Manifeste des méthodes agiles

- Signé par 17 personnalités
- 4 valeurs

L'équipe	« Personnes et interaction plutôt que processus et outils »
L'application	« Logiciel fonctionnel plutôt que documentation complète »
La collaboration	« Collaboration avec le client plutôt que négociation de contrat »
L'acceptation du changement	« Réagir au changement plutôt que suivre un plan »

Méthodes agiles : les 12 principes

- Satisfaction client
- Changement bienvenu
- Livraisons fréquentes
- Collaboration quotidienne
- Motivation et encouragements
- Communication face-à-face
- Logiciel sert de mesure

Les 12 principes (suite)

- Rythme soutenable
- Attention continue à la technique et à la conception
- Simplicité
- Auto-organisation
- Ajustements continus

Exemples

- Scrum
- Extreme programming (XP)
- Rapid Application Development (RAD)
- Adaptive software development (ASD)
- Crystal clear
- Dynamic software development method (DSDM)
- Feature driven development

Gestion de projets agiles avec Scrum

Cycle de formation « base »

Pierre NEIS

- Scrum Coach
  http://managingagile.blogspot.com/

1

LE JARGON DE SCRUM

Le Jargon

Sprint:	Est une itération
Backlog:	Est une liste de tâches ouvertes
Product Backlog:	Est une liste d'items ouverts pour livrer le produit
Sprint Backlog:	Est une liste de tâches ouvertes attribuées au Sprint
L'EQUIPE ou la Development TEAM:	C'est l'équipe de développement
La Scrum Team:	C'est l' EQUIPE + le ScrumMaster + le Product Owner
Estimation Meeting	C'est la réunion d'estimation
Sprint Planning Meeting	C'est la réunion de planification de Sprint
Daily Scrum ou Stand-up Meeting	C'est la réunion journalière de 15' où l'EQUIPE inspecte et adapte, coordonne son effort.
Sprint Review ou Revue de Sprint	C'est la réunion de fin de Sprint où tous les acteurs deu projet se retrouvent pour inspecter les délivrables du Sprint.
Rétrospective	C'est la réunion d'inspection et d'adaption de la Scrum Team.

Objectif

- Comprendre les fondamentaux de Scrum
- Savoir utiliser les outils de Scrum
- Être en mesure de démarrer votre projet Scrum

Périmètre

- Historique
- La théorie « Scrum »
- La Philosophie agile

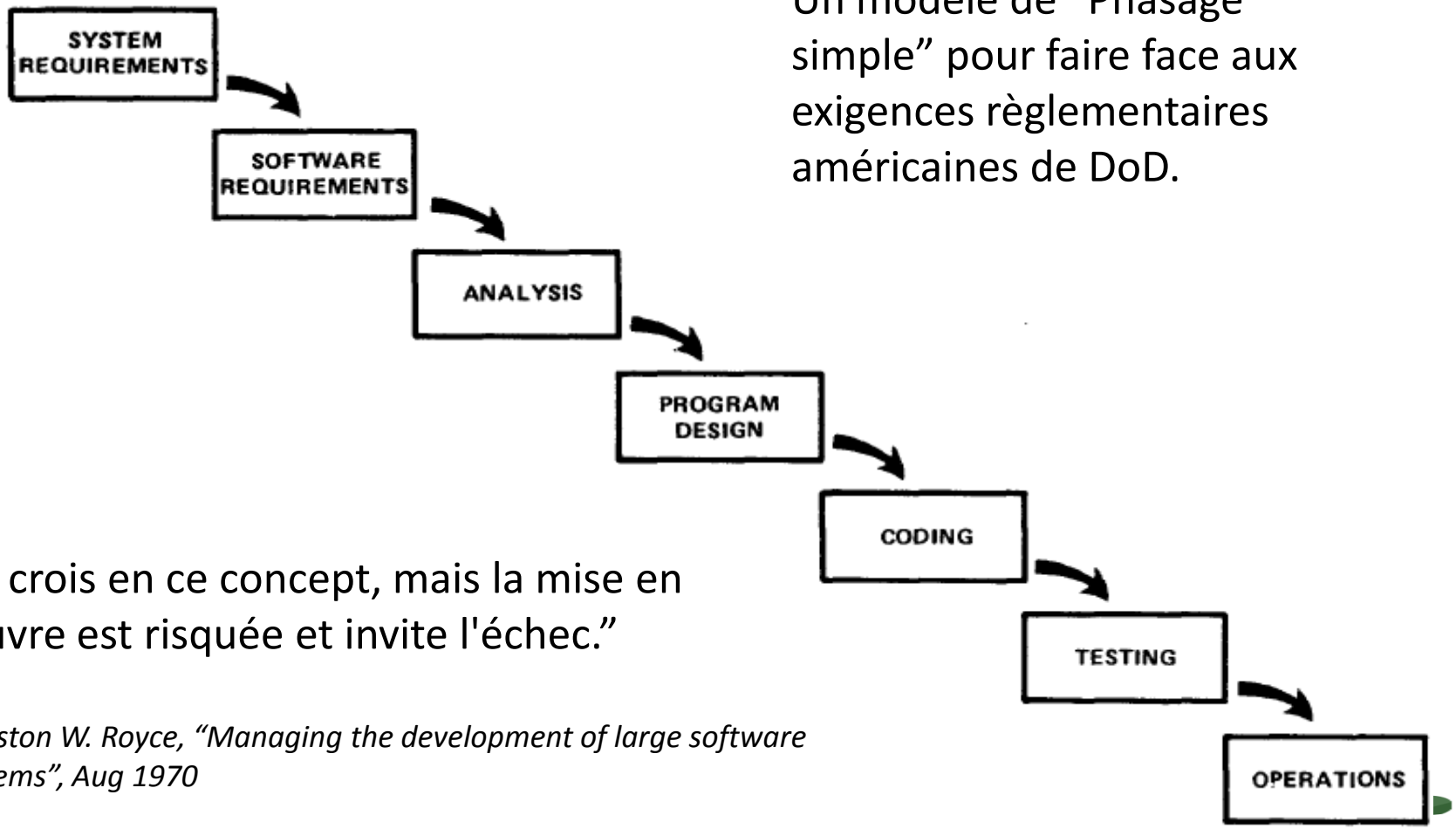
Scrum

- Les racines de Scrum se retrouvent dans la publication de Takeuchi et Nonaka dans "The New Product Development Game"
- Ken Schwaber et Jeff Sutherland la formalisent en 1996

- SCRUM est un framework de méthodologie
- SCRUM est un framework non prescriptif

Le modèle “Grandiose” de Winston Royce

Un modèle de “Phasage simple” pour faire face aux exigences règlementaires américaines de DoD.



“Je crois en ce concept, mais la mise en œuvre est risquée et invite l'échec.”

Winston W. Royce, “Managing the development of large software systems”, Aug 1970

Nous perdons la course de relais

HBR

JANUARY–FEBRUARY 1986

The New New Product Development Game

Hirotaka Takeuchi and Ikujiro Nonaka

The rules of the game in new product development are changing. Many companies have discovered that it takes more than the accepted basics of high quality, low cost, and differentiation to excel in today's competitive market. It also takes speed and flexibility.

This change is reflected in the emphasis companies are placing on new products as a source of new sales and profits. At 3M, for example, products less than five years old account for 25% of sales. A 1981 survey of 700 U.S. companies indicated that new products would account for one-third of all profits in the 1980s, an increase from one-fifth in the 1970s.[1]

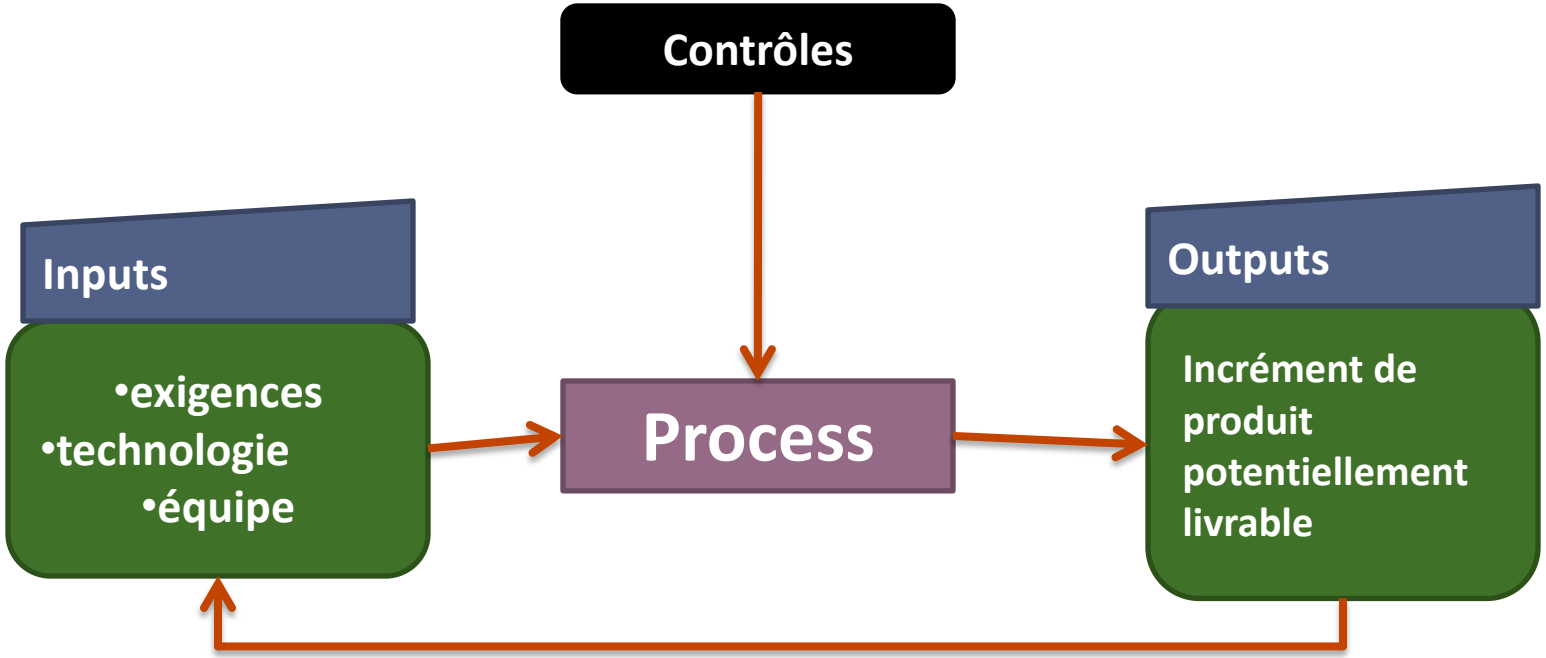
This new emphasis on speed and flexibility calls for a different approach for managing new product development. The traditional sequential or "relay race" approach to product development—exemplified by the National Aeronautics and Space Administration's phased program planning (PPP) system—may conflict with the goals of maximum speed and flexibility. Instead, a holistic or "rugby" approach—where a team tries to go the distance as a unit, passing

- " L'approche "course de relai" du développement de produit… peut entrer en conflit avec les objectifs de vitesse maximale et de flexibilité. A contrario, une démarche holistique ou « rugby » où une équipe essaie d'aller au loin comme une unité, passant la balle en arrière, peut mieux servir aujourd'hui les exigences de la compétivité. »

Hirotaka Takeuchi and Ikujiro Nonaka, "The New New Product Development Game", *Harvard Business Review, January 1986*

Est-ce que le développement logiciel est un processus défini?

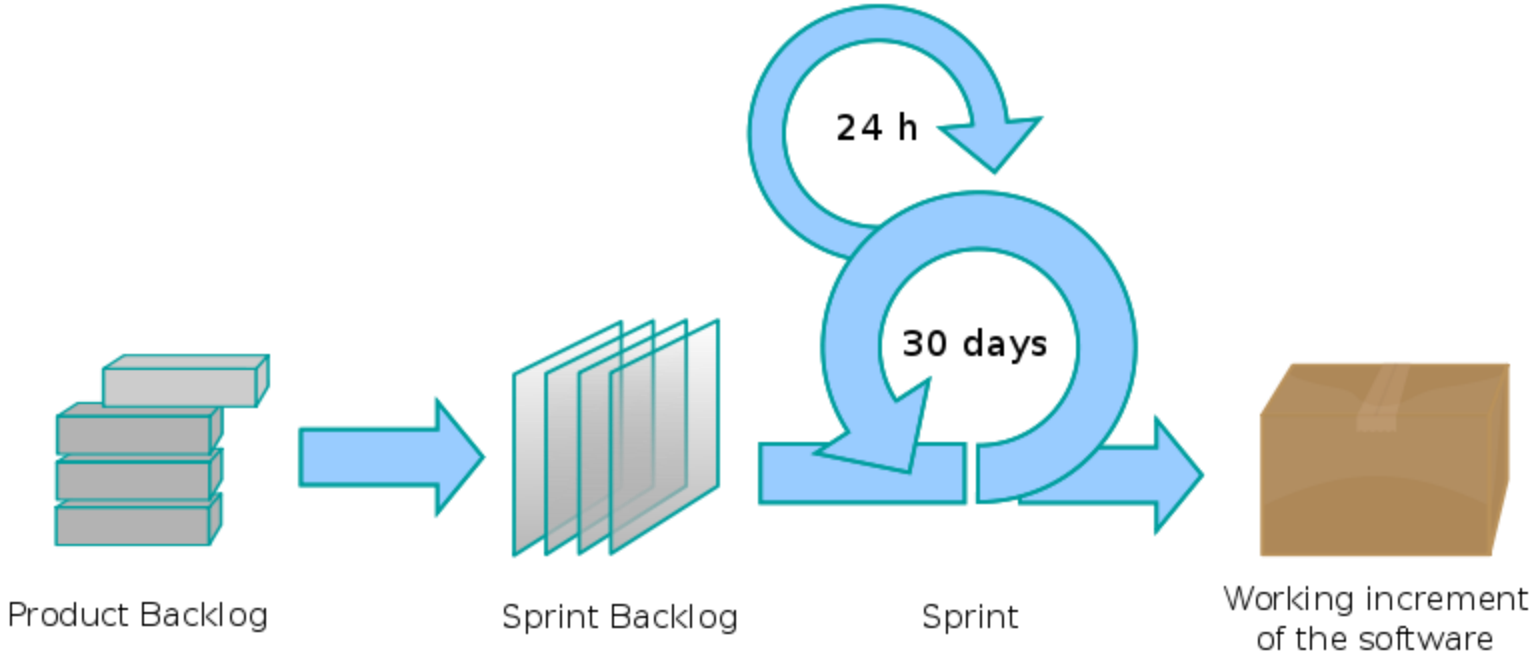
Le modèle empirique



Le modèle empirique est dépendant de fréquentes inspections et adaptations pour atteindre l'objectif.

La théorie SCRUM

Scrum est un processus empirique



Scrum repose sur 3 pieds

Transparence

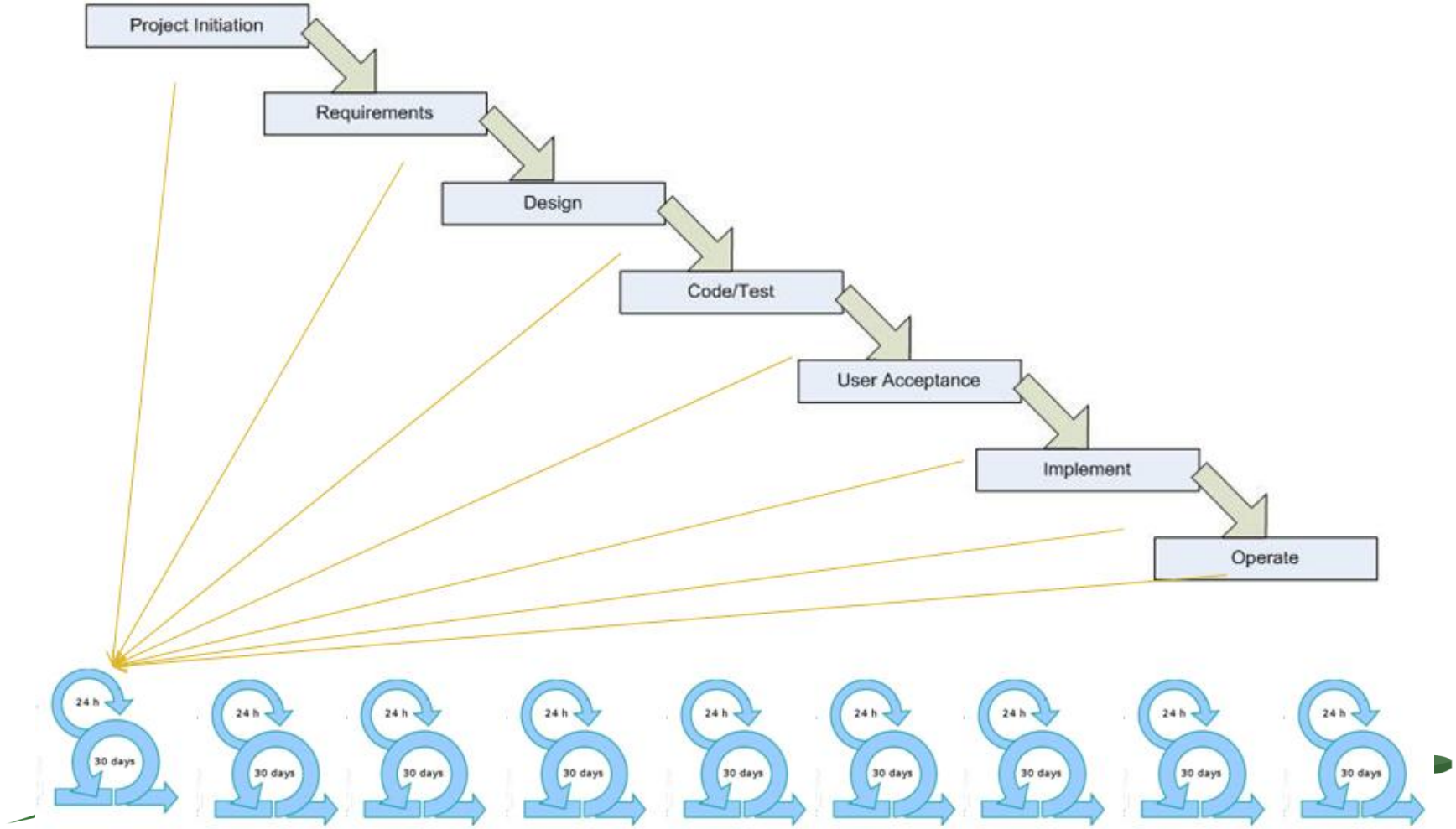
Inspection

Adaptation

10 Pratiques de base

1. Vision claire et partagée
2. Product Backlog entretenu
3. Product Backlog priorisé en fonction de la valeur métier
4. Items de backlog triés par l'équipe
5. Daily Scrums
6. Sprints non perturbés ni par le Management ni par le(s) client(s)
7. L'Equipe ne délivre que des items « terminés »
8. Revue de Sprint collaborative
9. Rétrospective concentrée sur l'amélioration du travail et du processus de l'équipe et de l'organisation
10. Burndown Charts (graphiques de reste-à-faire)

Scrum vs Modèle en “cascade”



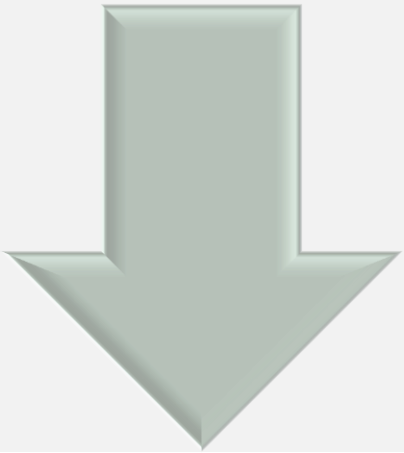
La Philosophie Agile

L’Agile Manifesto

Manifeste pour le développement Agile de logiciels

Les individus et leurs interactions

- Des logiciels opérationnels
- La collaboration avec les clients
- L'adaptation au changement



les processus et les outils

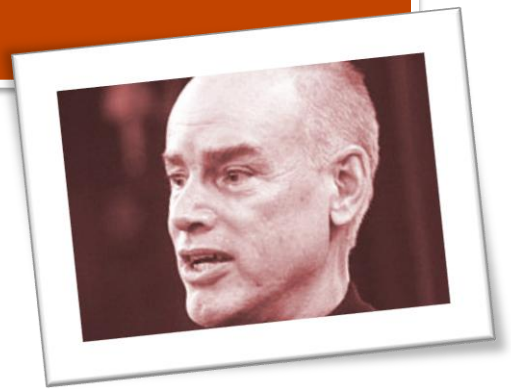
- une documentation exhaustive
- la négociation contractuelle
- le suivi d'un plan

Principes sous-jacents au manifeste

Priorité	
1	satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
2	Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
3	Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
4	Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
5	Réalisez les projets avec des personnes motivées.
6	La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.
7	Un logiciel opérationnel est la principale mesure d'avancement.
8	Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
9	Une attention continue à l'excellence technique et à une bonne conception renforcent l'Agilité.
10	La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
11	Les meilleures architectures, spécifications et conceptions émergent d'équipes auto organisées.
12	À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

CONTENU DE SCRUM

Introduction par Ken Schwaber



- Scrum n'est pas une méthodologie. Scrum ne fournit pas les réponses à la manière de construire des logiciels de qualité plus rapidement.
- Scrum est un cadre dans lequel le jeu du développement de produit est joué.
- Votre équipe joue et, le bon ou le mauvais deviennent très visibles.
- Votre équipe est dans un processus d'amélioration continue.

Le principe “Pull”

Équipes auto-gérées vs Organisation traditionnelle

Equipes auto-gérées	Organisation traditionnelle
Orientées client	Pilotée par le management
Force de travail multi-compétence	Force de travail constituée de spécialistes isolés
Peu de description de poste	Beaucoup de description de poste
Information largement partagée	Information limitée
Peu de niveau de management	De npmbreux niveaux de management
Orientée Ensemble du Métier	Orientée fonction/département
Objectifs partagés	Objectifs séparés
D'apparence chaotique	D'apparence organisée
Emphatique sur l'hypothèse d'atteindre le résultat	Emphatique sur la résolution de problème
Très fort engagement des "développeurs"	Très fort engagement du Management
Améliorations continues	Améliorations incrémentales
Autorégulées	Contrôlées par le Management
Basées sur des valeurs et des principes	Basées sur des politiques et des procédures

Source: "Leading self-directed work teams" by Kimball Fisher. Traduction libre Pierre NEIS.

Les Règles

Rôles, Artifacts et Time-boxes

3 Rôles Scrum Team plus 3 Rôles organisationnels

Les Rôles de l'Équipe Scrum

- La Development Team
- Le ScrumMaster
- Le Product Owner

Les Rôles organisationnels

- Le Management
- Le Client
- Les Utilisateurs

Les Rôles de l’Equipe Scrum

Le ScrumMaster

Sa fonction

- Protège l'équipe des turbulences
- Il n'est pas un membre de l'Équipe
- Il optimise la productivité de l'Équipe
- Il contrôle l'"Inspect-&-Adapt" de l'Équipe
- Il assure que les idéaux "agiles" soient bien compris et respectés par tous les participants au projet.
- Il n'est pas responsable des déliverables.

Sa Mission

- Protéger l'Équipe Scrum
- Lever les obstacles
- Exécuter le process
- Travailler avec le Product Owner
- Changer l'Organisation

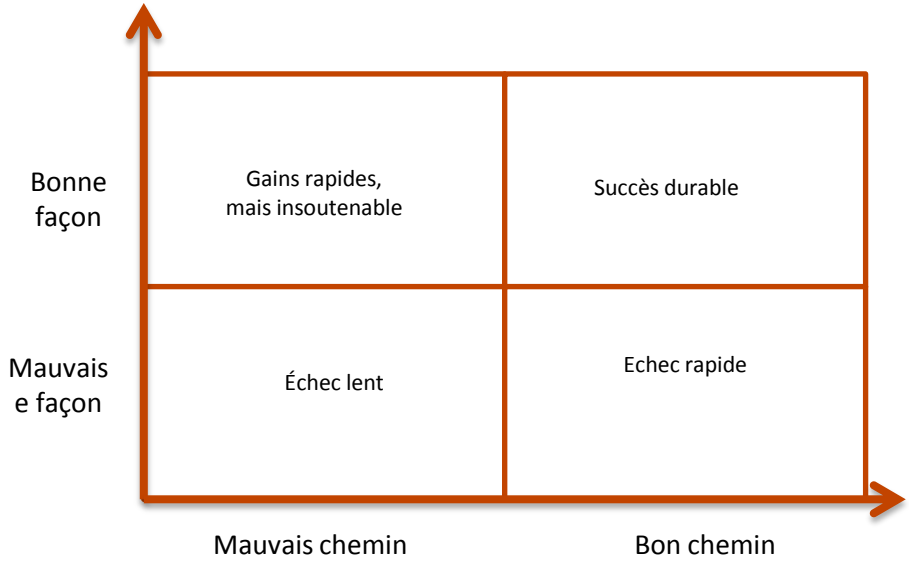
Le ScrumMaster

Agir de la bonne façon

Faire bien (Produit)

+ Produit

+ Process



- *Il forme et coache SCRUM*
- *Il régule les obstacles*
- *Il anime les réunions*
- *Il protège l'équipe*
- *Il est le gardien du process Scrum*

Le Product Owner

Sa fonction

- Il pilote le projet d'un point de vue métier
- Il communique une vision claire du produit et défini ses caractéristiques
- Il accepte ou rejette le produit à la fin de chaque Sprint
- Il s'assure que l'Équipe se concentre sur les items du Backlog de plus forte valeur ajoutée
- Il a le même objectif que l'Équipe
- Il est responsable du Retour sur Investissement et des livraisons.

Sa Mission

- Se concentre sur le retour sur investissement
- Construit et communique la vision
- Entretien le Product Backlog
- Rend compte de l'acceptance des déliverables
- Établi et maintien le Plan de Livraison

… dans Scrum?

Produit	***Process***
Sous la responsabilité du Product Owner	Sous la responsabilité du ScrumMaster

L’Équipe

Sa fonction

- Elle délivre le produit et elle est responsable de sa qualité
- Elle travaille avec les utilisateurs-finaux, le client, le Product Owner pour comprendre les exigences-métier.
- Elle s'engage volontairement
- Elle travaille continuellement avec le Product Owner pour définir la direction stratégique du Produit.

Constituer l'Équipe

- 5/9 personnes
- Multidisciplinaire
- Autogérée
- Cross-fonctionnelle / transverse
- Plus orientée compétence que fonction

Comment optimiser le travail de l'Équipe...

- Créer une règle de vie de l'Équipe
- Ne jamais utiliser le "VOUS"
- Être à l'heure
- Utiliser un "bâton de parole"
- Ne jamais être nominatif

Collaboration

- Le Product Owner n’est pas un ennemi
- D’autres équipes ont besoin de savoir que nous avons besoin d’elles.
- Nous avons tous le même objectif
- Une Équipe = un espace dédié à l’Équipe

Sa Mission

- Garantir la Qualité
- Livrer
- Livrer
- Livrer
- Estimer
- Estimer
- Estimer
- S’engager
- S’autogérer
- S’organiser .... Elle-même

Les Rôles Organisationnels

Le Client

Sa fonction

- Il demande le produit
- Il contracte l'organisation pour le développement de son produit
- Typiquement, il s'agit d'un responsable qui achète un développement de produit par un sous-traitant.
- Dans les projets internes, il s'agit principalement du sponsor au projet, c'est à dire la personne validant le projet et le budget.

Sa Mission

- Il commande le produit
- Il paye le développement du produit
- Il donne des feed-back et des révisions

Le Manager

Sa fonction

- Le management, la gestion, est primordial dans tout projet Scrum. Il permet à l'Équipe de constituer un environnement optimal pour le déroulement du projet Scrum.
- Le manager donne de la structure et de la stabilité.
- Il travaille de concert avec le ScrumMaster pour réorganiser l'organigramme de la structure et donner de la guidance si nécessaire.

Sa Mission

- Il s'assure que l'organisation puisse survivre en cas de défaillance.
- Il crée des règles et des lignes directrices.

L’Utilisateur Final

Sa fonction

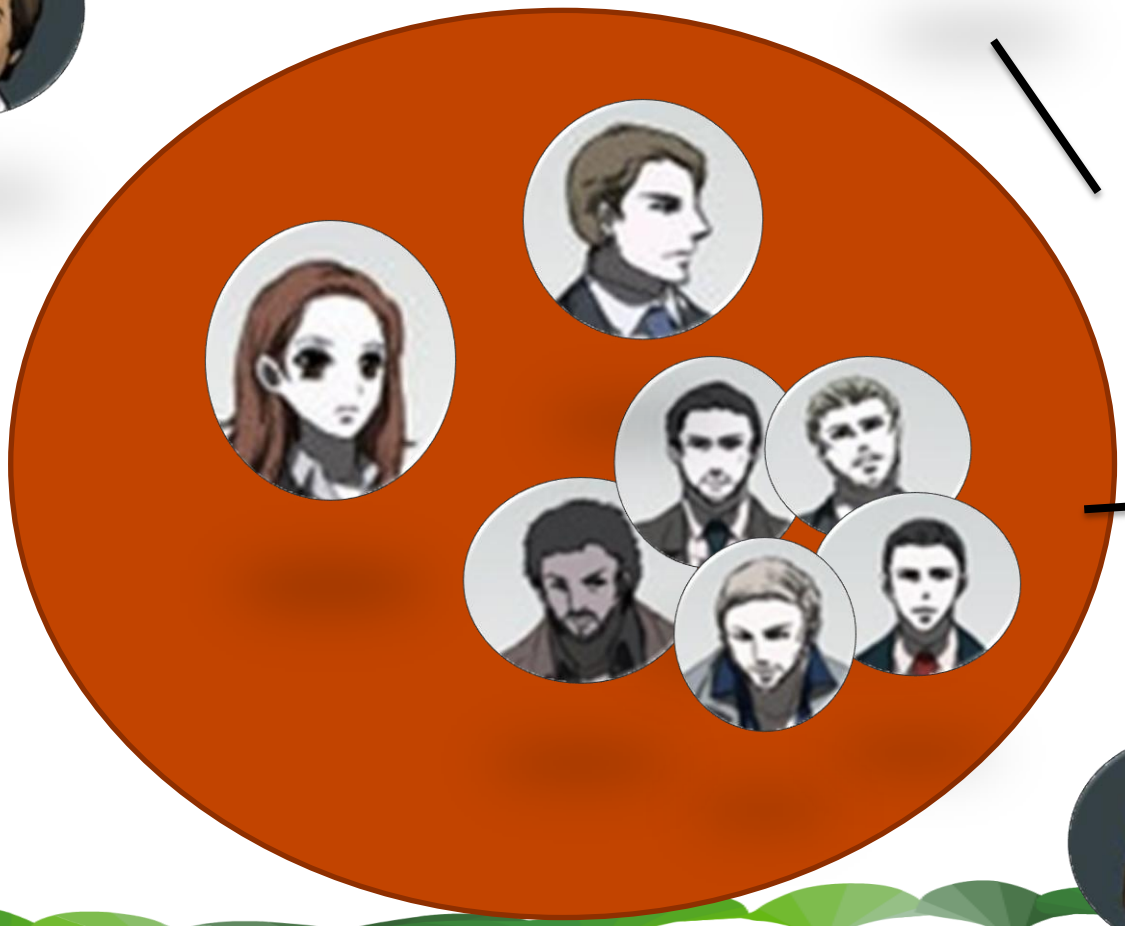
- Ce rôle peut être joué par un grand nombre de personnes.

- L'Utilisateur final est celui qui connaît les besoins et avec cette connaissance, il définit le produit en disant à l'équipe ce dont il a besoin comme fonctionnalités.

Sa Mission

- Il connaît ses besoins et ses exigences
- Il donne son feed-back lors des revues
- Il participe au Sprint Planning 1

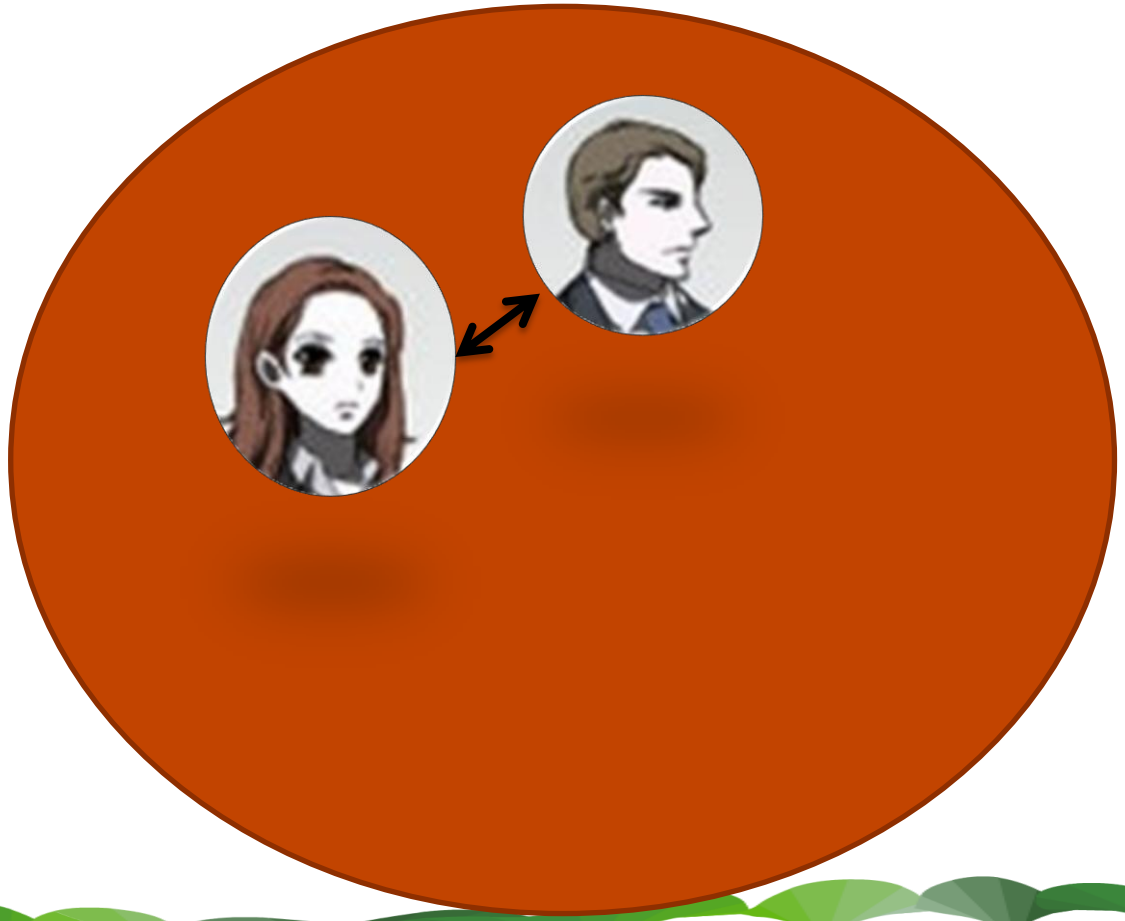
COMMENT CES RÔLES TRAVAILLENT-ILS ENSEMBLE?



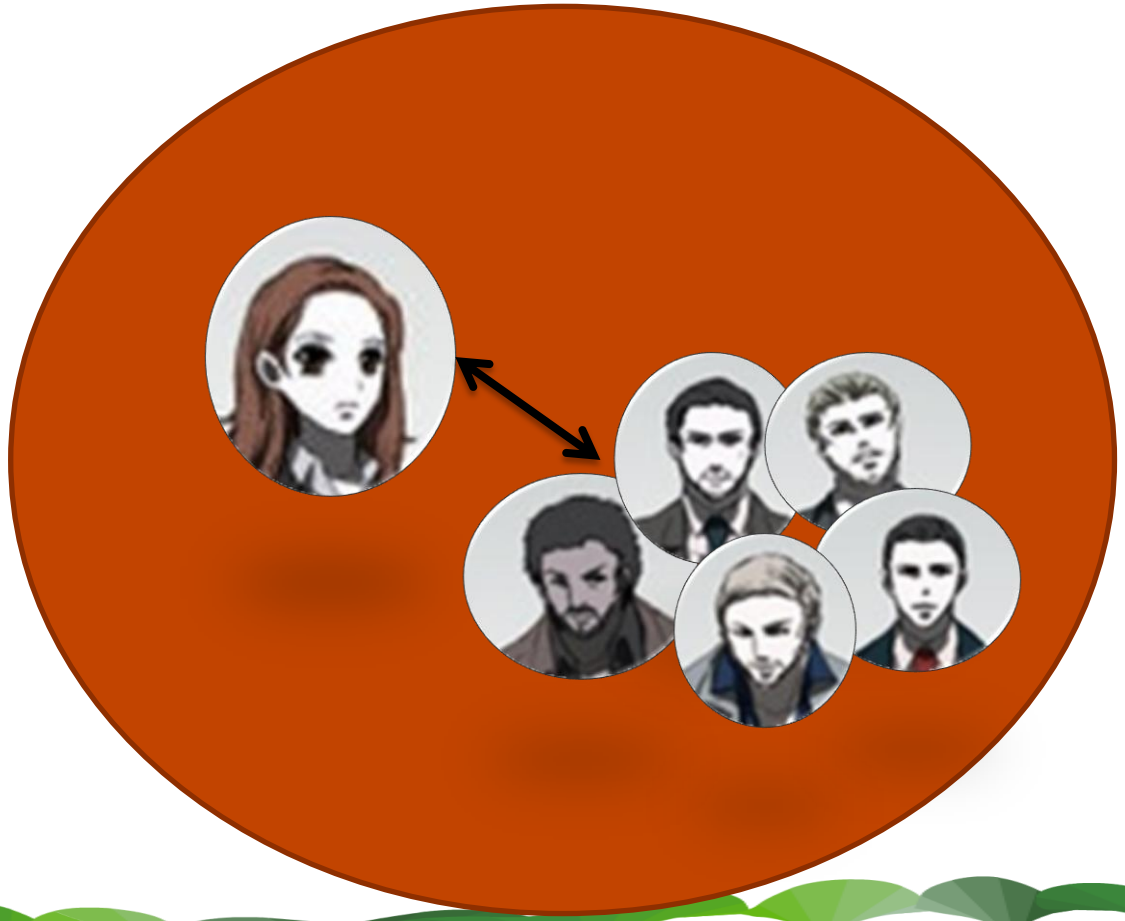
Rôles organisationnels

Scrum Team Roles

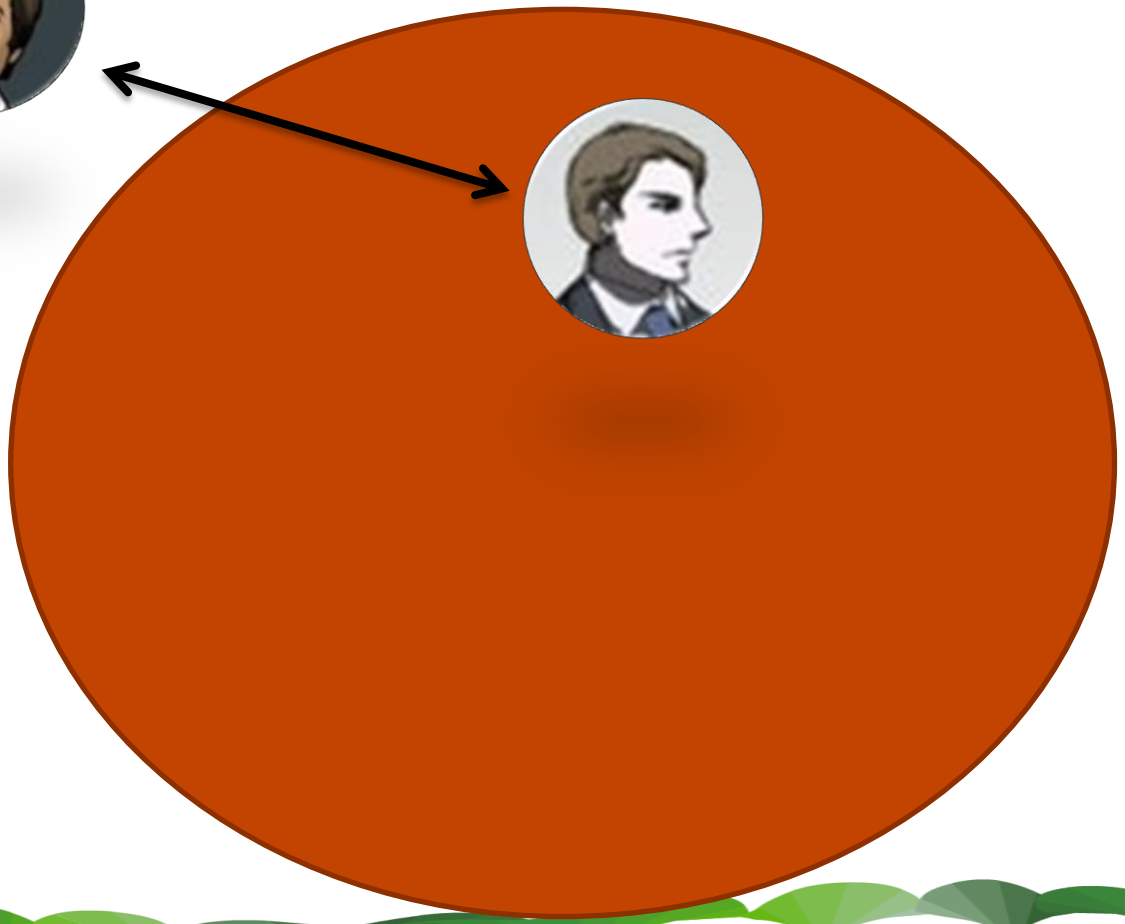
Le ScrumMaster travaille avec le Product Owner



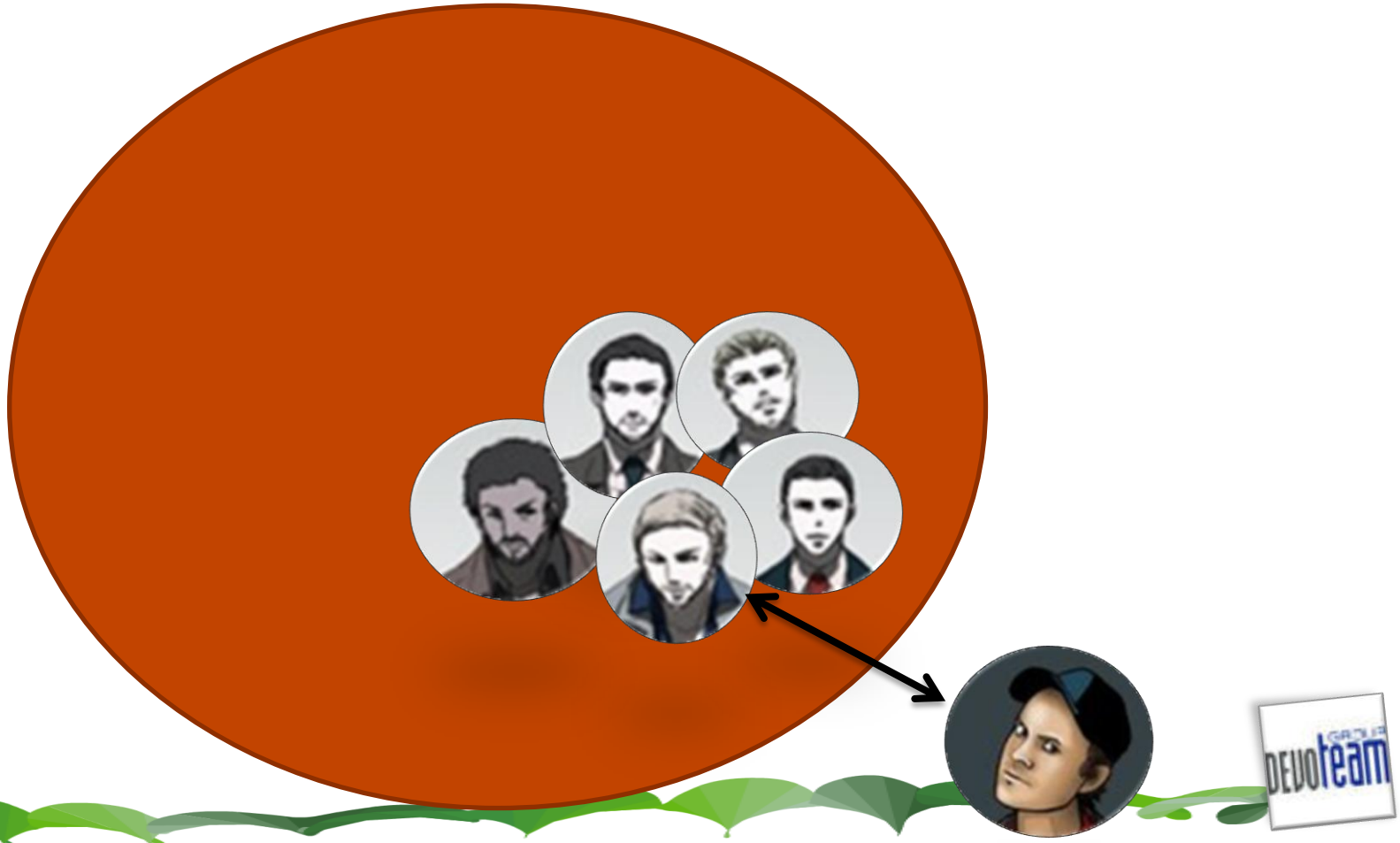
Le ScrumMaster travaille avec la Development Team



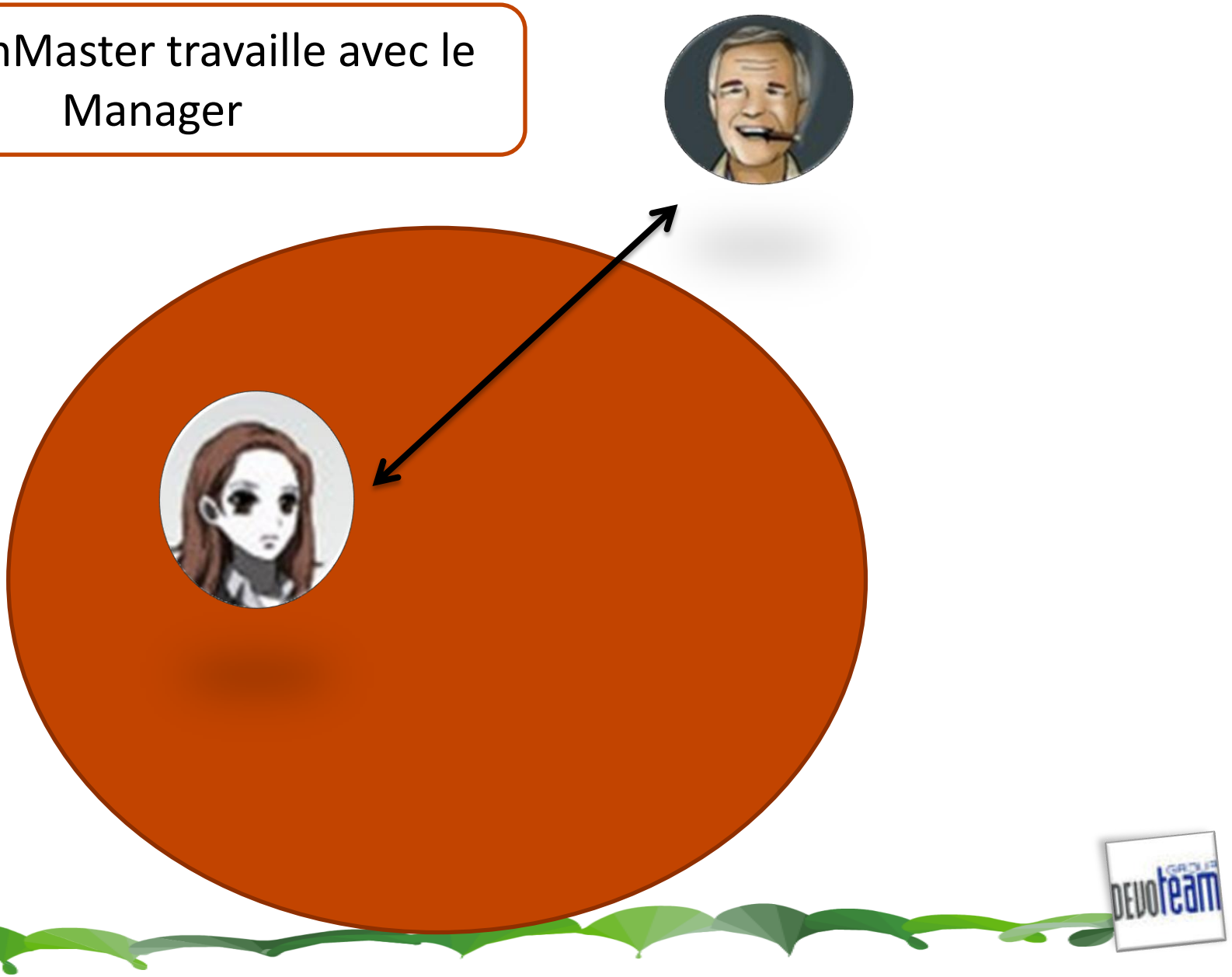
Le Product Owner travaille avec le client



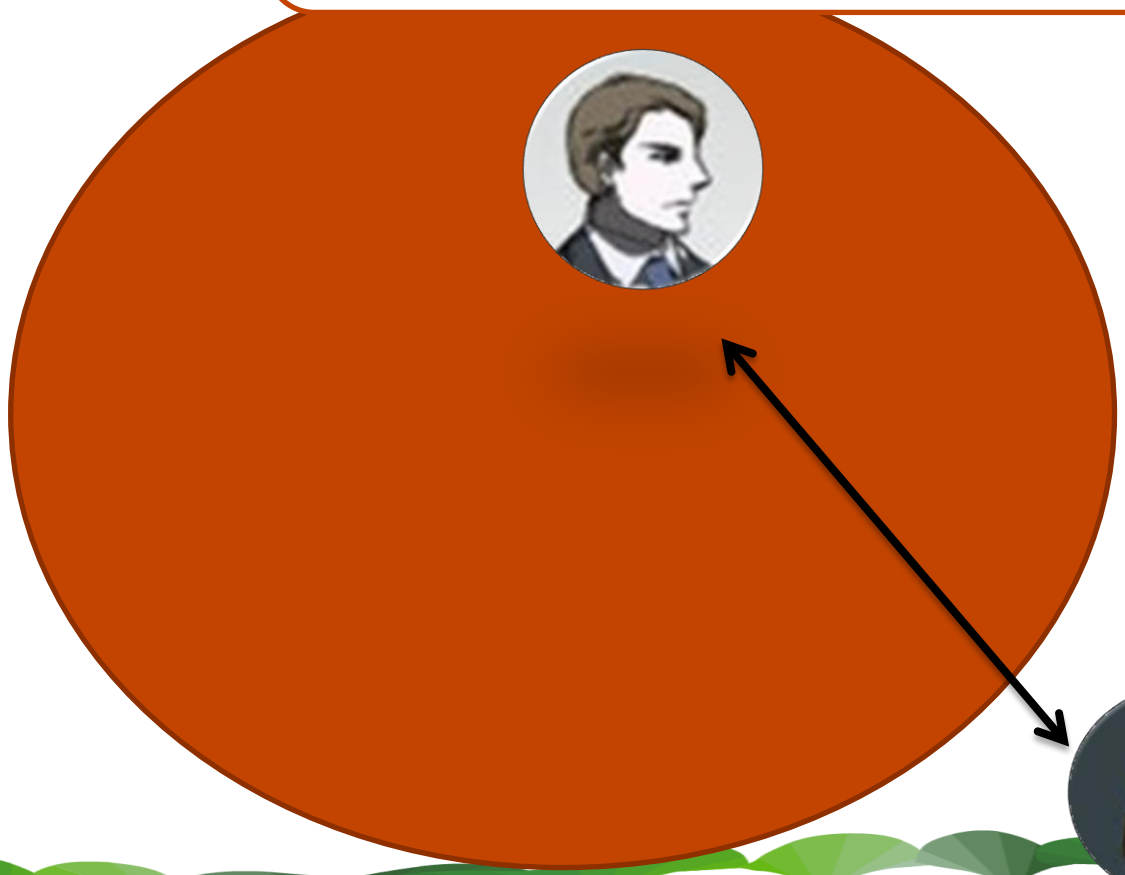
La Development Team travaille avec l'utilisateur final



Le ScrumMaster travaille avec le Manager



Le Product Owner a besoin de connaître ce que le marché (l'utilisateur final) souhaite.



LES ARTEFACTS

4 Artefacts

Product Backlog

Sprint Backlog

Release Burndown

Sprint Burndown

Sécurisation des Informations Personnelles

Modèle

Je voudrais une application bureau que je puisse utilise pour stocker toute mon information confidentielle tells que les numéros de série, les informations Carte de Crédit, les alias d'enregistrement sur les sites web, les mots de passe, etc. pour chaque item que je souhaite stocker, je dois définir le type de données (comme une date d'expiration). Bien entendu, le système devra être protégé par mot de passe et très sécurisé. Je souhaiterai effectuer des sauvegardes/restaurations online de sorte que je puisse récupérer mes informations à distance. Le produit devra posséder des options de recherche, etc....

Source: Mike Cohn, CSPO

User Stories

- **En tant que** *[rôle Utilisateur]*
- **Je veux une** *[FONCTIONNALITE]*
- **De sorte que je reçois** *[BUSINESS VALUE].*

User Story Card

- Une brève description textuelle des exigences
- + Risques
- + critères d'acceptation

AS A *Product Owner*

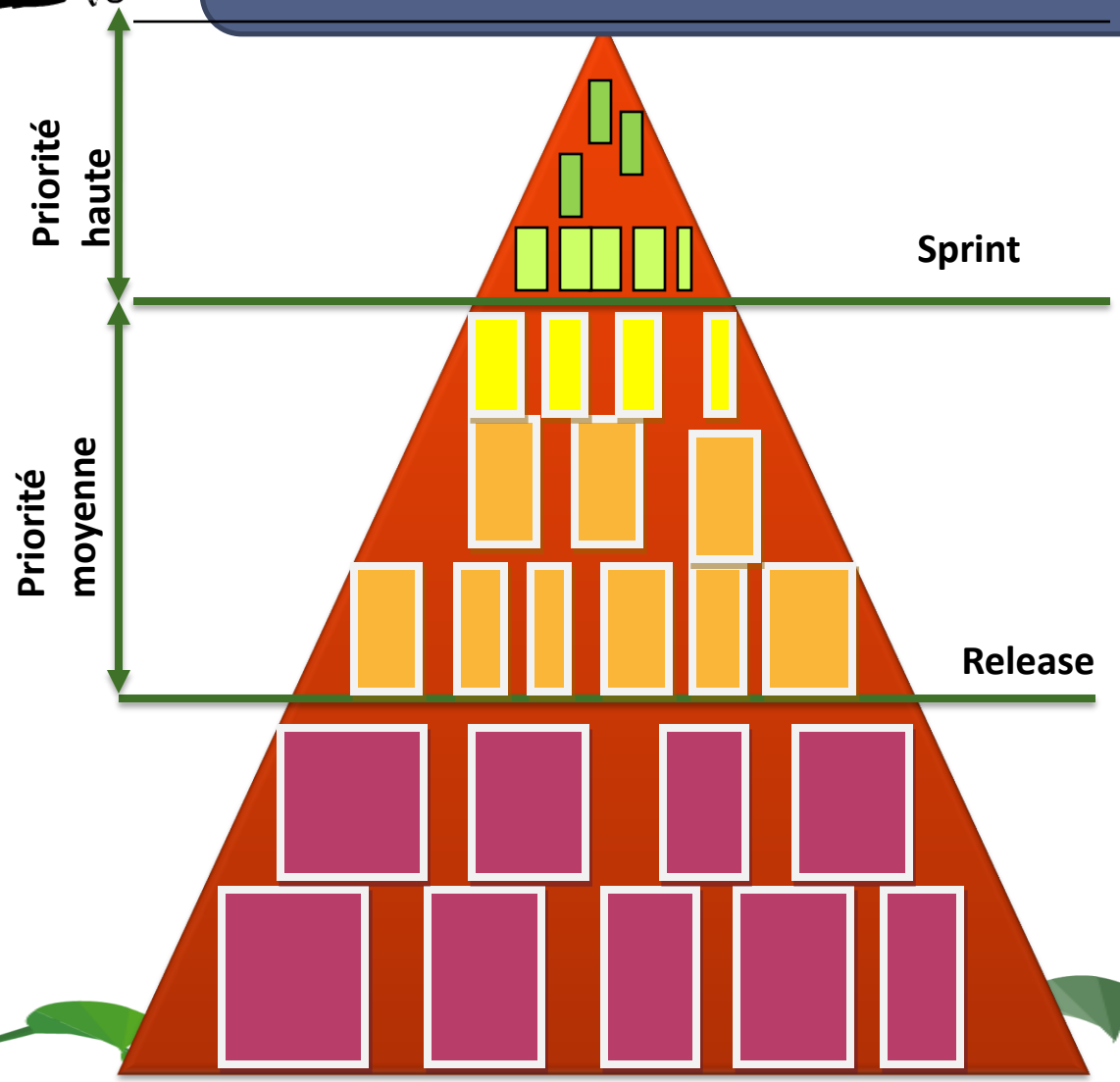
I CAN / I WANT *estimate Costs*

3 lines of Requirement Description

Les bonnes Stories sont INVEST

- **I** indépendant
- **N** négociable
- **V** valorisable
- **E** estimable
- **S** Sized to fit (à la bonne taille)
- **T** testable

Le Product Backlog?



Le Backlog est une liste de tâches ouvertes comme :

–les exigences

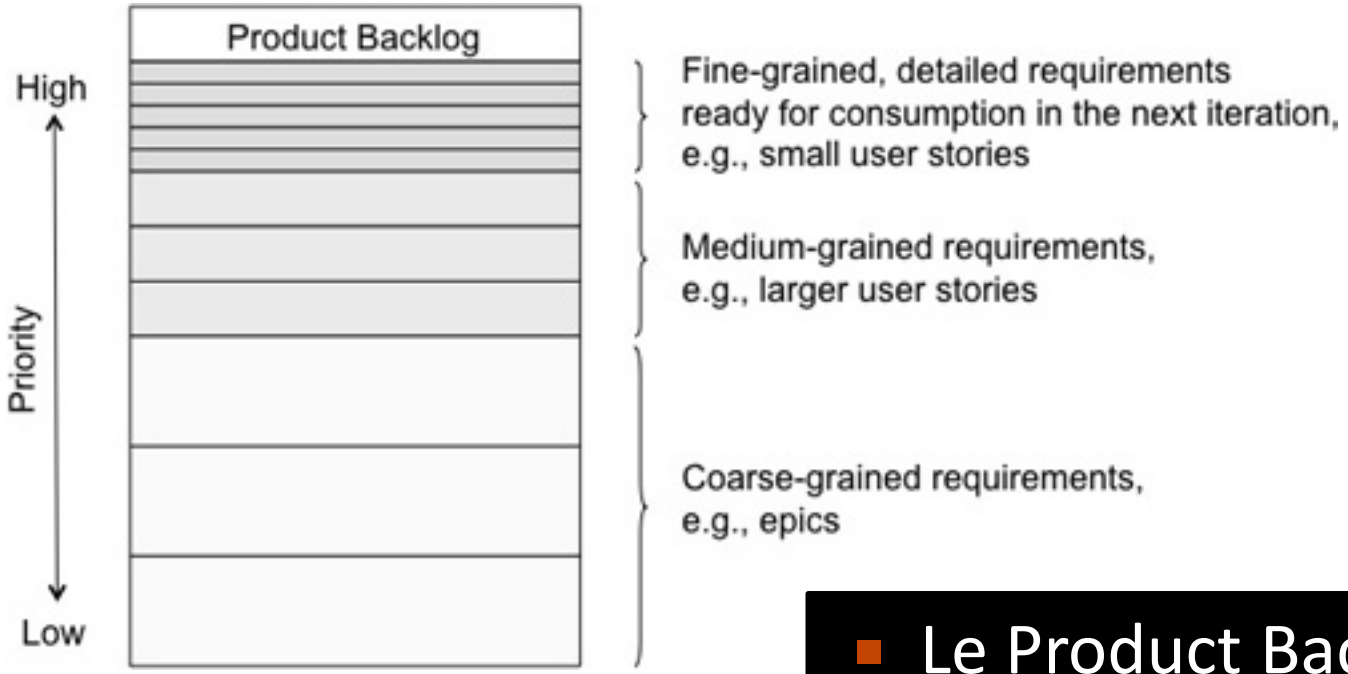
– une liste de tous les travaux souhaités pour le projet

–Idéalement exprimé de telle sorte que chaque objet a une valeur pour les utilisateurs ou les clients du produit

–Priorisé par le Product Owner

–Repriorisé au début de chaque Sprint

Le Product Backlog



- Le Product Backlog répond aux questions suivantes:

  Quoi? Quand? Pour Qui?

Un backlog de produit

Elément de backlog	Estimation
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

Sprint Backlog

Tâches pour transformer le Product Backlog en fonctionnalité-produit

Tâches estimées en heures

- Une tâche ne doit pas nécessiter plus d'un jour ou deux pour être Done.
- Les grandes tâches sont découpées ultérieurement dans le Sprint

Les membres de la Dev. Team s'engagent sur les tâches une fois que le Sprint a démarré

- Les tâches ne sont jamais assignées

Le Sprint Backlog est revu journellement

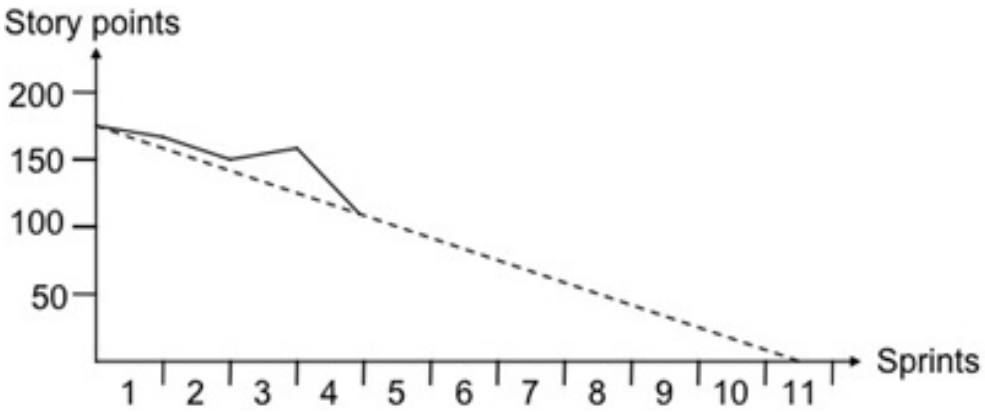
- De nouvelles tâches sont identifiées, d'autres sont modifées ou supprimées.
- Les heures restantes de travail pour chaque tâche sont mises à jour

Les équipes sont mesurées en fonction de leurs engagements

- Et pas sur le temps nécessaire pour les réaliser

Release Burn down Chart

Example de Burndown Chart (Schwaber and Beedle 2002)



Le Release burndown rend les tendances des progrès visibles.

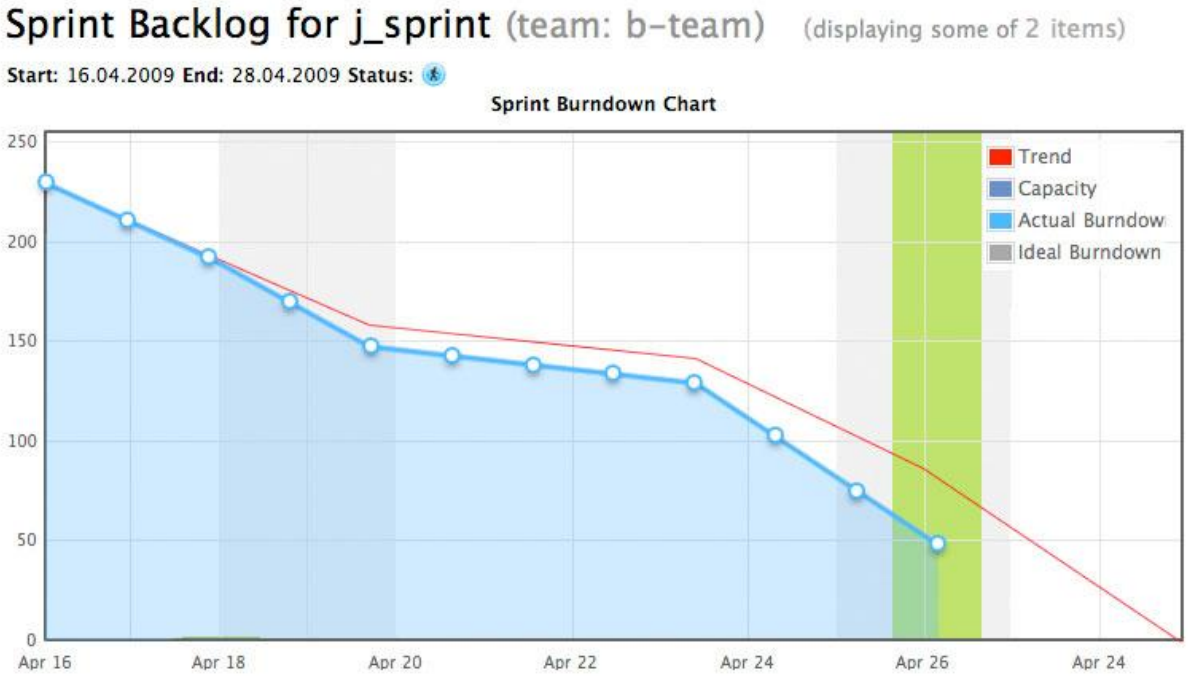
Le rapport est basé sur les informations suivantes:

- *le reste-à-faire du Product Backlog pour transformer la Vision en un produit gagnant.*
- *Le nombre de Sprints nécessaires ou restants.*
- *La vélocité.*

Le Release burndown regarde le passé pour comprendre ce que l'avenir est susceptible de détenir. Nous déterminons le taux d'avancement des sprints passés.

Sprint Burn-down Charts

- Le Sprint Burn-down chart montre
  - *combien d'efforts a été déployé en travaillant sur la tâche contenue dans le Sprint Backlog*
  - *Et compare cela à la depense idéale*



- Le tableau donne une tendance qui indique si l'équipe est susceptible de respecter son engagement (indicateur avancé)

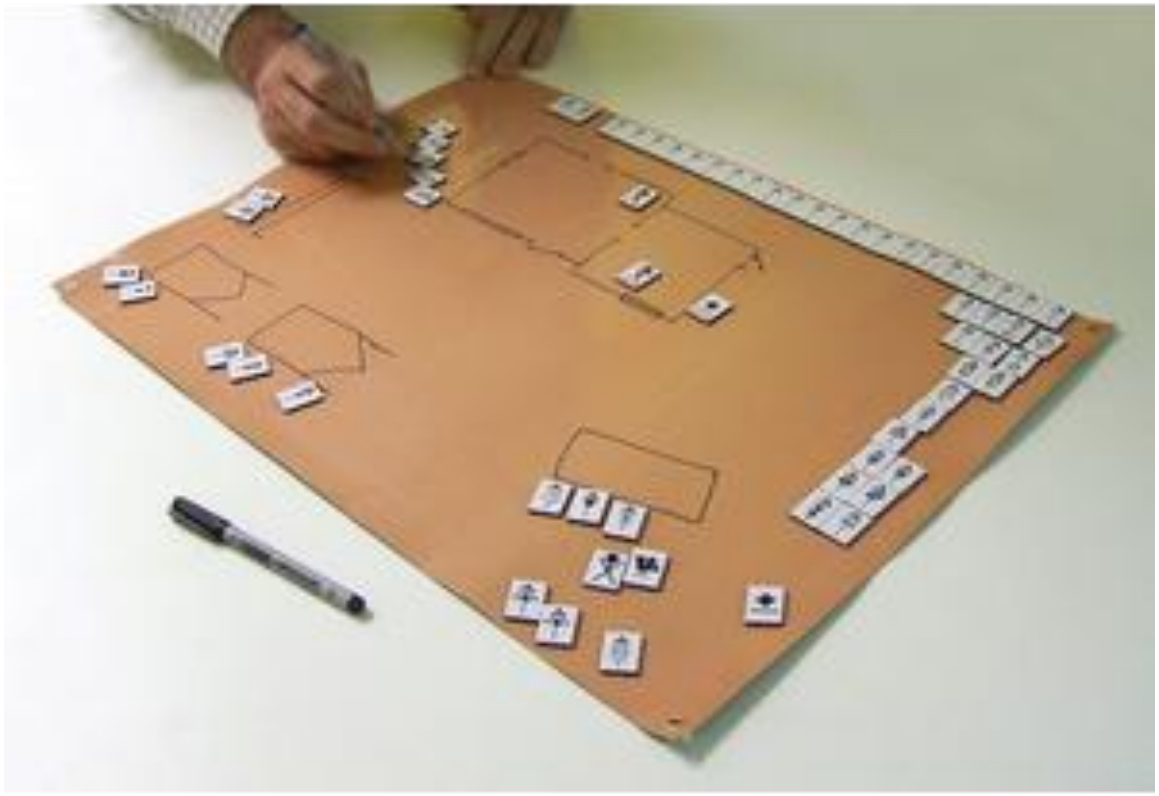
LES TIME-BOXES

Au niveau stratégique

Estimation Meeting

- Préparation du Sprint Planning
- Estimation formelle
- Passez au moins deux réunions par Sprint
- Estimer uniquement sur la taille et le temps

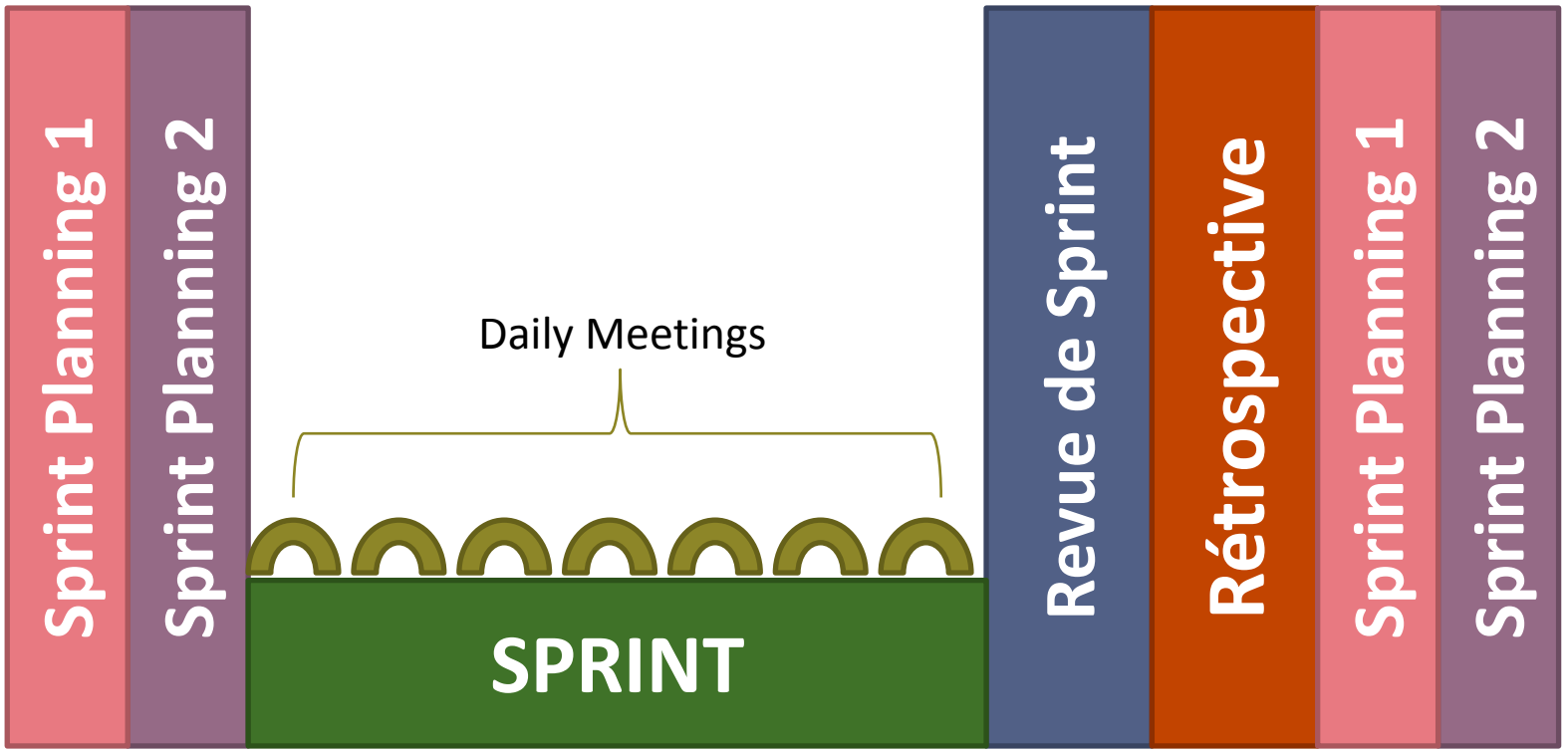
> Input pour Release Planning



Au niveau tactique

Les Meetings

- Le Quoi?
- Le Comment?
- La Synchronisation
- Les Résultats
- L'Amélioration


Sprint Planning 1
Sprint Planning 2
Daily Meetings
SPRINT
Revue de Sprint
Rétrospective
Sprint Planning 1
Sprint Planning 2

Sprint Planning Meeting

- Organisateur: Product Owner
- Participants: l'équipe (actif), le ScrumMaster (passif)
- Durée: 8 heures pour un Sprint de 4 semaines

- **2 PARTIES:**
  - Le QUOI?
  - Le COMMENT?
- **LE PRODUCT OWNER:**
  - *Présente le Product Backlog priorisé par le client et/ou les utilisateurs*
  - *Présente le Release Plan Initial*
  - *Présentation de la Vision*
- **L'ÉQUIPE:**
  - *Estime le Product Backlog en fonction de sa faisabilité (estimation fonctionelle)*
  - *Découpe le Product Backlog en Sprint Backlogs avec le Product Owner*
  - *Découpe le Sprint Backlog en tâches*
  - *Estime le Sprint Backlog*
- **LE PRODUCT OWNER ET L'EQUIPE:**
  - *Définissent l'objectif du Sprint*
  - *Valident la Definition of Done*

Pour chaque Item du Product Backlog (US)

- Quelle interface devons-nous designer?
- Quelle architecture devons-nous créer?
- Quelles tables devons-nous actualiser?
- Quels composants devons-nous mettre à jour ou créer?

Sprint Planning 2

Design

Definition of Done

Level of Done

Pour l'EQUIPE

- Le Code est conforme aux normes
- Le Code est
  - ✓ Propre
  - ✓ Refactoré
  - ✓ Testé unitairement
  - ✓ Validé (checked in)
  - ✓ Intégré (Built)
  - ✓ Dispose d'une suite de test unitaire qui lui est appliquée.
- Pour arriver à cela, l'environnement de développement est constitué :
  - ✓ D'une bibliothèque de code source
  - ✓ De codes standards,
  - ✓ Build automatisé,
  - ✓ D'un environnement pour les tests unitaires.

Definition of Done

Pour SCRUM

- Une Story/Item est “done” lorsque l’équipe a atteint son Level of Done

- Le Sprint/itération est “done” lorsque
  - tous les items sont “done”
  - et que le Sprint atteint son objectif
  - et que les critères d’acceptation sont adressés.

- La Release est “done”
  - ✓ “done” pour l’intégration
  - ✓ “done” pour la production

- Half done is not done

Daily Scrum

- *Synchronisation / Engagement sur les tâches*

Daily Scrum

- Organisateur: l'équipe
- Participants: l'équipe (actif), le ScrumMaster (passif), Product Owner (passif)
- Durée: 15 min

- C'est l'inspect-and-adapt de l'équipe: synchronisation et engagement
- Les 3 questions:
  1. *Qu'est-ce que tu as fait hier?*
  2. *Quels sont les problèmes que tu as rencontrés?*
  3. *Qu'est-ce que tu as prévu aujourd'hui?*

La Revue de Sprint

- Organisateur: Product Owner
- Participants: l'équipe (actif), le ScrumMaster (passif), le Management (actif), le client (actif), les utilisateurs (actifs)
- Durée: 4 heures pour un Sprint de 4 semaines

- C'est l'inspect-and-adapt des utilisateurs, du client et du management
- L'équipe présente les résultats du Sprint
- Utilisateurs/Client/ Management expriment leurs remarques et trouvent un compromis avec l'équipe
- Le Product Owner valide ou rejète les items du Sprint Backlog en fonction de la Definition of Done
- C'est le Product Owner qui a toujours le dernier mot...

Quand un membre de l'équipe dit « DONE", ça veut dire quoi?

- Le code est conforme aux normes, est propre, a été re-factoré, a été testé unitairement, a été vérifiée, a été built, et a eu une suite de tests unitaires qui lui est appliquée.

- Dispose d'un environnement de développement, pour cela il faut une bibliothèque de codes source, des normes de codage, des builts automatisés, et un environnement de tests unitaires

Sprint Review

- Présentation (par l'équipe)
- Feedback (par l'utilisateur final)

- C'est l'inspect-&-adapt de l'utilisateur permettant la création ou le changement des items du Product Backlog

La Rétrospective

- Organisateur: ScrumMaster
- Participants: l'équipe (actif), le ScrumMaster (actif), le Product Owner (actif en sa qualité de membre de l'équipe)
- Durée: 3 heures pour un Sprint de 4 semaines

- Analyse du Process Scrum:
  - *Comment cela c'est passé pendant le Sprint*
  - *Comment s'améliorer*
- Points principaux de vérification:
  - *La communication dans l'équipe*
  - *Les relations entre les membres de l'équipe*
  - *Les process et les outils*
  - *Les besoins en formation*

Rétrospective

- Nous faisons un point après l'action en nous posant deux questions:
  - Qu'est-ce qui a bien fonctionné?
  - Que devons-nous améliorer?

- Objectifs:
  - Apprendre du passé pour préparer l'avenir
  - Améliorer la productivité de l'équipe

Finalité de la Rétrospective

- Debriefing
- Amélioration
- Comprendre la réalité
- Apprendre
- “Input” pour le Sprint Planning

Où allons-nous à partir d’ici?