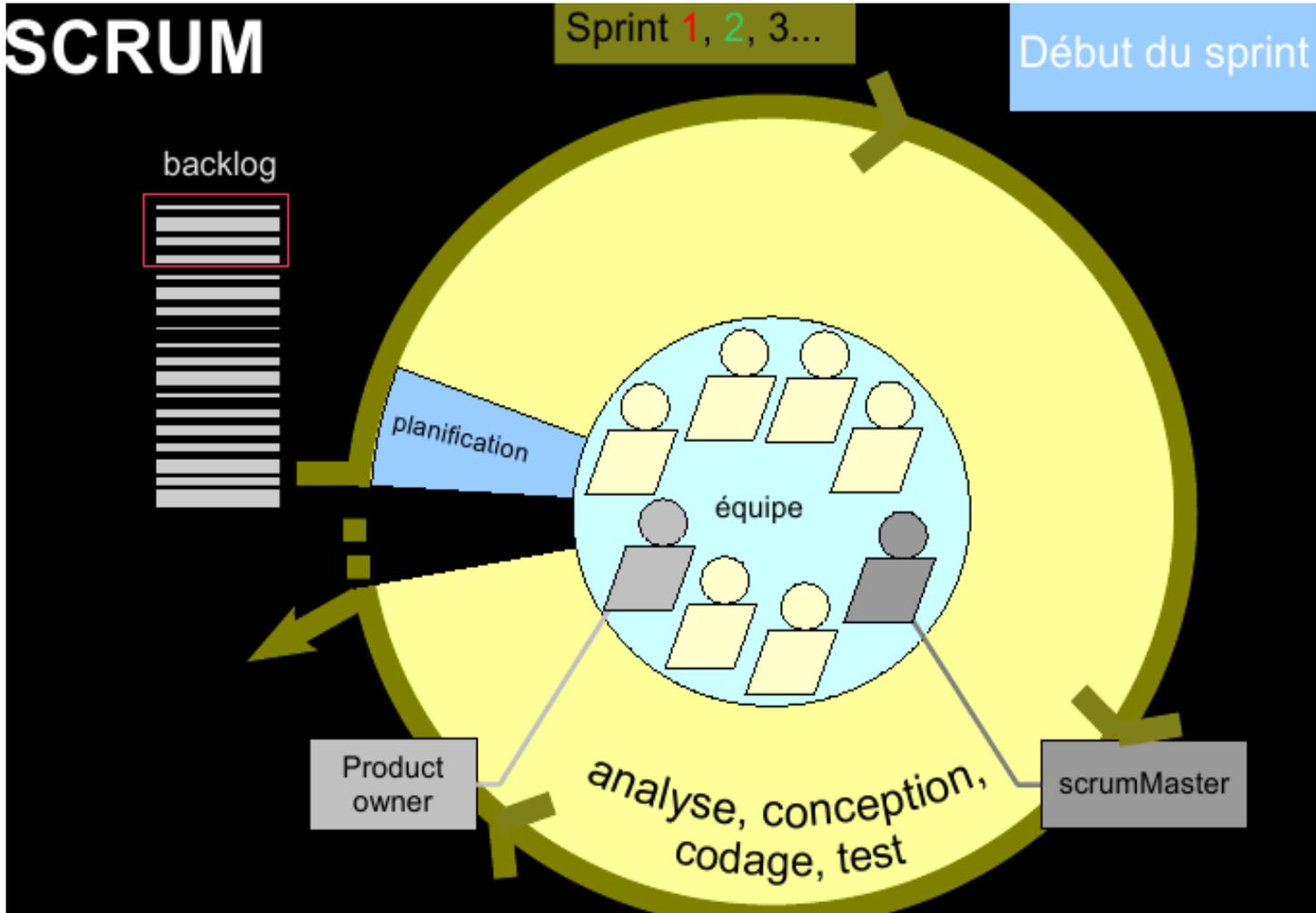
Scrum



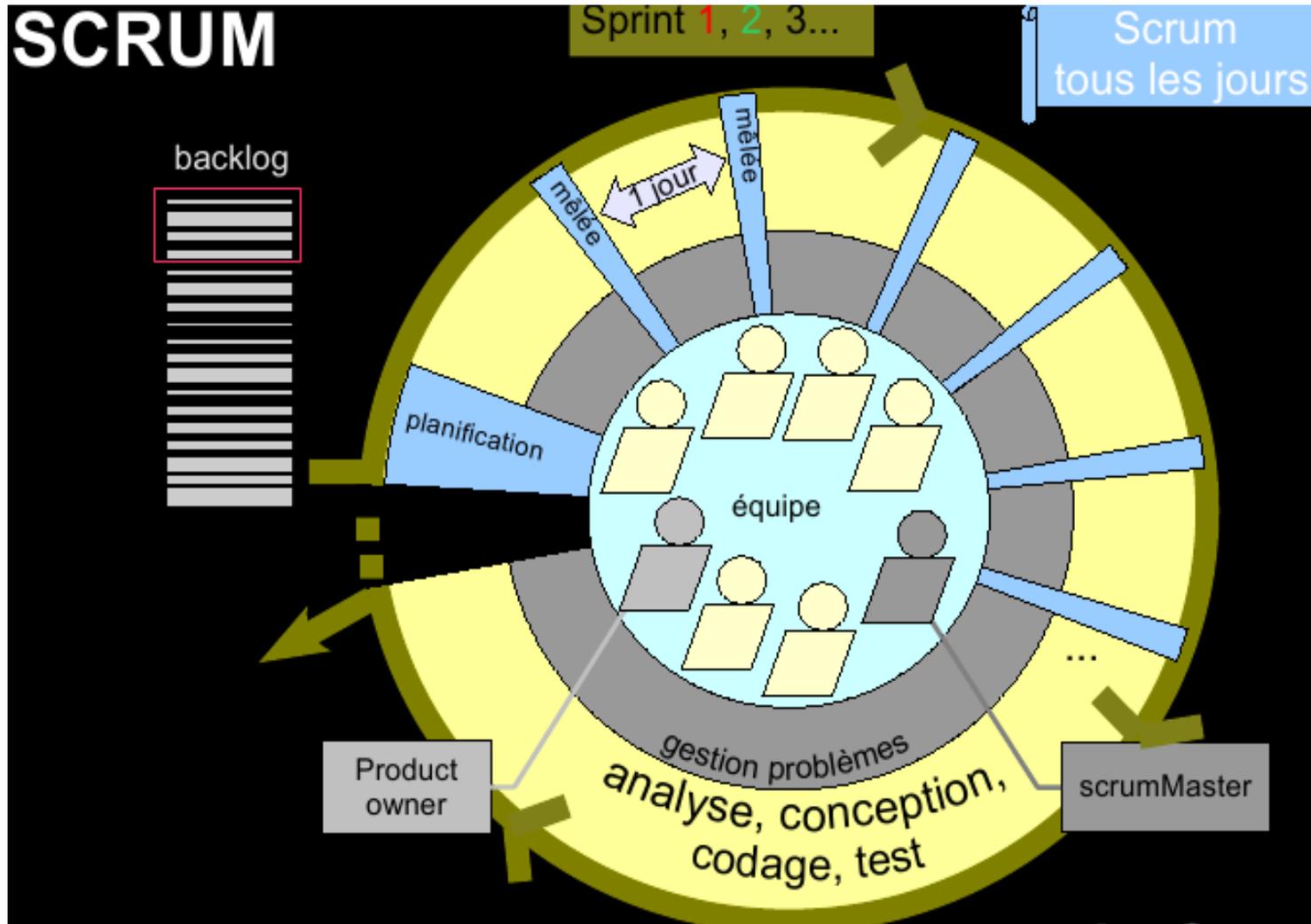
Scrum



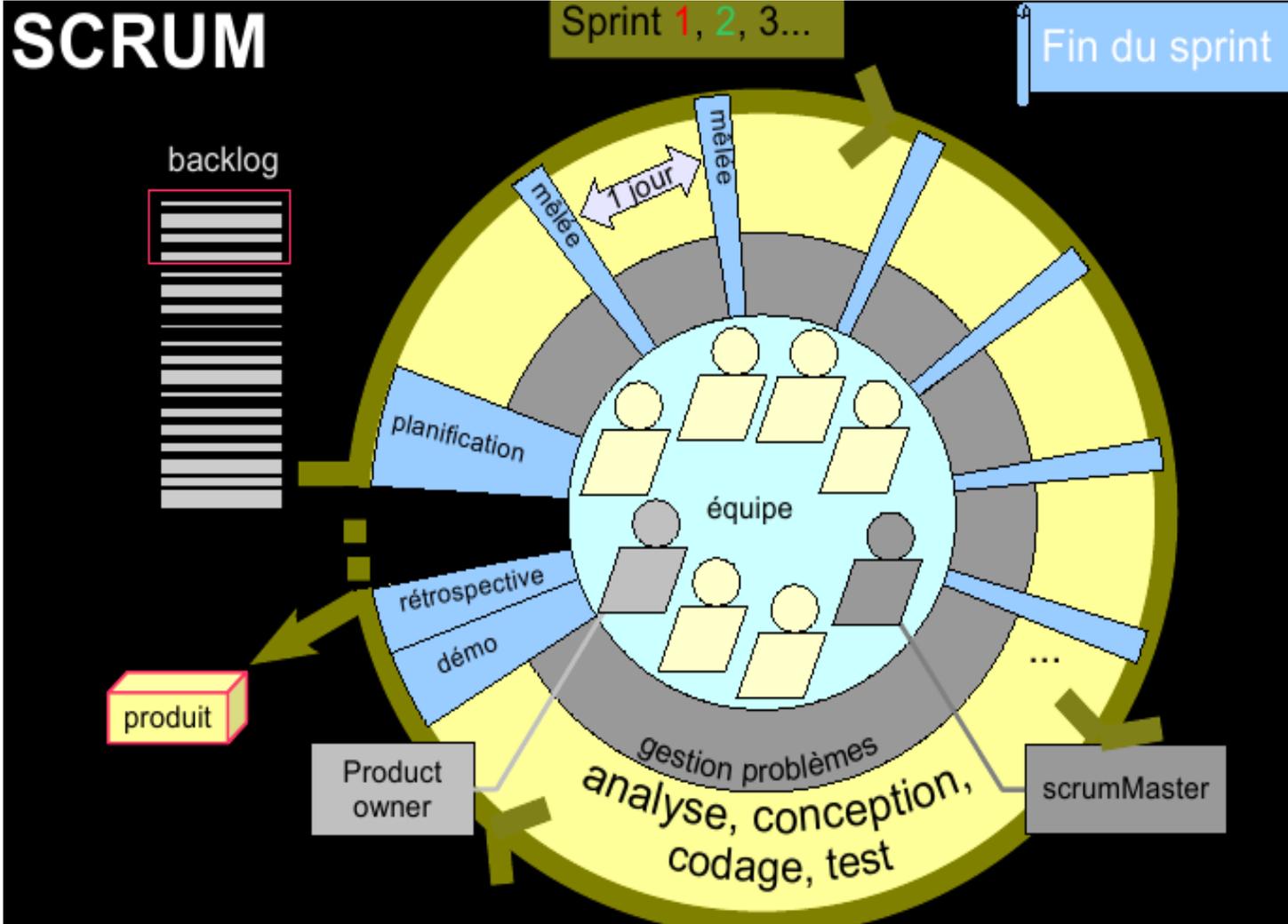
Scrum



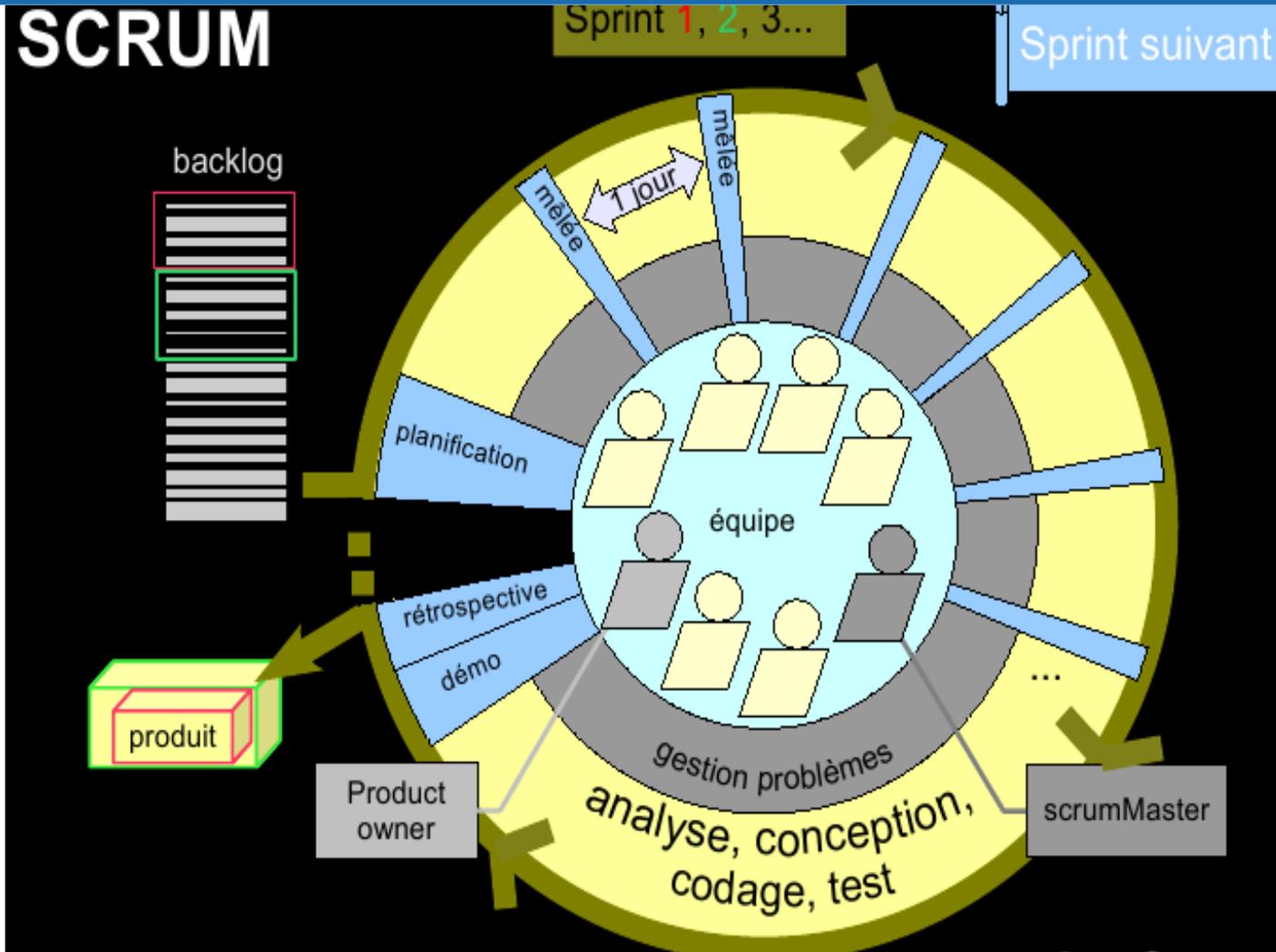
Scrum



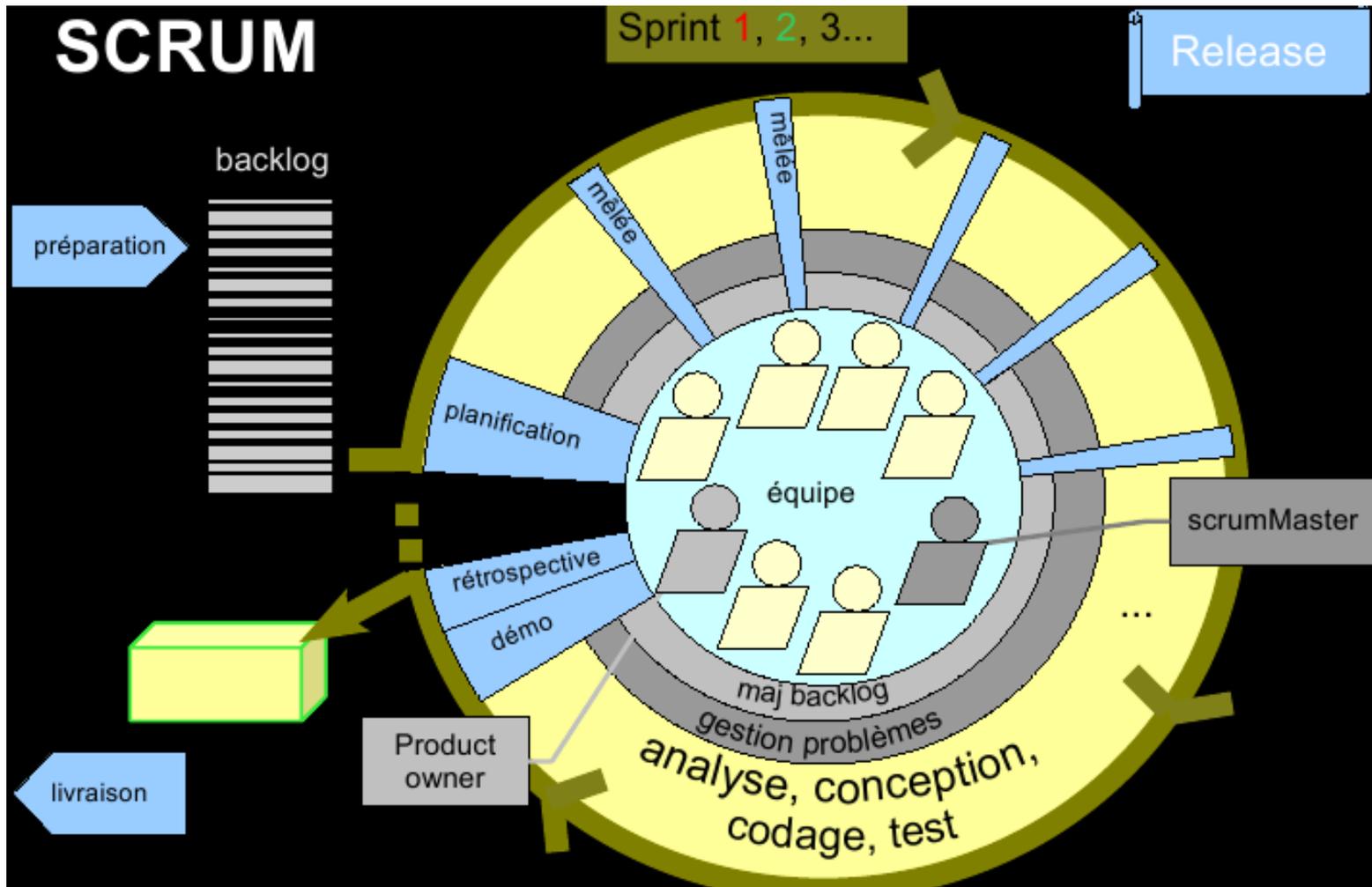
Scrum



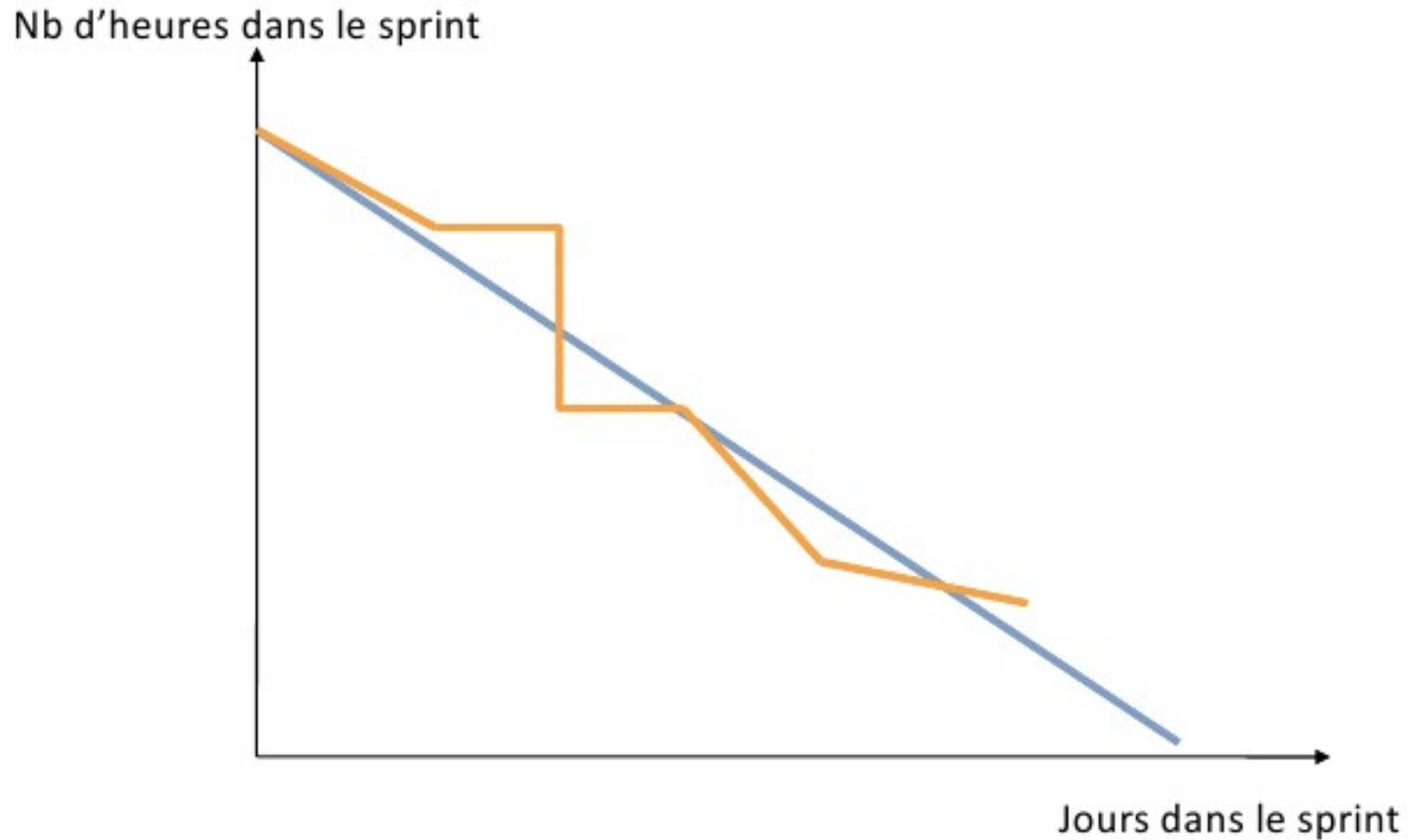
Scrum



Scrum



Burndown de sprint



Burndown de sprint

C'est le graphique qui montre l'évolution au cours du temps du nombre de points de complexité restant à traiter.

La durée du sprint est portée en abscisse, la complexité en ordonnée; la courbe de référence va linéairement de la complexité totale prévue pour le sprint (la somme des complexités de tous les items sélectionnés dans le Product Backlog) en début de sprint, jusqu'à zéro en fin de sprint.

Burndown de sprint

A la fin de chaque Daily Scrum, le Scrum Master reporte sur le Burndown Chart le niveau de complexité restant à traiter. C'est la somme des complexités de toutes les tâches qui ne sont pas encore dans la colonne « Done ».

Si la courbe suit la courbe de référence ou est en dessous, c'est bon, le planning du sprint est tenu. Dans le cas contraire, le rythme doit être renforcé. Dans le pire des cas, le planning du sprint ne pourra pas être tenu, le périmètre peut être renégocié ; le Product Owner doit dans ce cas jouer pleinement son rôle de décideur.

Burnup de sprint

C'est la courbe de progression de la business value réalisée.

Le Burnup Chart porte sur tout le projet. La durée du projet est portée en abscisse, la business value totale (somme des business values de tous les items du Product Backlog) en ordonnée ; la courbe de référence va linéairement de l'origine (0 business value réalisée en début de projet) jusqu'à la business value totale et maximum en fin de projet.

Burnup de sprint

A la fin de chaque sprint, le Scrum Master reporte sur le Burnup Chart la business value réalisée pendant le sprint.

Comme les items du Product Backlog sont priorisés en termes de business value, le Burnup Chart croit très rapidement au-dessus de la courbe de référence, pour s'infléchir à partir de la moitié du projet.

Burnup de sprint

Il n'est pas rare, sur un projet SCRUM, que 80% de la business value du projet soit atteinte à la moitié du temps. La question peut alors se poser légitimement : doit-on continuer le projet, et dépenser encore 50% du budget pour seulement 20% de business value restante ?

Burnup de sprint

C'est au Product Owner qu'il revient de prendre cette décision. Mais ce point peut être envisagé dès le début du projet et être contractualisé : on fixe un niveau de business value (75%, 80%) à atteindre à la moitié du projet. S'il est atteint, le projet est terminé et une partie des 25% ou 30% du différentiel business value – coût est versé à l'équipe sous forme de prime. Motivant, non ?

Courbe de température

Elle donne la possibilité, pour chaque équipier, d'indiquer régulièrement le niveau de son moral, en toute franchise, en toute indépendance et, encore une fois, en toute transparence.

Courbe de température

Les mesures sont portées sur le même support, en général une feuille de papier ou un tableau blanc – il faut beaucoup de tableaux blancs pour un projet SCRUM ! – à une fréquence convenue par tous.

La courbe de température permet de détecter les baisses de moral, les surchauffes des équipiers. Au SCRUM Master d'en identifier les causes et de prendre ensuite les mesures correctives adéquates.

Vélocité

Dans Scrum, la vélocité définit la part du product backlog dont l'équipe peut s'occuper pendant un sprint.

La vélocité peut s'estimer en regardant les sprints précédents, à supposer que la composition de l'équipe et la durée des sprints soient stable. Elle peut aussi être définie à chaque sprint avec un planning basé sur l'engagement de l'équipe.

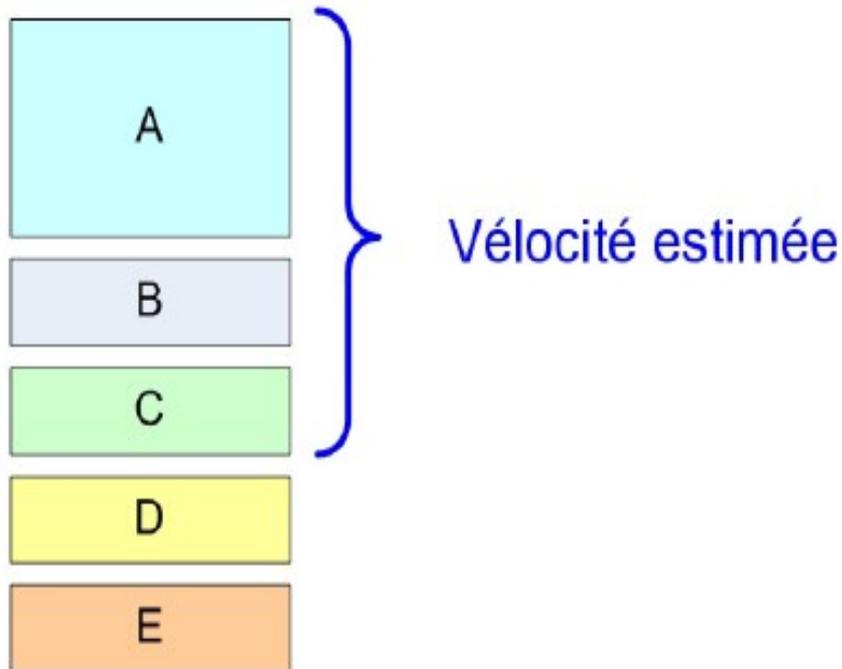
Une fois établie, la vélocité peut être utilisée pour planifier les projets et prévoir les dates de release et de terminaison du produit.

Comment le calcul de la vélocité peut avoir du sens alors que l'estimation des éléments du backlog est intentionnellement approximative ? La loi des grands nombres a tendance à moyennner l'approximation des estimations.

en anglais : velocity

Vélocité

Backlog Produit

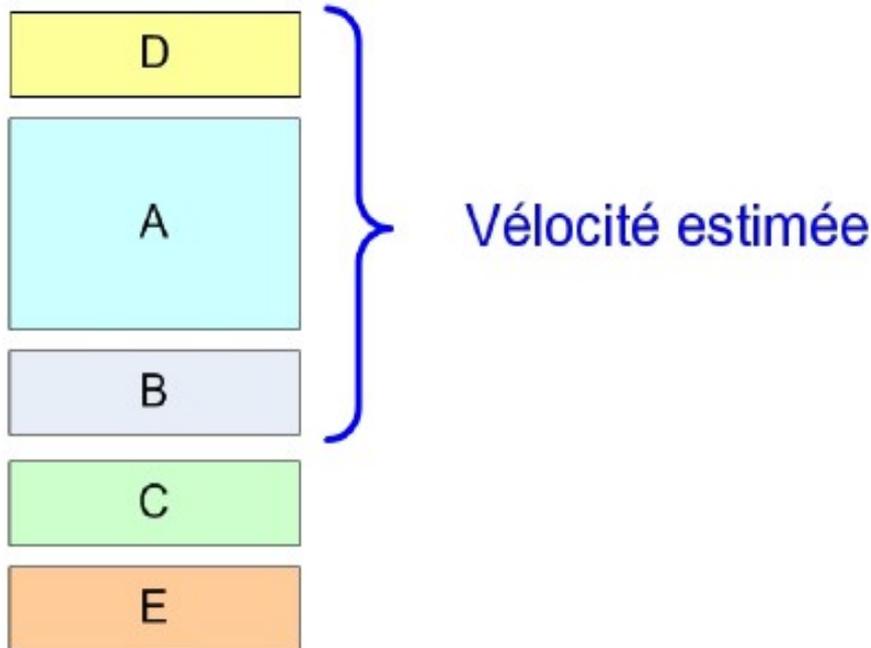


Le directeur de produit est déçu que l'histoire D ne soit pas incluse dans le sprint. Quelles sont ses options durant la réunion ?

Vélocité

Une option est de redéfinir les priorités. S'il donne à l'élément D le plus haut niveau d'importance, l'équipe sera obligée de l'ajouter en premier dans le sprint (et dans ce cas d'éjecter l'histoire C).

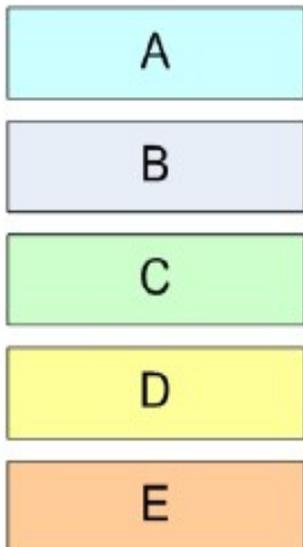
Option 1



Vélocité

La deuxième option est de changer la portée – réduire la portée de l’histoire A jusqu’à ce que l’équipe croie que l’histoire D va tenir dans le sprint.

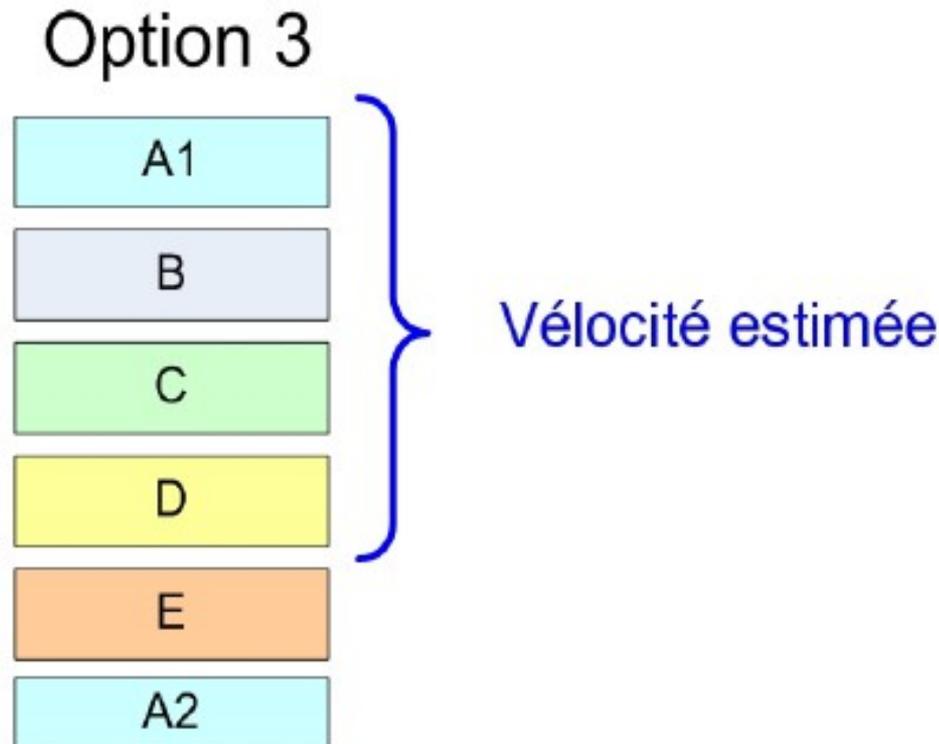
Option 2



Vélocité estimée

Vélocité

La troisième option est de partager une histoire. Le directeur de produit pourrait décider qu'il y a certains aspects de l'histoire A qui ne sont finalement pas si importants, et donc il partage A en deux histoires A1 et A2 avec des niveaux d'importance différents.



Vélocité

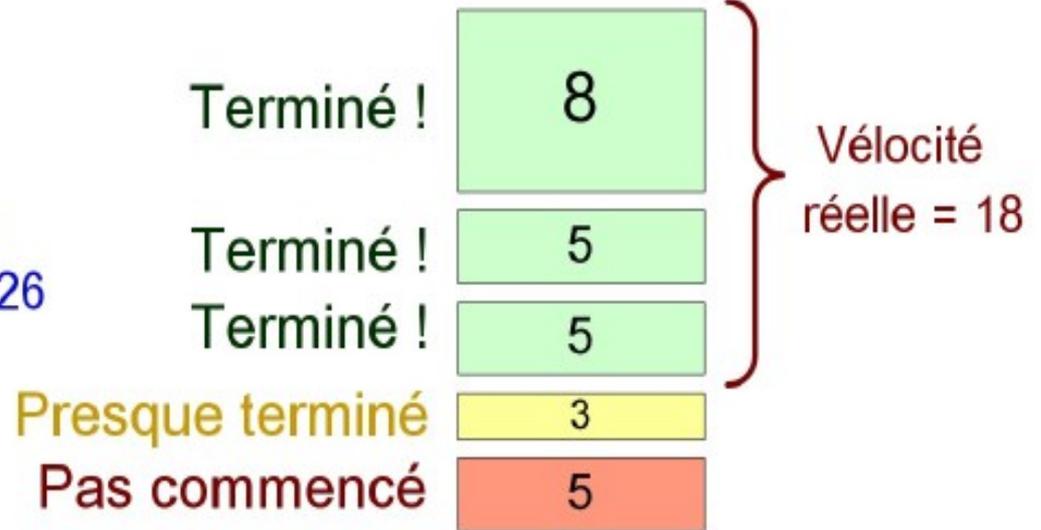
Vélocité estimée et vélocité effective

Début du sprint



Vélocité
estimée = 26

Fin du sprint



Vélocité
réelle = 18

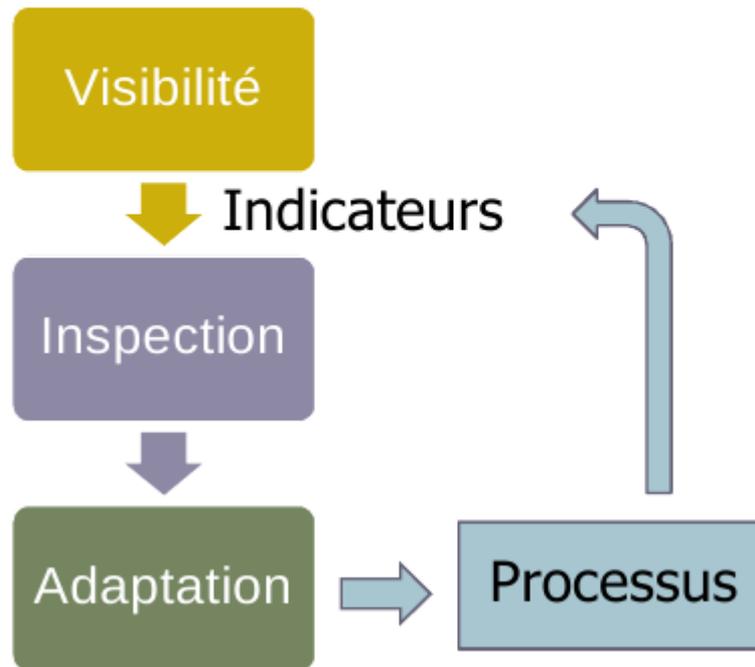
Estimations, mesures et indicateurs Agile

Présenter les nouveaux indicateurs associés aux méthodes agiles, et le processus pour les obtenir.



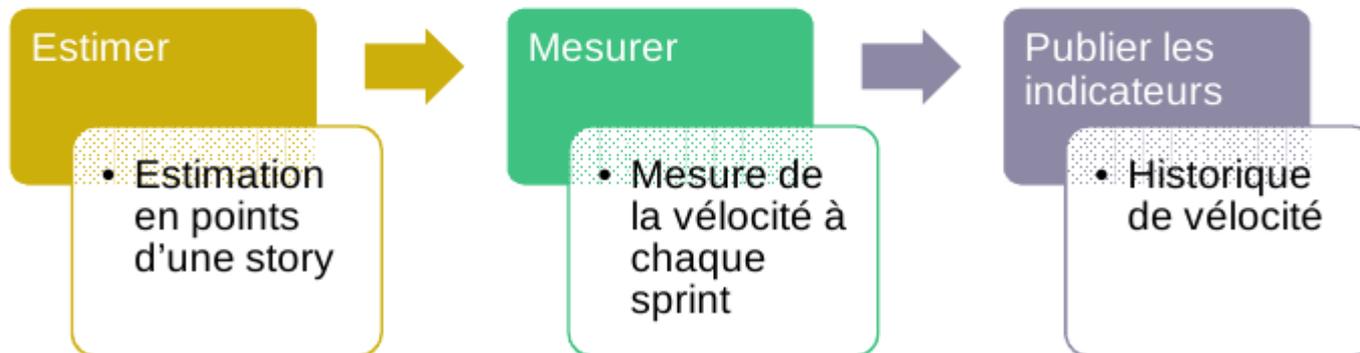
Estimations, mesures et indicateurs Agile

Pourquoi mesurer ?



Estimations, mesures et indicateurs Agile

Estimation, mesure, indicateur



Estimations, mesures et indicateurs Agile

L'estimation est collective

- En particulier, les estimations de taille ou de durée sont faites par ceux qui réalisent.

L'estimation se base sur des mesures

- Par exemple, la capacité de l'équipe est estimée à partir de la mesure de la vélocité sur les sprints passés.

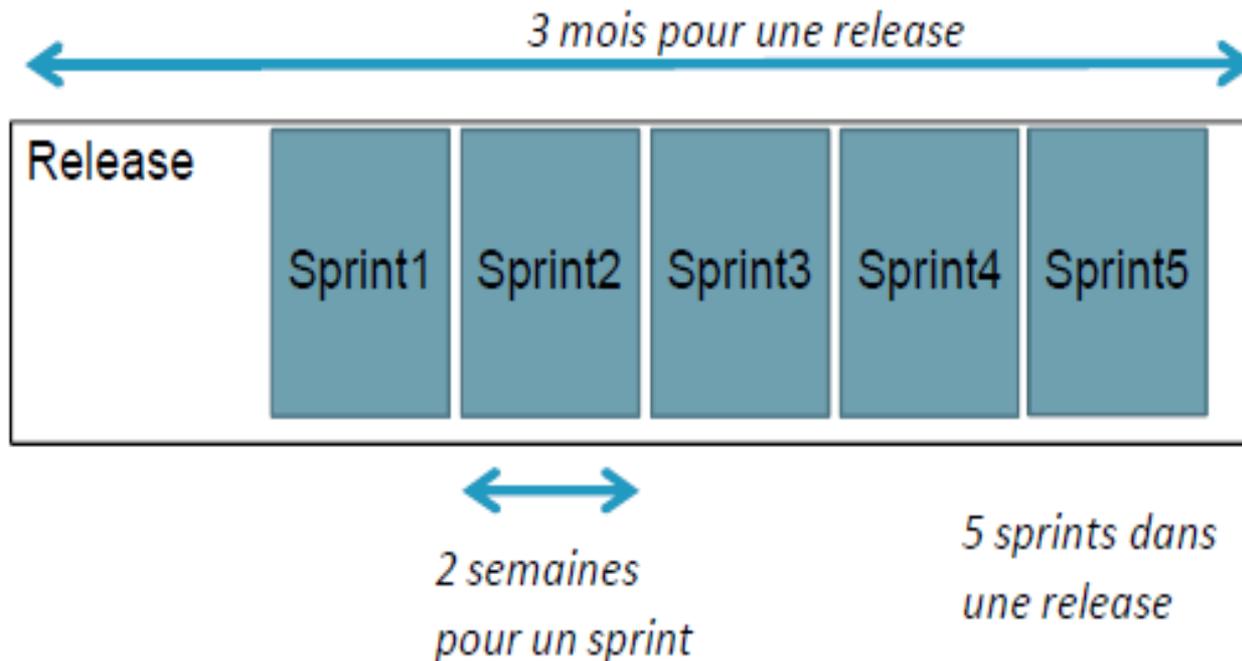
Les mesures clés portent sur des résultats visibles

- Un suivi de projet traditionnel porte sur l'avancement de tâches qui ne produisent pas de résultat visible, tandis que le suivi agile s'appuie sur les stories finies, qui sont visibles.

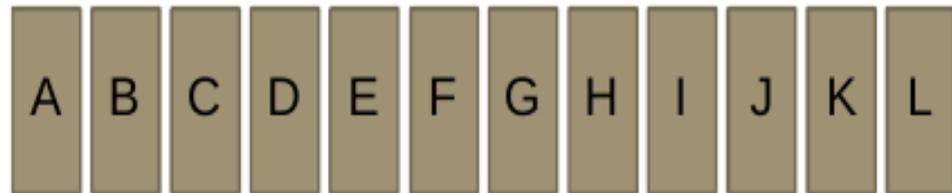
La transparence est la règle absolue

- Les indicateurs, qui sont mis à jour régulièrement, sont affichés et restent publics.

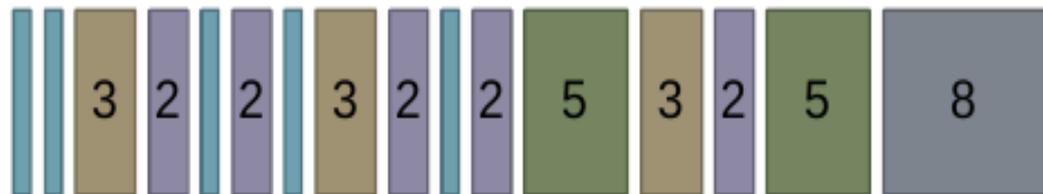
Estimations, mesures et indicateurs Agile



Estimations, mesures et indicateurs Agile

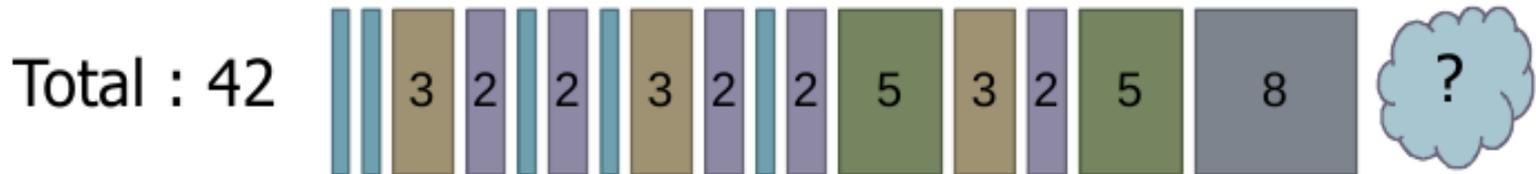


Toutes les stories ne sont pas de taille identique



Estimations, mesures et indicateurs Agile

Taille du backlog



Ce qui n'est pas estimé
n'est pas compté

Estimations, mesures et indicateurs Agile

Vélocité et capacité

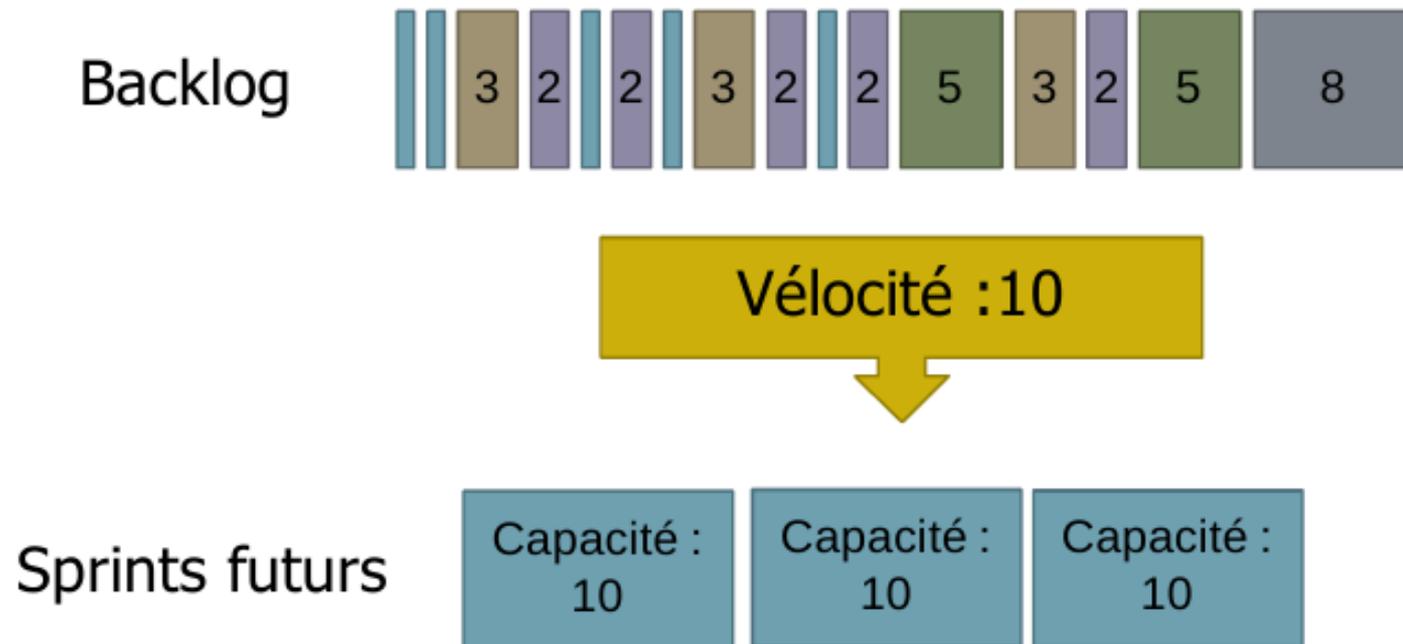
La vélocité est une mesure sur les sprints passés

La capacité est une prévision sur les sprints futurs

Pour estimer la capacité, le mieux est de se baser sur la vélocité

Estimations, mesures et indicateurs Agile

De la taille à la durée



Estimations, mesures et indicateurs Agile

Périmètre fixé

Taille du backlog : 42

Capacité de l'équipe : 10

Durée d'un sprint : 3 semaines

Date de fin : dans 15 semaines

Estimations, mesures et indicateurs Agile

Date fixée

Taille du backlog : 97

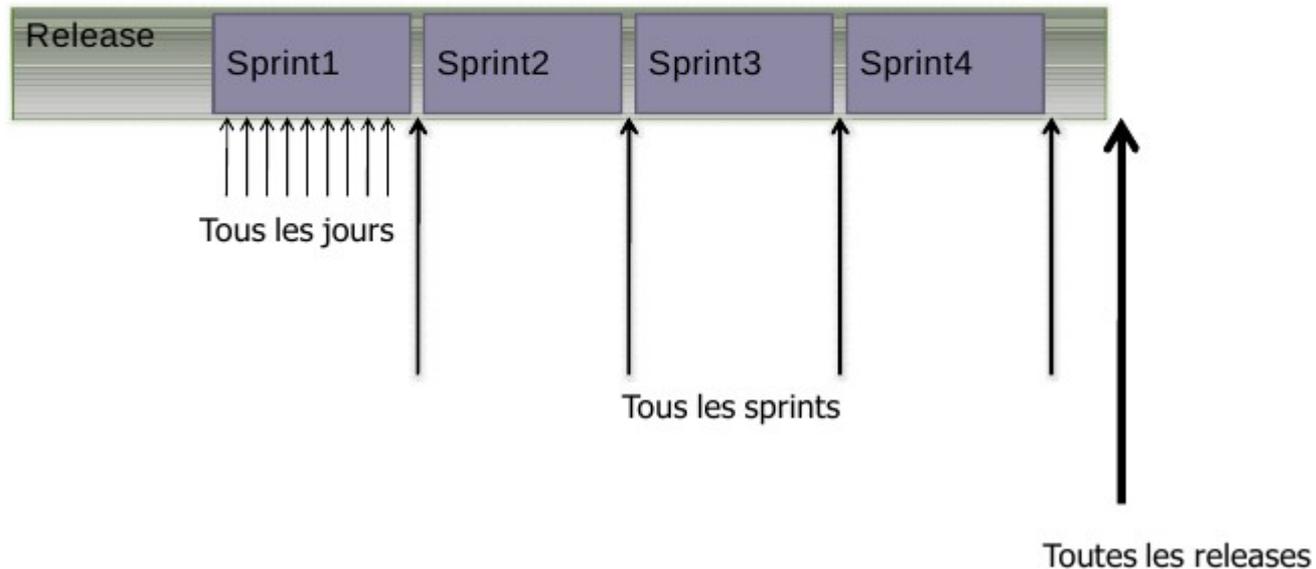
Release à date fixée : il reste 5 sprints

Capacité estimée de l'équipe :
17

Faisable : 85 sur 97

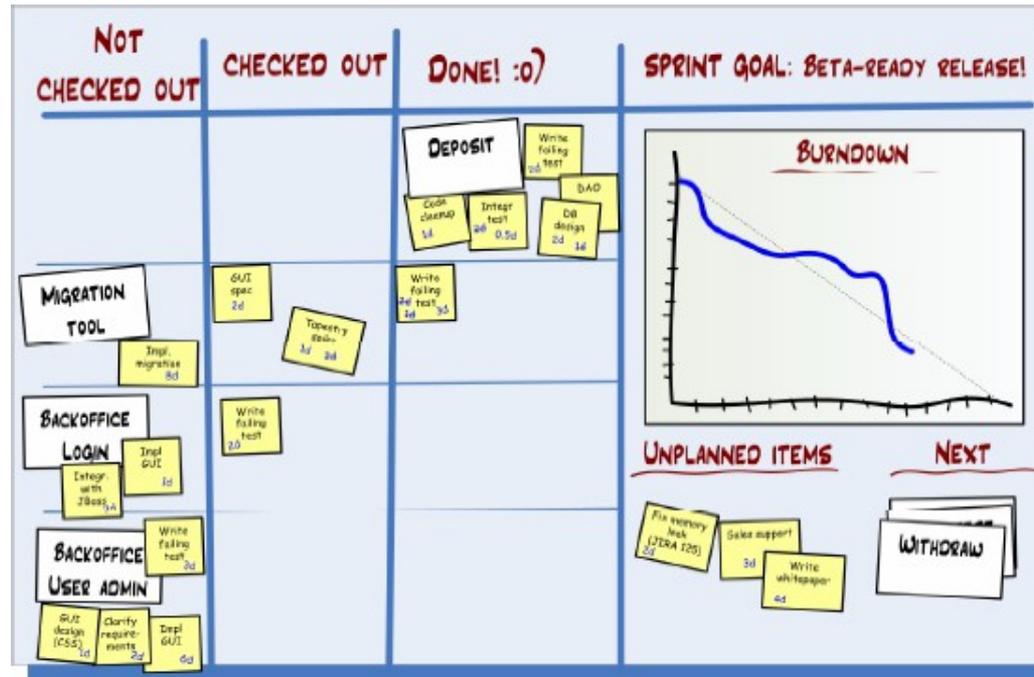
Estimations, mesures et indicateurs Agile

Mesures



Scrum – Indicateurs de projet 1/2

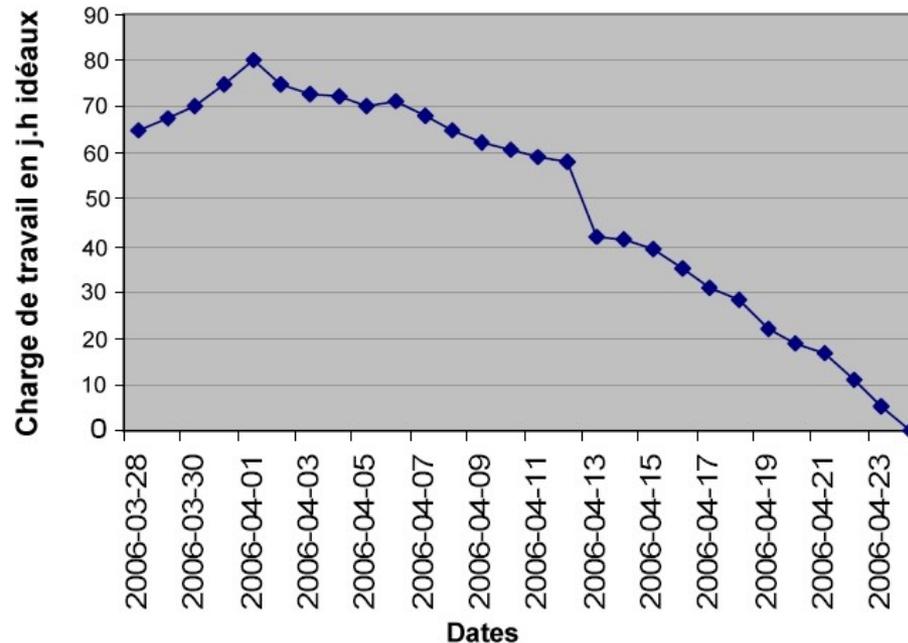
■ Le tableau des tâches



Source : « Scrum and XP from the trenches » de H. Kniberg, 2007

Scrum – Indicateurs de projet 2/2

■ Le burndown chart



Source : « Summary of Scrum », Signifikant Svenska A.B., 2007

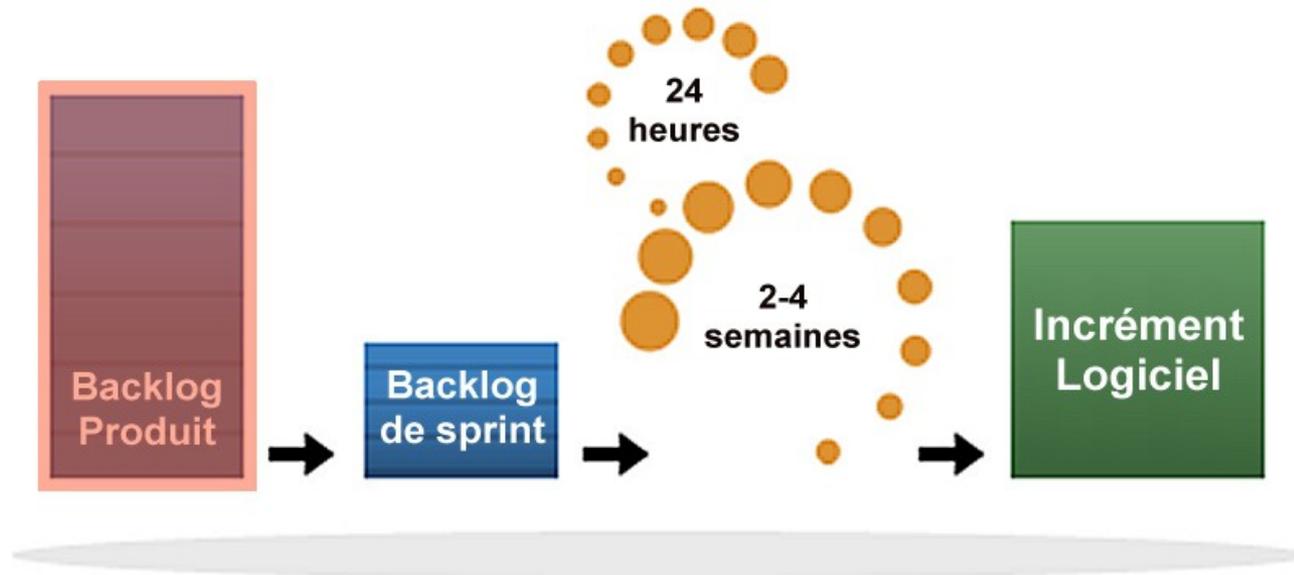
Plan

Le travail au quotidien

Le travail au quotidien

- L'organisation du travail au quotidien
- Le daily scrum

Scrum – Organisation 1/5

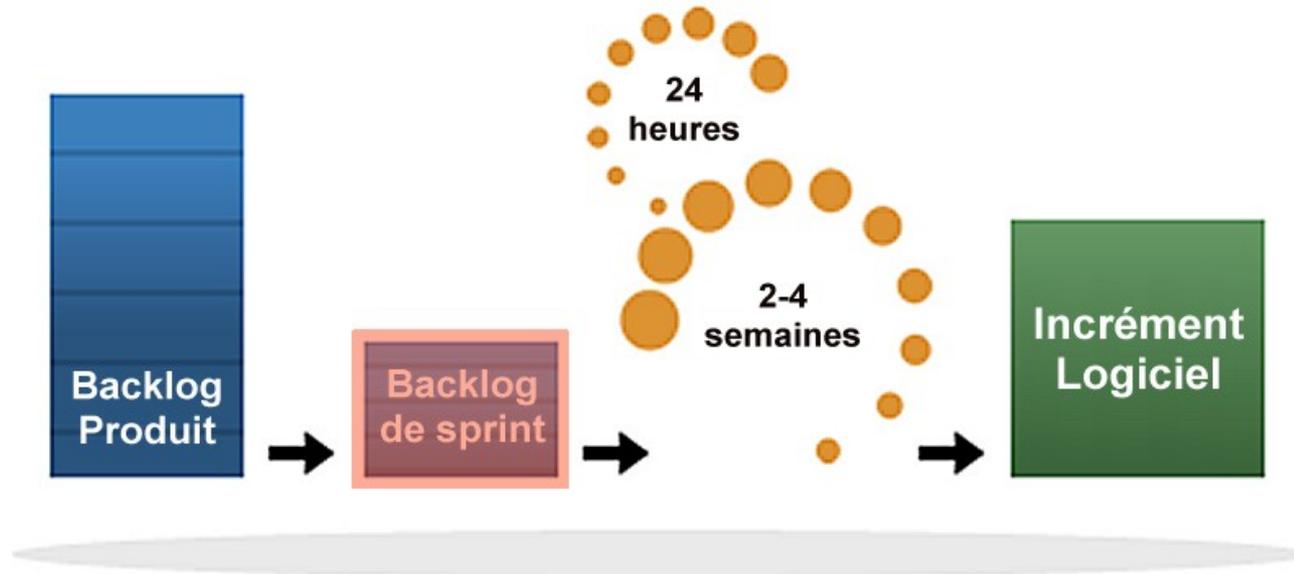


Source : www.scrumalliance.org

1. Backlog produit (ou catalogue des besoins)

- Besoins priorisés par le product owner
- Besoins évalués par l'équipe

Scrum – Organisation 2/5

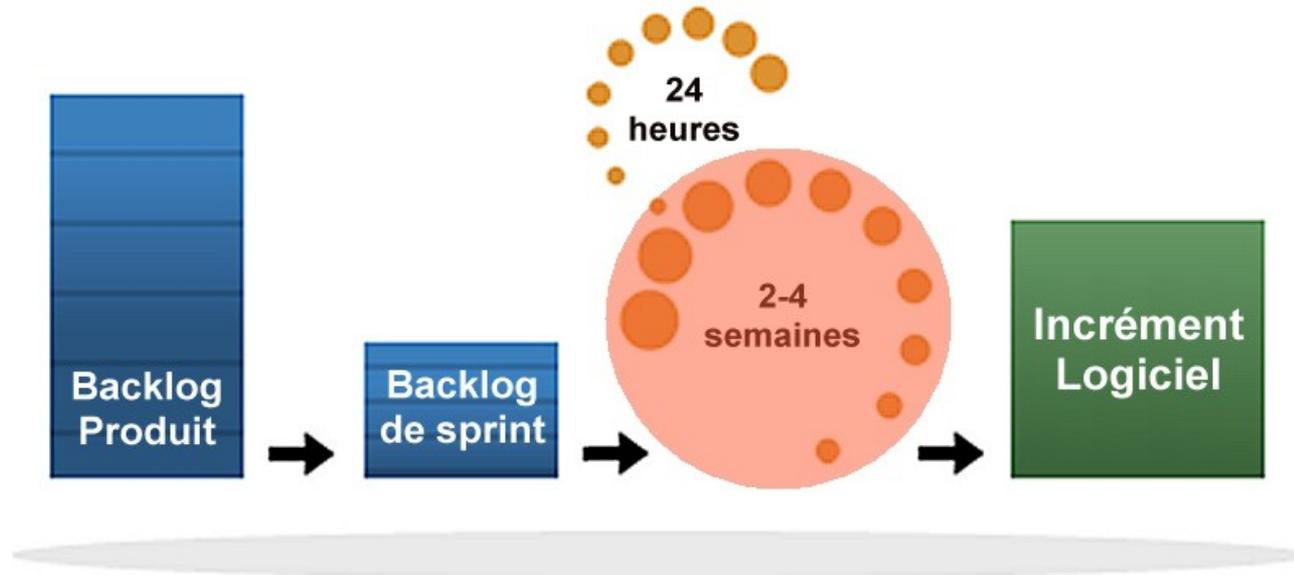


Source : www.scrumalliance.org

2. Backlog de sprint

- Extrait du backlog produit
- Besoins éclatés en tâches

Scrum – Organisation 3/5

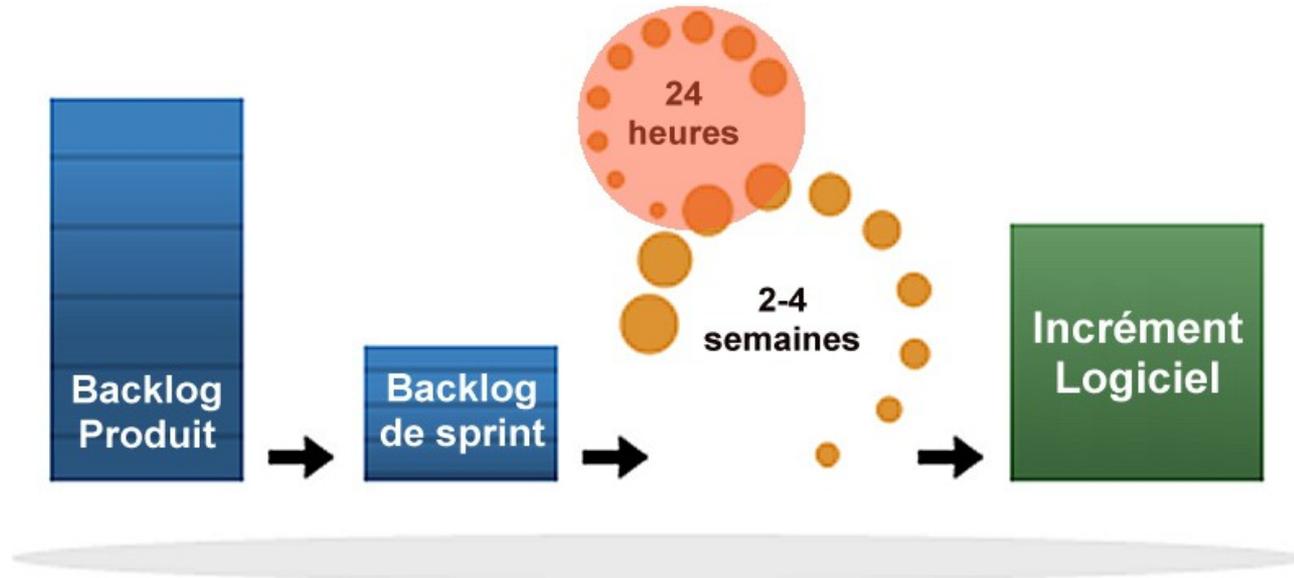


Source : www.scrumalliance.org

3. Sprint

- Développement des fonctionnalités du backlog de sprint
- Aucune modification du backlog de sprint possible

Scrum – Organisation 4/5

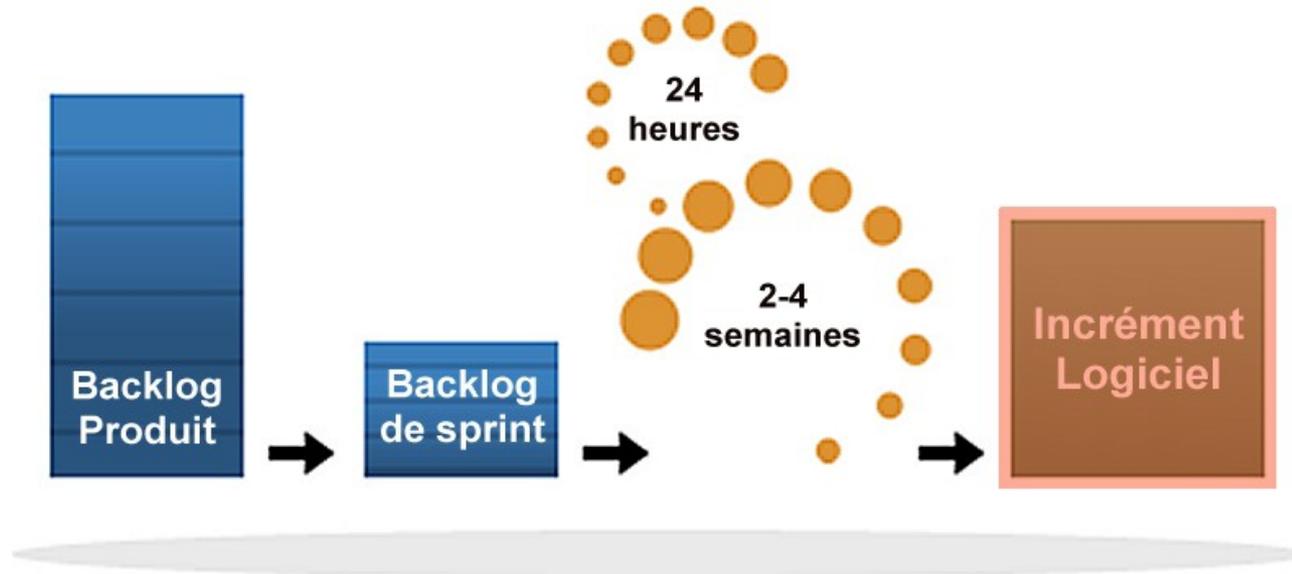


Source : www.scrumalliance.org

4. Mêlée quotidienne

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

Scrum – Organisation 5/5



Source : www.scrumalliance.org

5. Incrément logiciel : livré au product owner à la fin du sprint.

Daily scrum

- Chaque jour : $\leq 15\text{mn}$
- Participants : l'équipe, Product Owner, le Scrum Master + les autres
- Objectif : synchronisation de l'équipe
- 3 questions :
 - *Qu'est-ce que j'ai fait hier ?*
 - *Qu'est-ce que je compte faire aujourd'hui ?*
 - *Quelles sont les difficultés que je rencontre ?*

Plan

L'amélioration continue

L'amélioration continue

- **Le feedback**
- **La rétrospective**
- **Les retours d'expérience**

Revue & Retrospective du sprint

Sprint retrospective

- Réunion interne à l'équipe
- Objectif : comprendre les erreurs commises

Sprint review

- Valider le produit pendant le sprint
- Démonstration tâche réalisée
- Validation



Revue de sprint

- L'équipe présente ce qu'elle a fait pendant le sprint
- Se fait avec une démo des nouvelles fonctionnalités ou de l'architecture
- Informel
 - Préparation < 2h
 - Pas de slides
- Toute l'équipe participe
- On invite du monde



Rétrospective du sprint

- 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
 - ScrumMaster
 - Directeur produit
 - Equipe
 - Eventuellement clients et autres intervenants

Start / Stop / Continue

- Toute l'équipe collecte du feedback et discute sur ce qu'elle aimerait :

Commencer à faire

Juste une façon
parmi d'autres
de faire une
rétrospective.

Arrêter de faire

Continuer à faire

Suggestions

Si l'équipe sent qu'elle est s'est engagée au-delà de ses capacités, elle doit rencontrer le Propriétaire du produit afin d'éliminer ou bien de réduire la portée des certains éléments faisant parti du Carnet du Sprint. À l'opposé, si l'équipe croit avoir du temps disponible, elle peut travailler de concert avec le propriétaire du produit pour ajouter au Canet du Sprint des éléments additionnels du Carnet du produit.

Suggestions

Lorsqu'une équipe débute son apprentissage de Scrum, l'utilisation de Sprints d'une durée de deux semaines facilite l'apprentissage tout en limitant l'incertitude. Des Sprints de cette durée peuvent être synchronisés avec les autres équipes en combinant le résultat de deux sprints consécutifs (pour obtenir un cycle d'un mois).

Suggestions

Réunion de planification du sprint

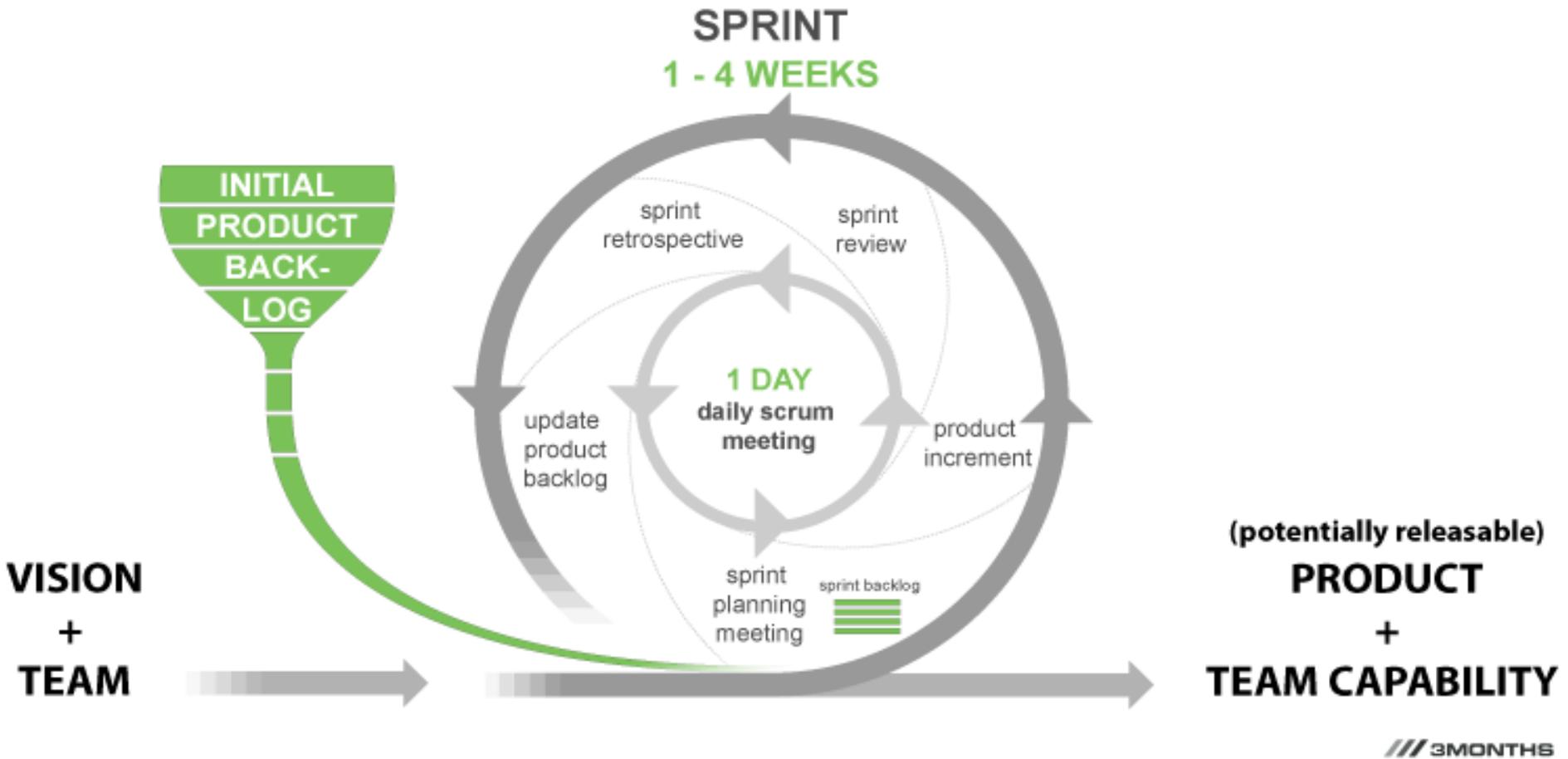
Habituellement, seulement environ 60-70% des tâches du Carnet de Sprint sont précisées en détail dans la réunion. Le reste peut être estimé sommairement et détaillé au cours du Sprint.

Suggestions

Tests

Les tests d'acceptation sont une autre propriété qui est souvent associée aux éléments du Carnet du produit. Ils peuvent souvent remplacer des descriptions textuelles détaillées par une définition testable de ce que le produit doit faire pour que l'élément soit considéré comme « complété »

Articulation de la méthode



Plan

Scrum à grande échelle

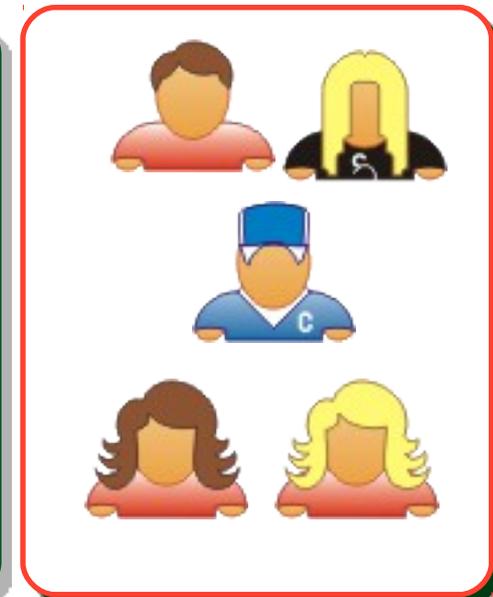
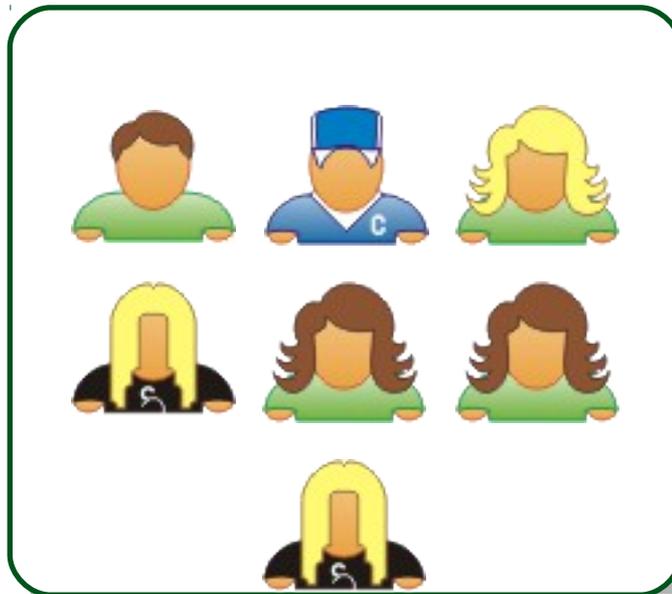
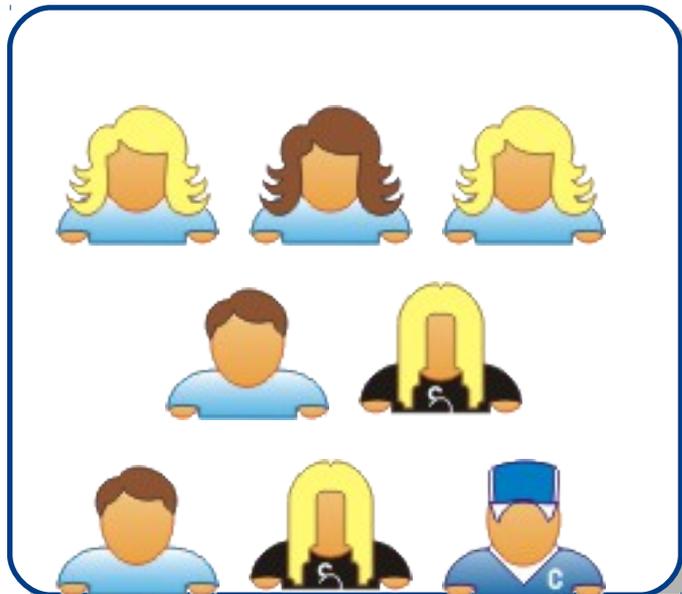
Scrum à grande échelle

- La scalabilité
- Les outils

Scrum à grande échelle

- **Une équipe typique c'est 7 ± 2 personnes**
 - Le changement d'échelle se fait avec plusieurs équipes
- **Facteurs dans la scalabilité**
 - Type d'application
 - Taille de l'équipe
 - Répartition géographique des équipes
 - Durée du projet
- **Scrum a été utilisé pour de nombreux projets de plus de 500 personnes**

Scalabilité avec un scrum de scrums



Scrum – Ingénierie logicielle

- Scrum est une méthode de gestion de projet
- Doit être complétée par des techniques d'ingénierie logicielle
- Complémentaire avec eXtreme Programming :
 - Test Driven Development
 - Intégration continue

Scrum – Equipes plus grandes

- **Principes :**
 - **Commencer par une équipe Scrum standard**
 - **Création de plusieurs équipes – essaimage**
- **Adaptation de la méthode :**
 - **Scrum des scrums**
 - **Rôle de team lead**
- **Problèmes à traiter :**
 - **Dispersion géographique**
 - **Développement off-shore**

Les outils

- **Outils traditionnels**
 - Tableau blanc et post-its
 - Excel – Backlog produit et backlog de sprint
- **Outils dédiés**
 - Outils commerciaux / Open source
 - Gèrent une charge de travail
 - Absence de PERT / Gantt
 - Intégration avec : IDE, contrôle de sources, gestion des tests, bug tracking, intégration continue.
- **Autres outils**
 - Connexion large bande
 - Wiki, webcams, messagerie instantanée...

Forces et questions ouvertes

Forces

Questions

Copilotage avec le client

- Propice à une confiance réciproque
- Transparence sur l'avancement (résultats rapides)
- Adaptabilité : le client peut modifier son besoin

- Le client est-il prêt à s'impliquer?
- Comment gérer le changement par rapport au mode classique? (organisation, comportements, contrat, ...)
- Comment éviter de retourner dans le mode classique au 1er point dur?

Equipe & responsabilité collective

- Plus de responsabilisation, plus de motivation
- Pas d'individualisme
- Amélioration permanente

- Capacité à travailler en équipe, à partager?
- Capacité à se responsabiliser, à s'impliquer (envie?)
- Capacité à faire l'effort d'être polyvalent? (alternance codage, tests, correctif)
- Comment gérer les ressources non adaptées à ce mode de fonctionnement?

Forces et questions ouvertes

Forces

- Orienté productivité et qualité
 - pas de travail superflu
 - Environnement adapté (IC)
 - Orienté tests (TDD)
- Plus de souplesse et de créativité
- Recherche de la simplicité
- Pilotage au quotidien

Questions

- Framework minimaliste > risque d'hétérogénéité? (si mauvaise communication dans l'équipe...)
- Comment justifier et estimer le surcoût lié aux tests et à la mise en place de l'environnement d'IC?
- Pour un 1er Scrum, comment estimer la charge de pilotage? (accrue par rapport à forfait "classique" + coût des outils si spécifiques agile)

Cadre de production, méthodologies

Perspectives

- Pas d'évolution, peu de critiques
- 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

Conclusion

- **Méthode de gestion de projet – développement logiciel**
- **A compléter avec des techniques d'ingénierie logicielle**
- **Rien de totalement nouveau**
- **Méthode à la mode. Conditions propices nécessaires**
- **Expérimentations prometteuses**
- **Principal bénéfice : des équipes motivées**

Où se renseigner ?

- www.mountangoatsoftware.com/scrum
- www.scrumalliance.org
- Www.scrum.org
- www.controlchaos.com
- scrumdevelopment@yahooogroups.com
- **En français :**

Le blog Scrum Méthodes Agiles :

scrum.aubryconseil.com

Questions ?