

Agile Scrum Foundation

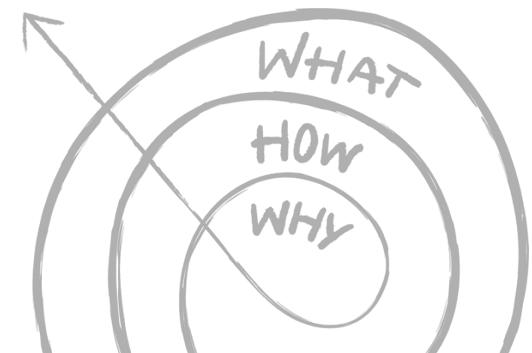
- A part of the management 3.0



Who Are You?



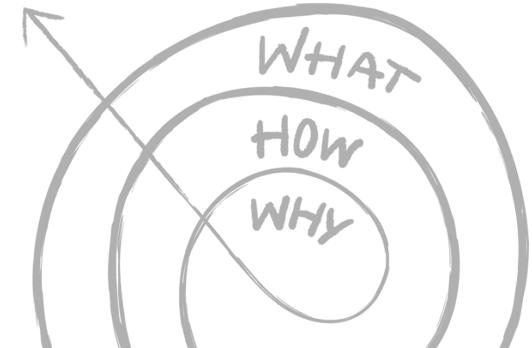
- Quelques mots sur vous
 - Votre rôle
 - Vos connaissances en gestion de projets agiles
 - Vos attentes
- Quelques mots sur votre formateur
 -



Votre formateur : Dominique CADOT



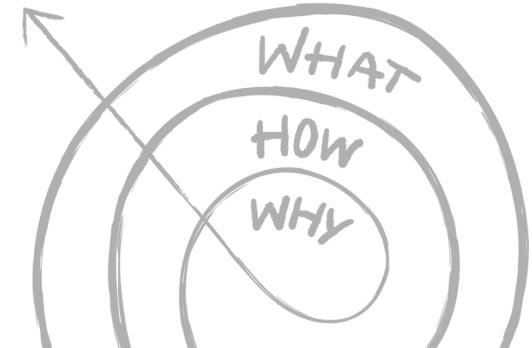
- **Titre** : Consultant Expert
- **Expérience en Agilité** : 2012
- **Expérience en entreprises** : 1995
- **Etudes** : Université de Mons (Psycho-Pédagogie) & **MBA Business Strategy**
- **Certifications** : Agile, Scrum Master, product Owner, Scrum Trainer, Lean SixSigma, ITIL, Prince2, ISO27002,...



Objectifs de ce module

Ce module va vous permettre de :

- Comprendre les bases et origines de l'Agilité et de Scrum.
- Planifier, initier et conduire un projet Scrum.
- Gérer et motiver les équipes Scrum dans la planification des Sprints, les revues et les rétrospectives.
- Identifier, impliquer et faire collaborer les différents intervenants dans un projet Scrum.
- Créer et gérer des stories dans un environnement Agile
- Exploiter les outils
- Comprendre les fondamentaux de l'agilité à l'échelle (Scaled Agile)



Plan de ce module

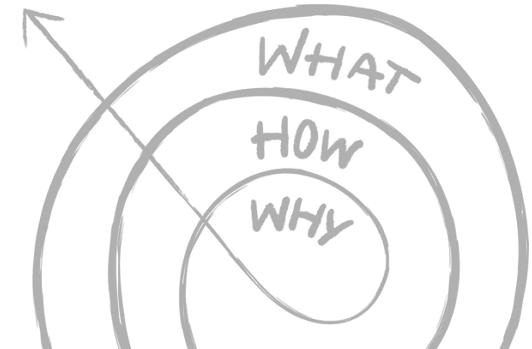
Voici les parties que nous allons aborder :

- **Partie 1 : Introduction & Scrum Basics**

- Les Rôles
- Le Processus
- Le Product Owner
- Le Scrum Master

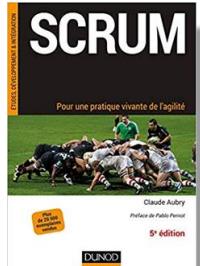
- **Partie 2 : Mise en œuvre**

- Le Produit
- Le Product Backlog
- Les Releases
- Les Sprints
- L'agilité à l'échelle



Introduction

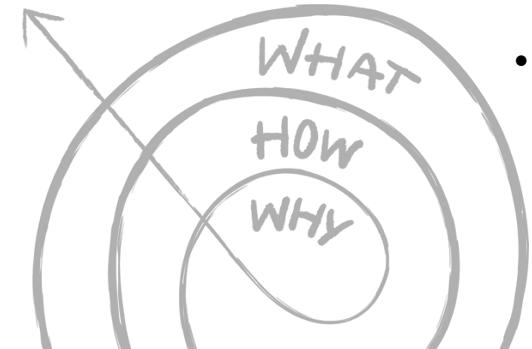
Quelques références



- *SCRUM – Le guide pratique de la méthode agile la plus populaire, Claude Aubry, DUNOD, 4ème édition*



- *The Scrum Primer, <http://www.scrumprimer.org/>*
- *Scrum : une méthode agile pour vos projets, Jean-Paul Subra, 3ème édition*
- *Vidéos Scrum Life : plus de 200 vidéos disponibles gratuitement sur Youtube*



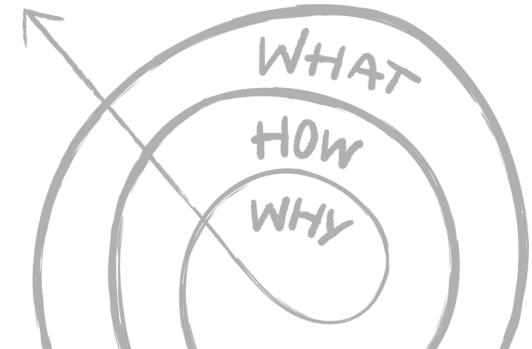
Scrum à l'origine

Scrum a été formalisé en 1995 par :

- **Jeff Sutherland** (Scrumalliance.org)
- <http://jeffsutherland.com/>

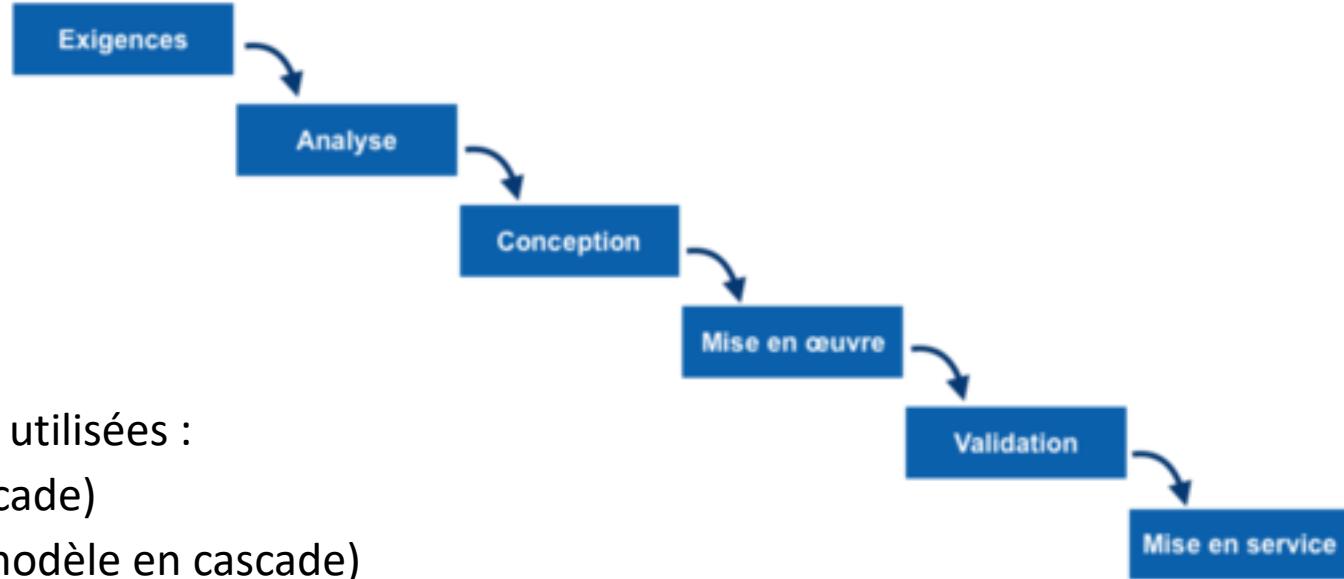


- **Ken Schwaber** (Scrum.org)
- <http://kenschwaber.wordpress.com/>

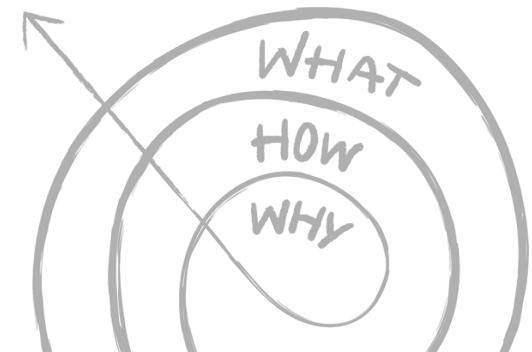


Introduction

Les modèles prédictifs

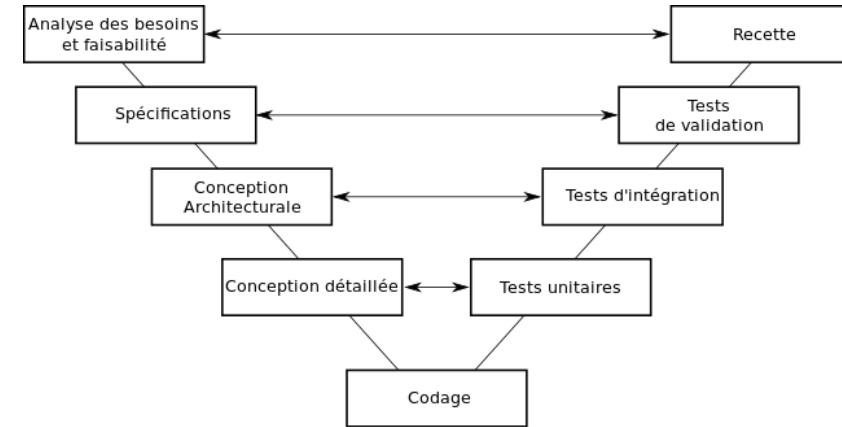


- Deux méthodes fréquemment utilisées :
 - Waterfall (modèle en cascade)
 - En « V » (adaptation du modèle en cascade)
- Ces deux méthodes se caractérisent par un attachement à tout planifier, « tout doit être prévisible », en tout début de projet.
- Un plan de management du projet décrit comment et quand le travail sera réalisé, les modalités de planification, d'exécution, de suivi et de clôture du projet.

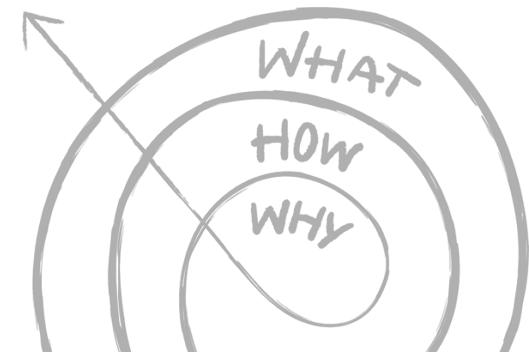


Introduction

Les modèles prédictifs



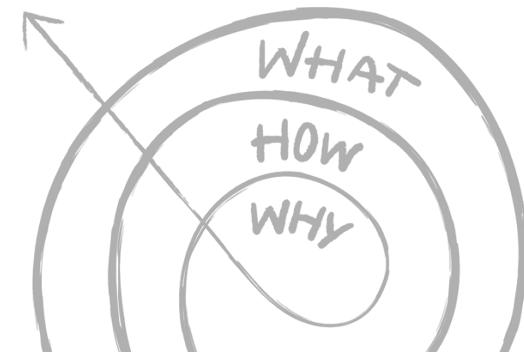
- Cette volonté persistante de vouloir piloter le projet par les plans (*Plan-driven development*) a conduit les acteurs d'un projet à s'opposer systématiquement à tout changement. (contenu, périmètre, processus, équipe,...)
- Un projet ne peut rester figé !
- Les approches prédictives se sont révélées trop « rigides ».



Introduction

Limites des approches classiques : (ex: Waterfall)

- Le cycle en cascade se caractérise par des phases séquentielles, qui se succèdent après la validation des livrables produits lors de la phase précédente;
 - Tous les besoins sont exprimés et recueillis lors de la première phase.
 - La conception du système doit être validée avant le démarrage des développements.
 - Les développements doivent être achevés pour permettre à l'équipe de testeurs de lancer ses campagnes de tests fonctionnels et techniques.
 - Une fois, et seulement une fois, que les anomalies ont été corrigées, on peut procéder à l'intégration globale finale et à la mise en production du système.



Introduction

Empiricism

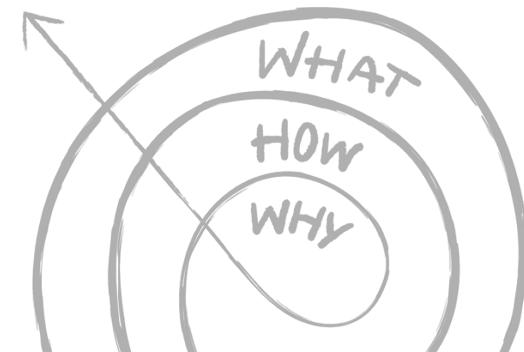
Empiricism: all of our knowledge comes through the use of the five senses.



A belief is a **basic empirical belief** if and only if it is a direct result of sensation.

Activité : Le Marshmallow Challenge

- **Durée de l'exercice :** 18 minutes
- **Objectif :** construire une tour en spaghetti la plus haute possible devant soutenir un marshmallow.
- **Matériel :** 20 Spaghettis – 1 marshmallow – 1 mètre de ficelle – 1 mètre de scotch – 1 paire de ciseaux – 1 minuteur (video)



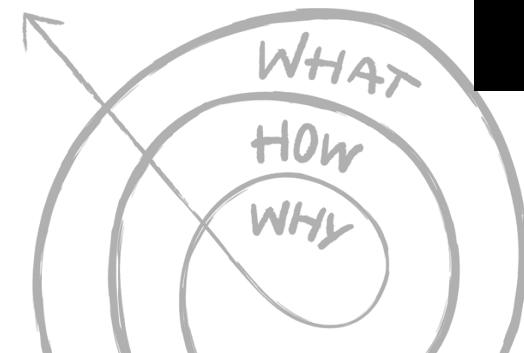
Introduction

Empiricism

Empiricism: all of our knowledge comes through the use of the five senses.

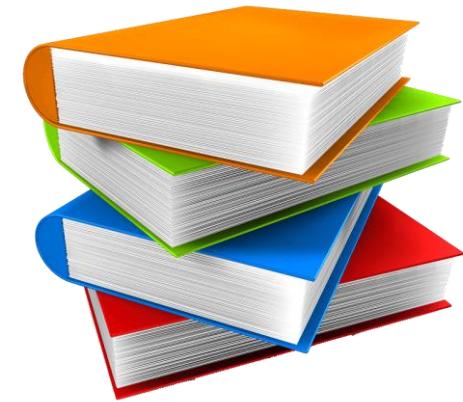


Activité : Le Marshmallow Challenge



Introduction

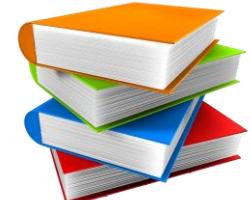
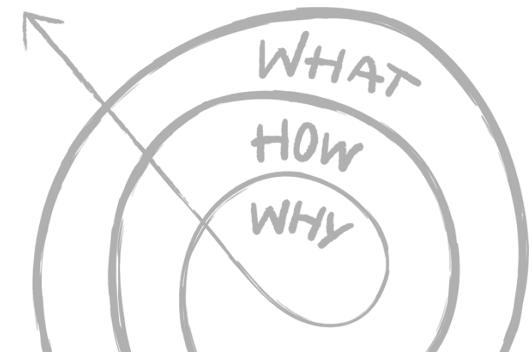
Commencer avec le « WHY » (Simon Sinek – Start With Why)



Introduction

L'Agilité, cadre, méthode ou philosophie de « Management » ?

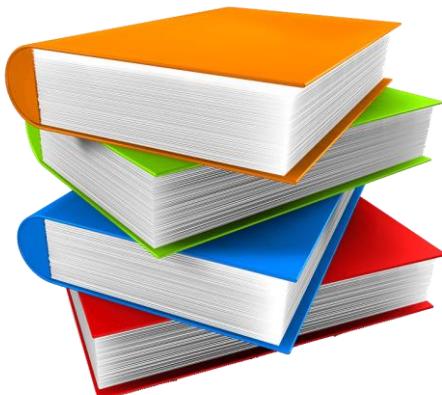
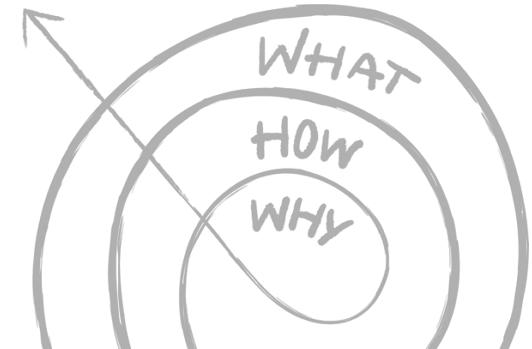
- *Vous voici en formation de « Management »; mais redéfinissons le sens du mot à sa juste valeur. Littéralement, le mot manager traduit de l'anglais donne gestionnaire.*
Il nous est possible d'être le gestionnaire d'un stock de boîte de haricots sans avoir de responsabilité en gestion de ressources humaines. Soit, nous pouvons être le manager de toutes sortes de choses.
- *L'agilité est avant tout un changement de mentalité au sein des entreprises et une réponse aux « nouvelles » générations Y et Z .*
- *Une responsabilisation d'équipe dans une relation basée sur la confiance. Ainsi qu'une remise en question permanente des développements et des processus ainsi que leur amélioration.*



Introduction

Frameworks « Agile » : Scrum en quelques mots

- SCRUM est un framework agile qui produit toutes les 1 à 4 semaines (au maximum) une partie fonctionnelle du produit : **Production d'un incrément potentiellement livrable.**
- Les exigences et les priorités sont définies dans le product backlog sous forme de **user story** et/ou **technical story**, ou encore des **bugs stories**
- L'équipe de développement est **auto-organisée** et **pluridisciplinaire**.
- A chaque fin de sprint, la décision de livrer le produit dans l'état ou de continuer à l'améliorer est sélectionnée sur demande expresse du client (**Release**)



Introduction

L'empirisme de Scrum

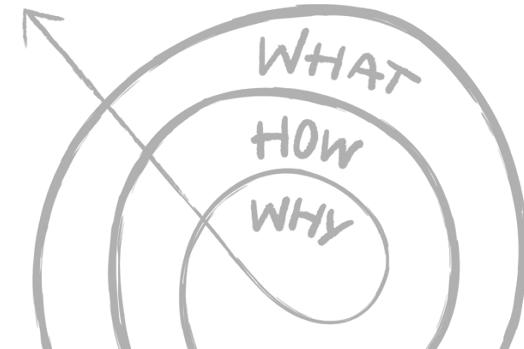
- Scrum est basé sur la théorie du contrôle empirique de processus
- L'empirisme soutient que les connaissances proviennent de l'expérience et d'une prise de décision basée sur des faits connus.
Scrum utilise une approche itérative et incrémentale pour optimiser la prédictibilité et pour contrôler le risque.
- Trois piliers soutiennent l'implémentation d'un contrôle empirique de processus :
 - **La transparence**
 - **L'inspection**
 - **L'adaptation**

Empiricism

Empiricism: all of our knowledge comes through the use of the five senses.



A belief is a **basic empirical belief** if and only if it is a direct result of sensation.



Introduction

Empiricism

Empiricism: all of our knowledge comes through the use of the five senses.



A belief is a **basic empirical belief** if and only if it is a direct result of sensation.

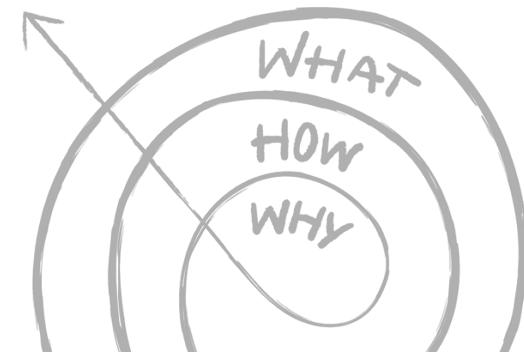
L'empirisme de Scrum

- **La Transparence selon Scrum :**

Les aspects importants du processus doivent être visibles à ceux qui sont responsables des retombées.

La transparence requiert la définition d'un standard commun pour ces aspects afin d'atteindre un niveau de cohésion.

En ce sens, la transparence est en relation notamment avec les critères de la définition du « **DONE** ».



Introduction

Empiricism

Empiricism: all of our knowledge comes through the use of the five senses.



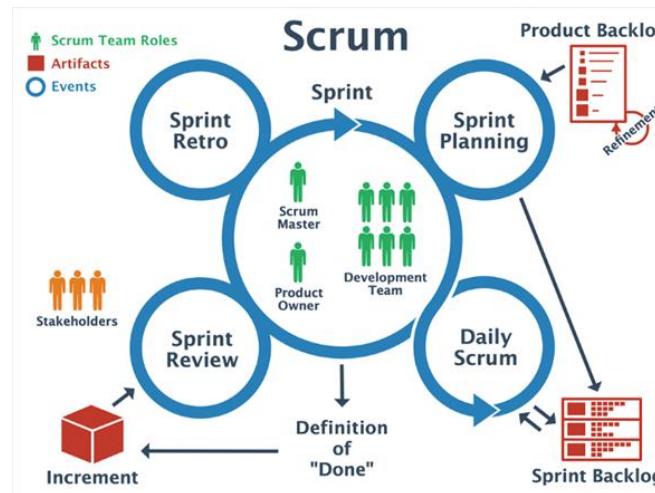
A belief is a **basic empirical belief** if and only if it is a direct result of sensation.

L'empirisme de Scrum

- **L'inspection selon Scrum :**

Les utilisateurs de Scrum doivent fréquemment inspecter les **artéfacts** Scrum et l'état d'avancement par rapport à un objectif de Sprint (**Sprint Goal**) afin de détecter les écarts indésirables.

Ces inspections sont bénéfiques lorsqu'elles sont effectuées de manière diligente sur les lieux du travail par les personnes qualifiées (Dev Team).



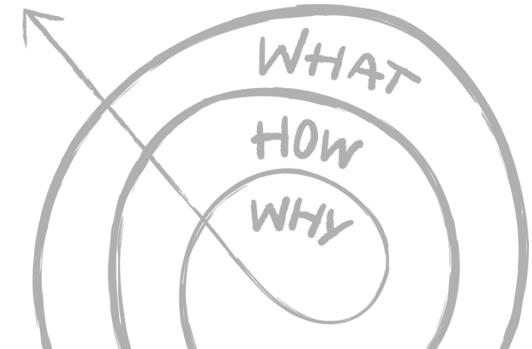
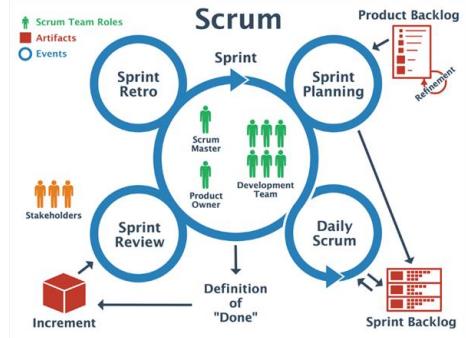
Introduction

L'empirisme de Scrum

- **L'adaptation selon Scrum :**

En cas de dérives hors des limites acceptables, des ajustements doivent être fait dès que possible afin de minimiser le risque d'autres dérives.

- Scrum Précise 4 occasions formelles d'inspection et d'adaptation :
 - ✓ Le Sprint Planning Meeting
 - ✓ Le Daily Scrum
 - ✓ Le Sprint Review
 - ✓ Le Sprint Retrospective



Introduction

Frameworks « Agile » : Dynamic Systems Development Method (DSDM)

- **Principes:**

La méthode DSDM s'appuie sur 9 principes de base :

Implication des utilisateurs durant tout le cycle de développement. Ils sont considérés comme des membres à part entière de l'équipe projet ;

Autonomie. L'équipe projet doit avoir un pouvoir de prise de décision concernant l'évolution des besoins ;

Visibilité du résultat. L'application doit être livrée le plus souvent possible afin de permettre un feed-back rapide. Les délais entre les livraisons doivent être le plus court possible ;

Adéquation. L'objectif est de livrer une application en adéquation avec le besoin métier du client ;

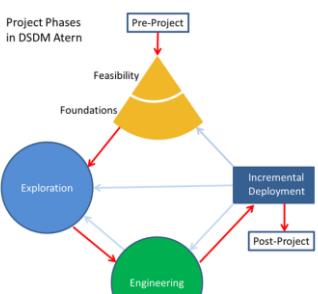
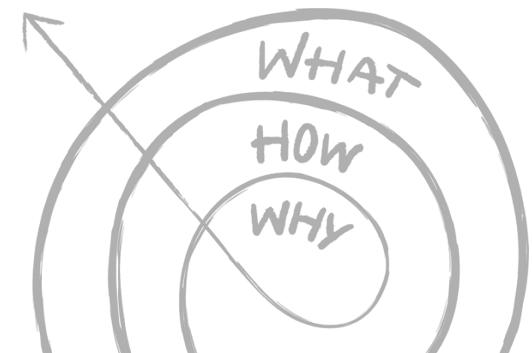
Développement itératif et incrémental. L'évolution du développement est basée sur le feed-back des utilisateurs ;

Réversibilité. Toute modification effectuée durant le développement doit être réversible ;

Synthèse. Un schéma directeur défini de manière préalable fixe les grandes lignes du projet, notamment son périmètre ;

Tests. Les tests sont continus durant tout le développement. Ils permettent de garantir le bon fonctionnement de l'application, à chaque étape du développement ;

Coopération. Les acteurs du projet doivent faire preuve de souplesse concernant les modifications des fonctionnalités demandées.



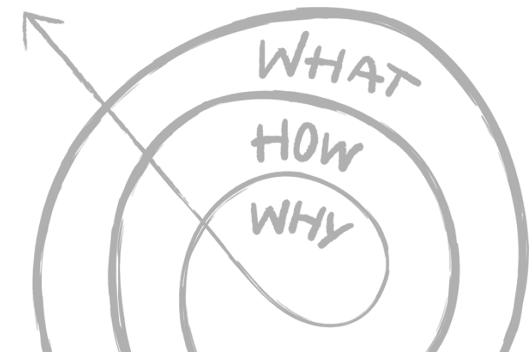
Introduction

Frameworks « Agile » : Extreme Programming

- **Principes:**

La méthode agile XP est issue d'un projet de la société Chrysler déployé au milieu des années 90.

- La première étape de la méthode agile XP consiste à définir les spécifications. Mais plutôt que de spécifier tout le projet dès le début, on va se concentrer sur les fonctionnalités les plus importantes. Cela demande naturellement de l'expérience et une analyse assez fine du besoin.
- Le projet va alors être découpé en modules et sous-modules. Les développements vont avoir lieu lors d'itérations pouvant avoir une durée de deux à cinq semaines maximum. Généralement, trois semaines est un bon choix. Une livraison est effectuée à l'issue de chaque itération et elle est testée. Si les tests sont concluants, on passe à l'itération suivante, sinon, l'itération recommence.
- Le suivi du projet se fait au fur et à mesure de l'avancement des itérations, ce qui permet de rapidement mettre en évidence les problèmes éventuels et de pouvoir prendre en compte de nouvelles demandes du client.
- XP est également axé sur le TDD ou Test Driven Developement comme expliqué ci-dessus.



Introduction

Frameworks « Agile » : Extreme Programming

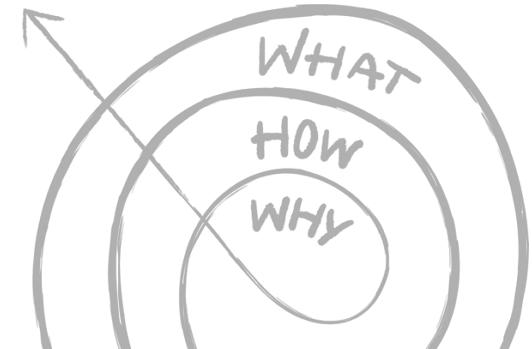
- **Principes:**

Le pair programming ou programmation en binôme.

La programmation en binôme (de l'anglais pair programming), parfois appelée programmation par pairs ou binômage, est une méthode de travail dans laquelle deux développeurs travaillent ensemble sur un même poste de travail.

La personne qui rédige le code est appelée conducteur (driver). La seconde personne, appelée observateur (observer), assiste le conducteur en décelant les imperfections, en vérifiant que le code implémente correctement le design et en suggérant des alternatives de développement. Les rôles s'échangent régulièrement pendant la séance de programmation1.

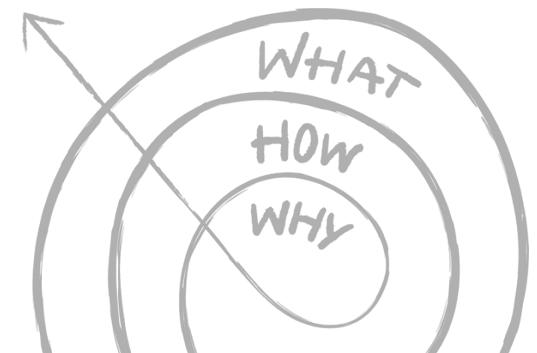
Cette pratique permet d'augmenter le niveau de maturité en compétence dans une équipe. Le principe est emprunté à l'approche du compagnonnage pédagogique.



Introduction

Le Manifeste Agile : 4 Valeurs

- Publié au début des années 2000 et inchangé depuis, il définit une attitude de réaction par rapport à des processus lourds et bureaucratiques.
- Valeurs relatives de l'Agilité :
 - **Les personnes et leurs interactions sont plus importantes que les processus et les outils.**
 - **Un logiciel qui fonctionne prime sur la documentation**
 - **La collaboration avec les clients est préférable à la négociation contractuelle**
 - **La réponse au changement passe avant le suivis d'un plan**



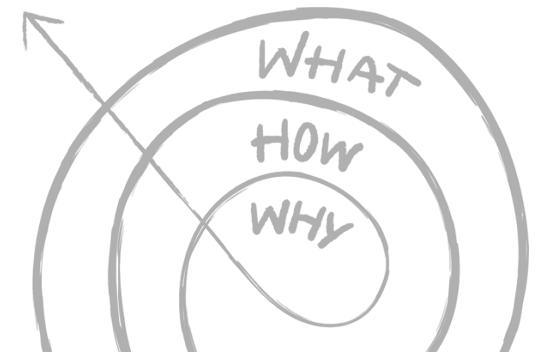
Attention !

Dans une transformation Agile, il sera essentiel que cette transition soit supportée par le Senior Management et Top Management

Introduction

Le Manifeste Agile : 12 principes

- **Satisfaire le client en livrant tôt et régulièrement des logiciels utiles, qui offrent une véritable valeur ajoutée.**
- **Accepter les changements, même tard dans le développement.**
- **Livrer fréquemment une application qui fonctionne.**
- **Collaborer quotidiennement entre clients et développeurs.**
- **Bâtir le projet autour de personnes motivées en leur fournissant environnement et support, et en leur faisant confiance.**
- **Communiquer par des conversations en face à face.**



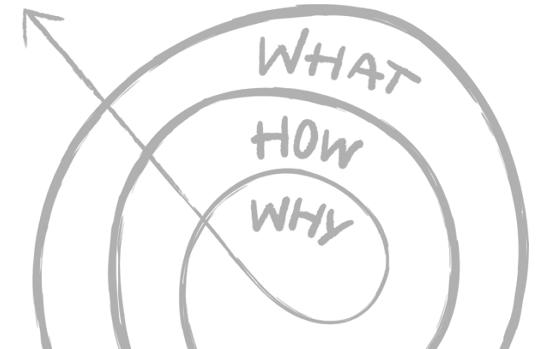
Attention !

Dans une transformation Agile, il sera essentiel que cette transition soit supportée par le Senior Management et Top Management

Introduction

Le Manifeste Agile : 12 principes

- Mesurer la progression avec le logiciel qui fonctionne.
- Garder un rythme de travail durable.
- Rechercher l'excellence technique et la qualité de la conception.
- Laisser l'équipe s'auto-organiser.
- Rechercher la simplicité.
- A intervalles réguliers, réfléchir au moyens de devenir plus efficace.

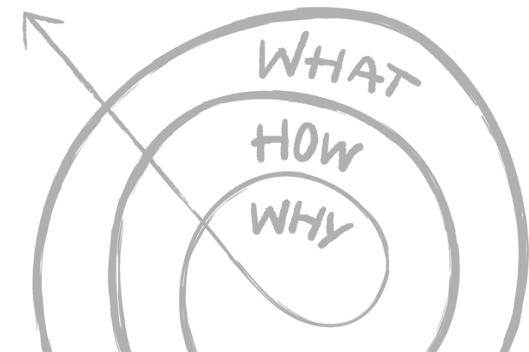


Attention !
Dans une transformation Agile, il sera essentiel que cette transition soit supportée par le Senior Management et Top Management

Introduction

Les 5 valeurs de Scrum

- Commitment (engagement dans ce qui a été convenu).
- Courage (environnement où l'échec est une opportunité d'apprentissage).
- Focus (objectif dans lequel chaque membre de l'équipe se reconnaît).
- Openness (ouverture au changement).
- Respect (respecter dans l'équipe et dans les engagements pris)



Introduction

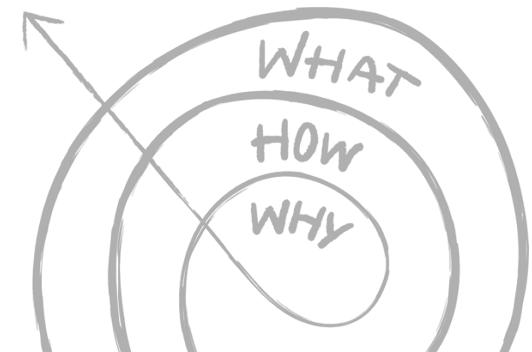
Concept de communication osmotique

Alistair Cockburn, l'un des initiateurs du principe du développement de logiciel "agile" a inventé la notion de communication osmotique pour identifier un type de communication intrinsèque à l'open space d'une équipe. La communication osmotique désigne une communication "implicite" entre les membres d'un espace.

Elle est exprimée de façon dite "ambiante", c'est à dire sans écoute active de la part des individus. Seules les informations essentielles sont donc retenues. Ainsi lorsqu'une personne pose une question, les personnes présentent dans la pièce peuvent y répondre tout en poursuivant leur travail.

Ce type de communication est permis par la colocation radicale vu que la relation de proximité entre les collaborateurs a un impact direct sur la productivité d'une équipe.

Le regroupement des collaborateurs en « war-room » ; terme agile qui désigne un espace dans lequel évolue une équipe centrale, elle permet un accès plus facile aux autres afin de mieux apprendre et collaborer ensemble. Les réponses aux problématiques quotidiennes sont facilement partagées et peuvent s'effectuer de manière synchrone.



Introduction

Mindset & Craftmanship



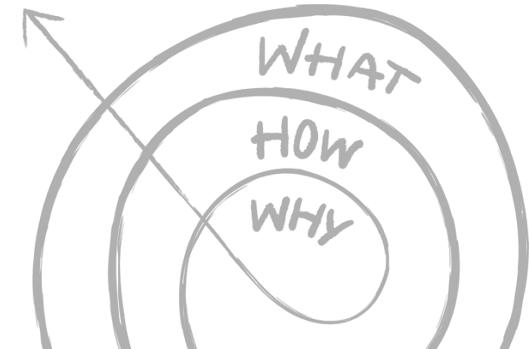
Introduction

Informations Radiators ou diffuseurs d'informations

Une War-room rassemble aussi en un seul lieu, accessible et visible de tous, les informations clés du projets (Risques, performance de l'équipe, Qualité, matrice de compétences etc.) Toutes ces informations sont qualifiées « D'Information Radiator » ou diffuseur d'information.

Parmi ces informations nous retrouvons :

- Un Task Board type Kanban
- Burndown ou Burnup chart
- Liste des impediments ou obstacles
- Métriques de qualités, de risques
- Les dates de release ou livraison



Plan de ce module

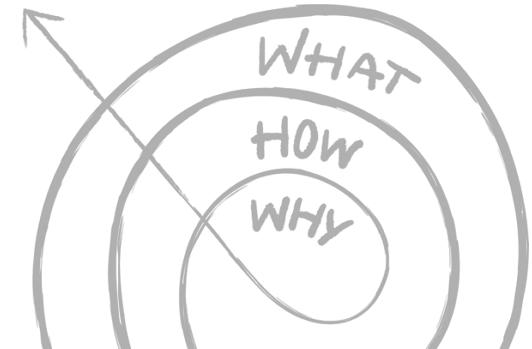
Voici les parties que nous allons aborder :

- **Partie 1 : Introduction & Scrum Basics**

- **Les Rôles**
- Le Processus
- Le Product Owner
- Le Scrum Master

- **Partie 2 : Mise en œuvre**

- Le Produit
- Le Product Backlog
- Les Releases
- Les Sprints
- L'agilité à l'échelle

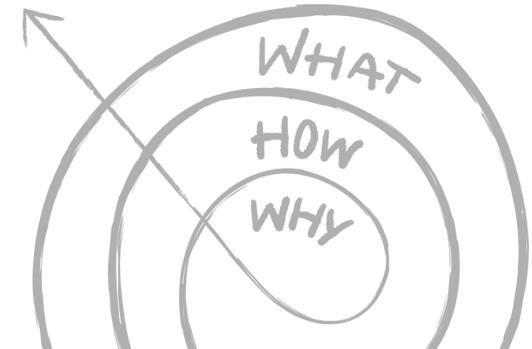


La Scrum Team / Team



Scrum = Mêlée du Rugby = Effort partagé

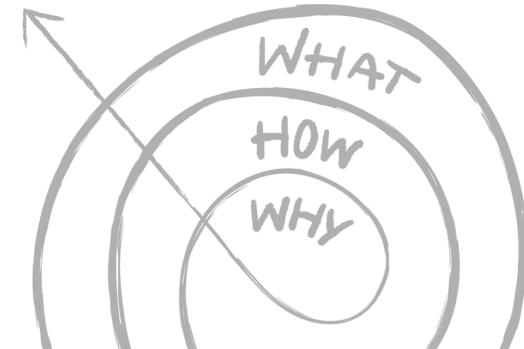
**Le terme « SCRUM TEAM » réuni les 3 rôles Scrum :
Scrum Master – Scrum Product Owner et Développeurs**



Scrum Basics

Le Scrum Master en 2 minutes

- Il est responsable de la bonne utilisation des pratiques, valeurs et règles de la méthode Scrum. Il est sans conteste l'expert Scrum. En ce sens, il est le manager du **framework**.
- En occupation, le rôle de Scrum Master correspond à un **1/3 temps** pour une équipe.
- Il est le coach de l'avancement du développement. Aussi, il s'engage à supprimer tous les freins à l'avancement du projet. Il peut aussi agir en tant que « **secrétaire** » du projet en créant des rapports d'avancement afin de tenir le client et/ou le management informé.
- **Le Scrum Master possède 2 grands axes de responsabilités**
 - Veiller au bon respect des règles, rôles, cérémonies et artefacts Scrum.
 - Faciliter le travail de la Dev Team en proposant des solutions aux problématiques pour lesquelles l'équipe n'a pas trouvé de solutions par elle-même.



Scrum Basics

Le Product Owner en 2 minutes

Il recueille les besoins du client et crée la vision du produit.

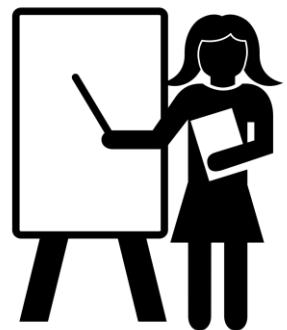
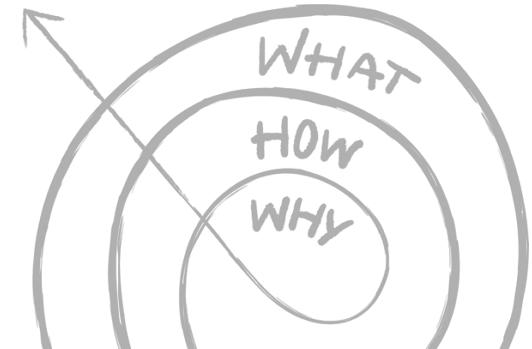
Il est responsable du produit ainsi que de la gestion, du contrôle et de la visibilité du Product Backlog.

Il crée les items du product backlog (Epics, Features, Story) et priorise celui-ci en fonction de différents critères (valeur business, risques, compétences, ...)

Il s'assure de la valeur fournie au client.

Il est sélectionné par :

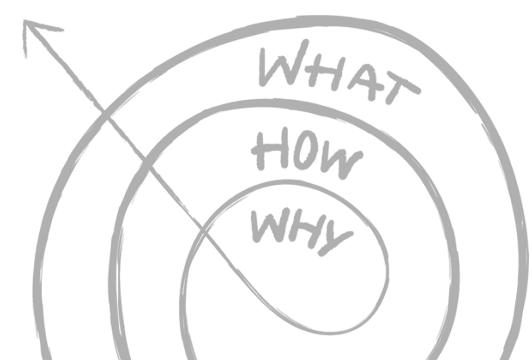
- Le management (MOE)
- Le client (MOA)



Scrum Basics

La Development Team en 2 minutes

- Elle est responsable du déroulement de chaque Sprint, c.-à-d., ce qui est mis en œuvre pour atteindre les objectifs du Sprint. (auto organisée et pluridisciplinaire).
- Au-delà du travail effectif, elle sera aussi responsable de la *définition du DONE* ainsi que du *Sprint Backlog*.
- Selon Scrum.org, la taille idéale pour la Team est de 3 à 9 membres (6 ± 3)
- **Elle est Impliquée dans:**
 - ❖ l'estimation de la charge de travail (valeur d'effort)
 - ❖ la définition du Sprint Goal
 - ❖ la création du Sprint Backlog
 - ❖ La création et la mise à jour du DOD (Definition Of Done)
 - ❖ La création du DOR (Definition Of Ready)
 - ❖ l'identification des freins à l'avancement du projet
 - ❖ le développement et le suivis des « charts»

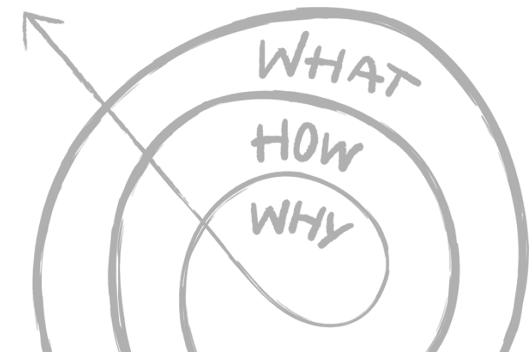


Scrum Basics

Le management et le client en 2 minutes



- **Le Management :**
 - Est représenté par la « MOE » (Maitrise d'œuvre)
 - Dans Scrum, le chef de projet n'existe pas et remplacé par le terme « management »
 - Le management est impliqué dans le choix du Scrum Master, du Product Owner et de l'équipe de développement pluridisciplinaire et auto-organisée. Le Management pourra gérer le calendrier des compétences afin de planifier les besoins en compétences flottantes.
- **Le Client:**
 - ❖ Est impliqué dans le choix du Product Owner et correspond au « **MOA** » (Maitrise d'ouvrage)
 - ❖ Fourni l'expression des besoins au Product Owner
 - ❖ Fourni ses critères d'acceptances et exigences pour « **la définition du Done** »
 - ❖ Est invité à chaque Sprint Review afin de fournir du feedback à la Dev Team
 - ❖ Communique régulièrement avec les membres de la Scrum Team



Plan de ce module

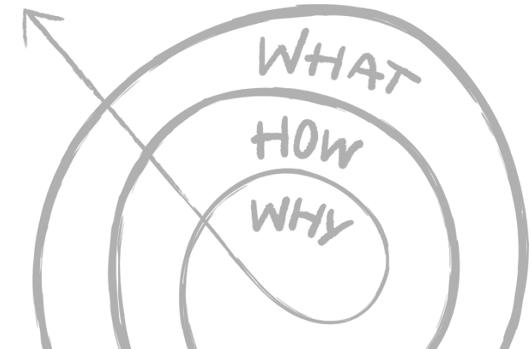
Voici les parties que nous allons aborder :

- **Partie 1 : Introduction & Scrum Basics**

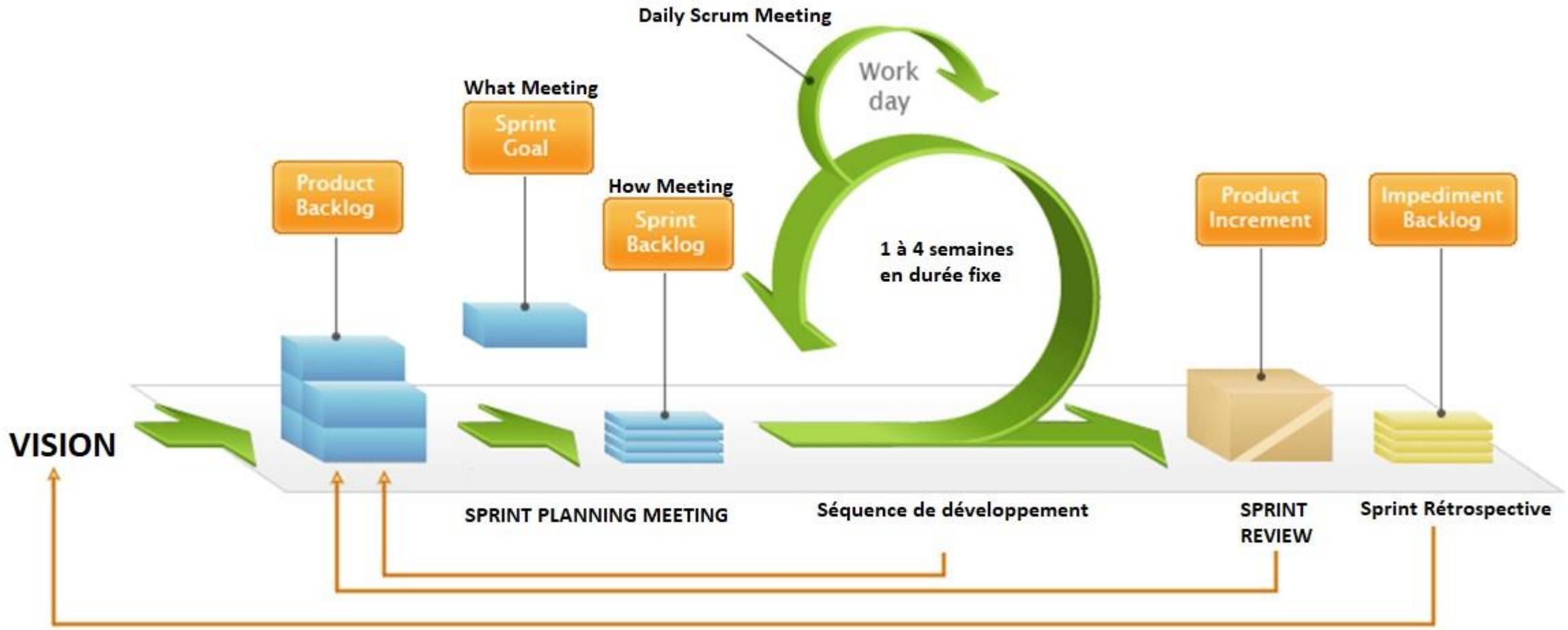
- Les Rôles
- **Le Processus**
- Le Product Owner
- Le Scrum Master

- **Partie 2 : Mise en œuvre**

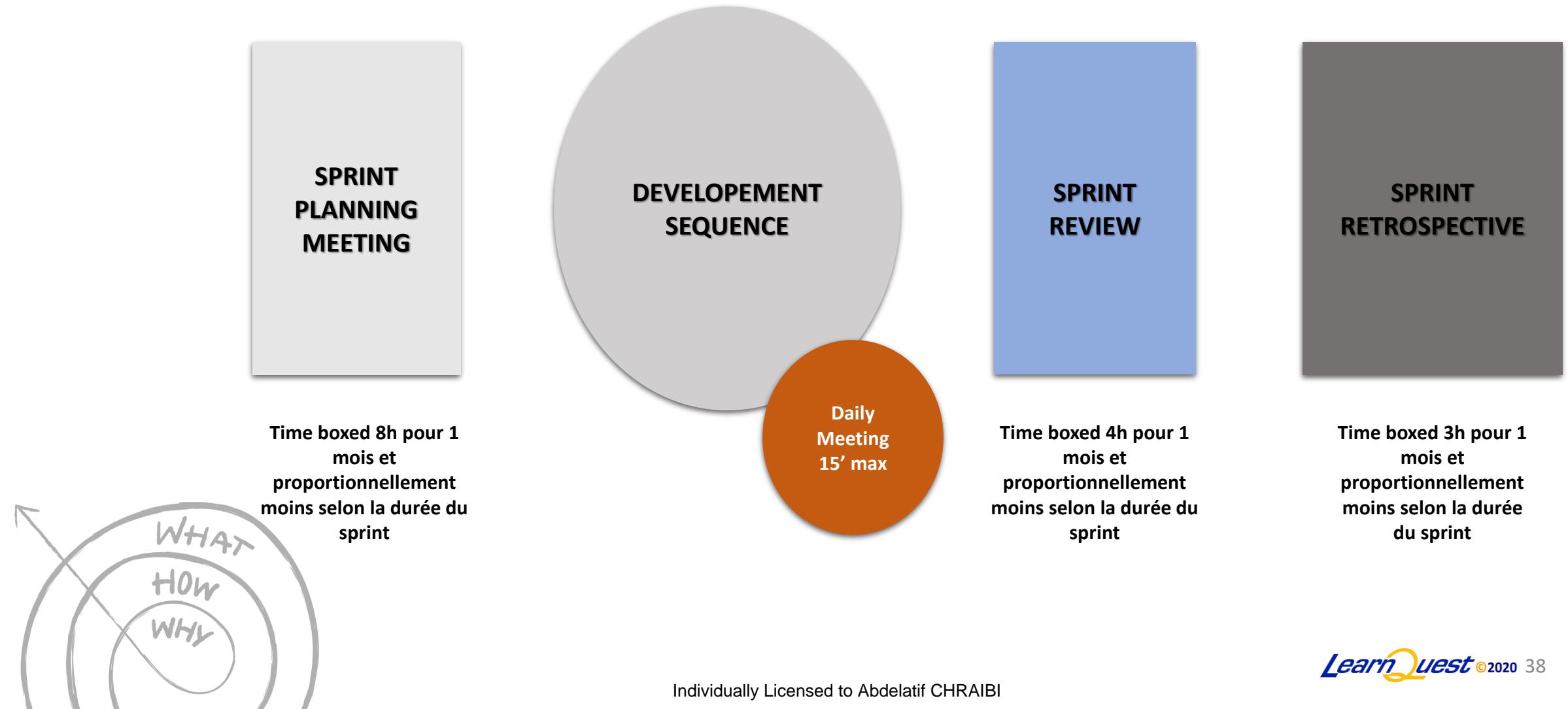
- Le Produit
- Le Product Backlog
- Les Releases
- Les Sprints
- L'agilité à l'échelle



Le Sprint en question... Les événements



Le Sprint en question... Les événements



Scrum Basics

Processus Scrum et timeboxing

- **Un processus Scrum consiste en 3 phases**



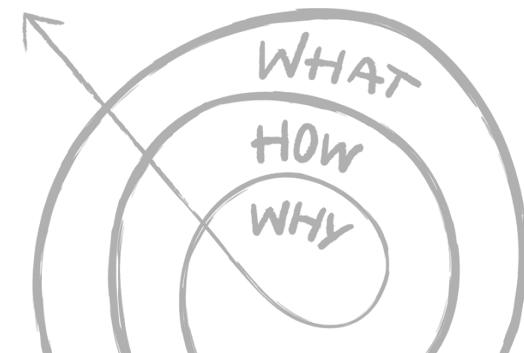
Scrum Basics

La phase initiale

- Cette phase aboutit à :
 - Définition des besoins chez le client
 - La conceptualisation et l'analyse du système
 - La mise en place d'un "**Product Backlog**", c'est à dire une liste de fonctionnalités restant à effectuer
 - La définition *approximative* de la **date de livraison**
 - La définition des **fonctionnalités** à livrer
 - La formation de la Scrum Team
 - L'analyse du **risque** et des **coûts**
- Il s'agit d'une phase **courte** mais qui requiert des compétences variées :
 - La connaissance du marché, des utilisateurs potentiels, recueil des besoins, aspects marketing, gestion des leçons apprises...

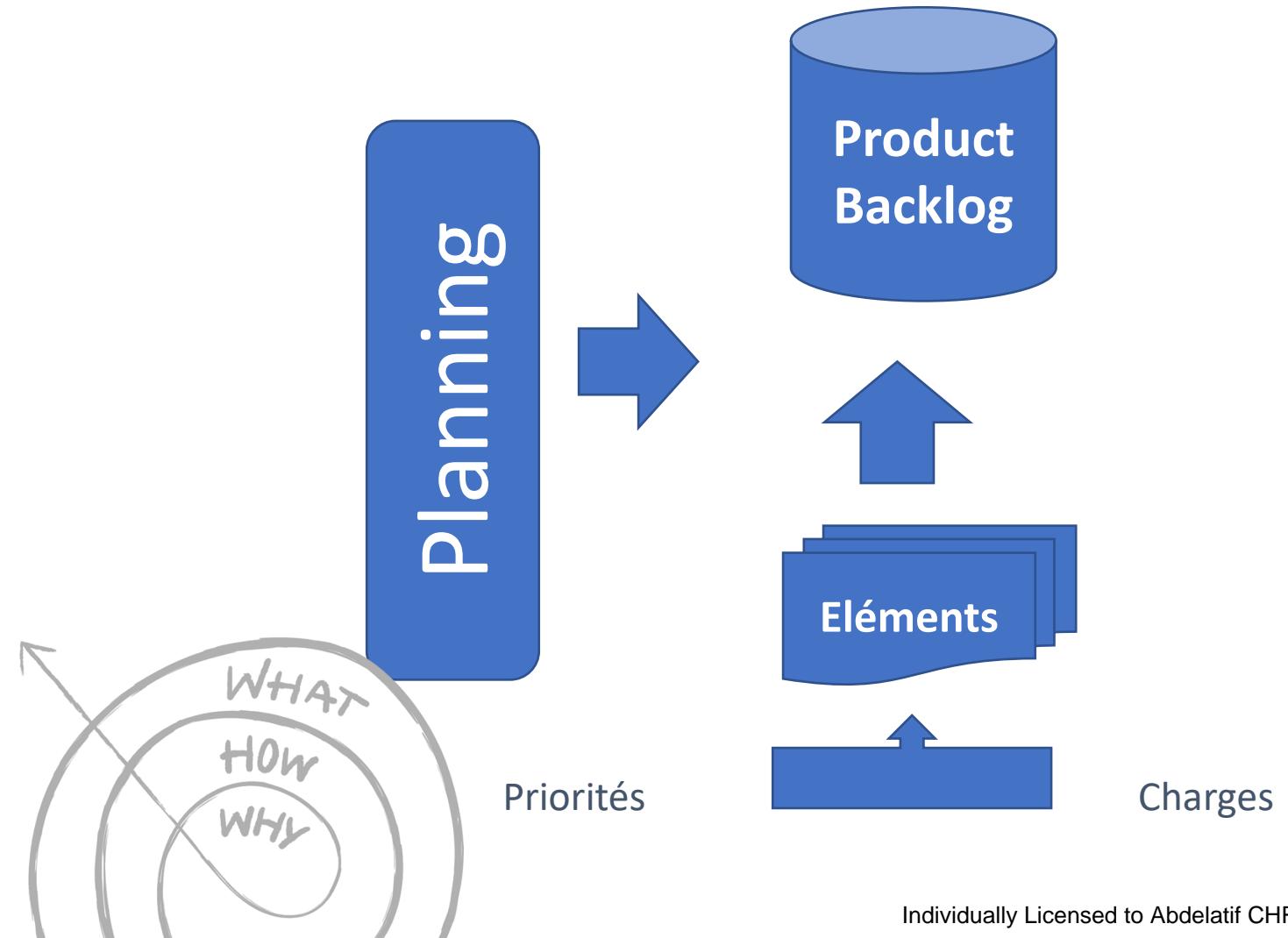


INITIALISATION DU
PROJET



Scrum Basics

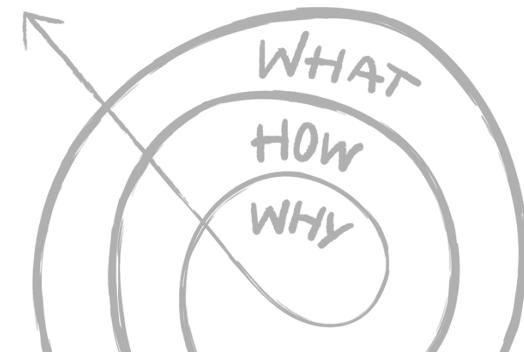
La Phase initiale



Scrum Basics

La phase de Sprint

- Il s'agit de la phase où le développement est à proprement parlé réalisé (analyse, conception, test, documentation, implémentation... de chaque élément).
- Les sprints sont guidés par une liste d' « items » provenant du Product Backlog formant le Sprint Backlog. Les tâches seront précisées par découpage des « users stories ». Il n'y a donc aucune tâche dans le Product Backlog.
- Durée de 5 à 20 jours ouvrables au maximum, soit 4 semaines.
- Une fois l'équipe en séquence de développement, l'équipe sera isolée de toute influence extérieure. Personne ne peut demander à l'équipe de travailler sur autre chose que ce sur quoi elle s'est engagée. Tout changement sera fait sur demande exclusive de la Dev Team.



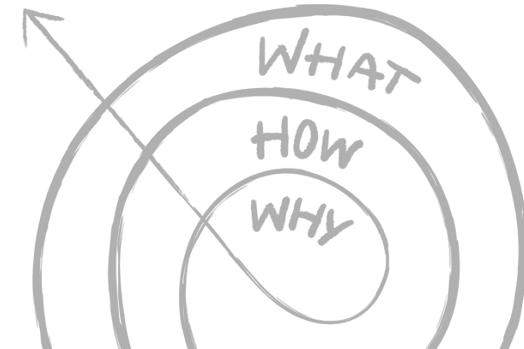
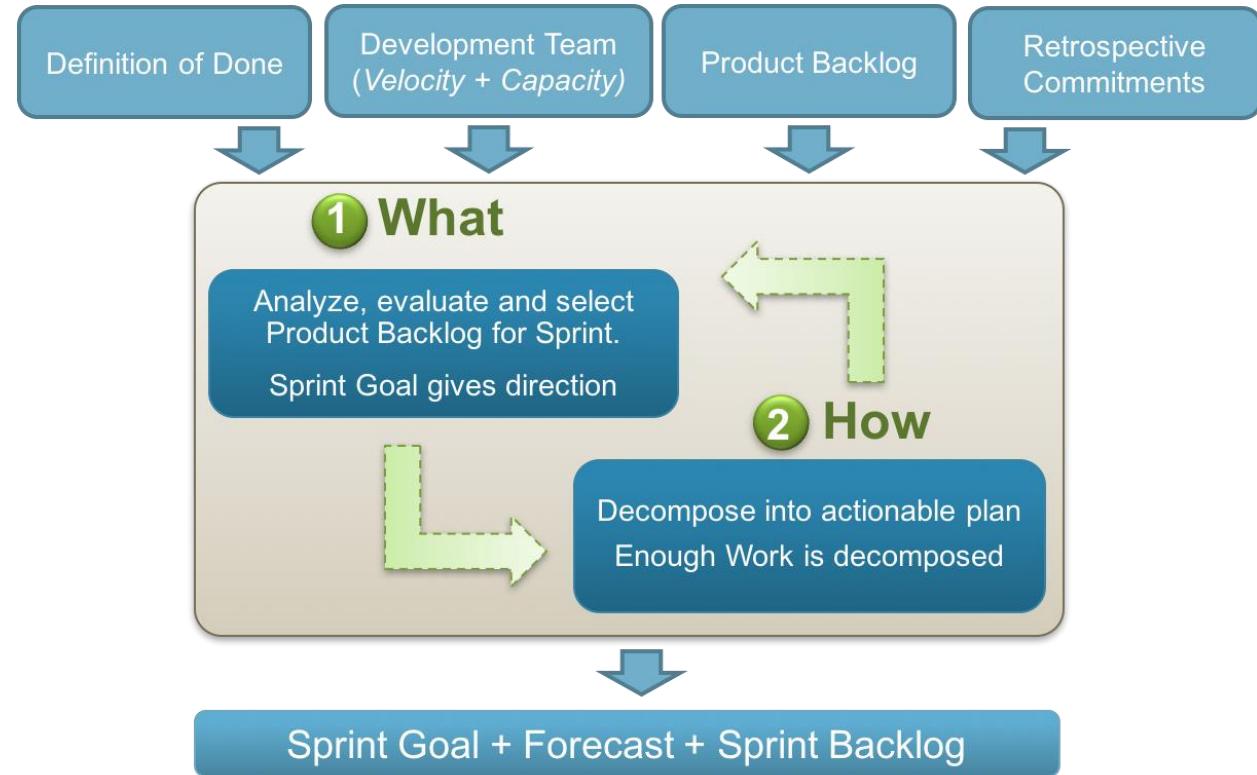
Scrum Basics

La phase de Sprint – Le Sprint Planning Meeting

- Pour rappel, cet événement est timeboxé à 8h00 pour un Sprint de 4 semaines.

- Le Sprint Planning répond à deux questions :

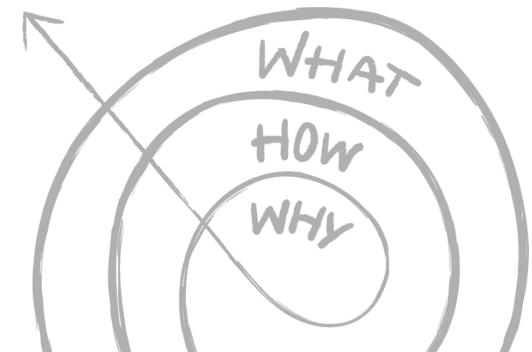
- Qu'est ce qui peut être terminé au cours de ce Sprint (**What Meeting**) ?
- Comment sera effectué le travail choisi (**How Meeting**) ?



Scrum Basics

La phase de Sprint – Le Sprint Planning Meeting en 2 thèmes

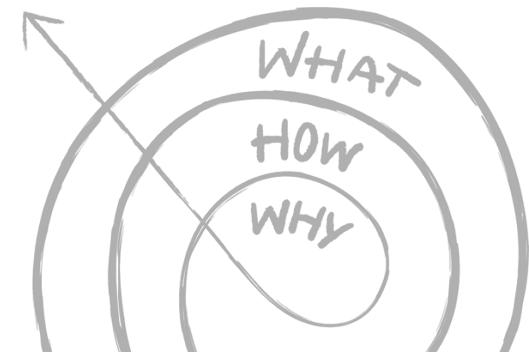
- Durant le **What Meeting**, La Development Team collabore avec le Product Owner pour envisager la ou les fonctionnalités à développer durant le sprint qui vient de débuter.
La Team discute de l'objectif qui devrait être atteint durant le Sprint et des Items du Product Backlog qui, s'ils sont terminés, permettront d'atteindre cet objectif. L'équipe Scrum collabore pour comprendre le travail requis.
- Pour planifier, il faut considérer le dernier incrément produit, la capacité de l'équipe (vitesse), les compétences présentes à un instant T et les risques identifiés. **Seule la Dev Team peut déterminer ce qu'elle peut accomplir durant le prochain Sprint.**
- Une fois que la Dev Team a déterminé les items du Product Backlog, la Scrum Team détermine le « **Sprint Goal** ».



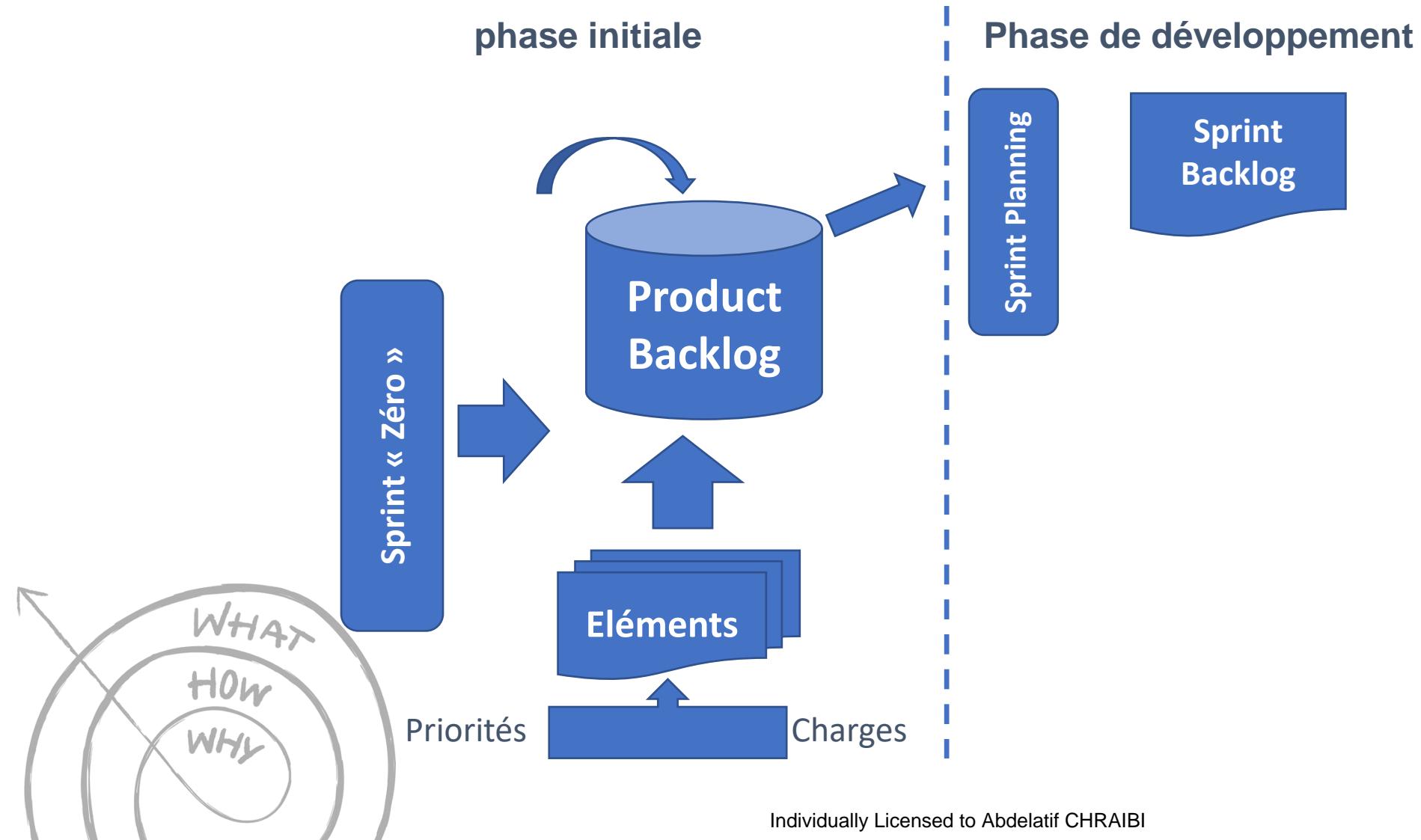
Scrum Basics

La phase de Sprint – Le Sprint Planning Meeting en 2 thèmes

- Une fois le Sprint Goal fixé, la Dev Team entre en ***How Meeting***.
- Durant ce thème, la Dev Team va planifier le travail pour transformer la ou les fonctionnalités en un incrément de produit « **DONE** ». Ainsi, les items du PB choisis et le plan conçu par l'équipe constituent le Sprint Backlog.
- La Dev Team commence généralement par concevoir le système et évalue le travail nécessaire afin de transformer le Sprint Backlog en un incrément fonctionnel du produit. La Dev Team pourra évaluer l'effort à fournir via par exemple un Planning Poker.
- La Dev Team s'auto-organise pour réaliser le travail consigné au SB, à la fois lors du Sprint Planning et aussi quand cela est nécessaire durant le Sprint. En ce sens, la Dev Team découpera en tâches les Stories afin de prévoir du travail pour chacun des membres durant les premiers jours du sprint. La suite se fera au fil de l'eau.



Scrum Basics

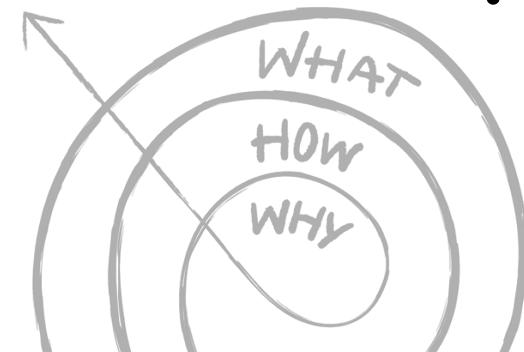


Scrum Basics

La phase de Sprint – La réunion journalière



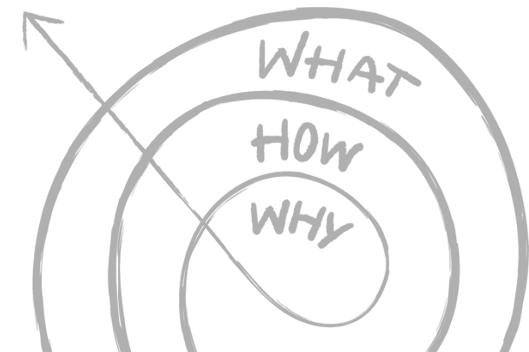
- Participation quotidienne à la réunion « **Daily Meeting** » aussi connue sous les noms de « **Stand-up Meeting** » ou « **Daily Stand-up** » ou encore « **Daily Scrum** »
 - **L'objectif de cette réunion :**
 - Partager les connaissances acquises
 - Faire un point sur l'avancement
 - Donner au management une certaine visibilité sur la progression du projet
 - Annoncer l'avancement afin de tenir à jour les artefacts
 - **Il s'agit d'un contrôle continu et empirique via 3 questions quotidiennes :**
 - qu'est ce qui a été fait depuis le dernier meeting ?
 - qu'est ce qui sera fait d'ici le prochain meeting ?
 - quels sont les obstacles qui gênent l'avancement du projet ?



Scrum Basics

La phase de Sprint – La réunion journalière

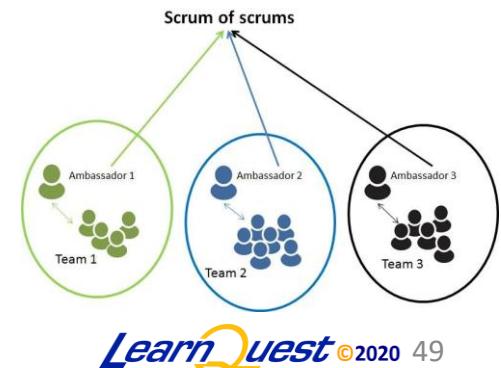
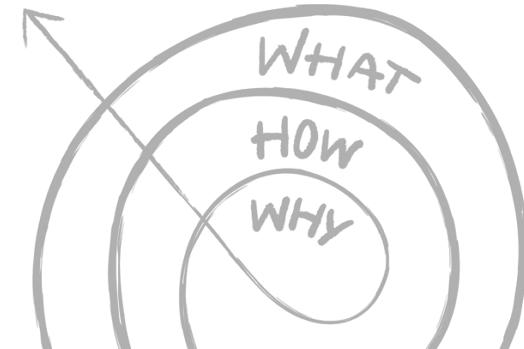
- Timebox de 15 min, peu importe la durée du sprint
- Si l'équipe n'a pas trouvé de solution à une problématique, le Scrum Master peut prendre les décisions et s'engage à apporter une solution à tout ce qui entrave le développement du projet.
- Il permet la synchronisation quotidienne de l'équipe et le partage des connaissances.
- Il peut avoir lieu dans certain cas, à n'importe quel moment de la journée (équipes externalisées et réparties)
- Ces réunions doivent se faire en face à face et non via échanges de mails.



Scrum Basics

La phase de Sprint – Le Scrum Of Scrum

- Le Scrum de Scrums est une technique Agile à grande échelle qui vous permet de connecter plusieurs équipes devant travailler ensemble pour livrer des solutions complexes.
- Il aide les équipes à développer et à livrer des produits complexes par le biais de la transparence, de l'inspection et de l'adaptation, le tout à grande échelle. Il est particulièrement efficace lorsque tous les membres de l'équipe Scrum, ultra performante, visent un objectif commun, se font confiance, se respectent et sont sur la même longueur d'onde.

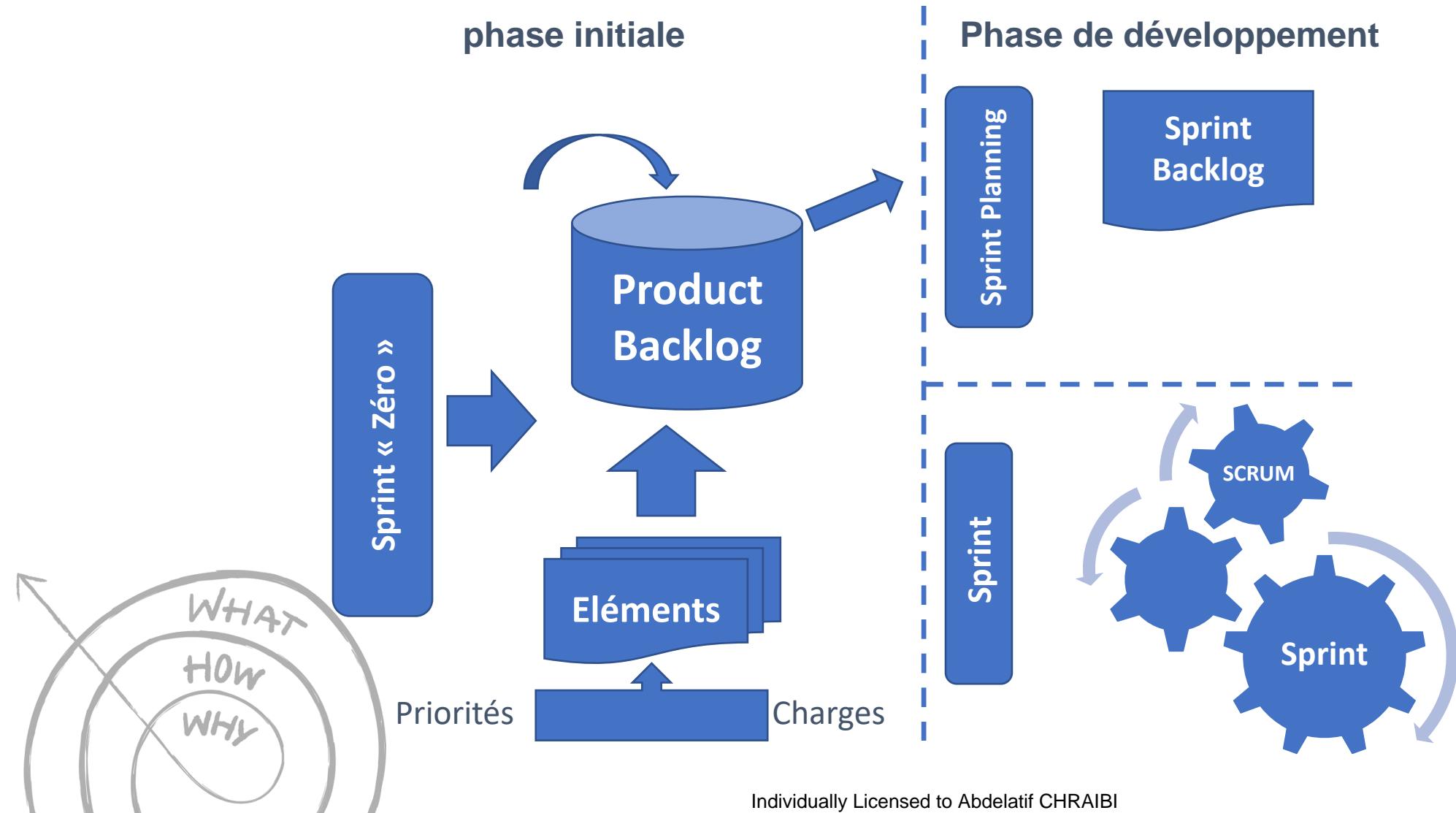


Scrum Basics

La phase de Sprint – Le Scrum Of Scrum



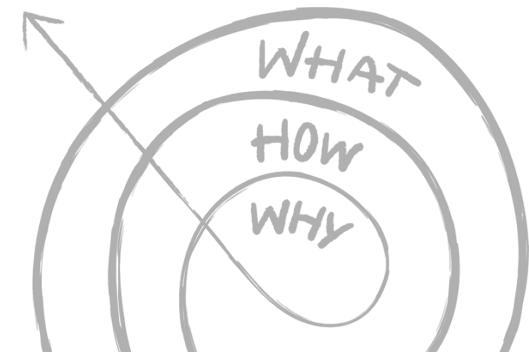
Scrum Basics



Scrum Basics

Réunion Post Séquence de Développement : Le Sprint Review

- Réunion formelle (Scrum Team, Client, Stakeholders, Experts, ...)
L'équipe présente:
 - la manière adoptée pour atteindre les objectifs
 - au client ce qui a été développé pendant la durée du sprint précédent via une démonstration
- Inspection de l'incrément du produit et adaptation du carnet de produit si nécessaire.
L'équipe SCRUM et les parties prenantes échangent sur ce qui a été fait durant le sprint.
- Les participants collaborent pour déterminer les prochains items qui pourraient être faits.
- La présentation de l'incrément est destiné à susciter des réactions et à favoriser la collaboration.

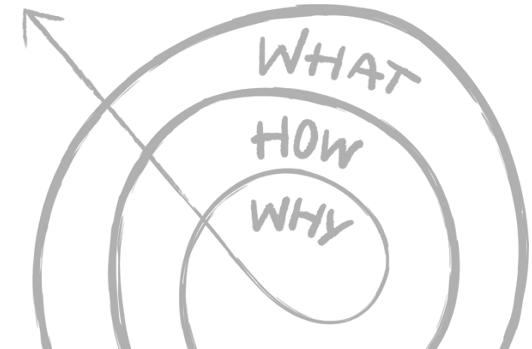


Scrum Basics

Réunion Post Séquence de Développement : Le Sprint Review



- **Le Sprint Review comprend les éléments suivants :**
 - Les participants incluent la Scrum Team, les stakeholders principaux que le PO a invité.
 - Le PO explique les items du PB qui ont été terminés et ceux qui ne l'ont pas été.
 - La Dev Team discute de ce qui s'est bien déroulé durant le Sprint, quels problèmes ont été rencontrés, et comment ces problèmes ont été résolus.
 - La Dev Team démontre le travail « DONE » et répond aux questions sur l'incrément.
 - Le PO discute du PB tel qu'il est. Il détermine des dates probables d'achèvement en fonction des progrès à ce jour.

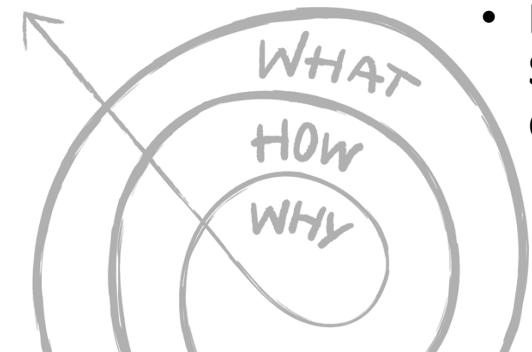


Scrum Basics



Réunion Post Séquence de Développement : Le Sprint Review

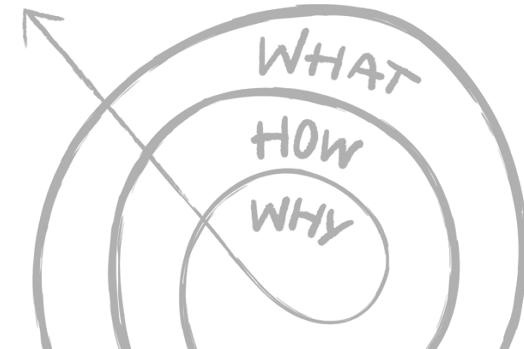
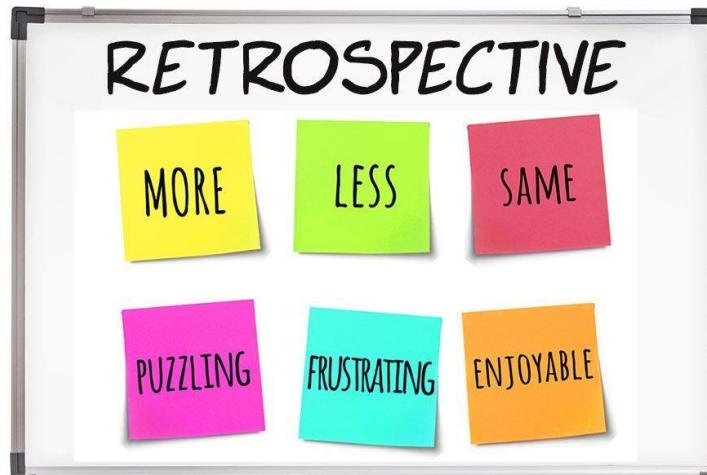
- **Le Sprint Review comprend les éléments suivants :**
- L'ensemble du groupe convient de ce qu'il faut faire pour la suite, de sorte que le Sprint Review fournit une contribution précieuse aux Sprint Planning subséquents.
- Une revue de la façon dont les conditions de marché ou un usage potentiel du produit pourrait avoir dicté ce qu'il conviendrait mieux de faire dorénavant.
- Une revue des délais, budget, fonctionnalités potentielles et conditions du marché pour la prochaine livraison du produit.
- **Le résultat du Sprint Review est un PB révisé qui définit les items probables pour le prochain Sprint. Le PB peut également complètement revu pour répondre à de nouvelles occasions d'affaires.**



Scrum Basics

Réunion Post Séquence de Développement : Le Sprint Retrospective

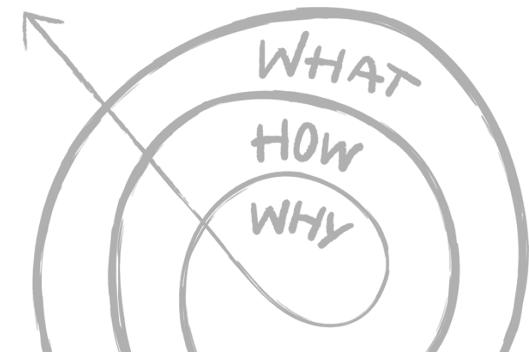
- Occasion pour l'équipe SCRUM de s'inspecter et de créer un plan d'améliorations qui sera mis en place au cours du sprint suivant.
- L'impédiment Backlog sera utilisé pour organiser les points d'améliorations. Ces éléments seront placés sur des cartes et priorisés à la façon de stories. Quelques cartes seront sélectionnées pour améliorations et placées dans le prochain Sprint Backlog.
- Une mise à jour de la **Définition Du Done** pourra également avoir lieu lors de cet événement.



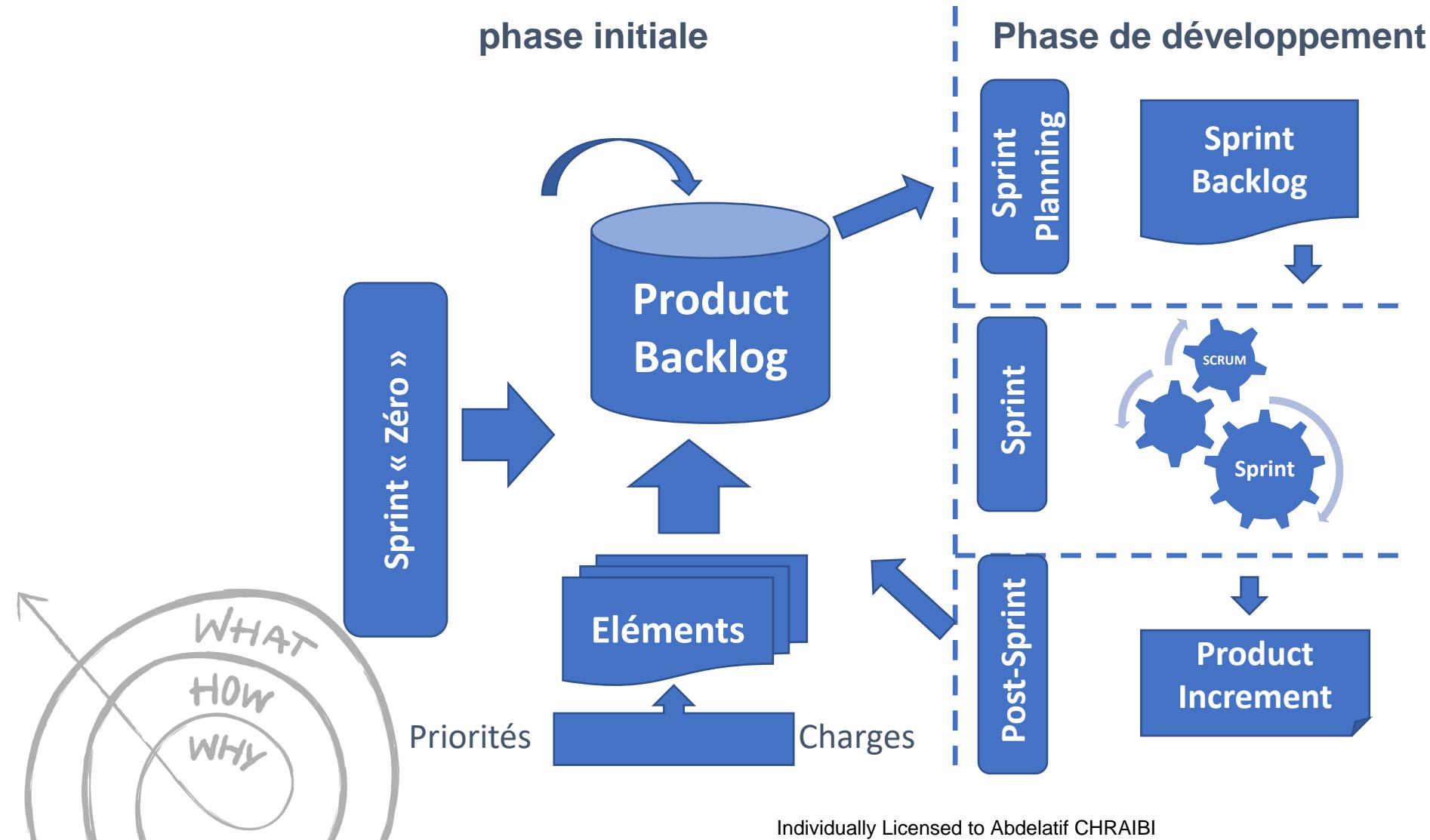
Scrum Basics

Réunion Post Séquence de Développement : Le Sprint Retrospective

- **Le but de la rétrospective de sprint est :**
- D'inspecter la manière dont le dernier Sprint s'est déroulé en ce qui concerne les personnes, les relations, les processus et les outils. (*attention, on ne traite plus du produit lors de cet événement*)
- D'identifier et ordonner les éléments majeurs qui se sont bien déroulés et les améliorations potentielles.
- De créer un plan pour améliorer les processus de travail de la Scrum Team
- Le sprint rétrospective fournit un événement dédié et axé sur l'inspection et l'adaptation



Scrum Basics

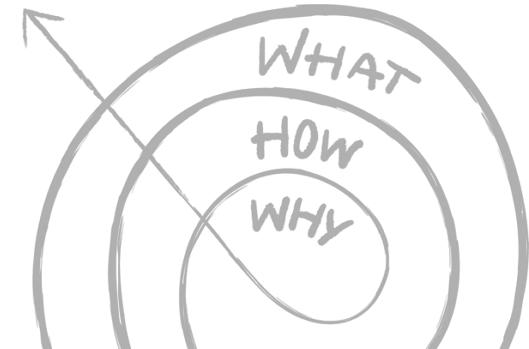


Scrum Basics

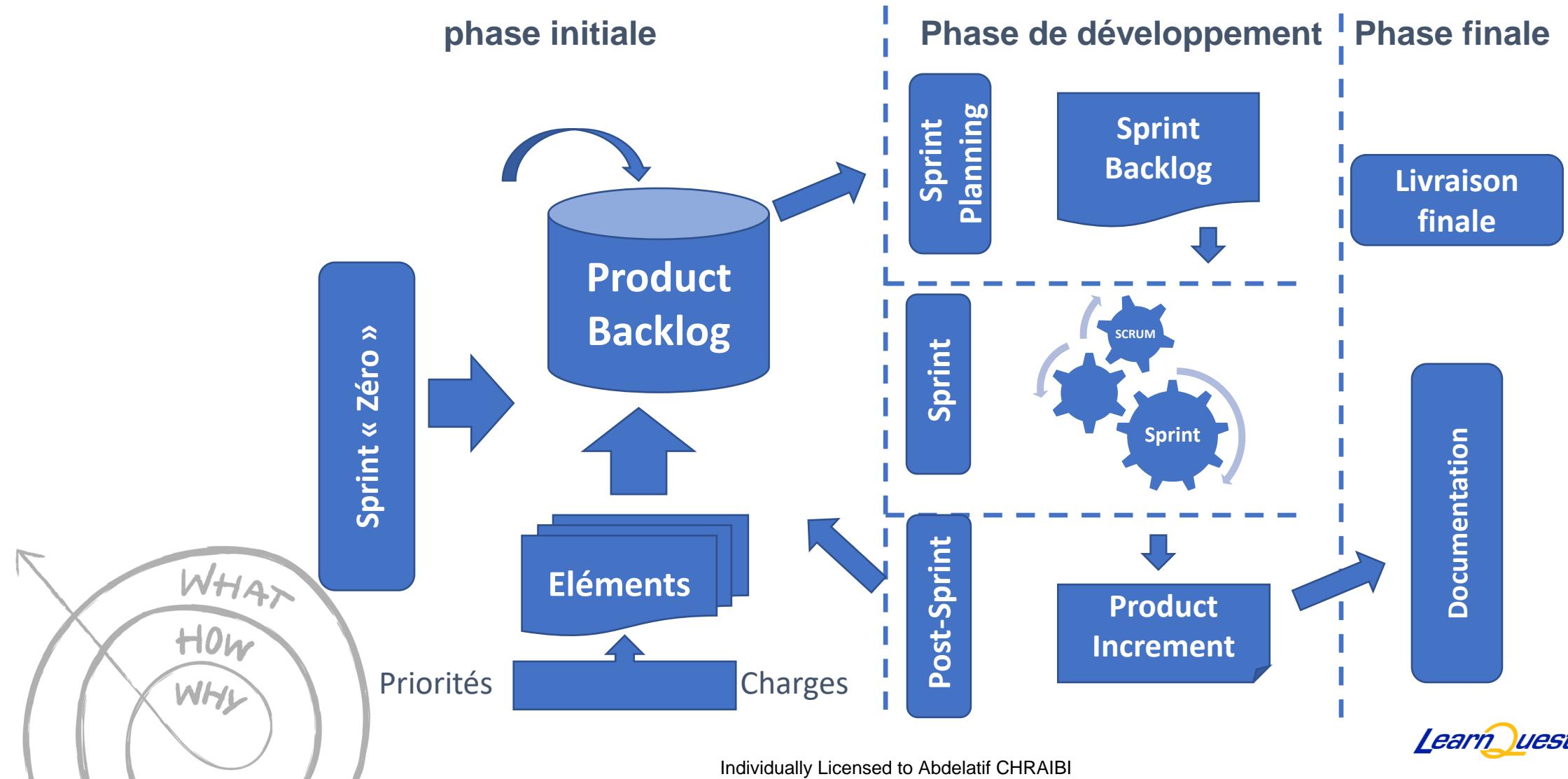
La phase de clôture

- **Préparation du produit pour une livraison:**

- Intégration des modules validés,
- tests systèmes effectués,
- documentation utilisateur réalisée,
- préparation de supports de formation (en fonction du cahier des charges),
- préparation de supports marketing (en fonction du cahier des charges),
- ...



Scrum Basics



Plan de ce module

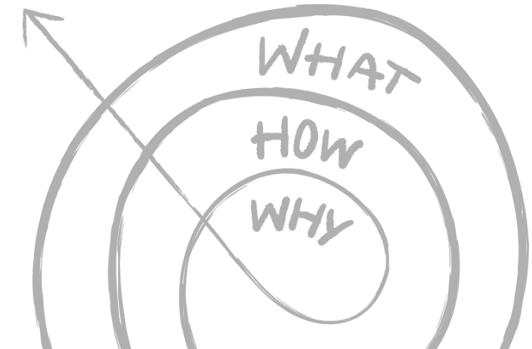
Voici les parties que nous allons aborder :

- **Partie 1 : Introduction & Scrum Basics**

- Les Rôles
- Le Processus
- **Le Product Owner**
- Le Scrum Master

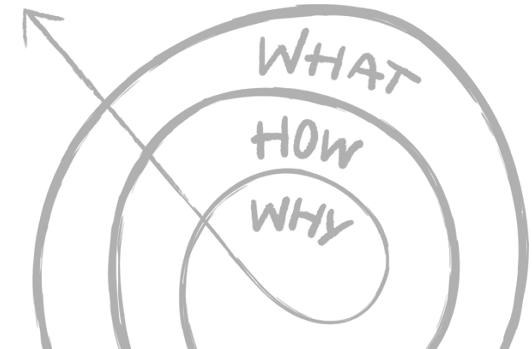
- **Partie 2 : Mise en œuvre**

- Le Produit
- Le Product Backlog
- Les Releases
- Les Sprints
- L'agilité à l'échelle



Scrum Basics

Le Product Owner : presque un Chef de projet ?



Gère le
product backlog

S'accorde
avec le Scrum
Master

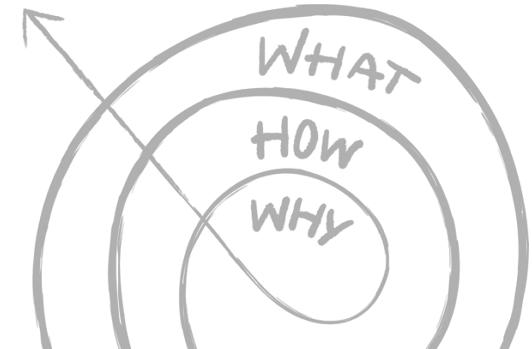
o Création de la
Vision du produit

o Challenge la
Dev Team pour
optimiser le ROI

Scrum Basics

Le Product Owner

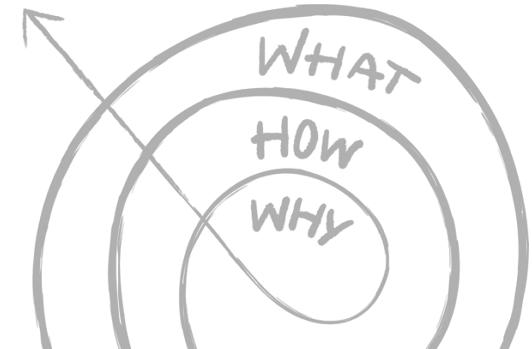
- **Le rôle du Product Owner**
- **Les caractéristiques**
- **Collaborations**
- **Ajuster le rôle du Product Owner pour de larges projets**
- **Erreurs courantes**



Scrum Basics

Le Product Owner : Son Rôle

- Il est l'unique responsable de la gestion du Product Backlog
- Il s'assure que le Product Backlog soit visible pour chacun
- Il s'assure de la valeur du travail fourni par l'équipe
- Il optimise le ROI et TCO



Scrum Basics

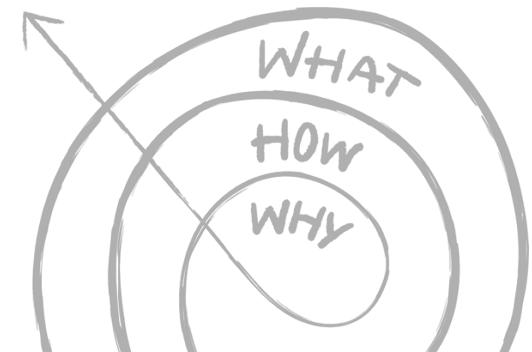
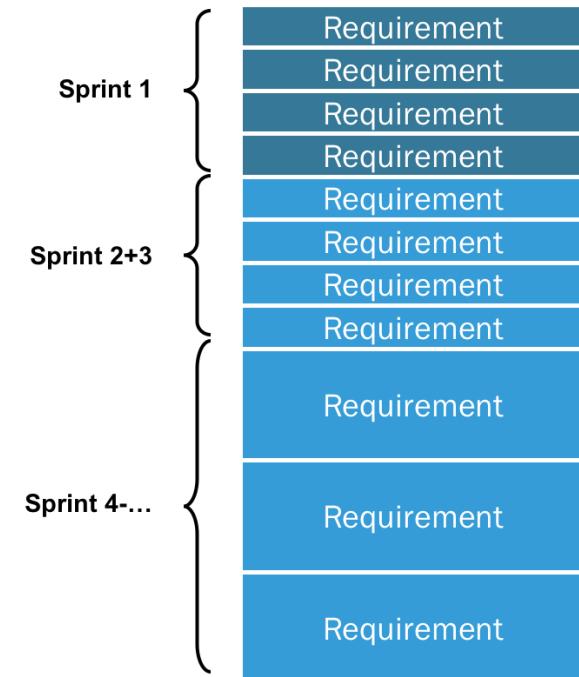
Le Product Owner : Son Rôle



Scrum Basics

La gestion du Product Backlog comprend :

- Exprimer clairement les items du Product Backlog
- Ordonner les items du Product Backlog pour mieux réaliser les objectifs et missions
- Optimiser la valeur du travail effectué par la Dev Team
- S'assurer que le Product Backlog est visible, transparent et clair pour tous et qu'il montre ce sur quoi l'Equipe de développement travaillera prochainement
- S'assurer que la Dev Team comprend adéquatement les items du Product Backlog



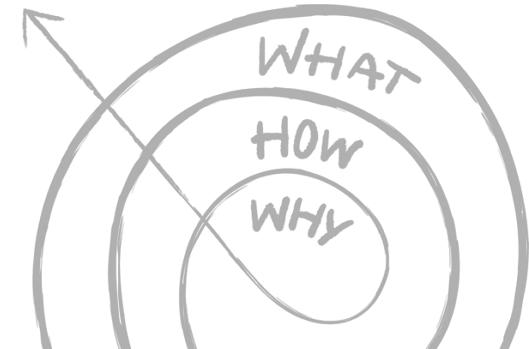
Scrum Basics

Le Product Owner : Caractéristiques

- Visionnaire (*C'est lui qui sur base de la définition des besoins va créer la vision du produit*)
- Leader et Team Player
- Communicateur et Négociateur
- Concerné et Engagé
- Disponible et Qualifié
- Souvent, il possède une expérience en Gestion de Projet MOA

The Vision Board

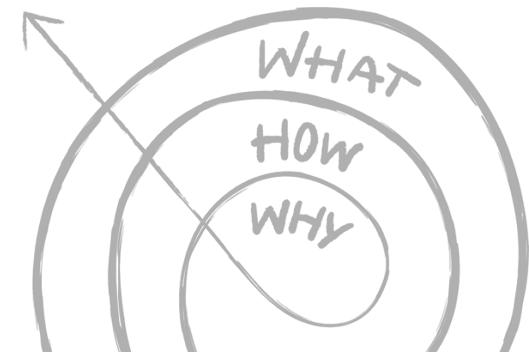
The Vision Board				
Vision Statement	Target Group	Needs	Product	Value
<p>Vision Statement Crisp summary of the vision / idea.</p>	<p>Target Group Which market segment does the product address? Who are the target users and customers?</p>	<p>Needs Which needs does the product fulfill? How does it create value for its users? Which emotions will it evoke?</p>	<p>Product What are the three to five top features? What are its unique selling points?</p>	<p>Value How is the product going to benefit the company? Will it, for instance, increase revenue, enter a new market, develop the brand, reduce cost, create valuable knowledge?</p>



Scrum Basics

Le Product Owner : Caractéristiques

- Le **Product Owner** peut réaliser toutes les tâches qui lui incombent ou les déléguer à la Dev Team. Toutefois, le PO demeure responsable de ces dernières.
- Le **Product Owner** est également une seule personne et non un comité. Cependant, il peut représenter un comité.
- Afin que le **PO** réussisse dans sa démarche, tous les intervenants de l'entreprise doivent respecter ses décisions. Ces décisions sont visibles dans le contenu et l'ordonnancement du Product Backlog
- Il est le **SEUL** point de référence en matière d'informations pour la Dev Team



Scrum Basics

Le Product Owner : Ajustement du rôle

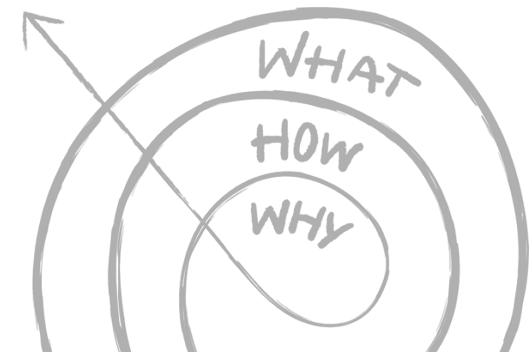
Pour les grands projets SCRUM

- Difficile pour un Product Owner de s'occuper de plus de 2 SCRUM Team
- Quid de projets comprenant plus de 2 SCRUM Team alors que le Product Owner est unique?

Introduction du rôle du Chief Product Owner

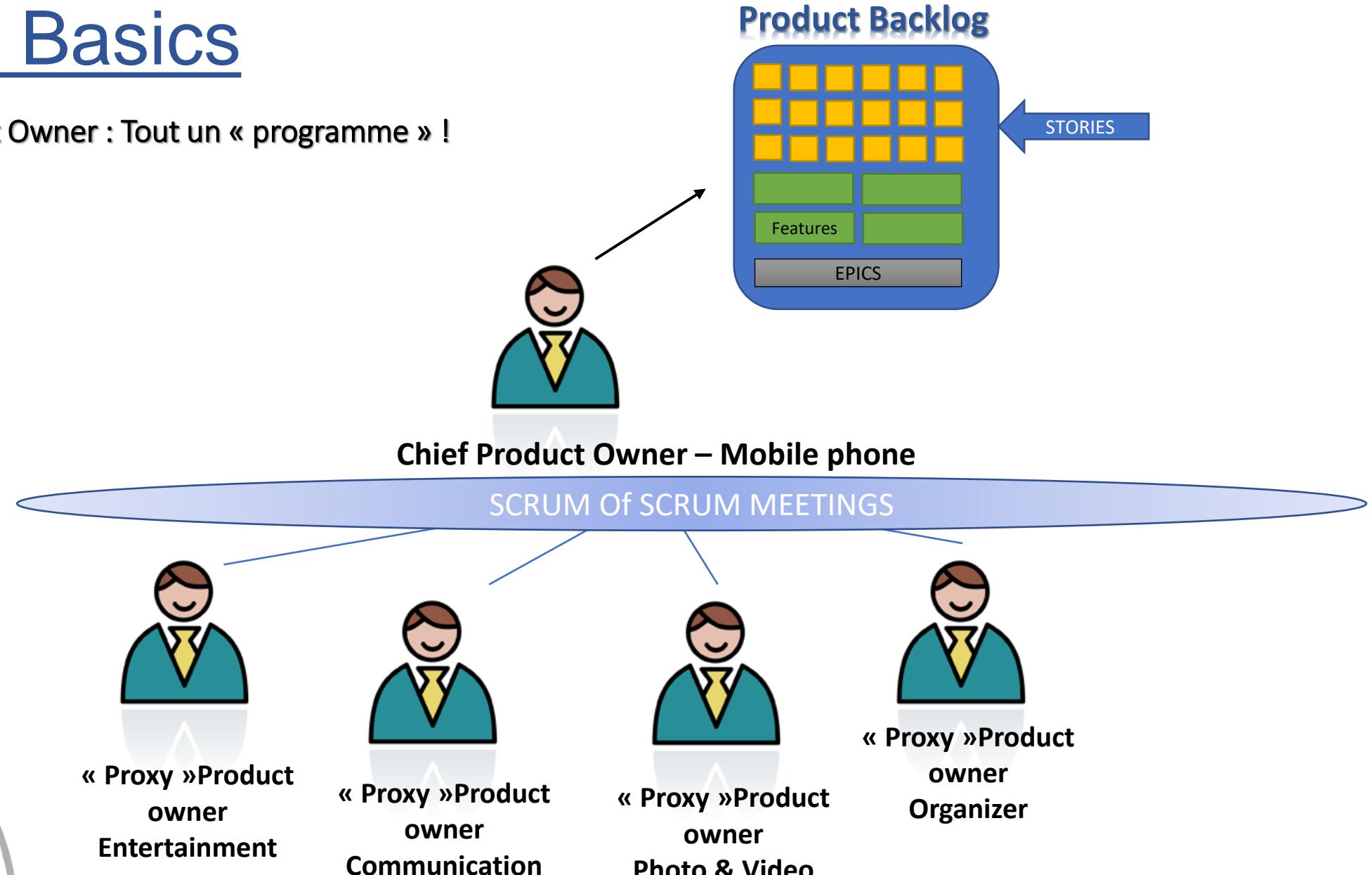
- Responsable de coordonner différents Product Owner
- Responsable de la vision du produit

« Dans certaines entreprises, le Product Owner d'un « lot » sera qualifié de Proxy Product Owner»



Scrum Basics

Le Chief Product Owner : Tout un « programme » !

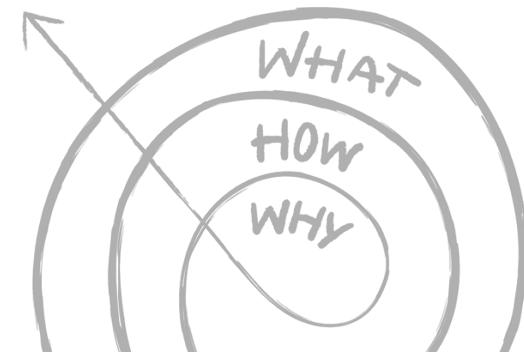


Scrum Basics



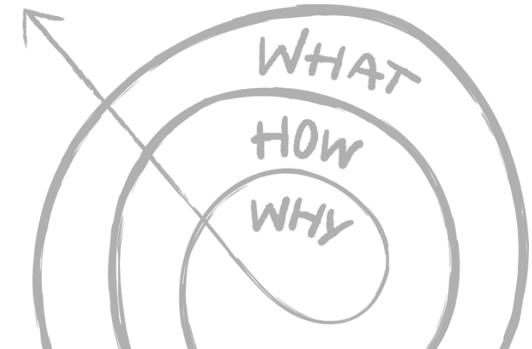
Le Product Owner : Les erreurs courantes

- **Ne pas donner assez de pouvoir au Product Owner**
 - Si il ne dispose pas d'assez d'autonomie, le PO peut être limité sur la valeur fournie au client et cela peut avoir un impact sur la qualité du produit.
- **Surcharger le Product Owner**
 - Cela n'est pas soutenable
 - Il peut devenir le bottleneck du projet
- **Surtout ne pas séparer les tâches du Product Owner entre différentes personnes**
 - Risques d'interprétations de la vision par les différentes personnes



Scrum Basics

Créer des stories : une affaire de Pro!



Scrum Basics

Créer des stories : I.N.V.E.S.T

- **I** : Indépendant – Liberté de prioriser la story sans contrainte
- **N** : Négociable – Flexibilité dans le périmètre et dans la solution technique (pas trop spécifique et précise)
- **V** : Valeur – La story doit apporter de la valeur aux utilisateurs
- **E** : Estimable – Il doit être possible d'estimer la charge de travail que représente la story
- **S** : Suffisamment Petite – pour être développée en un seul sprint
- **T** : Testable – Le résultat correspond-il bien à l'objectif précisé ?

En Tant Que (X)

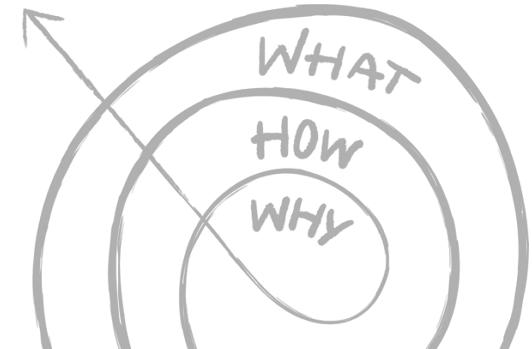
Je veux (Y)

Afin de (Z)

X est la personne ou le rôle qui va bénéficier de la fonctionnalité

Y est l'action liée à la fonctionnalité

Z est l'objectif ou le bénéfice de la fonctionnalité



Scrum Basics

Créer des stories : travailler avec des scénarios

En Tant que : Client de la banque

Je veux : Retirer du liquide d'un distributeur

Afin de : Ne plus faire la file à l'agence

* **Scénario 1** : Le compte est crédité

Etant donné que le compte est crédité

Et que la carte est valide

Et que le distributeur contient du liquide

Lorsque le client demande du liquide

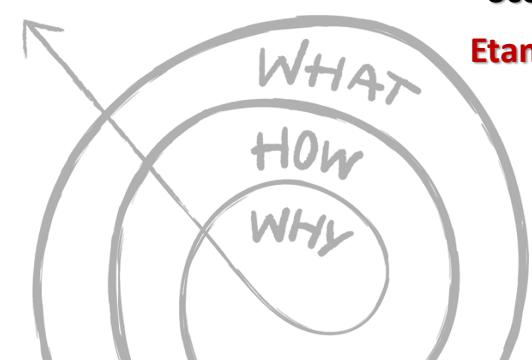
Alors on s'assure que son compte est débité

Et que le liquide est distribué

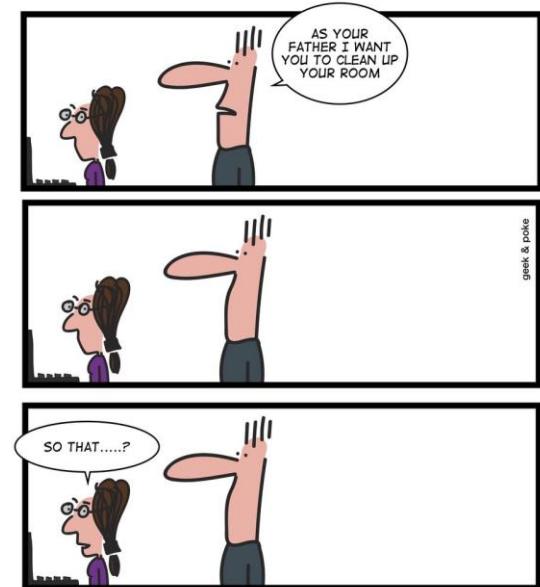
Et que la carte est rendue

***Scénario 2** : Le compte est débiteur au-delà de la limite du découvert autorisé

Etant donné que le compte est débiteur



AGILE FAMILIES



geek & pole

Scrum Basics

Créer des stories : Moi parler GHERKIN

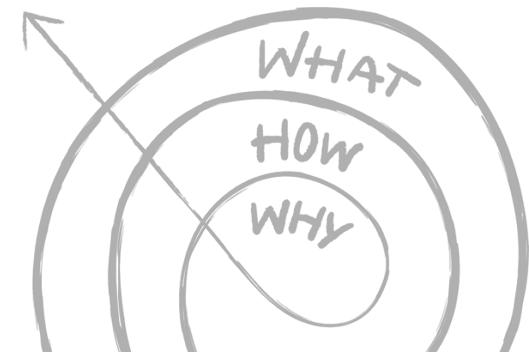
L'approche Gherkin a pour objectif d'intégrer une formalisation du développement pour obtenir automatiquement des tests fonctionnels en écrivant des scénarios de tests compréhensibles par des individus non techniques.

Cette approche sert 2 objectifs :

1. Documenter les fonctionnalités à développer
2. Permettre l'automatisation des tests

La définition d'un scénario de test en Gherkin se fait selon 3 étapes clés : **Given – When – Then**

- **Given** : Liste des conditions initiales nécessaires au test
- **When** : Décrit les actions à effectuer
- **Then** : Décrit le résultat attendu en cas de bon fonctionnement du produit

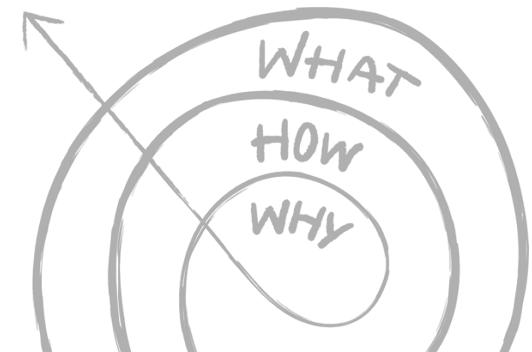


Scrum Basics

Créer des stories : Moi parler GHERKIN

Exemple Nominal

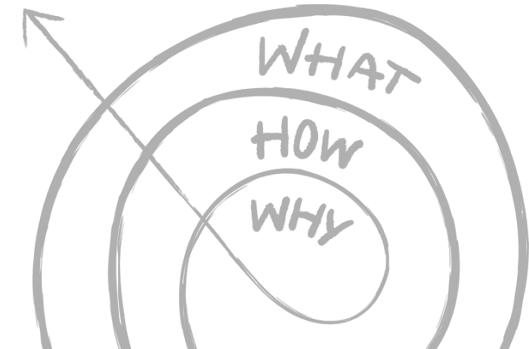
- **Fonctionnalité :** Authentification
- **Scénario :** Tentative d'authentification avec un compte valide
- **Etant donné que :** je dispose d'un compte utilisateur « Jean »
- **Quand :** j'accède à la page d'authentification /verif.html
- **ET :** Que je saisis mon identifiant « Jean » dans le champ « Login »
- **ET :** Que je saisis mon mot de passe dans le champ « Password »
- **ET :** Que je clique sur le bouton « connexion » du formulaire
- **Alors :** Je suis authentifié sur le site
- **ET :** Je suis redirigé vers la page « Mon Compte » à l'URL /moncompte.html



Scrum Basics

Créer des stories : Spécifier les tests

1. **Les tests unitaires :** Il s'agit d'un code source qui permet de vérifier un mécanisme de l'application. Un test = Un cas.
2. **Les tests d'intégration :** Il s'agit des tests permettant d'assembler tous les modules indépendants de l'application et de les tester.
3. **Les tests système :** Ils regroupent les tests de performances, de compatibilités, de sécurités, de volume, ...
4. **Les tests d'acceptance :** C'est un outil fourni par le Product Owner permettant de déterminer si une story est terminée. C'est un test qui peut être écrit conjointement avec le Product Owner, l'expert métier, des développeurs et le Scrum Master. Les tests d'acceptances sont concentrés sur le QUOI plutôt que sur le COMMENT. Ils permettent de garantir la conformité entre la fonctionnalité livrée et les règles fonctionnelles définies dans la user story.



Plan de ce module

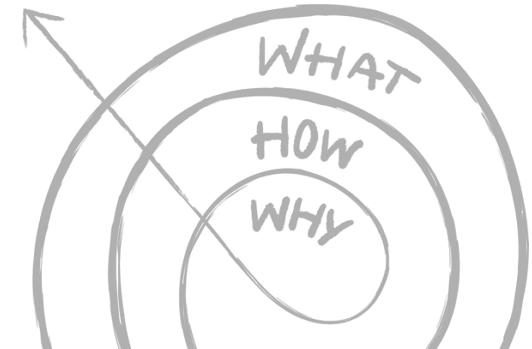
Voici les parties que nous allons aborder :

- **Partie 1 : Introduction & Scrum Basics**

- Les Rôles
- Le Processus
- Le Product Owner
- **Le Scrum Master**

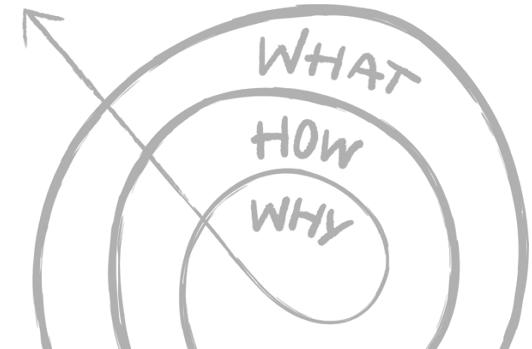
- **Partie 2 : Mise en œuvre**

- Le Produit
- Le Product Backlog
- Les Releases
- Les Sprints
- L'agilité à l'échelle



Scrum Basics

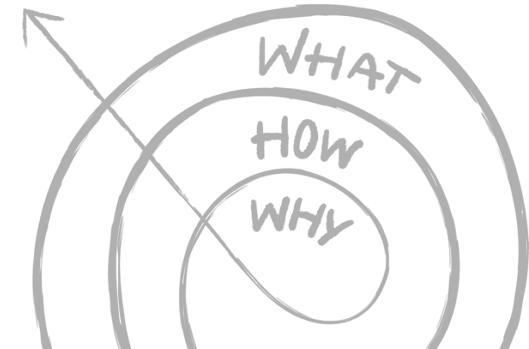
Le Scrum Master



Scrum Basics

Le Scrum Master

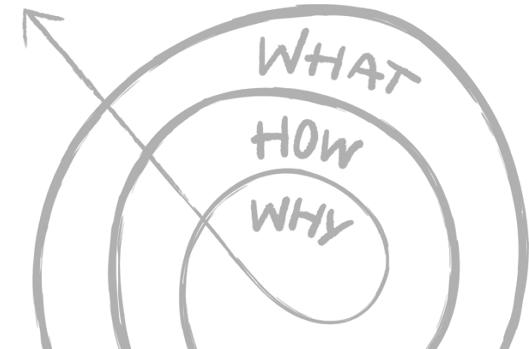
- **Le rôle du Scrum Master**
- **Les caractéristiques**
- **Erreurs courantes**



Scrum Basics

Le Scrum Master

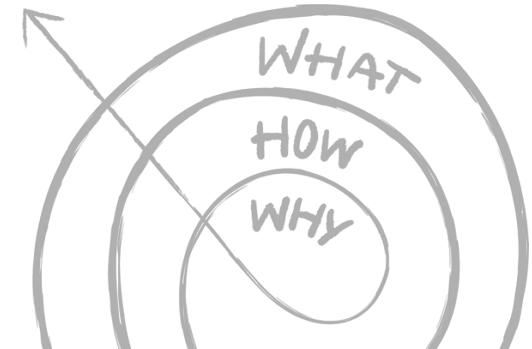
- Il est le garant de l'application du processus Scrum
- Il aide l'équipe à travailler de façon autonome et à s'améliorer constamment
- Il comprend et pratique l'agilité
- Il élimine les obstacles qui ralentissent l'équipe
- Il communique avec le management (ex. adoption de scrum)
- ...



Scrum Basics

Le Scrum Master : Les caractéristiques

- Bien connaître SCRUM, il est l'expert du « Framework »
- Talent de communication, de Team Leader, de coach, de facilitateur
- Pouvoir résoudre les conflits – problem solver
- Etre capable de guider sans imposer



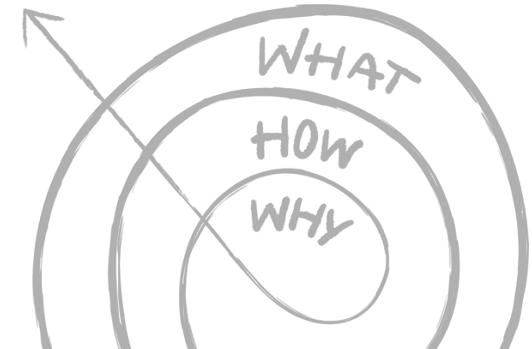
Scrum Basics

Le Scrum Master : Les caractéristiques

- Il ne dirige pas, il n'impose pas, il ne contraint pas
- Il fait partie de l'équipe
- Il est souvent un membre de la Development Team
- Il peut aussi changer de personne en cours de développement

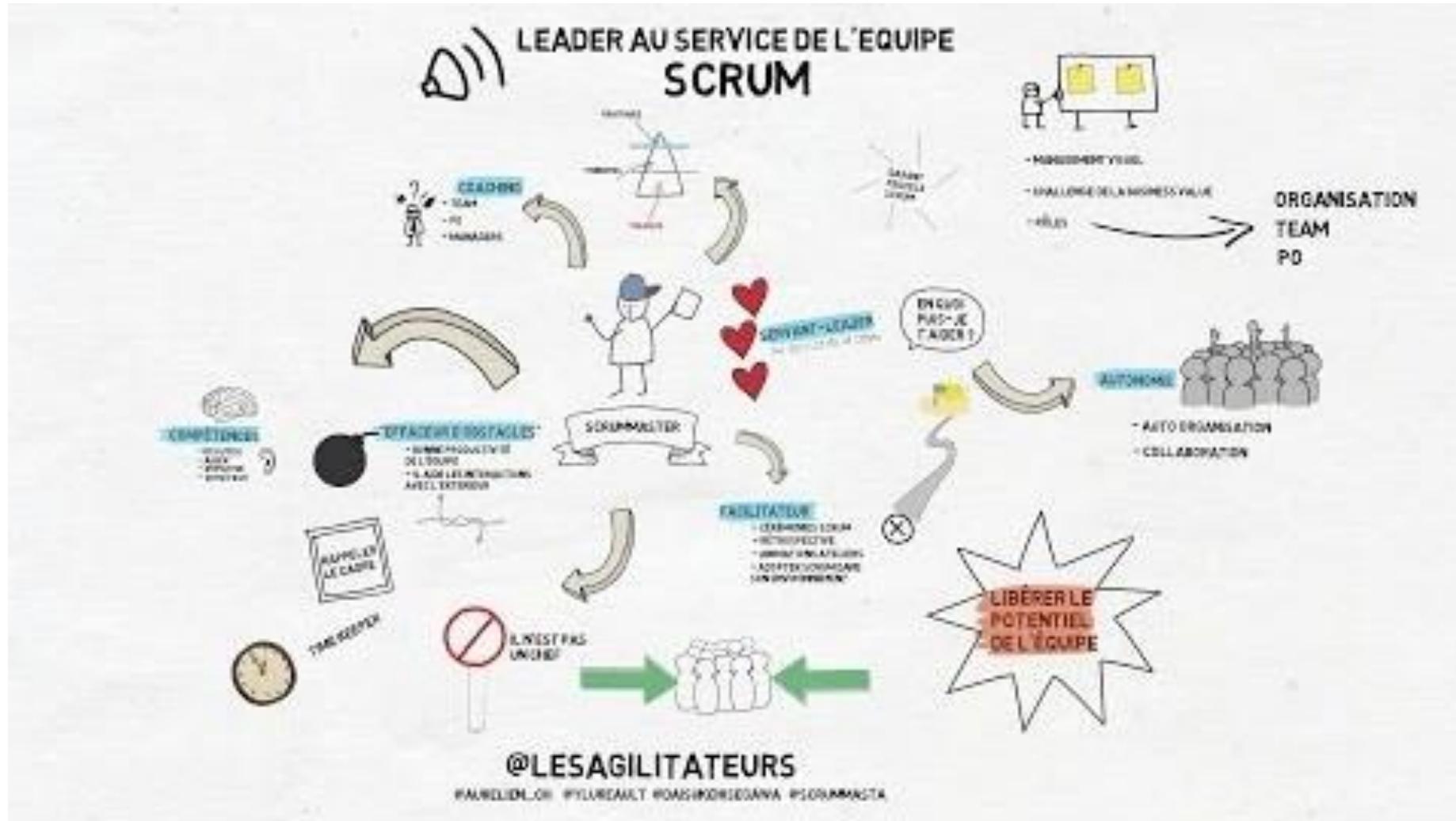
Le Scrum Master : Une erreur courante

- Attribuer les rôles de Scrum Master et Product Owner sur la même personne



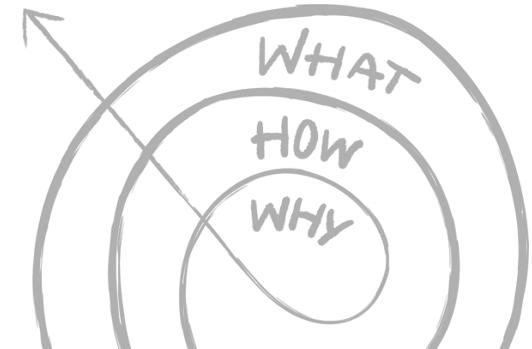
Scrum Basics

Le Scrum Master : Les caractéristiques

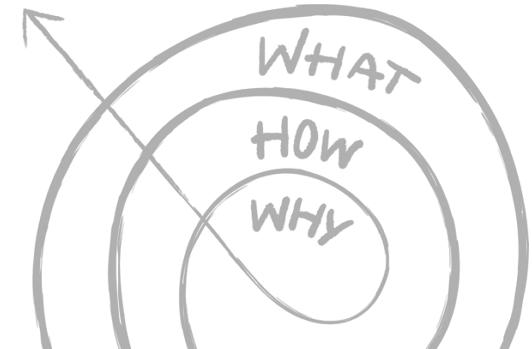




Avez-vous des questions sur
cette première partie ?

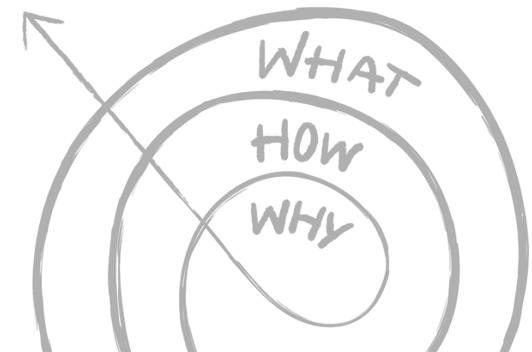


**KEEP
CALM
AND
PREPARE
FOR A TEST**



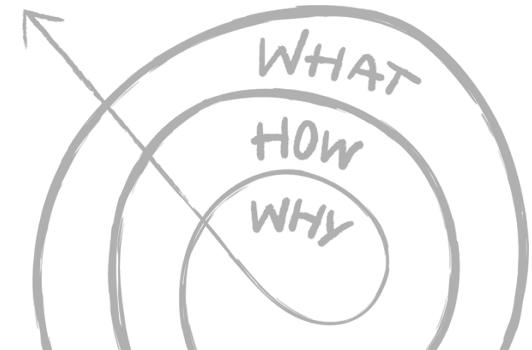
La Scrum Team est composée de plusieurs rôles. Quels sont-ils ?

1	Scrum Master, Stakeholder, Product Owner, Development Team
2	Scrum Master, Product Owner, Development Team
3	Scrum Master, Product Owner, Chef de projet, Development Team
4	Scrum Master, Development Team, Product Owner, Stakeholder, Chef de Projet



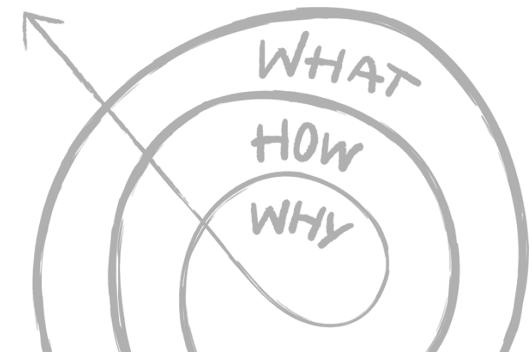
La Scrum Team est composée de plusieurs rôles. Quels sont-ils ?

1	Scrum Master, Stakeholder, Product Owner, Development Team
2	Scrum Master, Product Owner, Development Team
3	Scrum Master, Product Owner, Chef de projet, Development Team
4	Scrum Master, Development Team, Product Owner, Stakeholder, Chef de Projet



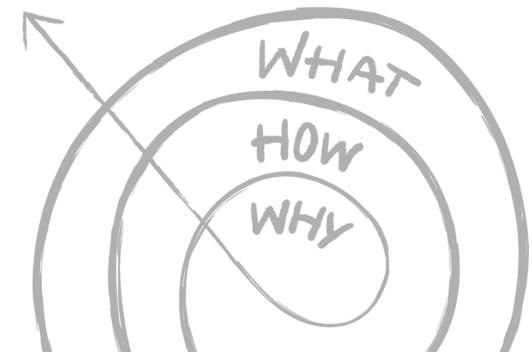
Quels sont les 3 piliers de l'empirisme ?

1	Inspection Adaptation Réaction
2	Inspection Transparence Acceptation
3	Transparence Inspection Adaptation
4	Transparence Itération Adaptation



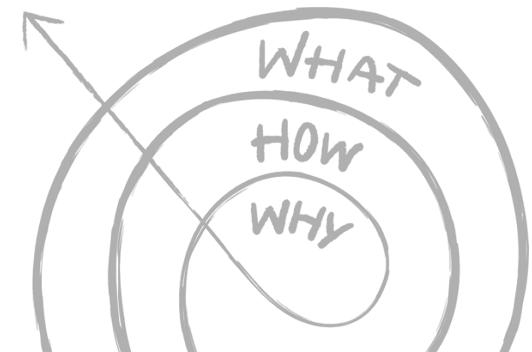
Quels sont les 3 piliers de l'empirisme ?

1	Inspection Adaptation Réaction
2	Inspection Transparence Acceptation
3	Transparence Inspection Adaptation
4	Transparence Itération Adaptation



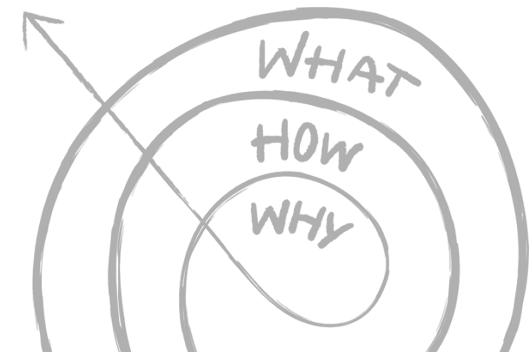
Scrum précise des événements formels pour inspecter et adapter. Quels sont ils ?

1	Sprint Planning Meeting, Sprint Review Meeting, Sprint Backlog review Meeting, daily Scrum Meeting
2	Daily Scrum Meeting, Sprint Planning Meeting, Sprint What Meeting, Sprint Review Meeting
3	Sprint Retrospective Meeting, Sprint Review Meeting, Sprint Planning Meeting, Daily Scrum Meeting
4	Sprint review Meeting, Sprint retrospective Meeting, Sprint Done Meeting, Daily planning Meeting



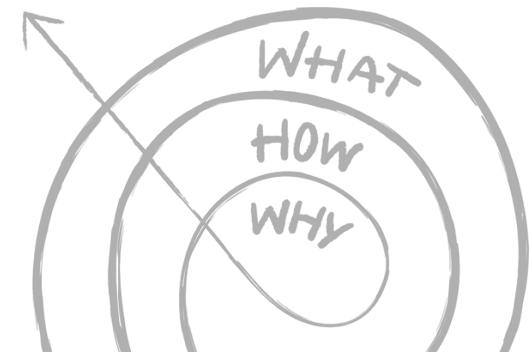
Scrum précise des événements formels pour inspecter et adapter. Quels sont ils ?

1	Sprint Planning Meeting, Sprint Review Meeting, Sprint Backlog review Meeting, daily Scrum Meeting
2	Daily Scrum Meeting, Sprint Planning Meeting, Sprint What Meeting, Sprint Review Meeting
3	Sprint Retrospective Meeting, Sprint Review Meeting, Sprint Planning Meeting, Daily Scrum Meeting
4	Sprint review Meeting, Sprint retrospective Meeting, Sprint Done Meeting, Daily planning Meeting



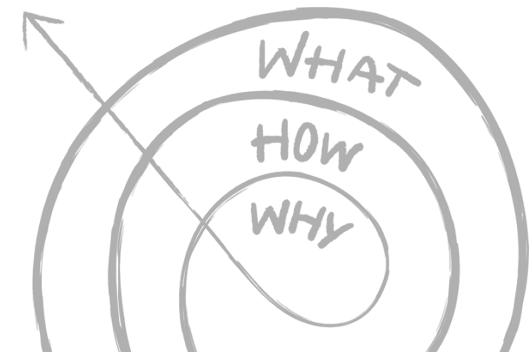
Scrum précise des « Timeboxes » pour chaque événement formel. Quelle proposition est correcte ?

1	Sprint Planning meeting 4h pour un sprint de 1 mois et proportionnellement moins selon la durée du Sprint
2	Scrum Review 3h pour un sprint de 1 mois et proportionnellement moins selon la durée du Sprint
3	Sprint de 3 à 4 semaines
4	Daily Scrum Meeting 15 minutes peu importe la durée du sprint



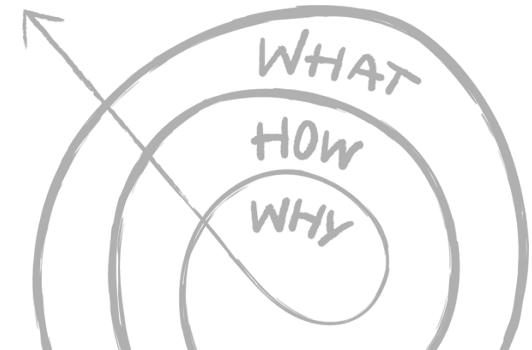
Scrum précise des « Timeboxes » pour chaque événement formel. Quelle proposition est correcte ?

1	Sprint Planning meeting 4h pour un sprint de 1 mois et proportionnellement moins selon la durée du Sprint
2	Scrum Review 3h pour un sprint de 1 mois et proportionnellement moins selon la durée du Sprint
3	Sprint de 3 à 4 semaines
4	Daily Scrum Meeting 15 minutes peu importe la durée du sprint



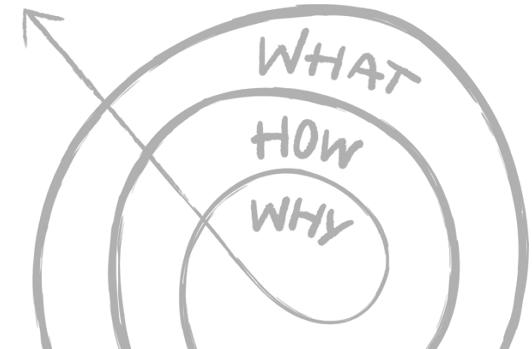
Quel est l'objectif principal du Sprint Retrospective ?

1	Pratiquer le refactoring du code
2	Trouver des améliorations pour l'équipe Scrum et les planifier pour le prochain Sprint
3	Intégrer les différents incrément en prévision de la release
4	Mettre à jour les différents artefacts pour le Sprint suivant



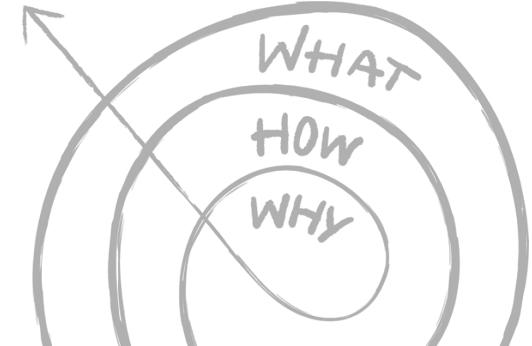
Quel est l'objectif principal du Sprint Retrospective ?

1	Pratiquer le refactoring du code
2	Trouver des améliorations pour l'équipe Scrum et les planifier pour le prochain Sprint
3	Intégrer les différents incrément en prévision de la release
4	Mettre à jour les différents artefacts pour le Sprint suivant





MISE EN OEUVRE



Plan de ce module

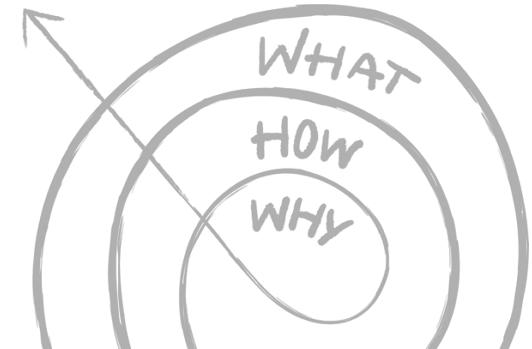
Voici les parties que nous allons aborder :

- **Partie 1 : Introduction & Scrum Basics**

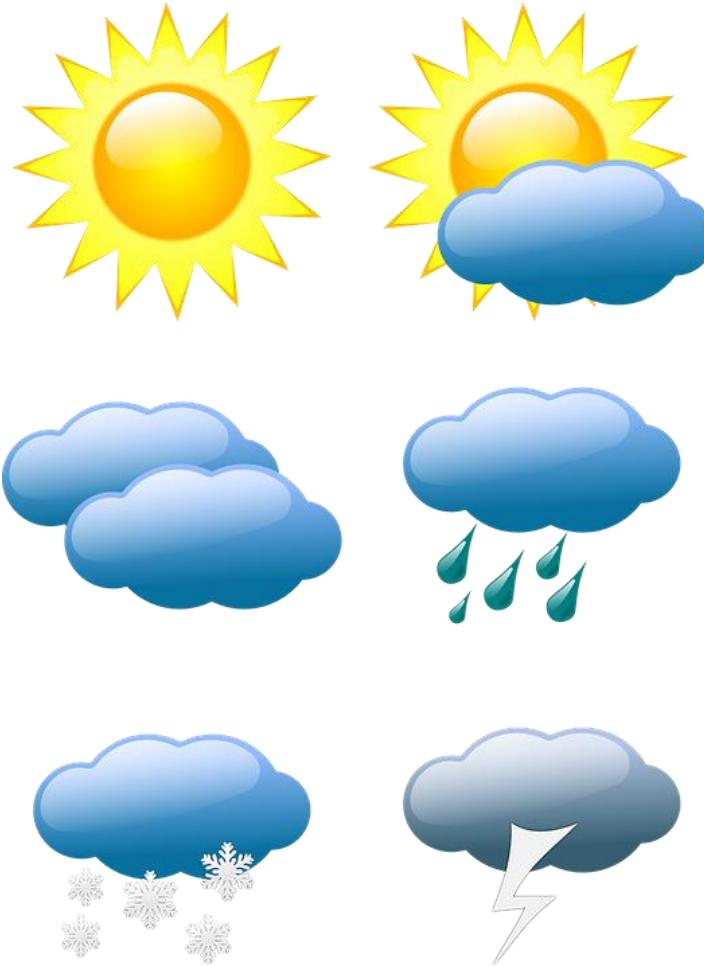
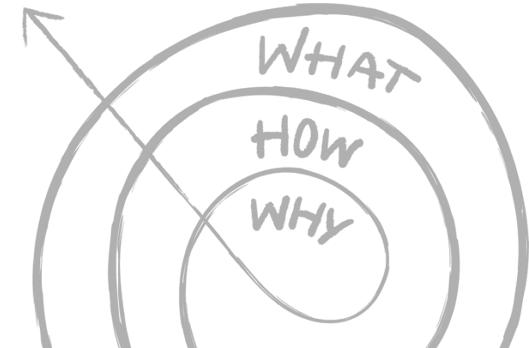
- Les Rôles
- Le Processus
- Le Product Owner
- Le Scrum Master

- **Partie 2 : Mise en œuvre**

- **Le Produit**
- Le Product Backlog
- Les Releases
- Les Sprints
- L'agilité à l'échelle

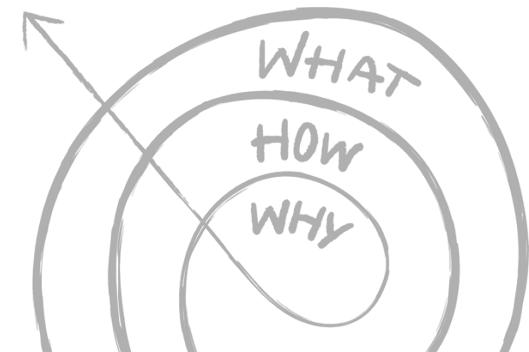


La météo du jour



Révision partie 1

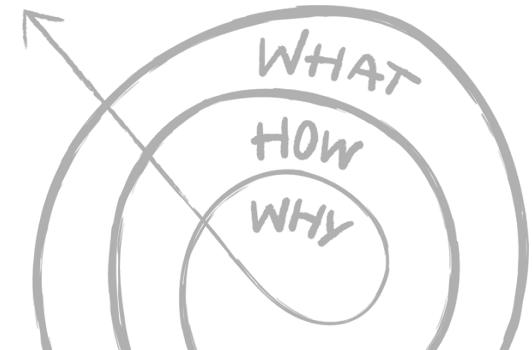
- Quels sont les Cérémonies SCRUM ? Quelles sont les timeboxes associées ?
- Qui est responsable du Product Backlog ? Du Sprint Backlog ?
- Quelle méthode utilise-t-on pour s'assurer du « bon état » d'une User story ?
- Gherkin, c'est quoi ?
- Que fait le chef de projet dans Scrum ?
- Qu'est ce qu'un artefact ?
- Qui crée le sprint goal ?
- Quels sont les 2 grands axes du rôle de Scrum Master ?



Recueil des besoins

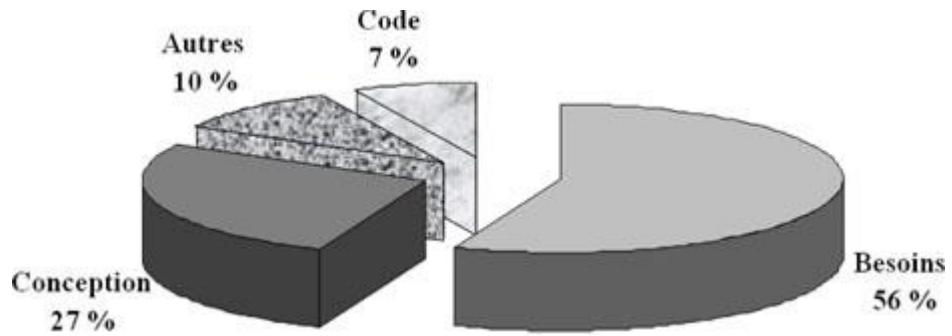
La description détaillée :

- L'analyse de l'existant
 - Forces/faiblesses de l'application
 - Quelles fonctionnalités améliorer/remplacer
- L'observation du comportement de l'utilisateur « en situation »

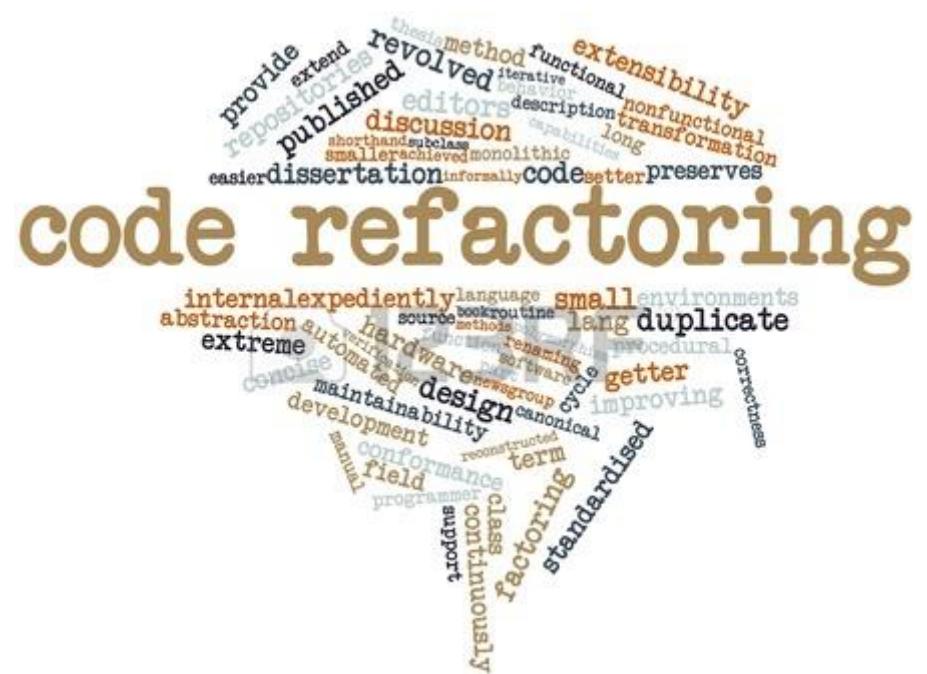
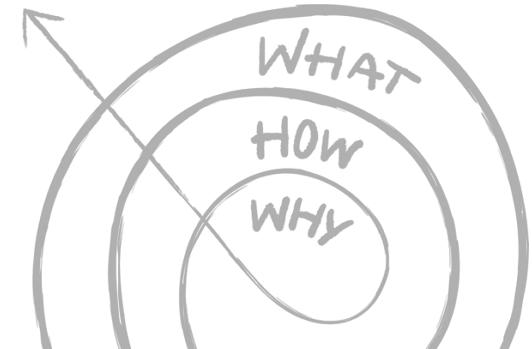


Recueil des besoins

- **Origines des défauts logiciels**

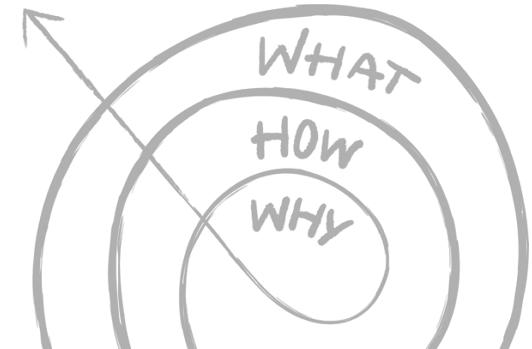
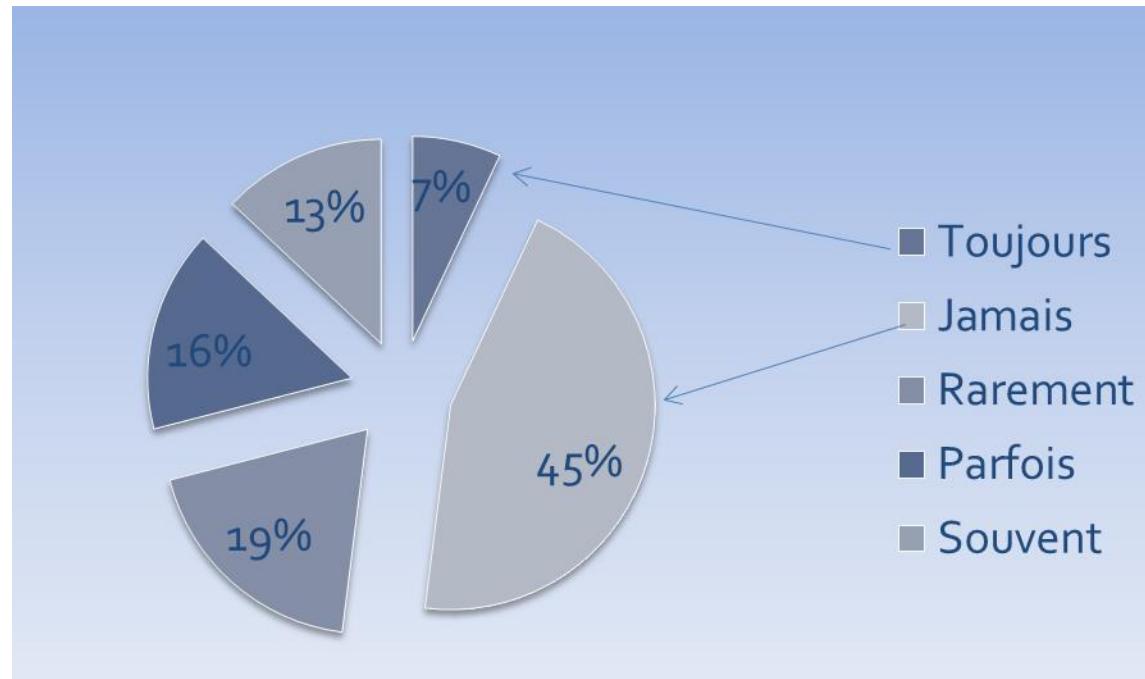


Source : http://librairie.immateriel.fr/fr/read_book/9782212136661/Chapter03



Définition des besoins

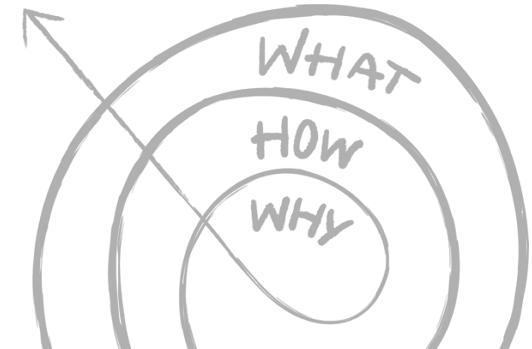
- Pourcentage de fonctionnalités implémentées réellement utilisées



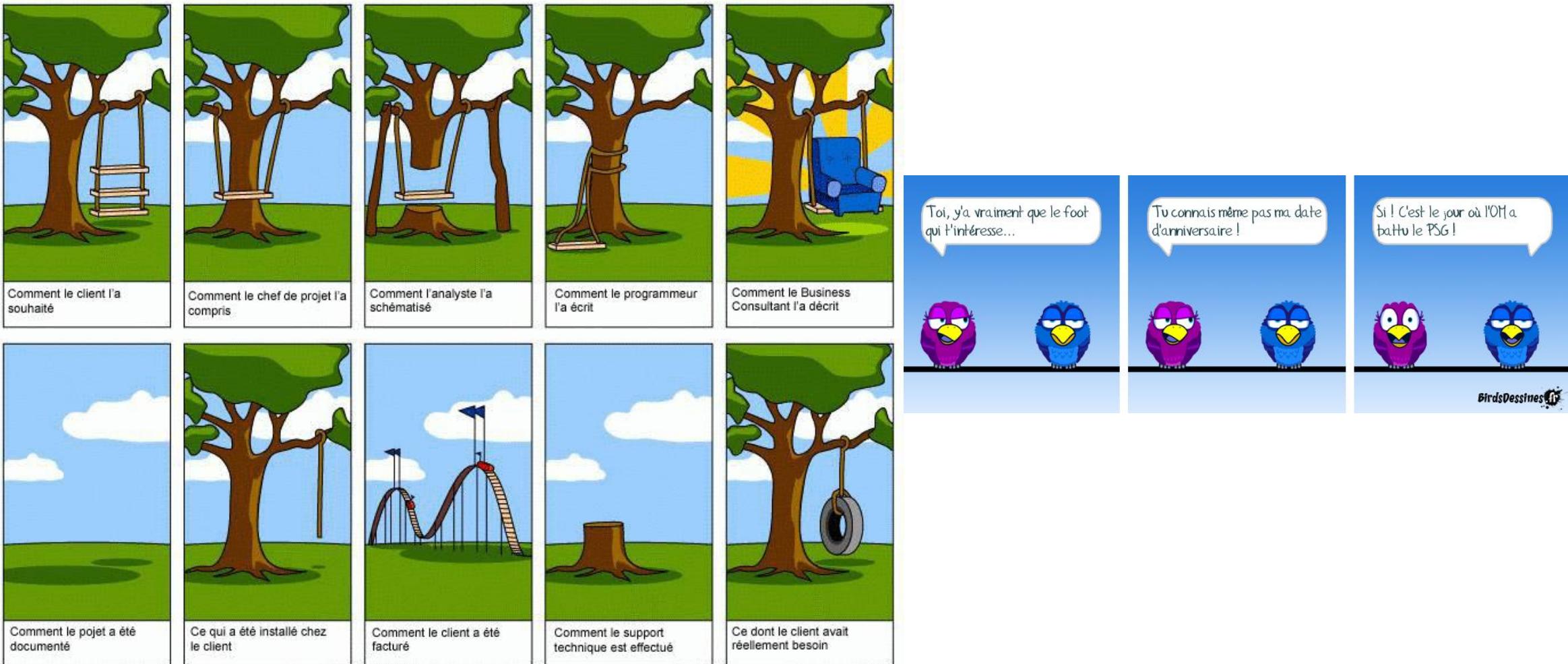
Définition des besoins

- Recueillir les besoins présentent 3 problèmes majeurs:

- Une mauvaise communication
- L'illusoire exhaustivité
- La défaillance du client

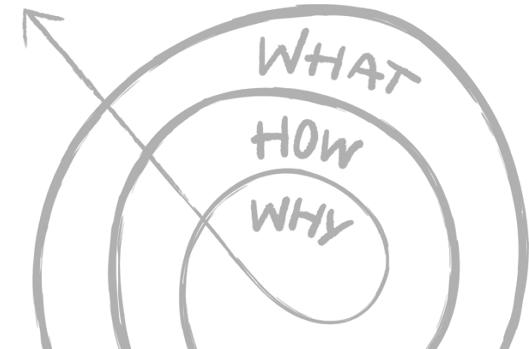
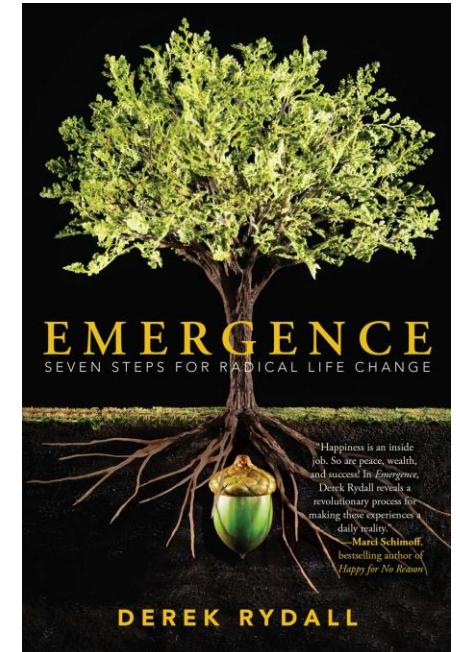


Produit et communication



LE PRODUIT : Faire émerger les besoins

- Après analyse, une idée (projet) peut s'avérer inutile, trop coûteuse, secondaire ou comportant trop de risques
- Il peut en résulter des priorités différentes entre les besoins
- Le marché peut être instable et l'organisation doit être réactive face à ses fluctuations
- L'introduction d'une nouvelle demande peut impliquer la renonciation d'un autre besoin



LE PRODUIT : Recueil des besoins

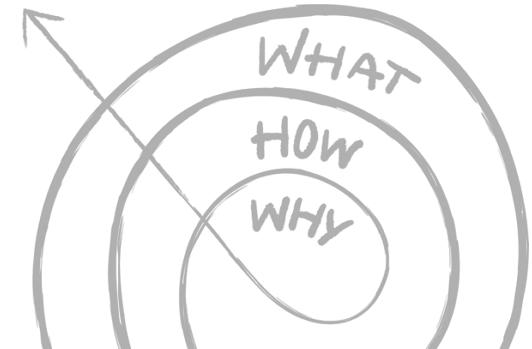
En amont du projet:

- Le brainstorming (Instancié à l'aide d'une mindmap)
- L'analyse de la concurrence (*benchmarking*)
- Différents Serious Games comme la méthode LEGO SERIOUS PLAY
- Le Story Telling
- ...

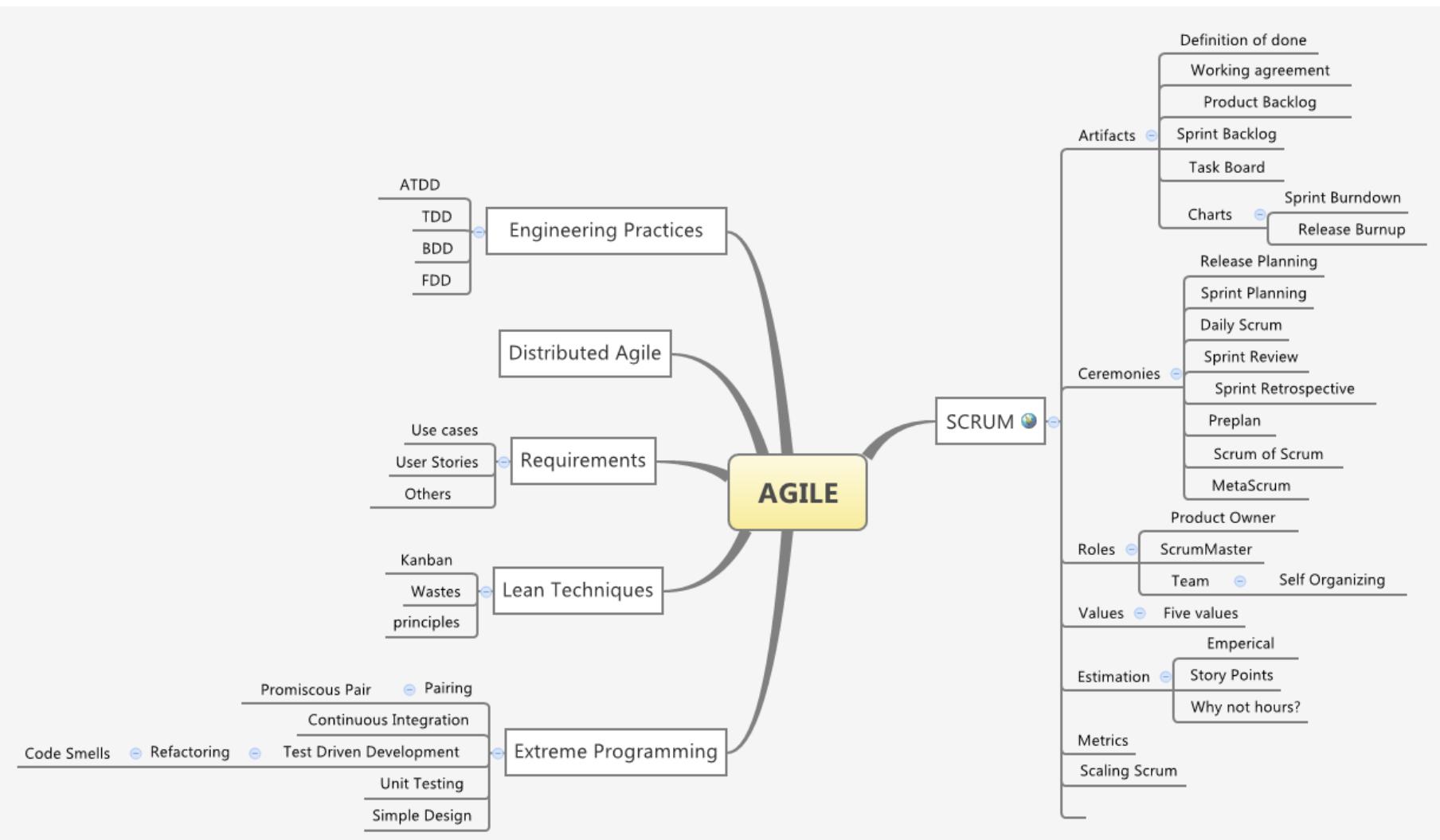
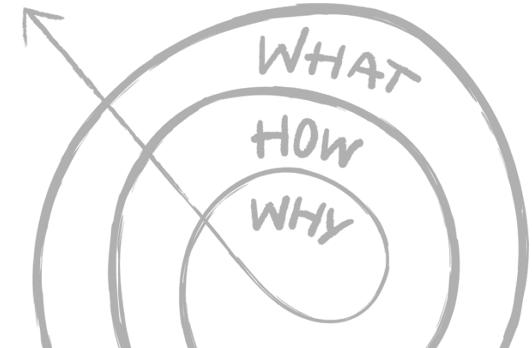


Affiner les besoins:

- Interviews (Questionnaires, croisement des résultats, ...)
- Workshops



Le mind Mapping



Recueil des besoins

Techniques d'interview : Les 9 cases de Solution Selling

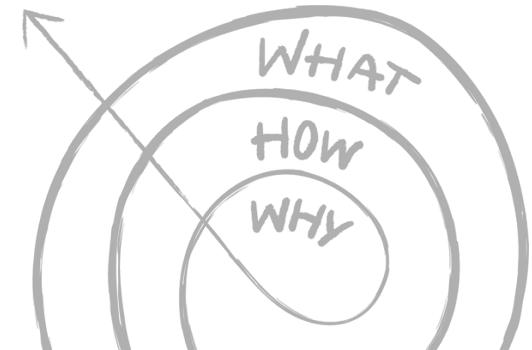
	Quel est le problème ?	Qui est impacté ?	Visualiser la solution
Questions ouvertes « Dites moi... » « Racontez moi... » « Et puis... »	?	?	?
Contrôle Combien ? Quand ? Où ?	?	?	?
Validation « Si je comprends bien... » Si non, revenir aux questions ouvertes. Si Oui, passer à la question suivante	?	?	?



Formaliser les besoins

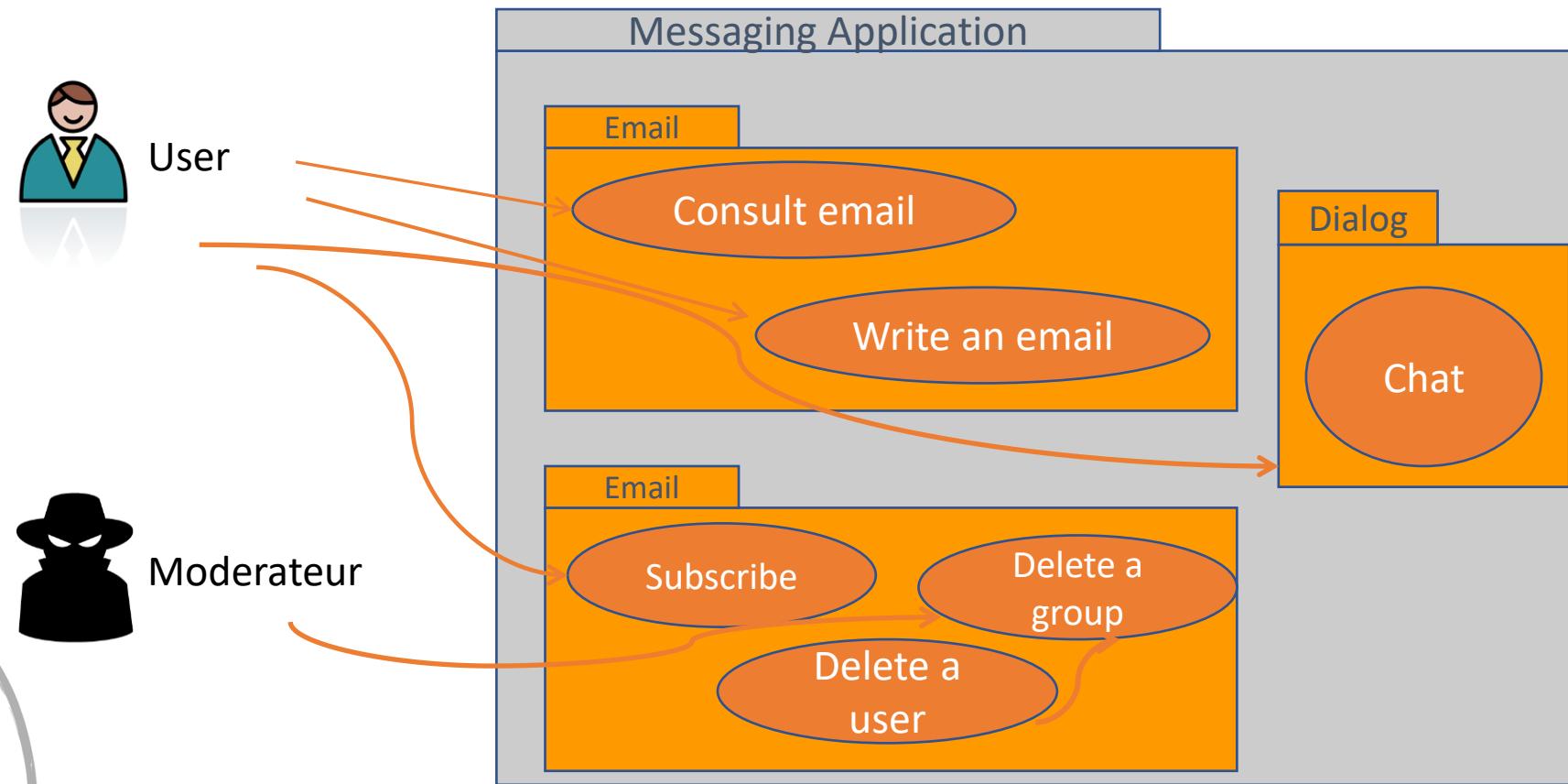
2 possibilités:

- **Use Cases (Cas d'utilisations)**
 - Modélisation UML
 - Description des échanges entre le système et les acteurs externes (utilisateurs ou autres systèmes) pour un contexte particulier
 - Description des différentes étapes pour les séquences:
 - nominales
 - d'exception
- **User Stories**
 - En tant que ..., je veux ... afin de ...



Formaliser les besoins

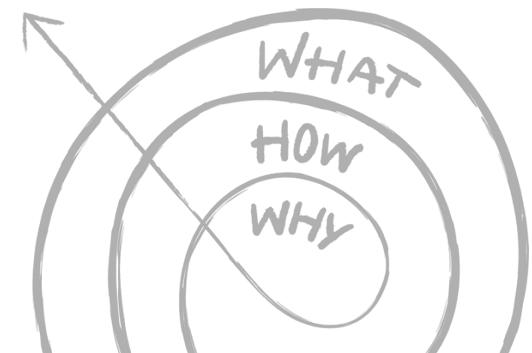
Diagramme de cas d'utilisation :



Préambule du Product Backlog

Use Case vs. User Story

User Story	Use Case
Brève description vue par l'utilisateur	Séquence d'action du système et ses interactions avec les autres acteurs
Format court, stimule la discussion orale	Format riche en information, peu de place à l'oral
Peut être une partie d'un Use Case	Somme d'un scénario
Utilisé pour la spécification et/ou la planification	Uniquement utilisé pour la spécification
Emergence rapide au travers d'ateliers collaboratifs	Long travail d'analyse et de formalisation
Grande visibilité	Visibilité réduite (due au volume)
Difficile à lier les unes aux autres	Liaison et vision globale



Plan de ce module

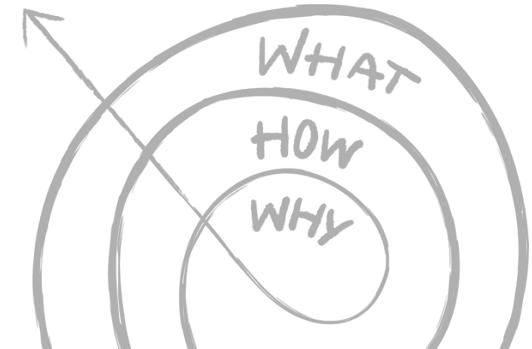
Voici les parties que nous allons aborder :

- **Partie 1 : Introduction & Scrum Basics**

- Les Rôles
- Le Processus
- Le Product Owner
- Le Scrum Master

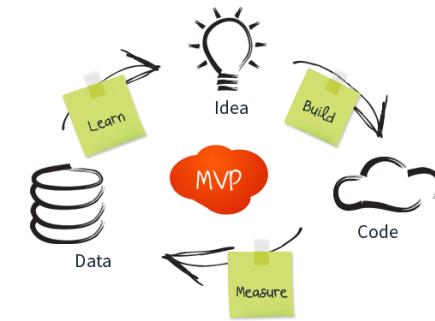
- **Partie 2 : Mise en œuvre**

- Le Produit
- **Le Product Backlog**
- Les Releases
- Les Sprints
- L'agilité à l'échelle



Préambule du Product Backlog

Mission MVP : Une question d'expert ? (La minute Agile)



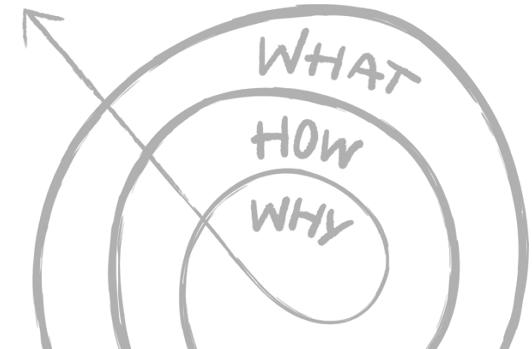
Préambule du Product Backlog

Mission MVP : Une question d'expert ?

- La version minimale d'un produit peut aussi en faire son succès.

Selon vous, quel produit web a été un véritable succès grâce à sa concentration sur les besoins essentiels des utilisateurs ? Et pourquoi ?

G . . . E

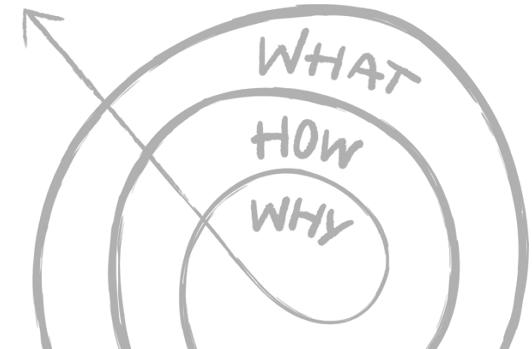
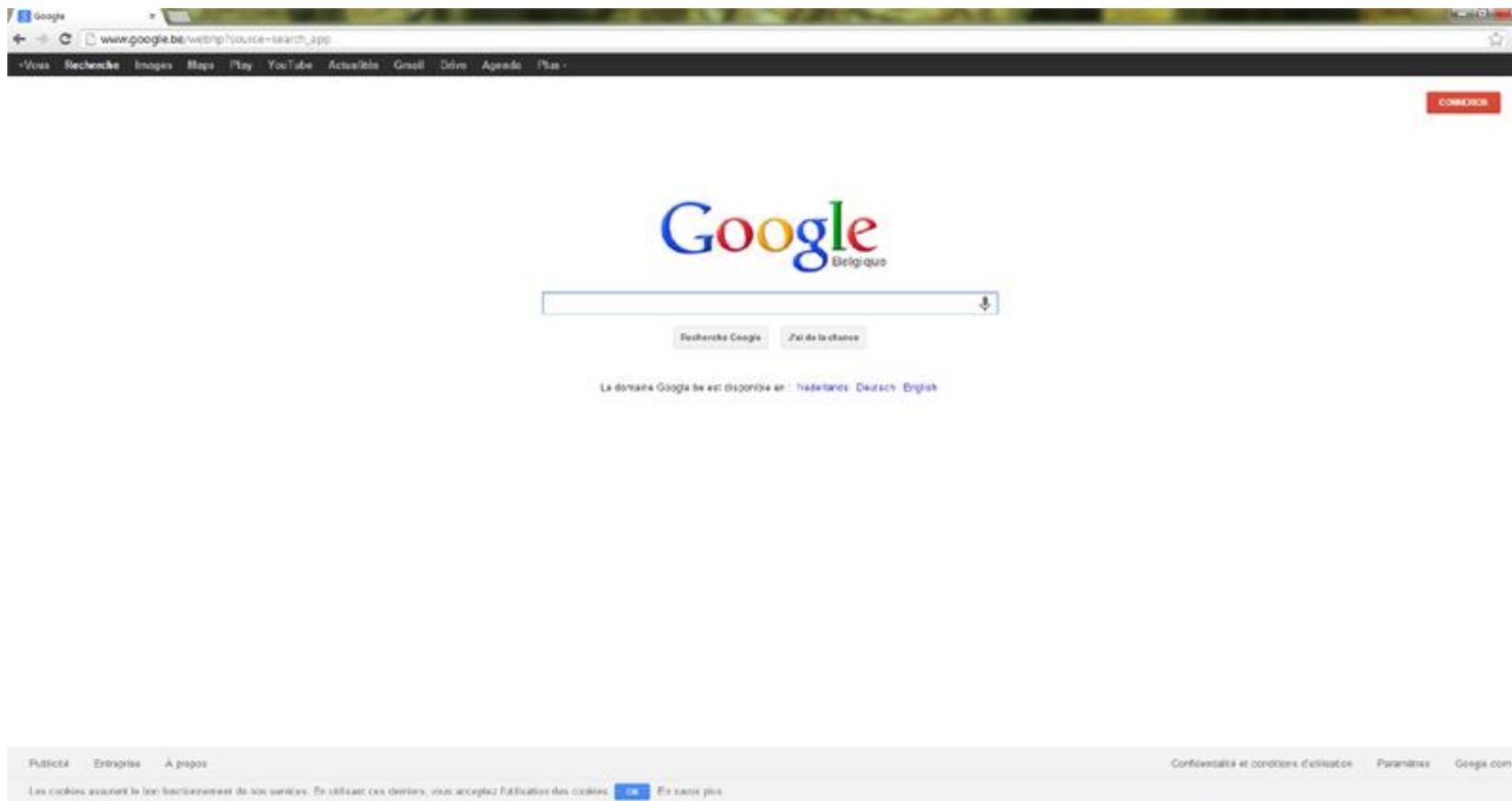


Préambule du Product Backlog

Mission MVP : Une question d'expert ?

Simplicité

Google.com

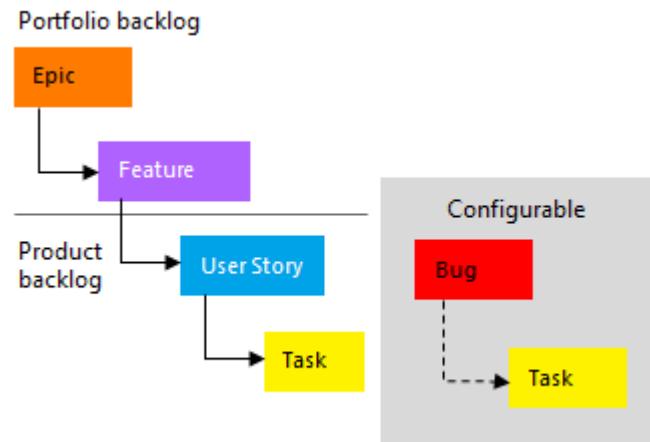


Préambule du Product Backlog

Un Product Backlog est constitué de différents “items”. Ils se différencient par leur niveau de granularité. Le plus macro est qualifié d'**Epic** qui peut être affiné en **Features** qui peuvent être affinées à leur tour en **Stories** (user, technical, bug).

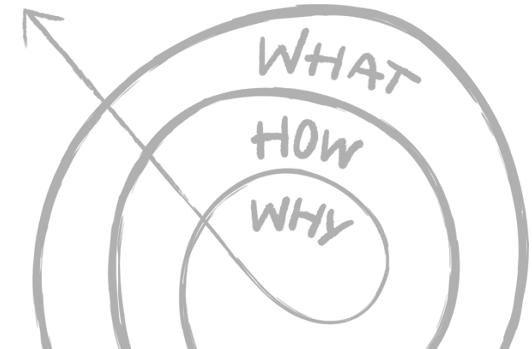
Attention !!!

En matière de bug, nous avons des bugs blocants qui seront traités en priorité. Si les bugs sont découverts par le client, nous parlerons de **défault négligé**.



Qualités d'une User story
(pour rappel - INVEST)

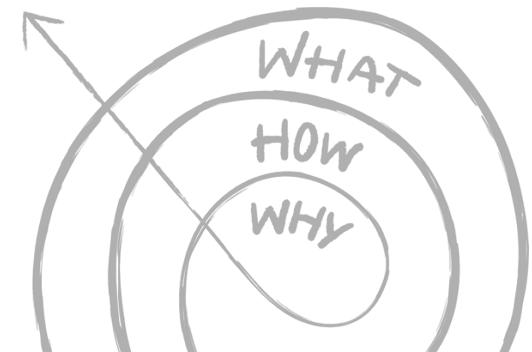
- INDEPENDENT
- NEGOCIABLE
- VALUABLE
- ESTIMABLE
- SMALL
- TESTABLE



Le Product Backlog

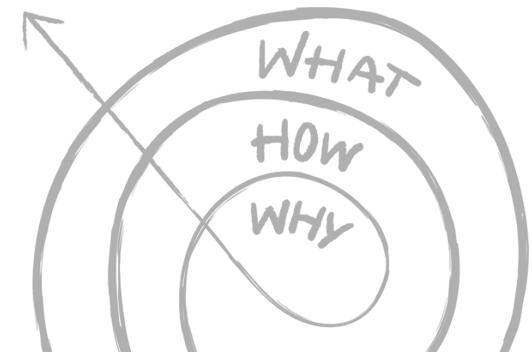
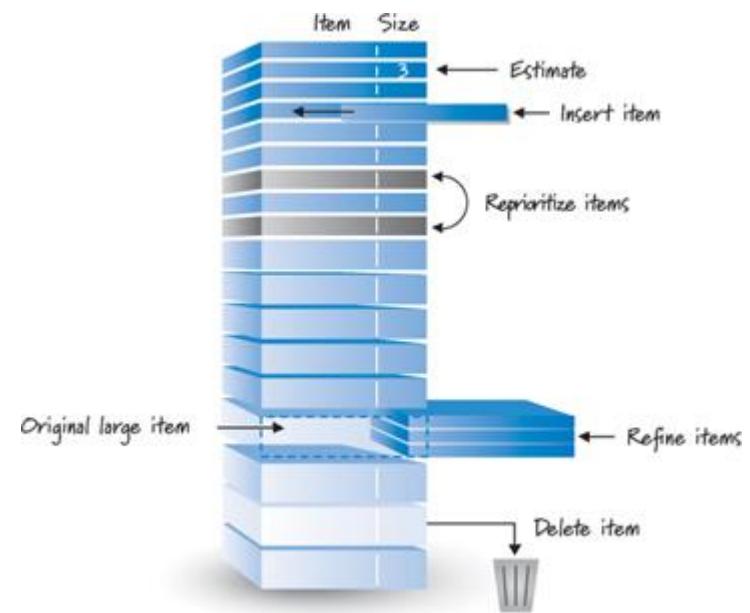
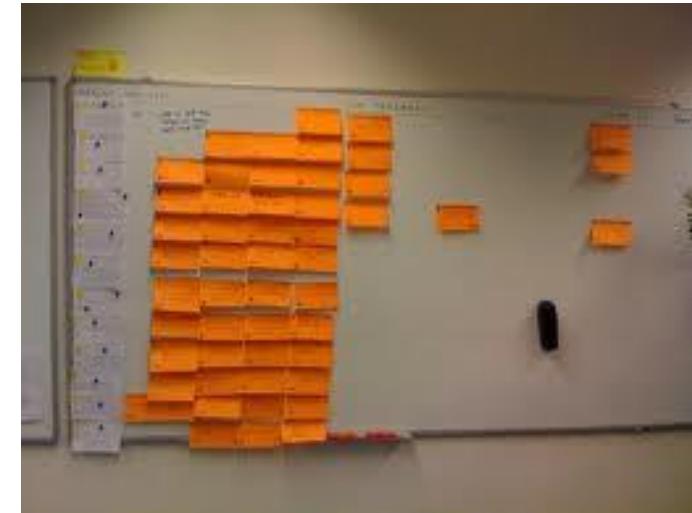
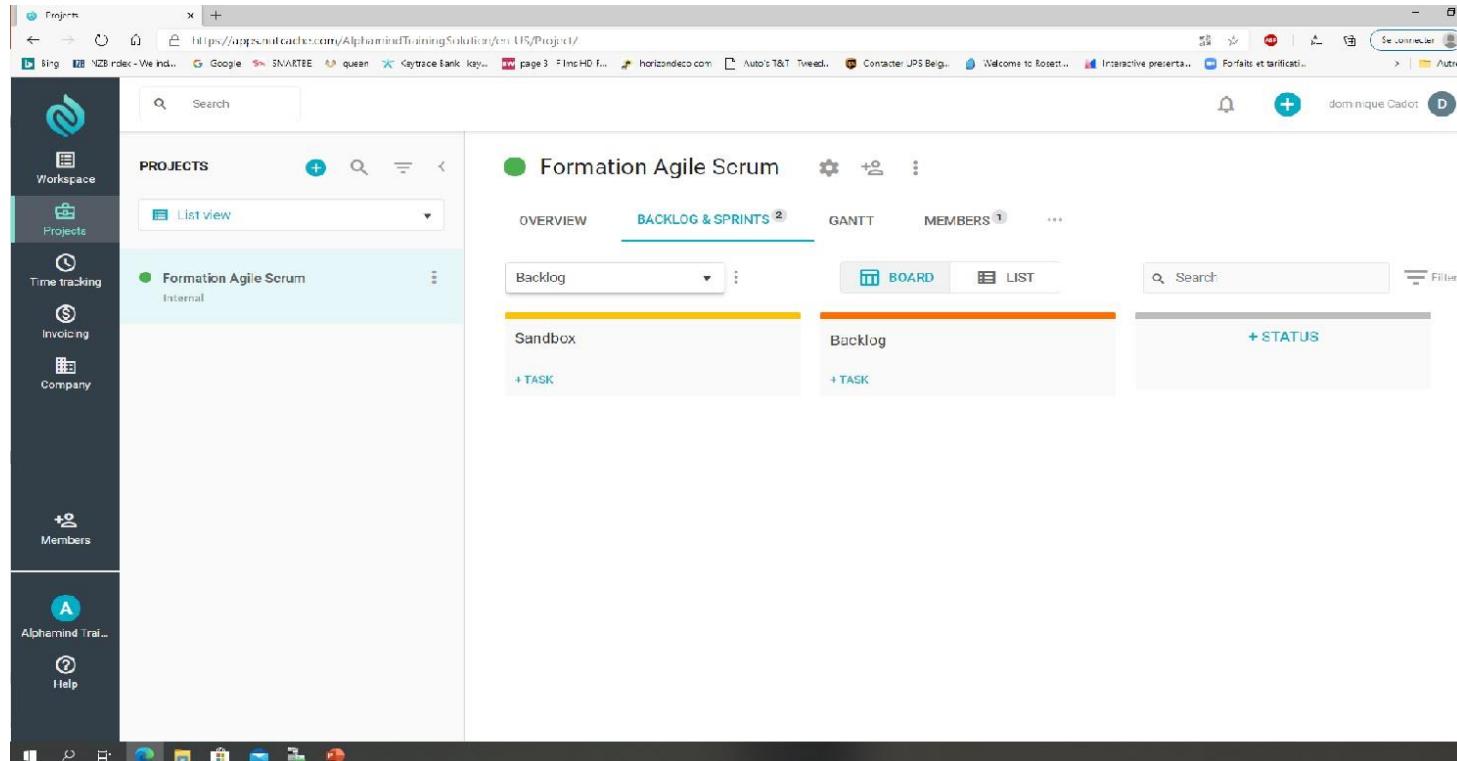
Mise en oeuvre du Product backlog

- **Les qualités du Product Backlog**
- **Entretenir le Product Backlog**
 - Décrire les éléments
 - Affiner les Epics et Macro users stories (Grooming ou refinement)
 - Prioriser le Product Backlog
 - Préparation pour le Sprint Planning
 - Estimation des éléments
- **Exigences non-fonctionnelles**
- **Erreurs courantes**



Le Product Backlog

Structure et outils.



Le Product Backlog

JIRA

Dashboards Projects Issues Agile Bonfire

GIVE GREENHOPPER FEEDBACK! Nicholas Muldoon Administration

+ Create Issue Quick Search

Angry Nerds

Quick Filters: Only My Issues Recently Updated

Sprint 1 Progress: 13 days left

Upcoming Sprint 1 Start Sprint

- NERD-1 As a Front-Ender I would like to stop supporting IE6 so I can enjoy my life
- NERD-20 As a Hacker I would like to hack the mainframe and enter the Matrix
- NERD-11 Hall of Fame
- NERD-6 As the Founder I want to have the last say so I can get my way

Issues 4 Estimate 16

Upcoming Sprint 2

- NERD-17 As an Agilista I want to iterate on the rate so I can rate the iteration
- NERD-7 As a Bug I want to make like hard for the Angry Nerds
- NERD-18 As the Founder I want to find the thing I found
- NERD-23 Specifically, an insect of the order Hemiptera, known as the "true bugs".
- NERD-10 Achievements Section
- NERD-25 Bacterium or any microorganism that causes illness and has a superficial resemblance to
- NERD-3 As the Dev Manager I would like to look busy so I can keep my job

Issues 7 Estimate 45

Backlog Add Sprint

- NERD-16 As an Agilista I want to be lean to make the green, quicker
- NERD-15 As an Outsourcer I want to work more hours and get less done
- NERD-24 An insect, spider or other small pest excluding rodents
- NERD-8 As a Bug I want to fly in the face of progress
- NERD-9 As a Bug I want to be like The Beatles so I can be fab

Angry Nerds / NERD-1
As a Front-Ender I would like to stop supporting IE6 so I can enjoy my life

Estimate: 3

Details Description (0) (0) (3) (2)

Status: In Progress 0/2 Completed

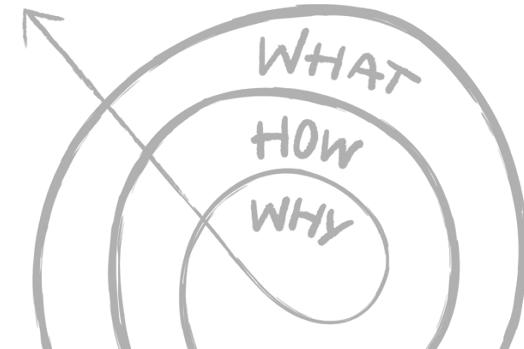
Session Status Actions

- Blitz Test Started
- Pre-Release Blitz Test Created

Create Session

Structure et outils.

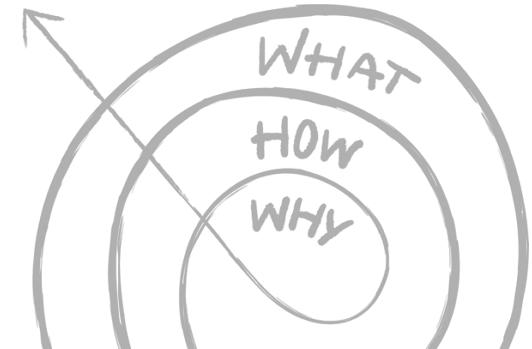
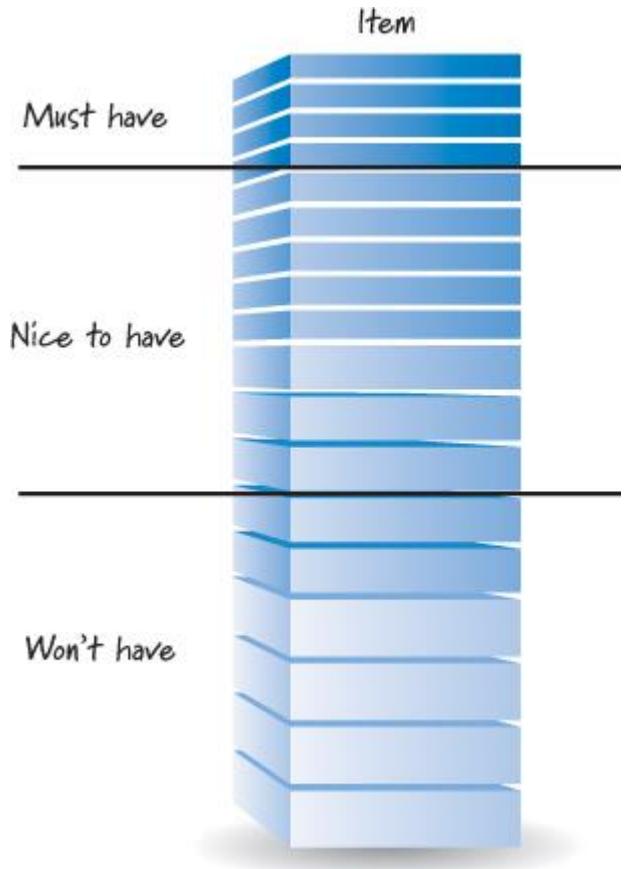
Gestion du Product Backlog
Avec Jira et GreenHopper



Le Product Backlog

Les qualités du Product Backlog

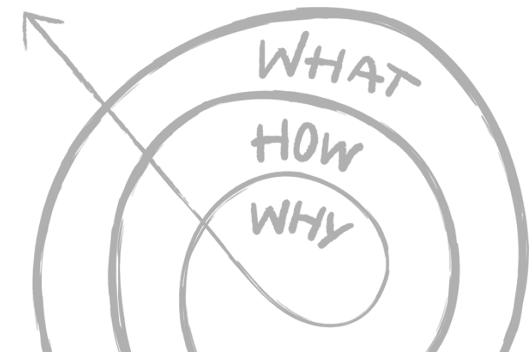
- 4 Qualités: **DEEP**
 - Detailed Appropriately (détailé de façon approprié)
 - Estimated (Estimé)
 - Emergent (émergent)
 - Prioritized (priorisé)



Le Product Backlog

Entretien le Product Backlog

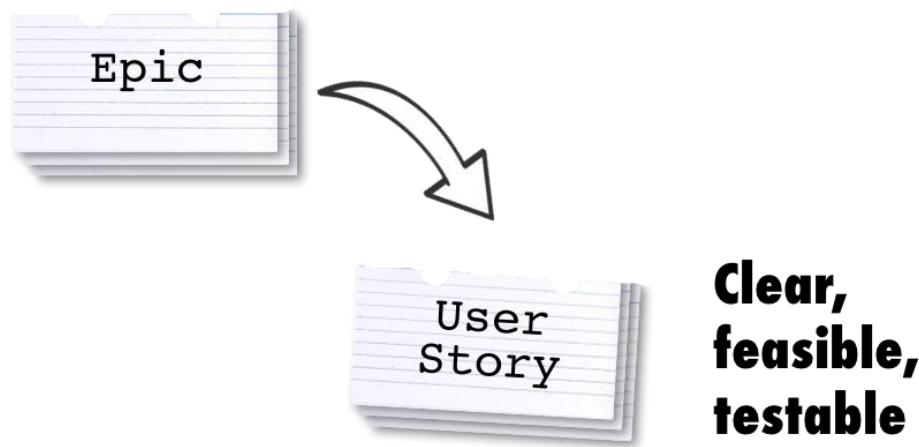
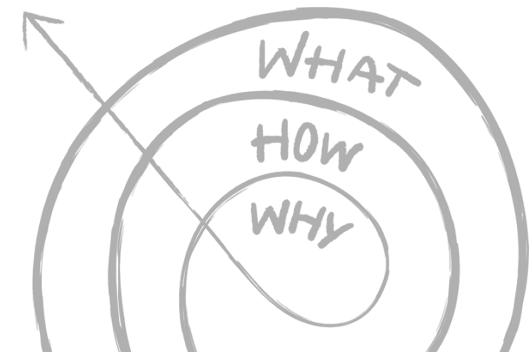
- **Processus continu comprenant:**
 - Ajout, modification, suppression d'éléments
 - Priorisation (plus important en haut)
 - Les éléments hautement prioritaires sont décomposés et raffinés en vue de la réunion du planning de Sprint
- **Processus collaboratif:**
 - Toute l'équipe Scrum y **participe**
 - Communication en face-à-face plutôt qu'au travers de documents
- **Responsabilité du Product Owner**



Le Product Backlog

Détailler le Product Backlog de manière appropriée

- **Les éléments du product backlog sont des user story**
 - *En tant que [rôle], je veux [action] afin de [but]*
- **Exemple:**
 - As a user closing the application, I want to be prompted to save anything that has changed since the last save so that I can preserve useful work and discard erroneous work.

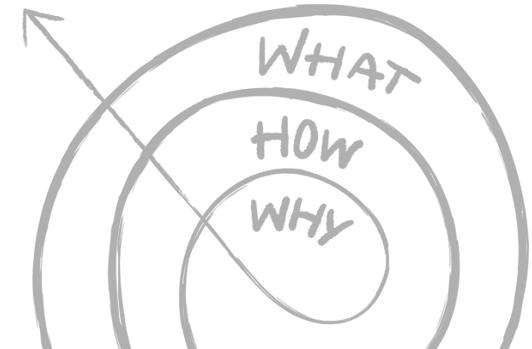
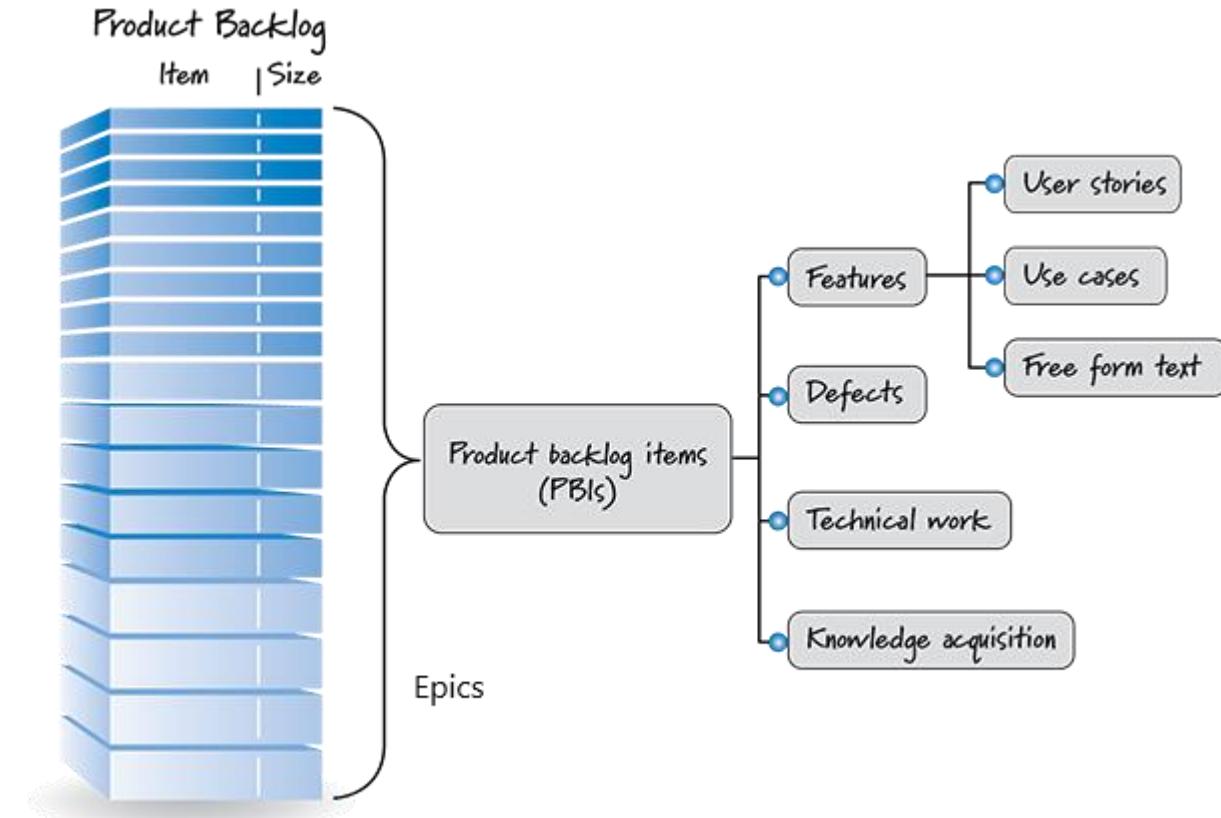


**Clear,
feasible,
testable**

Le Product Backlog

Détailler le Product Backlog de manière appropriée

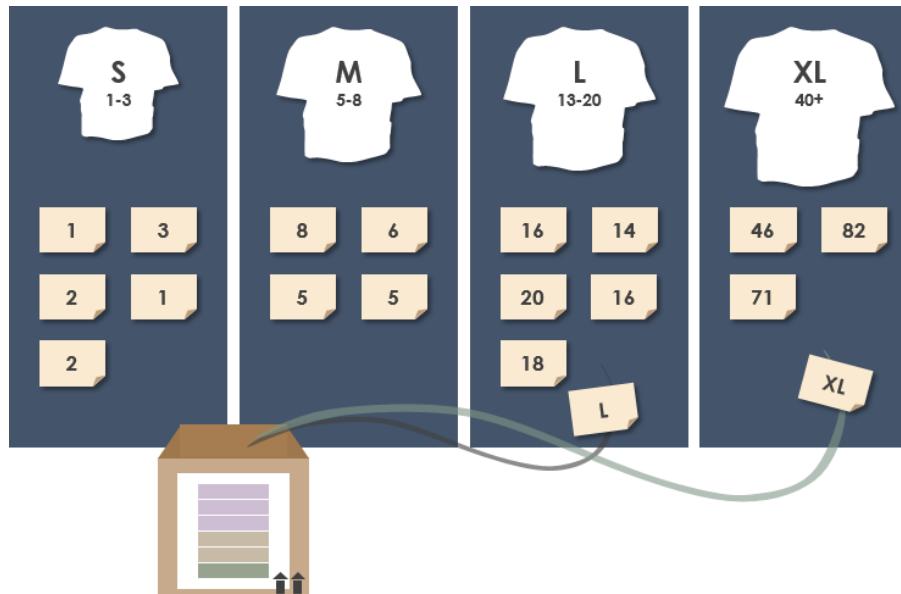
- Les éléments les plus prioritaire sont les éléments les plus détaillés.
- Implique que les exigences sont décomposées et raffinées durant l'entièreté du projet.



Le Product Backlog

Estimer les éléments

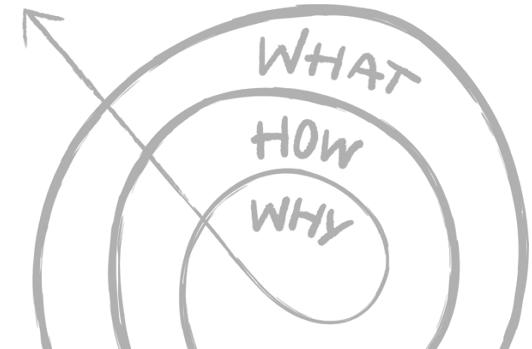
- Les éléments du Product Backlog sont généralement pré-estimé en taille de T-shirt afin d'évaluer leur complexité brute.
- Il faut prendre en considération que connaître la charge de travail permet de prioriser les éléments



Le Product Backlog

Priorisation du Product Backlog selon :

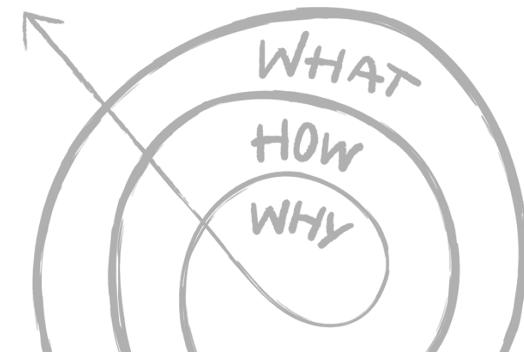
- **Bénéfice financier attendu**
 - « Quel est le bénéfice financier à développer cette fonctionnalité ? »
- **Coût de développement estimé**
 - « Quel est le coût de développement ? De maintenance ? De formation des utilisateurs ? »
 - « Quel est le coût d'un changement ? »
- **Opportunité d'apprentissage pour l'équipe**
 - « L'implémentation de cette fonctionnalité nous permet-elle d'apprendre ou de développer de nouvelles compétences ? »
 - « Cette fonctionnalité va-t-elle améliorer la productivité de mon personnel ? »



Le Product Backlog

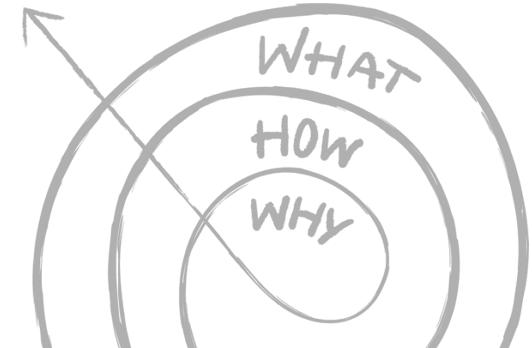
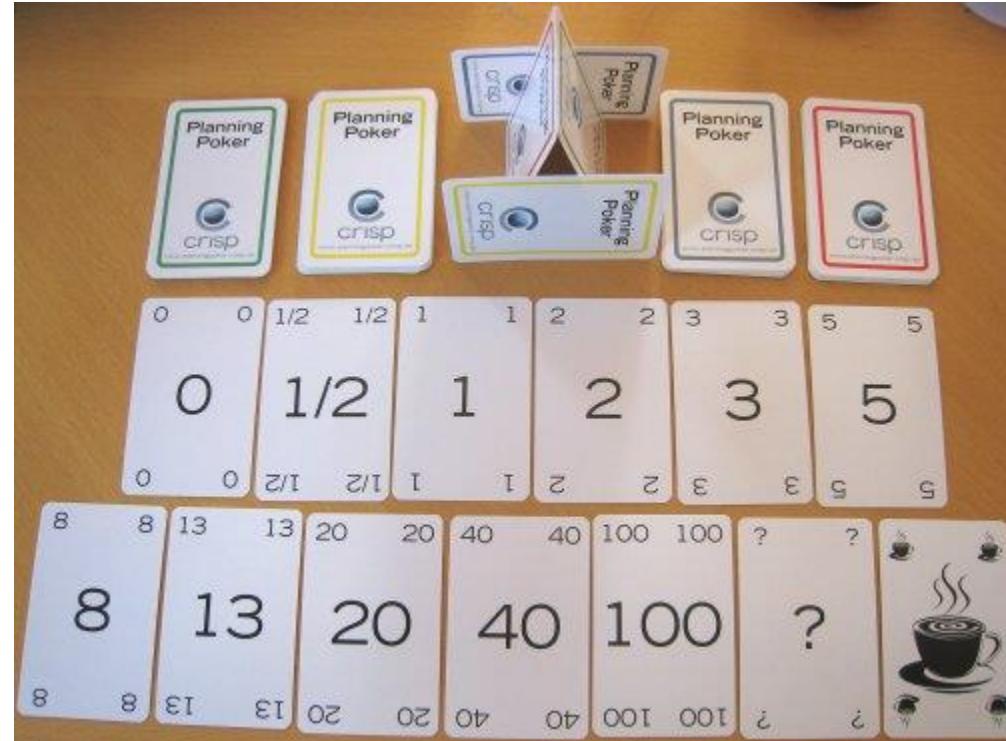
Priorisation du Product Backlog selon :

- **Le risque de développement**
 - « Cette fonctionnalité nous expose-t-elle à davantage de risques ?
Ou, au contraire, nous permet-elle de nous confronter au risque ? »
 - « Cette fonctionnalité va-t-elle impacter les processus métier ?»
 - « Cette fonctionnalité va-t-elle susciter de la frustration ou de la résistance ? »
 - « Quel est le préjudice si cette fonctionnalité n'est pas implémentée ? »



Les Stories

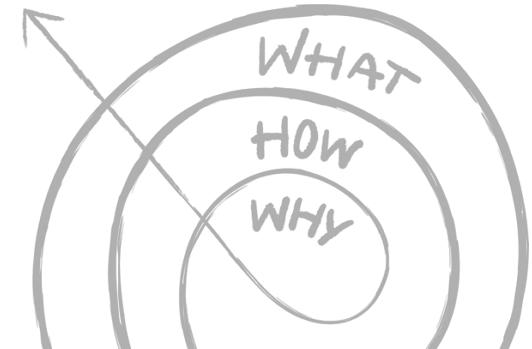
Le Planning Poker : un outil d'estimation pour l'effort



Les Stories

Le Planning Poker : un outil d'estimation pour l'effort

- **Estimation**
- **Estimation Agile**
- **Planning Poker**
 - Préparation de la réunion
 - La réunion
 - Variations
 - Exemple
 - Avantages et inconvénients

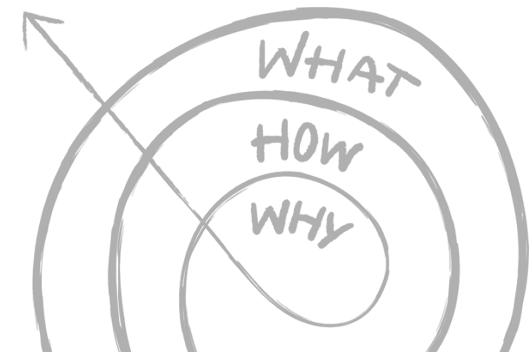


Les Stories



Le Planning Poker : un outil d'estimation pour l'effort

- L'estimation est une des parties plus difficiles d'un projet de logiciel.
- Quelques données:
 - Presque $\frac{2}{3}$ de projets dépasse significativement leurs estimations de coût.
 - 64 % des fonctionnalités incluses dans les produits sont rarement ou jamais utilisées.
 - Un projet moyen peut dépasser sa grille-horaire de 100%.

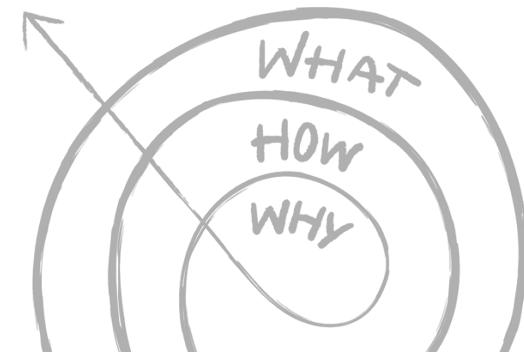


Les Stories



Le Planning Poker : un outil d'estimation pour l'effort

- **Estimation Agile**
 - 2 niveaux d'estimation
 - Estimation de la version
 - Estimation de l'itération
 - Concentré en fonctionnalités et pas en activités
 - Tous les membres de la Dev Team aident à l'estimation.
 - Seul les membres de la Dev Team ont le dernier mot sur les estimations



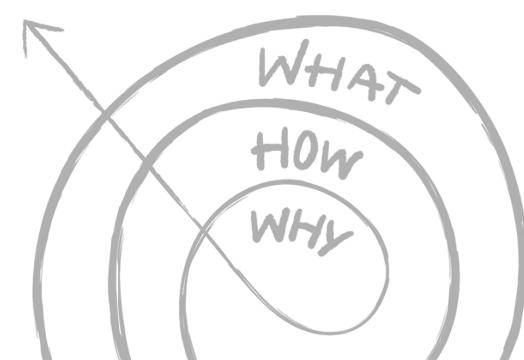
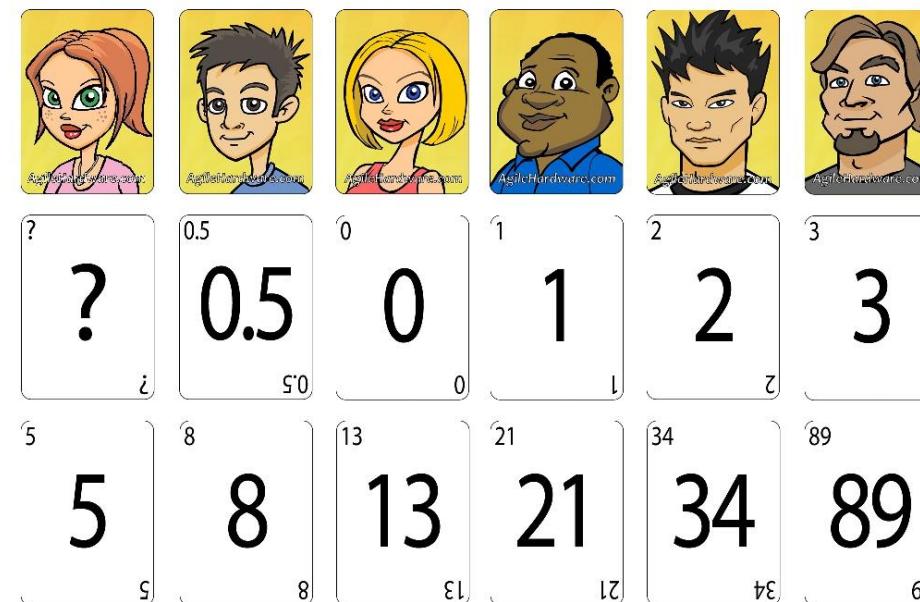
Les Stories

Le Planning Poker : un outil d'estimation pour l'effort



Le planning Poker en question

- Cette méthode essaie de faire des réunions plus courtes et plus productives, en les rendant plus amusantes et dynamiques.



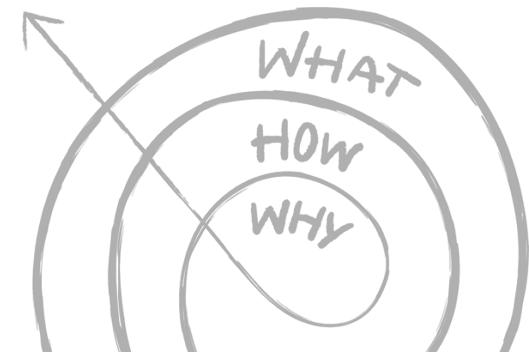
Les Stories

Le Planning Poker : un outil d'estimation pour l'effort



Préparation de la réunion

- Les «exigences d'experts » doivent connaître parfaitement chacune des User stories.
- Chaque « user story » devrait avoir une granularité de pas plus de 10 jours de travail.
- Un jeu de cartes est préparé pour chaque membre de l'équipe.
- Le jeu est composé de quelques cartes, chacune d'elles représentant une estimation.



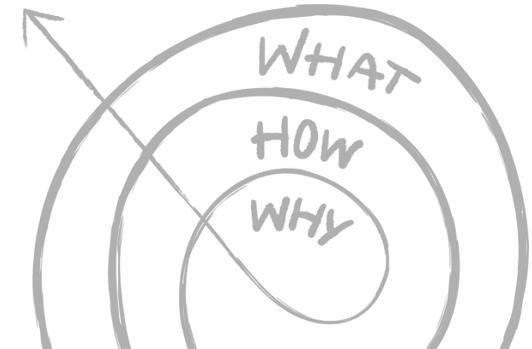
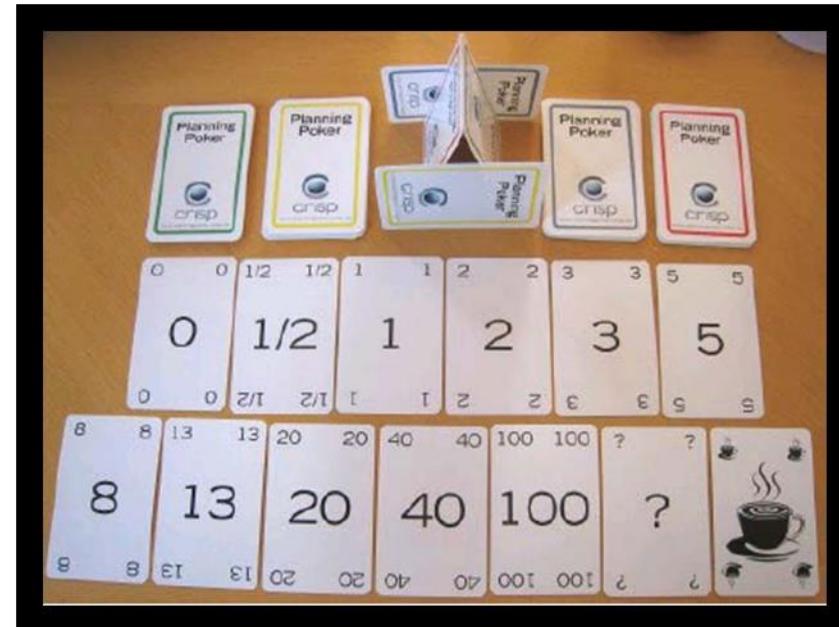
Les Stories



Le Planning Poker : un outil d'estimation pour l'effort

Préparation de la réunion

- Exemples de valeurs d'estimation pour les cartes :
- 0, 1, 2, 3, 5, 8, 13, 20, 40, 100.
- 1, 2, 3, 5, 8, 13, BIG.
- $\frac{1}{2}$, 1, 2, 3, 4, 5, 6, 7, ∞



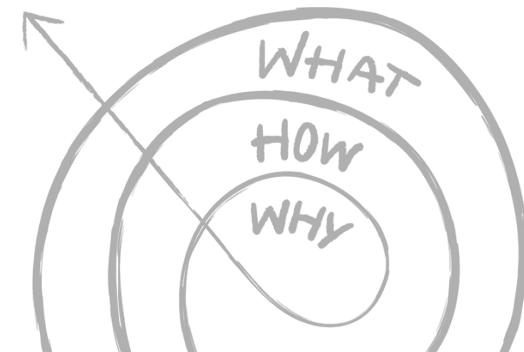
Les Stories

Le Planning Poker : un outil d'estimation pour l'effort



La réunion

- Un jeu est donné à chacun des membres.
- L'animateur expose une « user story » en pas plus de 2 minutes.
- L'animateur octroie du temps pour des questions sur la « user story ».
- Chacun des membres choisit une carte sans la montrer.
- Une fois que tout le monde a choisi, toutes les cartes sont montrées **en même temps**.
- Dans ce premier tour, il est probable que les estimations diffèrent sensiblement.



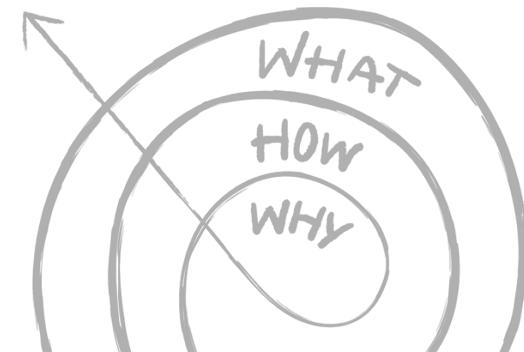
Les Stories

Le Planning Poker : un outil d'estimation pour l'effort



La réunion

- Dans le cas où les estimations diffèrent, les estimateurs de hautes et basses valeurs exposent leurs raisons.
- Quelques minutes pour l'équipe discuter de la story et de l'estimation.
- Encore une fois, chaque membre établi discrètement une estimation, et ils montrent les cartes simultanément.
- Si les estimations sont toujours différentes, le même processus peut être répété.

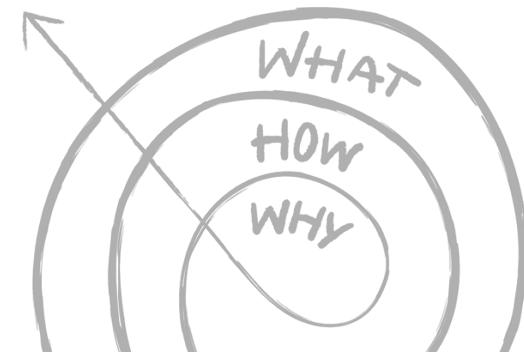




Le Planning Poker : un outil d'estimation pour l'effort

La réunion

- Lorsque les estimations convergent, le processus se termine et on estime la prochaine « user story ».
- Dans le cas où les estimations ne convergent lors de la 3ème Manche, il y a quelques options :
 - Mettez la “User Story” sur le côté et recommencez plus tard
 - Demandez au Product Owner de décomposer la story en parties plus petites..
 - Prendre l'estimation la plus élevée, la plus basse ou faire la moyenne.



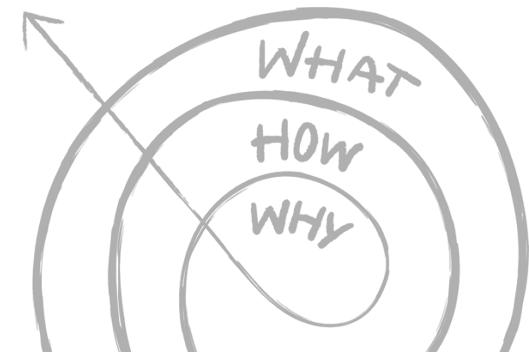
Les Stories



Le Planning Poker : un outil d'estimation pour l'effort

Variations

- Bien sûr, c'est une méthode ouverte.
- Il y a quelques variations qui peuvent être appliquées :
 - Utiliser 2 cartes au lieu de 1.
 - Demander à n'importe quel membre son avis sur son estimation, pas nécessairement les plus élevées et les plus basses.
 - Utilisez plus ou moins tours.
 - ...



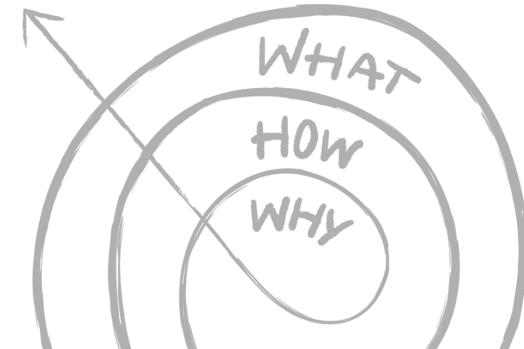
Les Stories

Le Planning Poker : un outil d'estimation pour l'effort



Exemple

- User story: En tant que utilisateur, je veux encoder la liste de mes clients afin de les utiliser ultérieurement.
- Équipe de 7 membres.
- Premier tour :



Les Stories



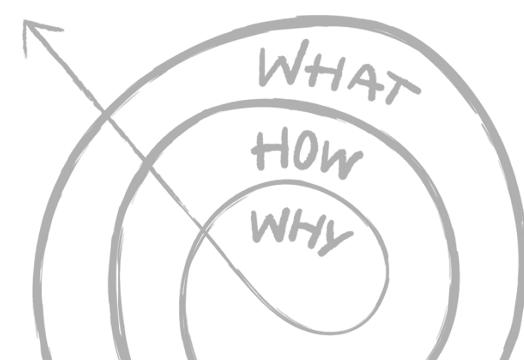
Le Planning Poker : un outil d'estimation pour l'effort

Exemple



3ème et 6ème membres exposent leurs motifs, leurs estimations.

2nd tour:



Les Stories

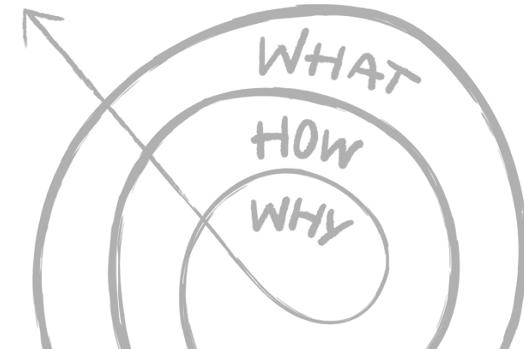


Le Planning Poker : un outil d'estimation pour l'effort

Exemple



- Tous les membres ont convergé à l'exception du 3^{ème}
- Un nouveau cycle d'expositions et de vote est possible.
- Il est également possible de prendre 3 ou 5 comme l'estimation.

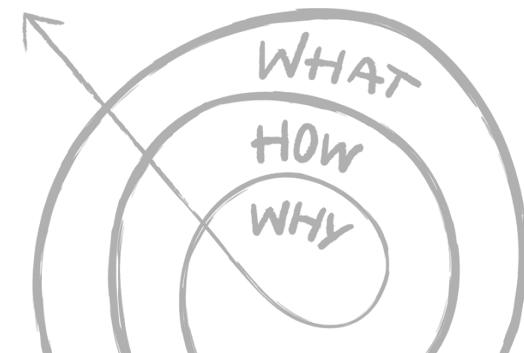




Le Planning Poker : un outil d'estimation pour l'effort

Avantages

- Plusieurs avis d'experts.
- Le dialogue entre les membres permet des estimations plus précises.
- Des études ont montré qu'en moyenne les estimations et discussions de groupe mènent à de meilleurs résultats.
- C'est amusant !



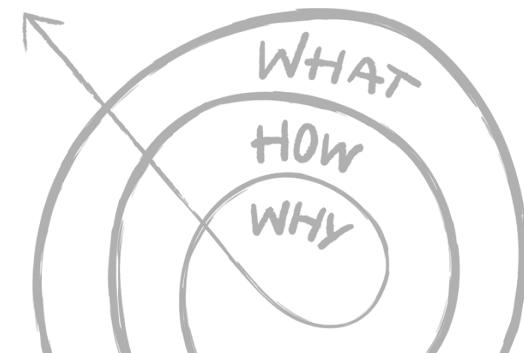
Les Stories



Le Planning Poker : un outil d'estimation pour l'effort

Inconvénients

- Une réunion avec toute l'équipe coûte cher.
- Le modérateur doit être prudent et doit contrôler la réunion, donc elle ne doit pas être trop longue.
- Certains facteurs peuvent interférer dans les estimations : personnalités dominantes, la politique de l'entreprise...
- Des discussions peuvent finir dans les estimations polarisées.



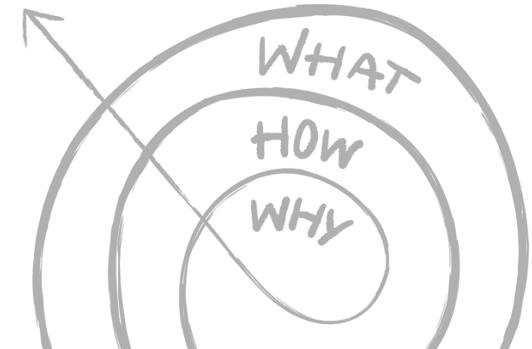
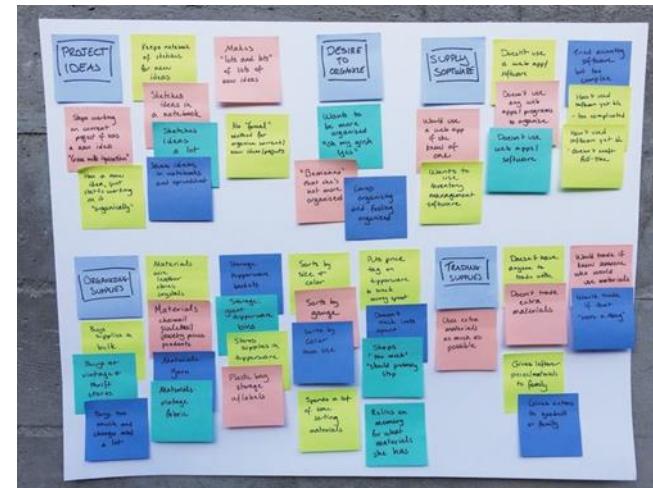
Les Stories



Estimations par affinités

Les fonctionnalités ici sont regroupées par affinités, c'est-à-dire qu'elles présentent des similarités entre-elles, qui doivent être évaluées. Une fois les fonctionnalités classées, les regroupements peuvent être associés à une estimation numérique, ou être utilisés tels quels.

L'estimation peut être complexe dans la mesure où elle demande une analyse fine des fonctionnalités pour pouvoir faire des recouplements. Cette méthode d'estimation est donc plutôt adaptée à un petit nombre de fonctionnalités, entre 20 et 50 maximum.



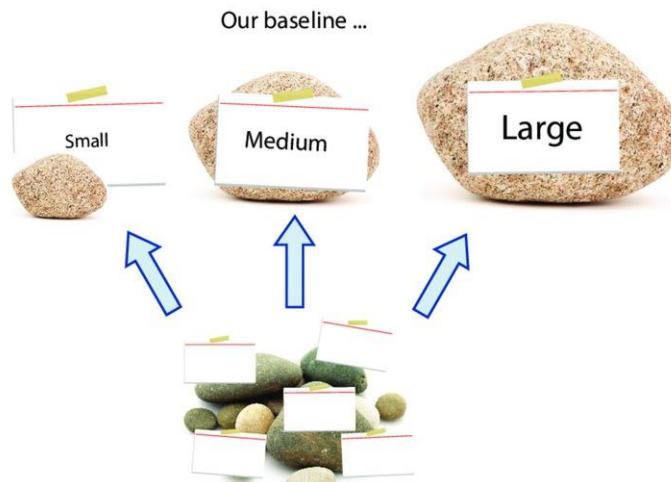
Les Stories



Estimations par Triangulation

La triangulation des stories base entièrement l'estimation des nouvelles stories sur les estimations des stories précédentes dans votre carnet de commandes. Il est incontestable que la nouvelle story est/soit plus facile, plus difficile, ou le même effort que les stories précédentes dans votre Backlog. Par conséquent, vous disposez de nombreux points de référence pour la comparaison.

Par exemple, si la nouvelle story a une taille, une complexité et un risque similaires à une story précédente évaluée à 8 points d'effort, alors la nouvelle story doit être estimée à 8 points d'effort. Si elle est environ 50% plus difficile qu'une story à 2 points précédente, alors la nouvelle story doit être estimée à 3 points d'histoire.



against which we size our remaining stories

Individually Licensed to Abdelatif CHRAIBI

Les Stories



Estimations en jour idéal

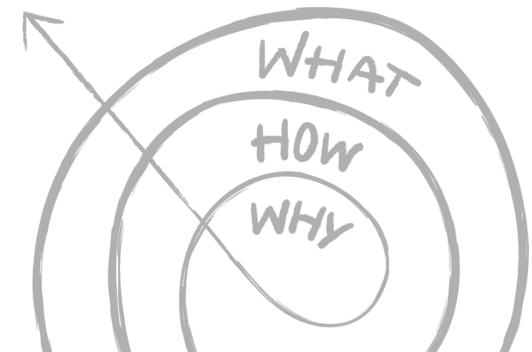
Pour comprendre le concept, je vous propose l'exercice suivant :

Une Story a été estimée à 8 heures idéales. Nous considérons que les développeurs sont actifs 6h sur une journée de 8 heures. Quel est le temps nécessaire pour terminer la story ?

1,25 jours

1,33 jours

2 jours



Prioriser les items

Le MoSCoW

La méthode **MoSCoW** est une technique visant à prioriser des besoins ou des exigences en matière d'assistance à maîtrise d'ouvrage et de développement logiciel. L'objectif est que le maître d'œuvre et le maître d'ouvrage s'accordent sur l'importance des tâches à réaliser par rapport aux délais prévus.

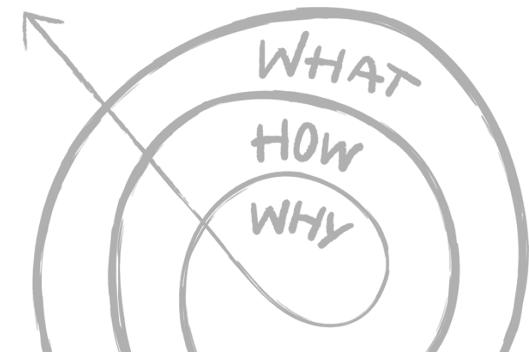
Les lettres majuscules de l'acronyme **MoSCoW** signifient (en anglais) :

M - MUST have this, c'est-à-dire 'DOIT être fait' (Vital).

S - SHOULD have this if at all possible, c'est-à-dire 'DEVRAIT être fait dans la mesure du possible' (Essentiel).

C - COULD have this if it does not affect anything else, 'POURRAIT être fait dans la mesure où cela n'a pas d'impact sur les autres tâches' (Confort).

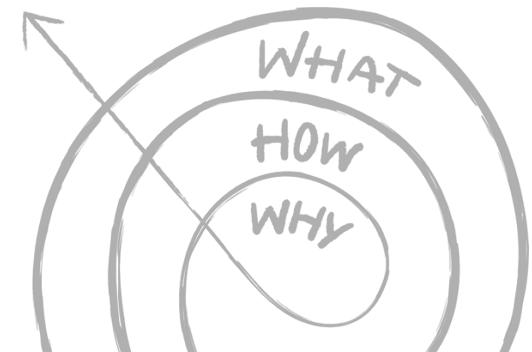
W - WON'T have this time but WOULD like in the future, 'NE SERA PAS fait cette fois mais sera fait plus tard' (Luxe, c'est votre zone d'optimisation budgétaire).



Prioriser les items

Le MoSCoW

- 1.Les utilisateurs peuvent se connecter au logiciel.
- 2.Les utilisateurs doivent pouvoir demander un nouveau mot de passe par email s'ils l'ont oublié.
- 3.Les utilisateurs peuvent créer des tâches.
- 4.Un utilisateur peut envoyer un email au logiciel et cet email sera attaché à la bonne tâche.
- 5.Lorsqu'un utilisateur clique sur un n° de téléphone enregistré dans le logiciel, le numéro est automatiquement composé.



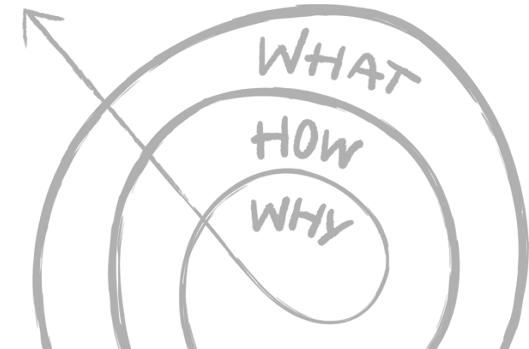
Prioriser les items

Le MoSCoW

- **Must** : il est obligatoire de s'authentifier compte tenu de la politique sécurité du client
- **Should** : il est fortement souhaitable de pouvoir être capable de se reconnecter à l'application, mais cela ne fait pas pour autant partie du chemin critique des fonctionnalités de l'application.
- **Must** : c'est la raison d'être d'un logiciel de "Gestion de tâches" !
- **Could** : fonctionnalité intéressante mais qui reste du domaine du confort... on peut travailler sans.
- **Won't** : excellente idée mais qui pourrait être traitée dans une version ultérieure de l'application.

Il est normal d'avoir des avis différents car il vous manque l'essentiel pour prioriser : **la valeur pour le client**

Source : http://fr.wikipedia.org/wiki/M%C3%A9thode_MoSCoW



Prioriser les items

La définition du DONE

- Les critères sur la définition du « done » sont définis avant le premier Sprint par :
 - Le Product Owner
 - Le Scrum Master
 - La Development Team

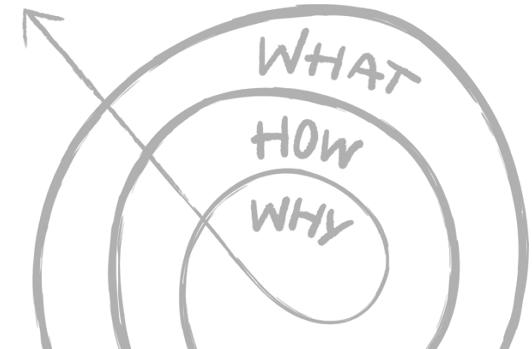
Typiquement, un élément du produit backlog est considéré comme achevé lorsque :

- il est implémenté
- il est testé
- il est documenté



Definition Of Done

	Développement	Migration des données (structures + données)
	Support IE7 + FF3	Test Seleniums écrits
	Support IE6	Test Seleniums passé avec succès
	Support "Navigateurs Home Page"	Test Unitaires écrits
	Déployé sur Staging	Test Unitaires passé avec succès
	Tests de régression ok (tous les tests passent)	Multilingue et traduction ok
	Documentation (dossier d'hébergement,...)	Démarches à effectuer auprès de l'infrastructure (pour la Prod ou autres. Ex: url, connexion db,ftp,...)
	Dépendance avec d'autres acteurs	Visualiser sur le mur



Prioriser les items

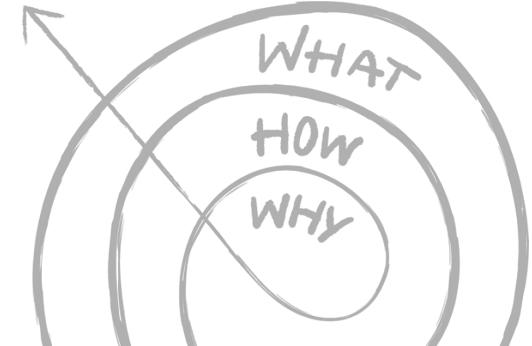
La définition du DONE

- On distingue 3 niveaux de responsabilité concernant la définition du Done.
 1. Création de la DoD : L'entièrre organisation de développement via les 3 rôles Scrum.
 2. Responsabilité de la DoD une fois créée : comme elle représente le travail attendu; c'est la Scrum Team qui en est responsable.
 3. Validation via la DoD : Le product Owner



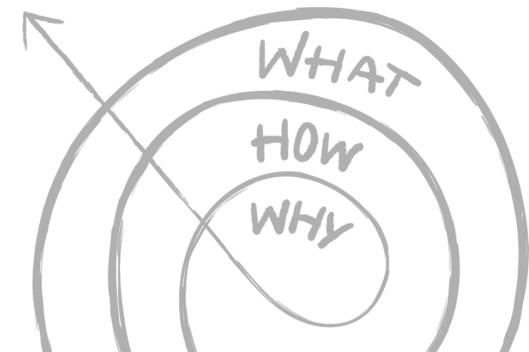
Definition Of Done

	Développement	Migration des données (structures + données)
	Support IE7 + FF3	Test Seleniums écrits
	Support IE6	Test Seleniums passé avec succès
	Support "Navigateurs Home Page"	Test Unitaires écrits
	Déployé sur Staging	Test Unitaires passé avec succès
	Tests de régression ok (tous les tests passent)	Multilingue et traduction ok
	Documentation (dossier d'hébergement,...)	Démarches à effectuer auprès de l'infrastructure (pour la Prod ou autres. Ex: url, connexion db,ftp,...)
	Dépendance avec d'autres acteurs	Visualiser sur le mur



Exigences non fonctionnelles

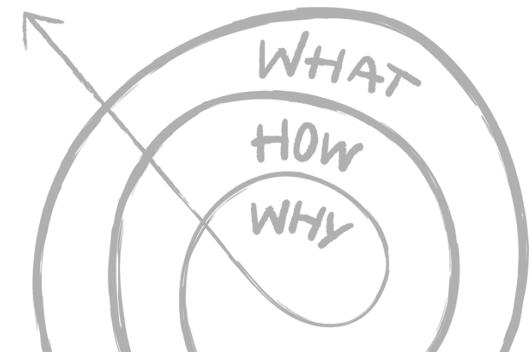
- Ex.: **performance, robustesse,...**
- **Influence:**
 - Interface utilisateur
 - Architecture
 - Choix technologiques
 - ...



Exigences non fonctionnelles

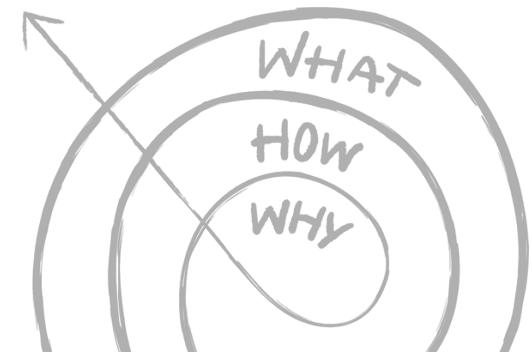
- **Description sous forme de contraintes:**
 - Ex.: Le système doit pouvoir répondre à n'importe quelle requête en moins d'une seconde
- **Distinction entre les exigences globales et locales**
 - Globales: doivent être définie avec la vision du produit
 - Locales: peut être définie et attachée avec la user story affectée

Ces exigences feront partie des critères d'acceptance du client et seront donc liées à la définition du Done (DoD)



Etendue du Product Backlog dans de larges projets

- **Utiliser un seul Product Backlog**
 - Permet d'éviter les problèmes de synchronisation et de vision
- **Préparer les éléments pour les 2 à 3 Sprints suivants**
 - Le nombre d'éléments détaillés est donc plus important
- **Proposer différentes vues pour les différentes teams SCRUM**



Gestion des rôles et des responsabilités

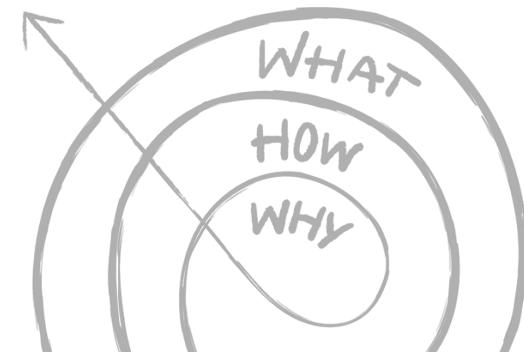


- **Le « Product Owner »**

Le propriétaire de produit est la seule personne responsable de la gestion du carnet de produit.

La gestion du carnet de produit comprend :

- Exprimer clairement les items du carnet de produit ;
- Hiérarchiser les items du carnet de produit pour mieux atteindre les objectifs et les missions ;
- S'assurer de la valeur du travail que l'équipe de développement réalise ;
- S'assurer que le carnet de produit est visible, transparent et clair pour tous et qu'il décrit bien le travail à venir pour l'équipe Scrum ;
- S'assurer que l'équipe de développement comprend suffisamment les items du carnet de produit.



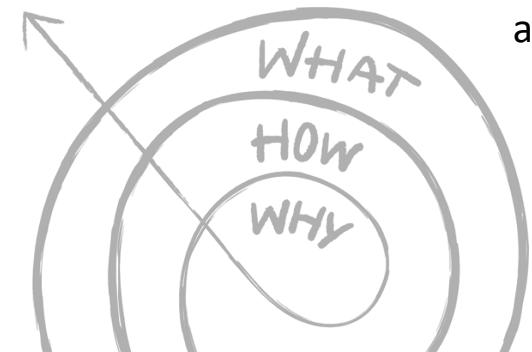


Gestion des rôles et des responsabilités

- **La « Development team »**

Les équipes de développement sont structurées et habilitées par l'organisation à organiser et gérer leur propre travail. Les synergies qui en résultent permettent d'optimiser l'efficacité globale des équipes de développement. Les équipes de développement ont les caractéristiques suivantes :

- Elles sont auto-organisées. Personne (pas même le Scrum Master) ne peut dicter à l'équipe comment traduire le carnet de produit en incrément de fonctionnalités potentiellement livrables ;
- Elles sont pluridisciplinaires, ayant toutes les compétences nécessaires pour mettre en œuvre un incrément du produit ;
- Scrum ne reconnaît aucun autre titre, pour les membres de l'équipe de développement, que celui de développeur, indépendamment du travail réalisé par la personne ; il ne peut y avoir aucune exception à cette règle ;

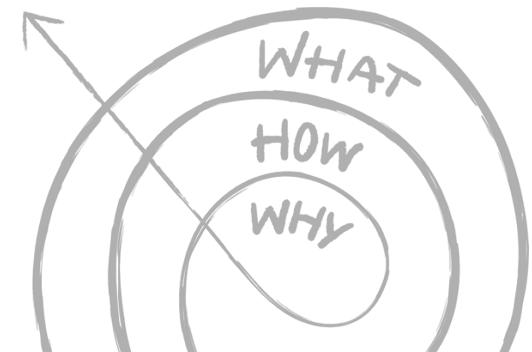


Gestion des rôles et des responsabilités



- **La « Development team »**

- Individuellement, les membres d'une équipe peuvent être spécialisés dans certaines compétences ou dans certains domaines, mais la responsabilité appartient à l'équipe de développement dans son ensemble ; et,
- Les équipes de développement ne contiennent pas de sous-équipes dédiées à un domaine en particulier comme les tests ou l'analyse métier (fonctionnelle).





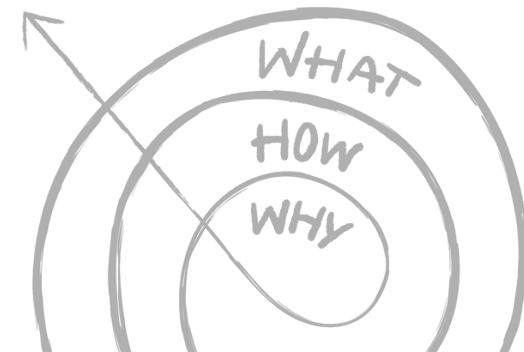
Gestion des rôles et des responsabilités

- **Le « Scrum Master »**

Le Scrum Master aide les personnes extérieures à l'équipe Scrum à comprendre lesquelles de leurs interactions sont utiles et lesquelles ne le sont pas. Le Scrum Master apporte son aide pour changer ces interactions afin de maximiser la valeur créée par l'équipe Scrum.

Le Scrum Master sert le propriétaire de produit de plusieurs façons, notamment :

- En trouvant des techniques pour la gestion efficace du carnet de produit ;
- En communiquant clairement la vision, les objectifs et les items du carnet de produit à l'équipe de développement ;
- En enseignant à l'équipe de développement comment créer des items de carnet clairs et concis ;
- En comprenant la planification à long terme du produit dans un environnement empirique;
- En comprenant et pratiquant l'agilité ;
- En facilitant les événements Scrum à la demande ou lorsque c'est nécessaire.



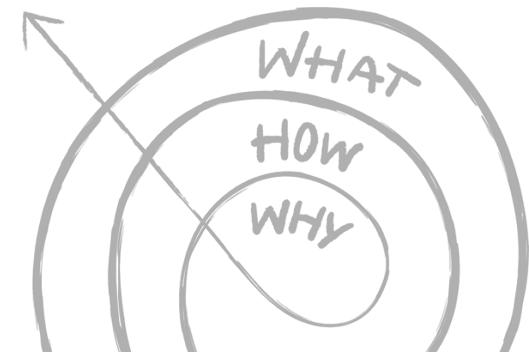
Gestion des rôles et des responsabilités



- **Le « Scrum Master »**

Le Scrum Master sert l'équipe de développement de plusieurs façons, notamment :

- En aidant l'équipe de développement à apprendre comment s'auto-organiser et développer sa transversalité ;
- En enseignant et en menant l'équipe de développement à livrer des produits de haute valeur ;
- En supprimant les obstacles nuisant au progrès de l'équipe de développement ;
- En facilitant des évènements Scrum à la demande ou lorsque c'est nécessaire ;
- En accompagnant l'équipe de développement dans les environnements organisationnels dans lesquels Scrum n'est pas encore complètement adopté et compris.



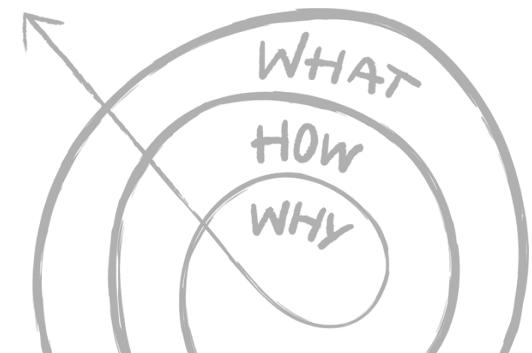
Gestion des rôles et des responsabilités



- **Le « Scrum Master »**

Le Scrum Master sert l'organisation de plusieurs façons, notamment :

- En menant et en accompagnant l'organisation dans ses efforts d'adoption de Scrum;
- En planifiant des mises en oeuvres de Scrum au sein de l'organisation ;
- En aidant les employés et les parties prenantes à comprendre et à adopter Scrum ainsi que le développement empirique de produits ;
- En provoquant le changement qui est susceptible d'accroître la productivité de l'équipe Scrum ;
- En travaillant avec d'autres Scrum Master dans le but d'accroître l'efficacité de l'application de Scrum au sein de l'organisation.



Plan de ce module

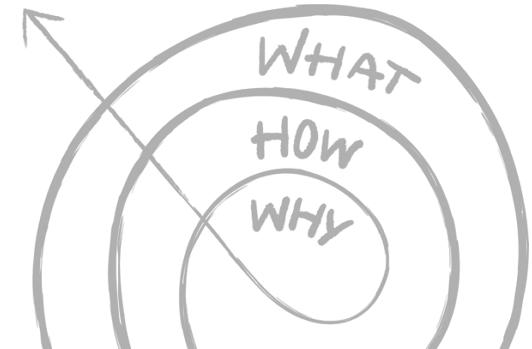
Voici les parties que nous allons aborder :

- **Partie 1 : Introduction & Scrum Basics**

- Les Rôles
- Le Processus
- Le Product Owner
- Le Scrum Master

- **Partie 2 : Mise en œuvre**

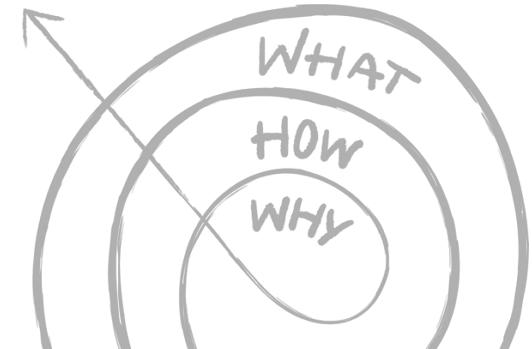
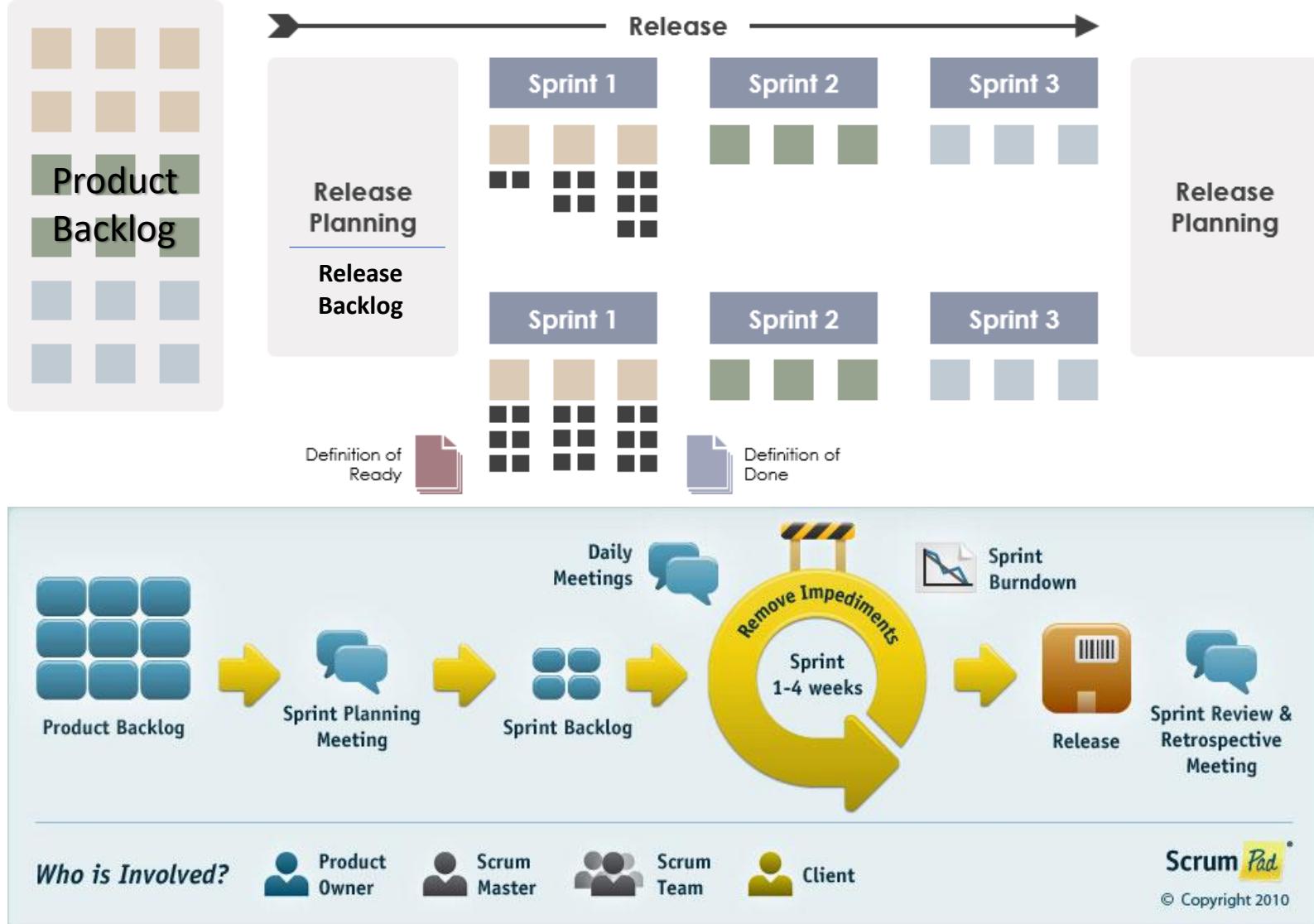
- Le Produit
- Le Product Backlog
- **Les Releases**
- Les Sprints
- L'agilité à l'échelle



Le Release Plan

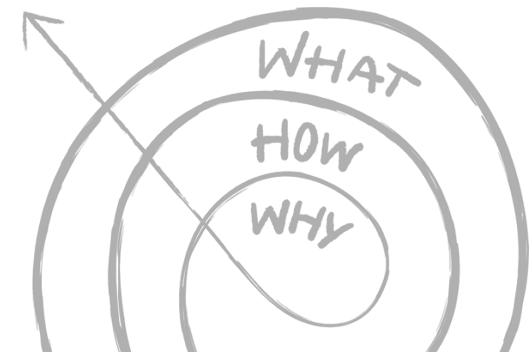


Le Release Plan



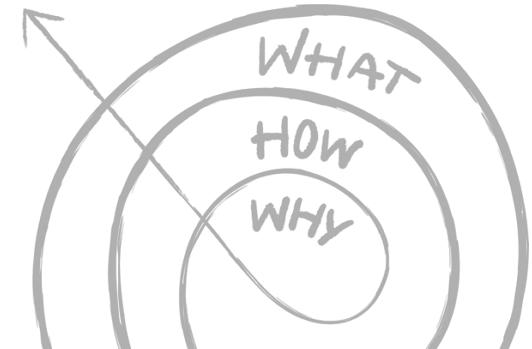
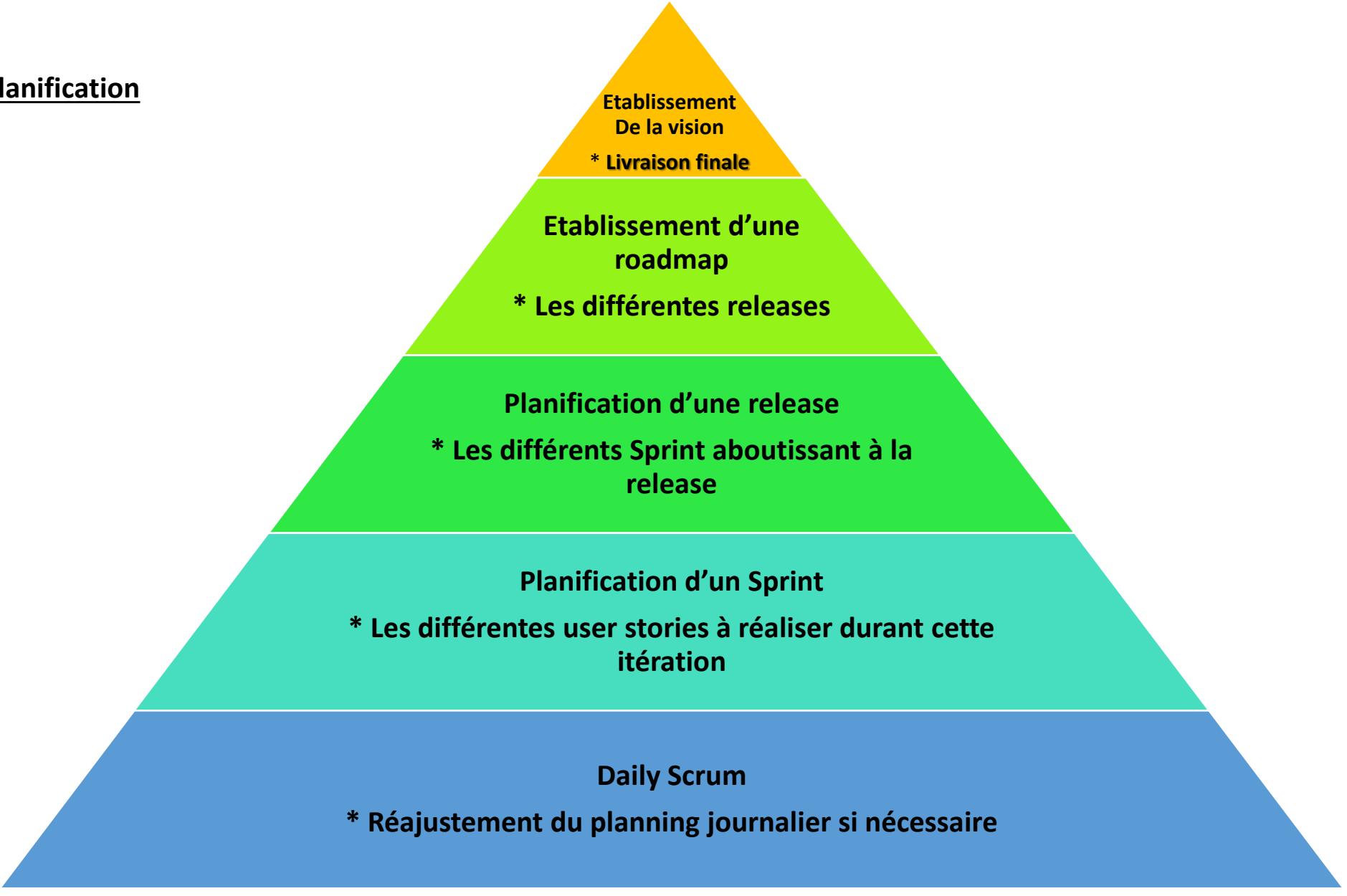
Le Release Plan

- **Planification au fil de l'eau**
 - ❑ Acceptation de l'incertitude
 - ❑ La fiabilité des estimations augmente avec les enseignements des itérations précédentes
 - ❑ Prévision définitive d'une date de livraison du produit final



Le Release Plan

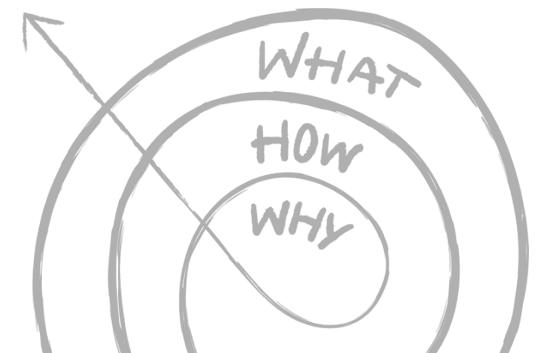
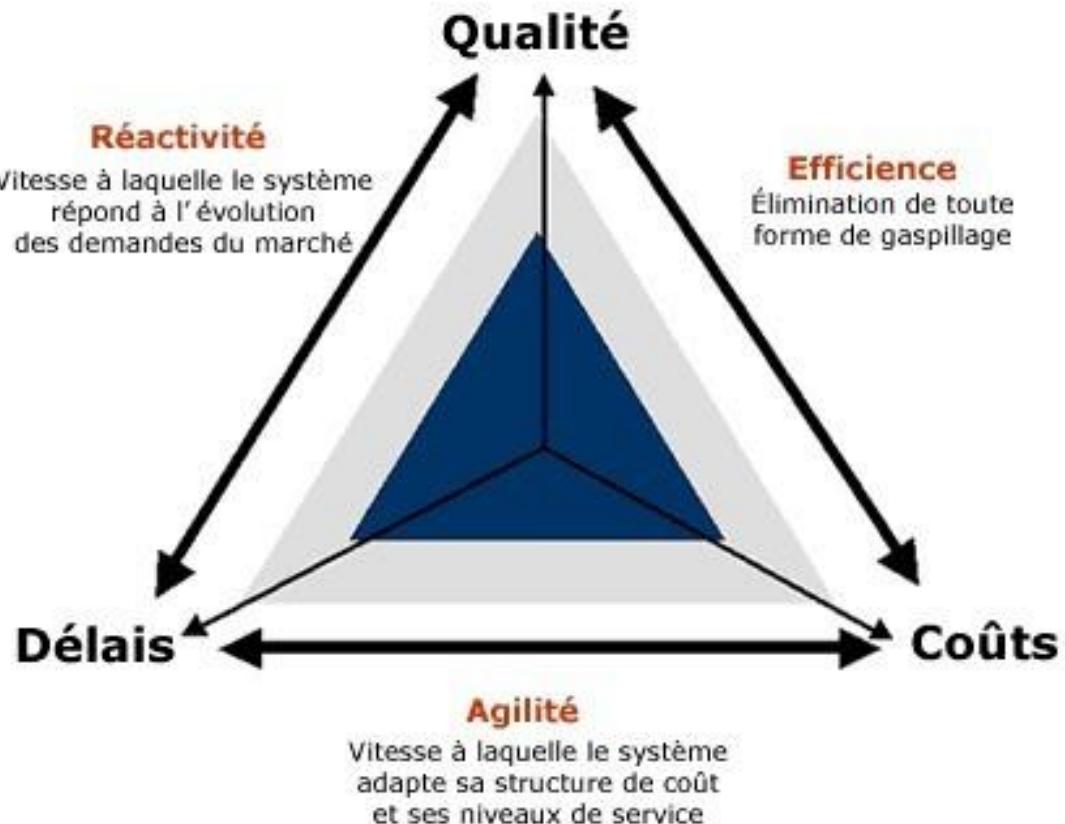
Problématique de la planification



Les Releases

- Trois variables de décision à considérer pour planifier une release

- Coûts
- Délais
- Qualité (fonctionnalités)



Les Releases

- Trois variables de décision à considérer pour planifier une release

- Temps

La date de lancement est-elle fixe? Est-ce une contrainte ?

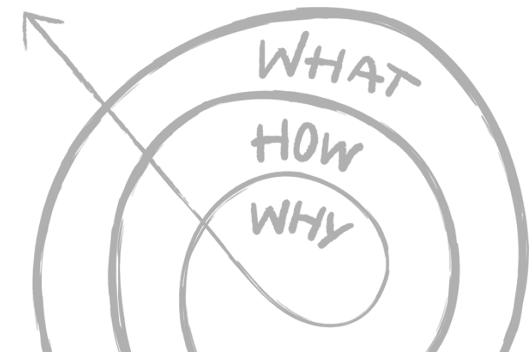
- Coûts

Le budget de développement est-il fixe? Est-ce une contrainte ?

- Fonctionnalités

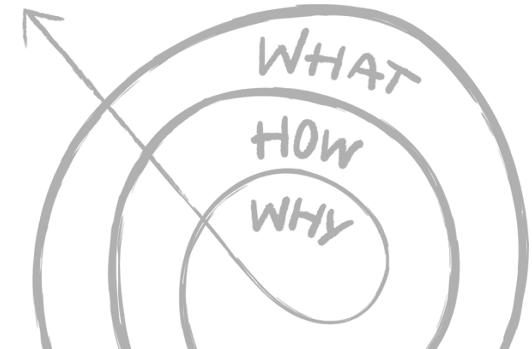
Un certain niveau de fonctionnalité doit-il être assuré pour une release?

- Si une variable est soumise à modification, elle aura un impact sur les autres



Les Releases

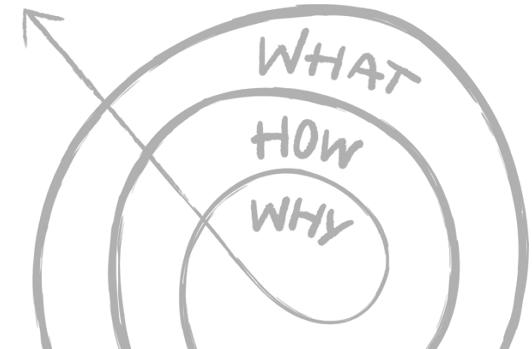
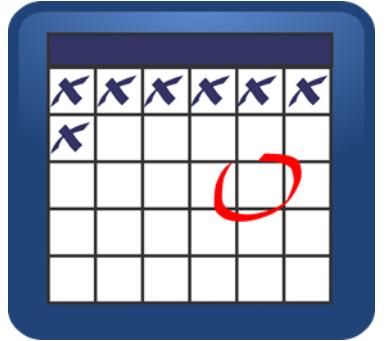
- **Fixer un prix (enveloppe budgétaire)**
 - Arrêt du développement lorsque:
 - Toutes les fonctionnalités sont implémentées ou,
 - Le budget est consommé
 - Lorsque le PO ne trouve plus de sens à continuer le projet
- **Estimation d'une enveloppe globale est indispensable**
 - Evaluer l'opportunité du projet
 - Cadre général pour une planification plus détaillée



Les Releases

- **Fixer une date de lancement**

- Déterminer une *fenêtre d'opportunité*
- Déterminer une "timebox" identique pour toutes les releases, celle-ci pourra être modifiée sur demande express du client afin d'avancer ou de retarder une mise en production.
- Améliore l'innovation
 - Améliore les estimations (coûts, planning...)
 - La détermination du budget est direct si la composition de l'équipe reste stable
(Hypothèse: travail est coût principal)



Plan de ce module

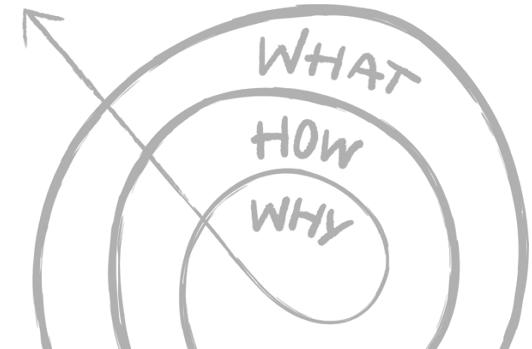
Voici les parties que nous allons aborder :

- **Partie 1 : Introduction & Scrum Basics**

- Les Rôles
- Le Processus
- Le Product Owner
- Le Scrum Master

- **Partie 2 : Mise en œuvre**

- Le Produit
- Le Product Backlog
- Les Releases
- **Les Sprints**
- L'agilité à l'échelle

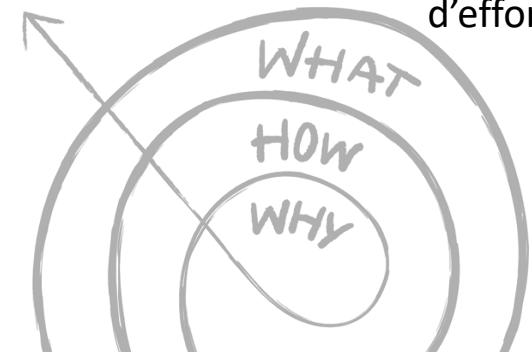


Artefacts de la Scrum Team

Le Sprint Backlog

- **Sprint Backlog**
 - ❑ Contient les différents éléments à implémenter dans le sprint à venir
 - ❑ Les **Users Stories** retenues sont évaluées en point d'effort (story point) en relation avec la vélocité de l'équipe
 - ❑ Les **Users Stories** seront découpées en tâches dans un premier temps pour les premiers jours du Sprint et en suite elles le seront au fil de l'eau durant le sprint de développement
 - ❑ Les tâches sont évaluées en temps et pas en points d'effort

Sprint Backlog			
Forecast	To-Do	In-Progress	Done
<p>Fix My Profile</p> <p>5</p>		<p>aliquip</p>	<p>ipsum</p> <p>duis</p> <p>sit</p> <p>ipsum</p>
<p>Filter Service Tickets</p> <p>8</p>	<p>dolor</p> <p>ipsum</p> <p>culpa</p>	<p>vale</p> <p>culpa</p>	<p>aliquip</p>
<p>Quick Tips</p> <p>3</p>	<p>ipsum</p> <p>sit</p> <p>duis</p> <p>duis</p>		



Artefacts de la Scrum Team

Le Sprint Backlog

Website Launch ★ ... Share

Board Timeline Calendar Conversations Progress Files

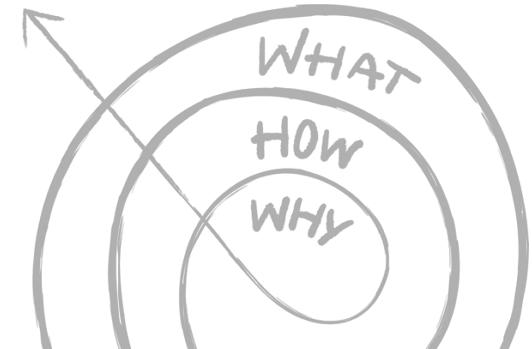
Show Project Description

View: All Tasks

Backlog Ready to do In progress Done + Add column

Backlog	Ready to do	In progress	Done
QA new feature	CSS/HTML validated	Sync with Google Analytics	Font and image licensing
Create short URLs	Videos have updated branding	Cookies notice	Page hierarchy
Optimize campaign tracking	Tracking codes inserted	Terms and privacy policy updated	Sharing icons updated and working
Final QA	Mobile-friendly site parallels	SEO keyword optimizations	Scripts and images are
Forms submit data properly	Auto-responses working	RSS metadata	

capture de Asana : www.asana.com



Artefacts de la Scrum Team

Le Sprint Backlog

Fabrikam Fiber Team Sprint 1

June 1 - June 12
6 work days remaining

Backlog **Board** Capacity

Group by **Backlog items** Person **Christie Church**

To do 66 h In progress 9 h Done

Unparented 1 not started (8 h)

Hello World Web Site

364 Slow response on information form Raisa Pokrov... 15 h State Committed	374 Apply fix elsewhere as needed Christie Church 12	383 Fix performance issues Christie Church 3
+ Add an information form		
360 Change initial view Christie Church 8 h State Committed Effort 13	385 Add switching context Christie Church 8	

Capture Azur & TFS Microsoft

www.tuleap.org

tuleap

My Personal Page Projects Help More Admin

BrainstormingApp Top Tracking Planning / Release 1.2 / Sprint 3

Admin

Sprint 3 Sept 15 – Oct 05

Content Burndown Card Wall Artifact

Sept 15 – Oct 05 40% 11 days to go Some cards might be hidden 62% 4/12 cards open

Search for user, title, ...

Fullscreen

BACKLOG ITEMS TODO ON GOING UNDER REVIEW DONE

story #1269 Communicates with the members of the meeting
details Remaining effort: in Story Points: 2 Status: On Going

task #1367 Invitation to discuss
details Assigned to: -

bug #1260 Do not show error message if login fail
details Assigned to: - Severity: Low impact Submitted on: 2 months ago Submitted by: -

task #1364 Board members directory
details Assigned to: -

Microsoft Visual Studio 2010 Team Web Access

Home Work Items Source Build Planning board Task board Settings Help

Dominic Danis Visual Studio 2010 Video (VS2010) Sprint 1

New Work Item

Card Selection: All | None Card Filters: Show resolved work items

Visual Studio 2010 Video

Release 1

Done: 0 Not done: 28 T. Work items: 28 T. Estimate: 34h Tasks: 19 T. Remaining: 85h User Stories: 9 T. Story Points: 29

Release

Sprint 1

124 As a store administrator I should be able to examine contact details for a customer. Sprint 1 2 pts +

535 Add address to detail section Sprint 1 4 h +

534 Add search contact box Sprint 1 8 h +

125 Design web front-end for viewing customer details Sprint 1 3 h +

126 Implement web front-end for viewing customer details Sprint 1 2 h +

117 As a customer I should have to enter a strong password when creating a new account. Sprint 1 3 pts +

533 Validate that the password never been used Sprint 1 2 h +

532 Validate expression Sprint 1 2 h +

119 Add RegEx to account creation Web page Sprint 1 6 h +

118 Write regular expression for strong password Sprint 1 10 h +

114 As a customer I should be able to remove items from my shopping car cart. Sprint 1 2 pts +

531 Add cancel button Sprint 1 4 h +

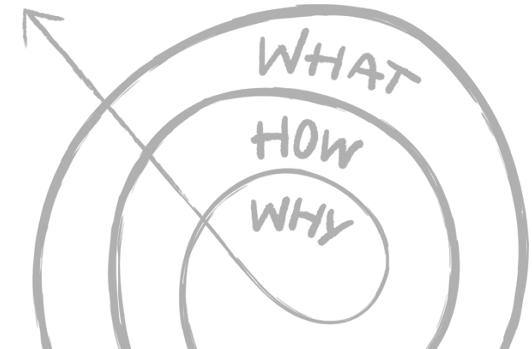
550 Create confirmation window Sprint 1 6 h +

529 Update cart statistics Sprint 1 8 h +

116 Bind cart removal function to back end Sprint 1 5 h +

115 Design UI for item removal Sprint 1 8 h +

Powered by Urban Turtle - (Trial, Version: 3.0) - Licensed to: Dominic Danis - Upgrade - Contact Us

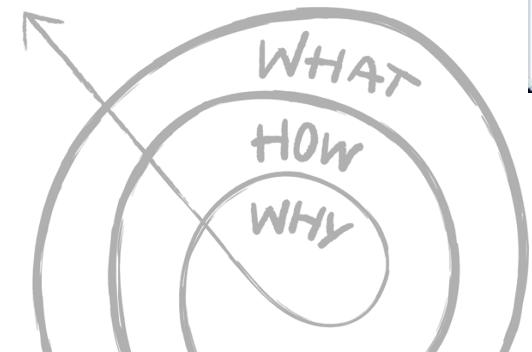


Artefacts de la Scrum Team

Le Sprint Backlog

The screenshot shows a Jira Team Scrum Board for Sprint 3. The board is divided into four columns: To Do, In Progress, In Review, and Done. Each column contains several Jira issues with their respective descriptions and status icons.

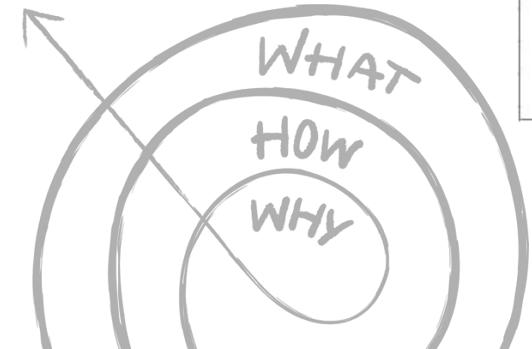
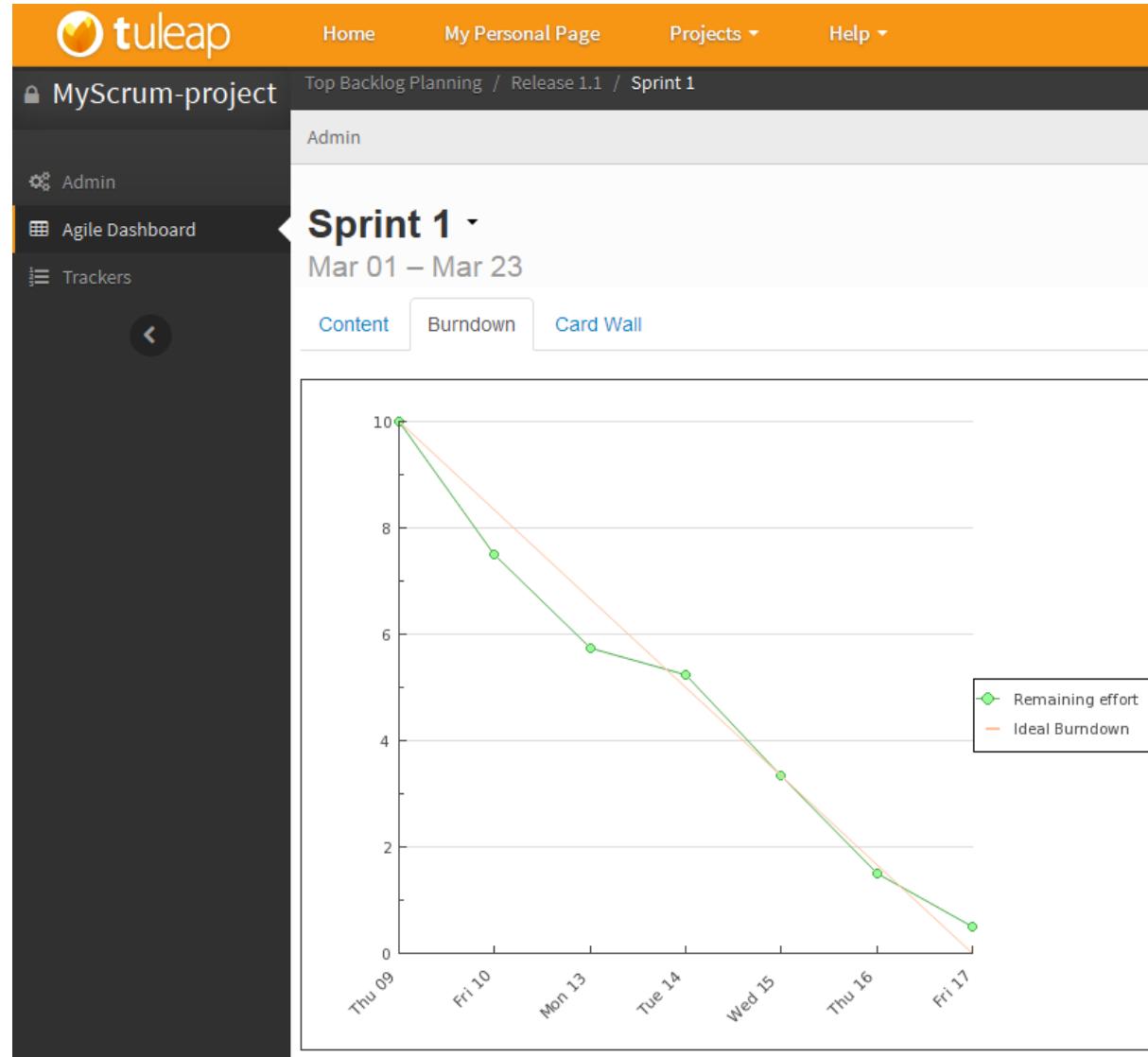
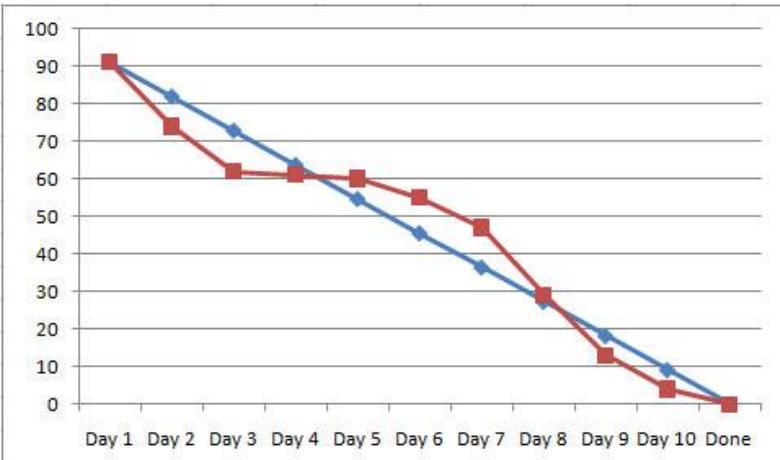
To Do	In Progress	In Review	Done
TIS-28 Research options to travel to Pluto	TIS-27 Add Phobos and Deimos Tours as a Preferred Travel Partner	TIS-58 Add feedback button to the plugin sample code	TIS-9 After 100,000 requests the SeeSpaceEZ server dies
TIS-8 Requesting available flights is now taking > 5 seconds	TIS-10 Bad JSON data coming back from hotel API	TIS-45 Email non registered users to sign up with Teams In Space	TIS-16 Establish relationship with local office supplies company
	TIS-25 Engage Jupiter Express for outer solar system travel		TIS-7 500 Error when requesting a reservation
	TIS-20 Engage Saturn Shuttle Lines for group tours		TIS-11 Register with the Mars Ministry of Labor



Artefacts divers

Le Burndown Chart (Ce qu'il reste à faire)

Le diagramme Burndown pour une release à portée fixe permet de voir la quantité de travail restant dans chaque sprint pour atteindre le but de release. L'axe vertical de ce diagramme correspond aux unités qui ont permis de dimensionner les Items du Product Backlog. L'axe horizontal est rythmé par les sprints.



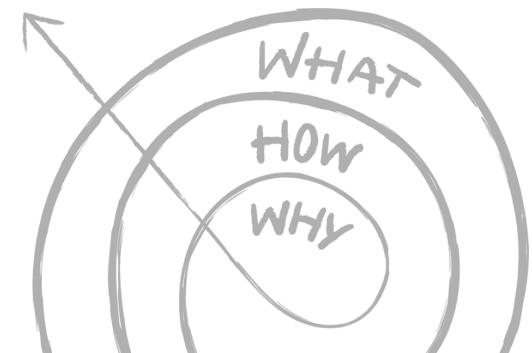
Artefacts divers

Le Burndown Chart (Ce qu'il reste à faire)

En ce qui concerne le Burndown Chart à barres :

Les éléments retirés sont placés sous la barre de base.

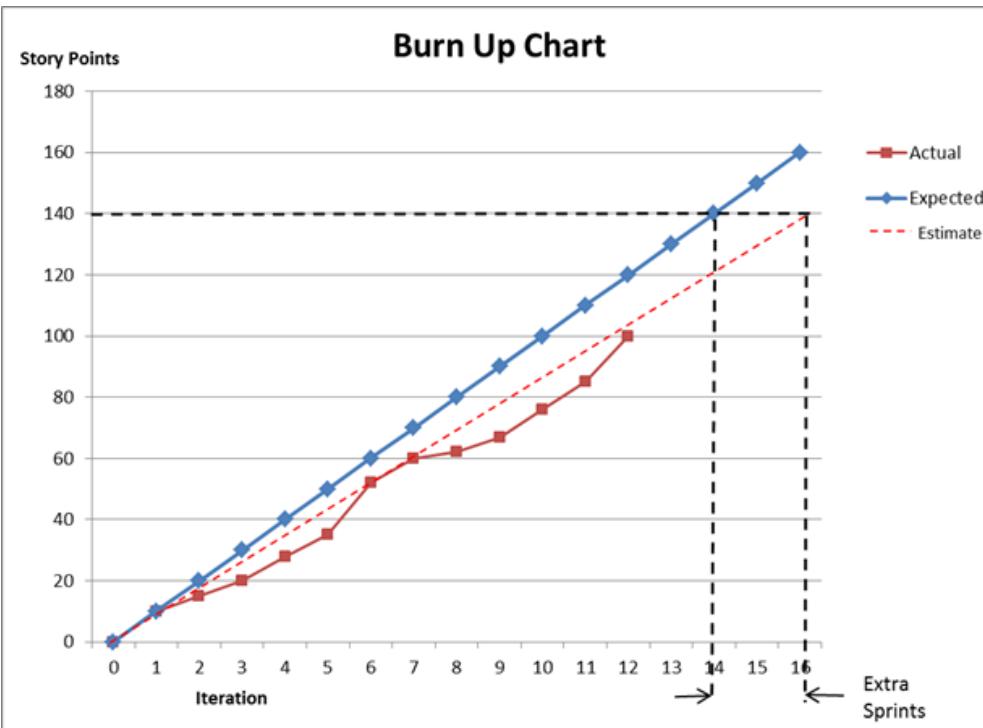
La progression se trouve au dessus de la barre



Artefacts divers

Le Burnup Chart (Ce qui a été fait)

Le diagramme burnup d'une release à portée fixe permet de visualiser le rythme auquel le travail réalisé vient rejoindre la ligne d'objectif de Sprint en Sprint. L'abscisse et l'ordonnée du diagramme sont les mêmes que celles du diagramme burndown.



JIRA Dashboards Projects Issues Boards Create Search

AGILE REPORTS Project burnup Epic burnup

Agile board BURNUP board

Story Points

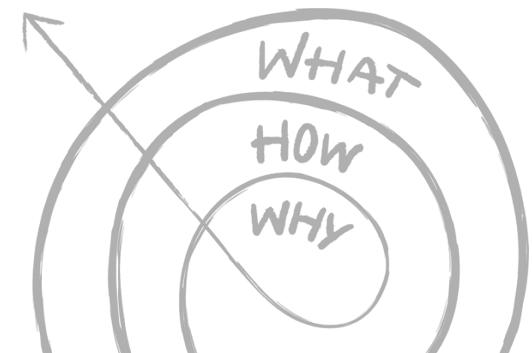
Active sprints Past sprints Scope MAX velocity - 8 AVG velocity - 5.7 MIN velocity - 3

Sprint: TeamA Sprint 7 State: closed Start date: 2017-12-28 End date: 2018-01-04 Velocity: 6 Burnup: 17

Past sprints Sprint count 3 Add Sprint filter

Future sprints Sprint length 7 Remaining work 24 Remaining work growth

Velocity	End Date	Sprints
3 MIN	2018-03-06	8
5.7 AVG	2018-02-13	5
8 MAX	2018-01-30	3



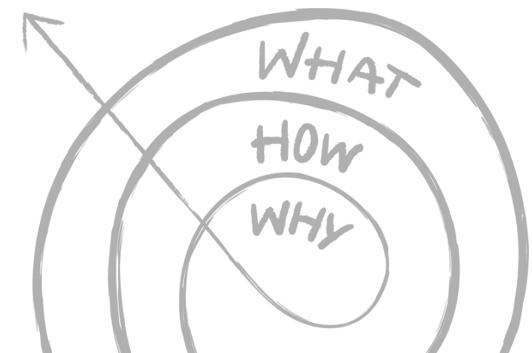
Artefacts divers

Le Calendrier NIKO-NIKO

- **Le Niko-Niko Chart**

- Concrètement, le calendrier Niko-Niko se présente sous la forme d'une feuille divisée en colonne, chaque colonne correspondant à un membre de l'équipe.
- Chaque jour, à l'issue de sa journée de travail, chacun membre met dans sa colonne un smiley représentant son ressentie de la journée.
- Un smiley qui fait la tête représente donc une mauvaise journée, tandis qu'un smiley souriant représente une bonne journée et un smiley neutre une journée mi-figue, mi-raisin.
- Ce smiley a de plus une valeur globale sur la journée, dans le sens ou il englobe aussi bien l'aspect humain que technique, et il est strictement personnel.

Sprint 33	Mon	Tue	Wed	Thur.	Fri	Mon	Tue	Wed	Thur.	Fri
Lucinda	:)	:)	:)	:)	:)	:)	:)	:)	:)	:)
Theodore	:S	:S	:S	:S	:S	:S	:S	:)	:)	:S
Virginia	:)	:S	:S	:S	:)	:)	:)	:)	:)	:)
Olga	:)	:)	:)	:)	:)	:)	:)	:)	:)	:)
Elliot	:)	:)	:)	:)	:)	:)	:)	:)	:)	:)
Suresh	:)	:)	:)	:)	:)	:)	:)	:)	:)	:)
Emile	:)	:S	:)	:)	:)	:)	:)	:)	:)	:)



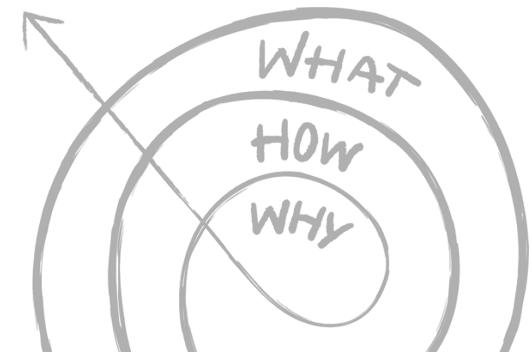
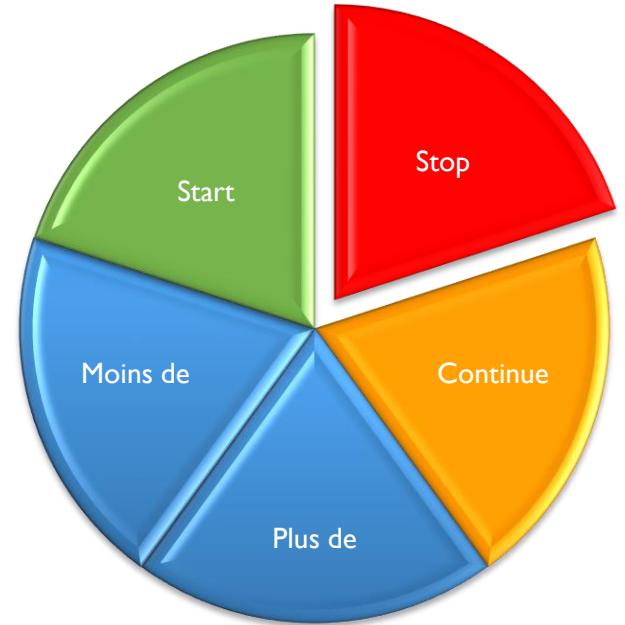
Outils de rétrospective

Le « Start – Stop – Continue – Plus de – moins de »

Obtenir un retour d'information sur le processus appliqué par l'équipe afin d'apprendre ce qui doit être poursuivi et ce qui doit être ajusté pour que l'équipe aille de l'avant. C'est un moyen rapide de mener une discussion sur le processus en lui-même.

Durée de jeu : 10 minutes pour les plus petits groupes jusqu'à 25 minutes pour les plus grands groupes. La durée dépend aussi de la longueur de la période sur laquelle porte la rétrospective.

1. Tracez un cercle sur une affiche, divisez-le en 5 sections et inscrivez les mots "commencer", "arrêter", "continuer", "plus de" et "moins de" autour de chaque section.
2. Demandez : que devrions-nous commencer à faire que nous n'avons peut-être pas encore fait ? Que devrions-nous arrêter de faire qui ne nous aide pas ? Qu'est-ce qui fonctionne et que nous voulons continuer à faire ? Que devrions-nous faire en plus ? en moins ?"
3. Encouragez les commentaires pour chaque catégorie. Clarifiez pour quelle catégorie s'applique le commentaire et inscrivez les commentaires (ou résumez-les) sur l'affiche. Utilisez le silence, attendez les commentaires.



Outils de rétrospective

Le Speed boat

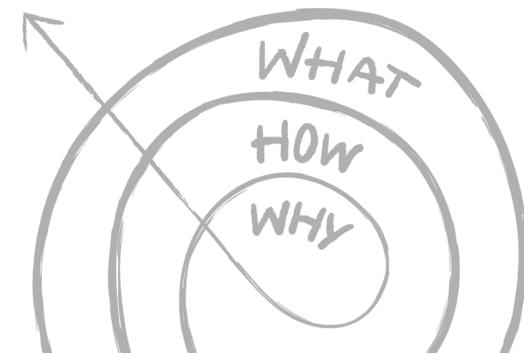
Le speed boat est un exercice ludique destiné à inciter les membres de l'équipe à participer. Il est particulièrement efficace car il est très visuel et imagé.

Pour commencer, on dessine sur un tableau un voilier voguant sur la mer, des ancre, le vent qui le fait avancer, un beau soleil qui brille dans le ciel, et des récifs.



- *Le voilier représente l'équipe projet.*
- *Le vent représente tous les éléments qui ont fait avancer l'équipe durant l'itération.*
- *Les ancre représentent les facteurs qui ralentissent l'équipe ou diminuent sa productivité.*
- *L'île représente les objectifs à atteindre.*
- *Le soleil représente ce que l'équipe a aimé, les actions ou les attitudes qui méritent des remerciements.*
- *Les récifs représentent les futurs obstacles identifiés, et que l'équipe risque de rencontrer dans les prochaines itérations.*

Chaque participant dispose d'un paquet de post-it. L'exercice commence par la définition des objectifs de chacun. Chaque membre de l'équipe écrit ses objectifs au sein du projet sur ses post-it.



Outils de rétrospective



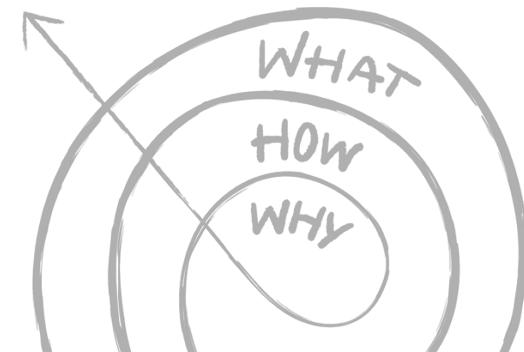
Le Speed boat

Les post-it représentant les éléments positifs sont collés sur le vent qui gonfle les voiles et les éléments négatifs sont collés sur les ancre (et plus l'ancre s'enfonce profondément dans l'eau et plus l'impact est négatif). Les post-it représentant les obstacles sont regroupés sur les récifs et les remerciements sur le soleil.

Le facilitateur de la rétrospective agile reprend ensuite chaque post-it et le lit à haute voix afin que tout le monde puisse en prendre connaissance. Chaque participant peut expliquer rapidement l'objet du ou des post-it dont il est l'auteur. Les post-it sont regroupés par sujet et exposés à tous. Si le nombre de participants est important, il est préférable de regrouper les post-it par sujet au fur et à mesure que leurs auteurs viennent les coller.

Une fois que tout le monde a pris connaissance des différents sujets évoqués sur les post-it, un vote est organisé. L'objectif est de décider quels sont les sujets devant être traités prioritairement. Il est intéressant à ce stade de limiter le nombre de sujets, de façon à rester efficace et à ne pas trop disperser les efforts à fournir.

Lorsque les priorités ont été établies, il est possible d'élaborer un plan d'actions à mettre en place pour l'itération suivante.



Outils de rétrospective

Le Speed boat

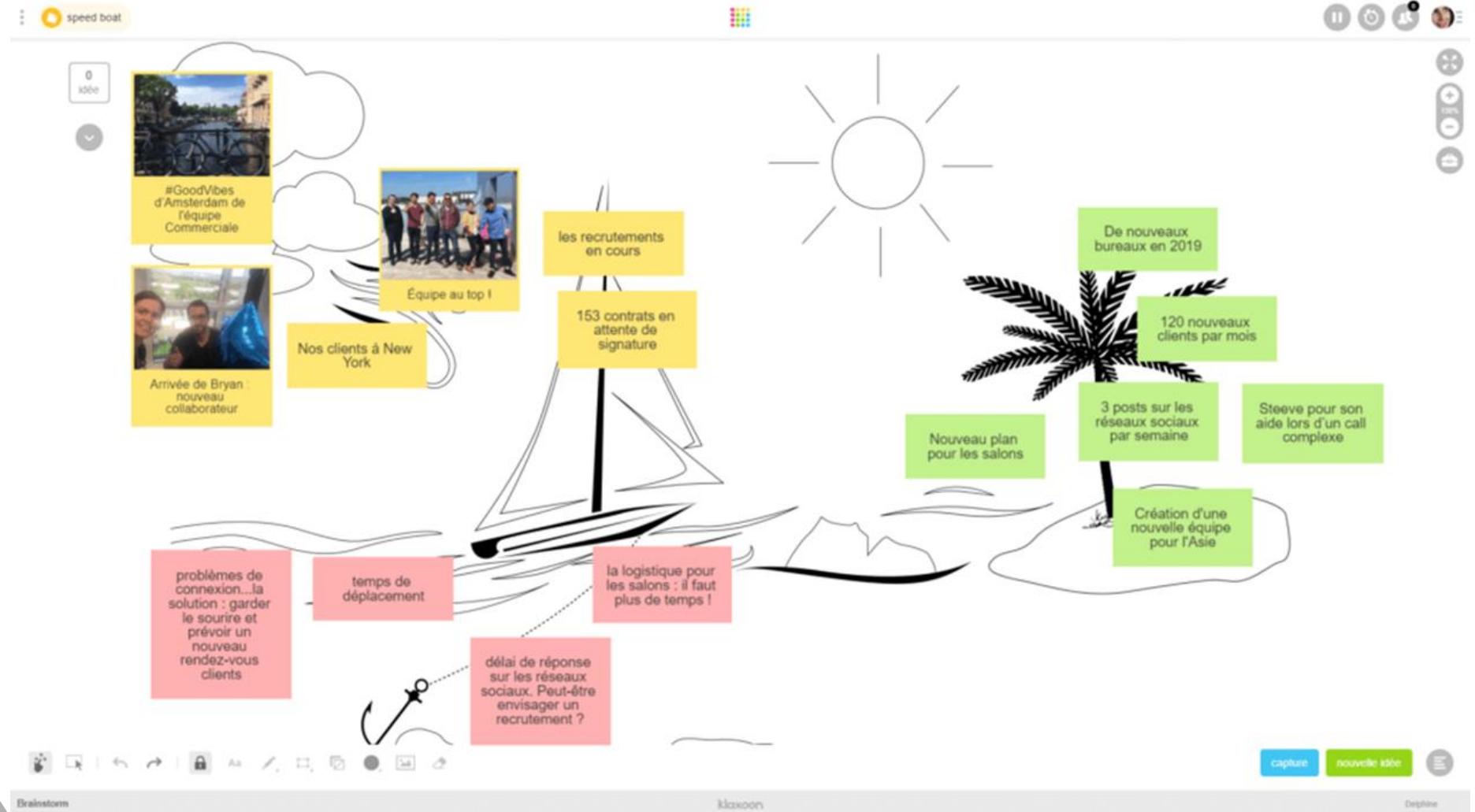
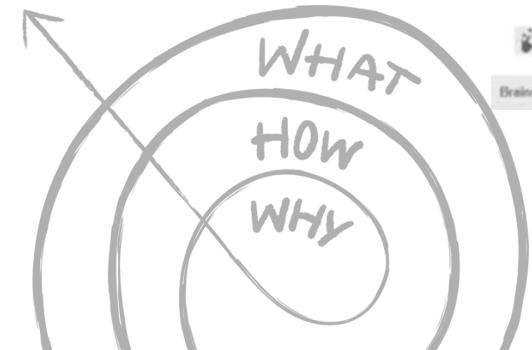
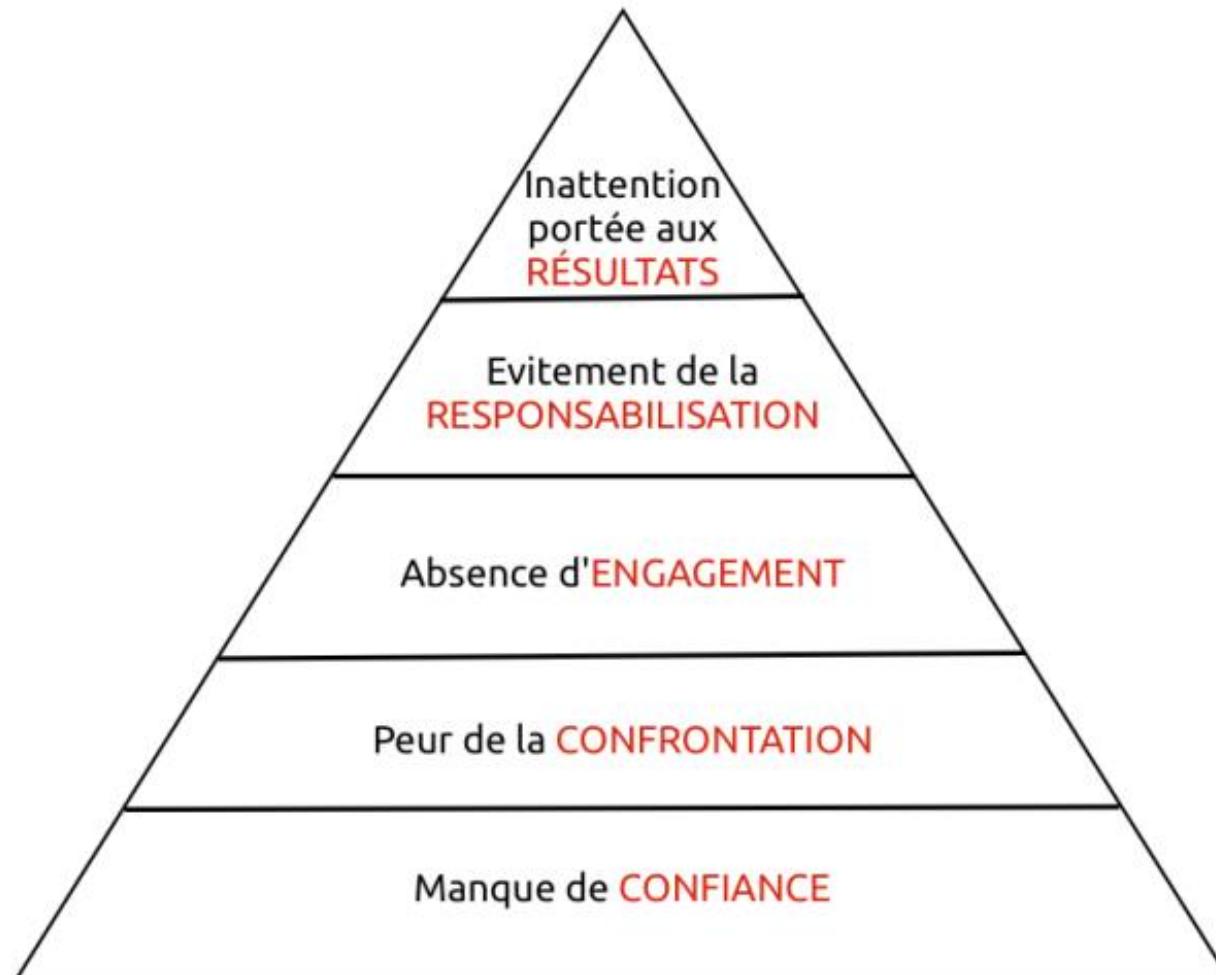


Image provenant du site
www.klaxoon.com

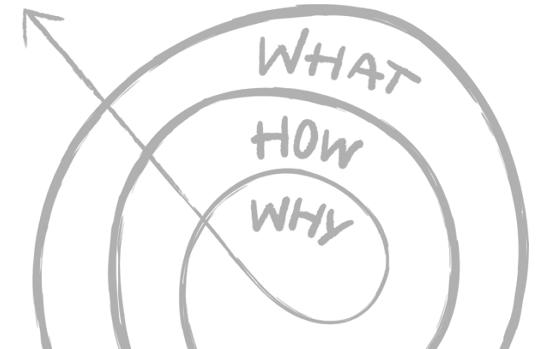


Outils de management

La Pyramide de Lencioni



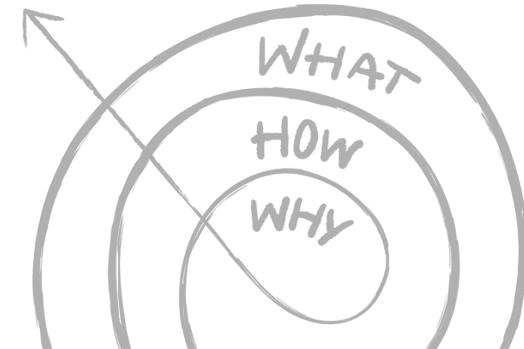
Les 5 dysfonctionnements de l'équipe (Lencioni)



Outils de management

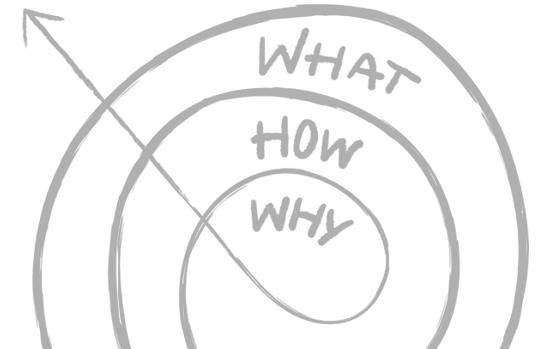
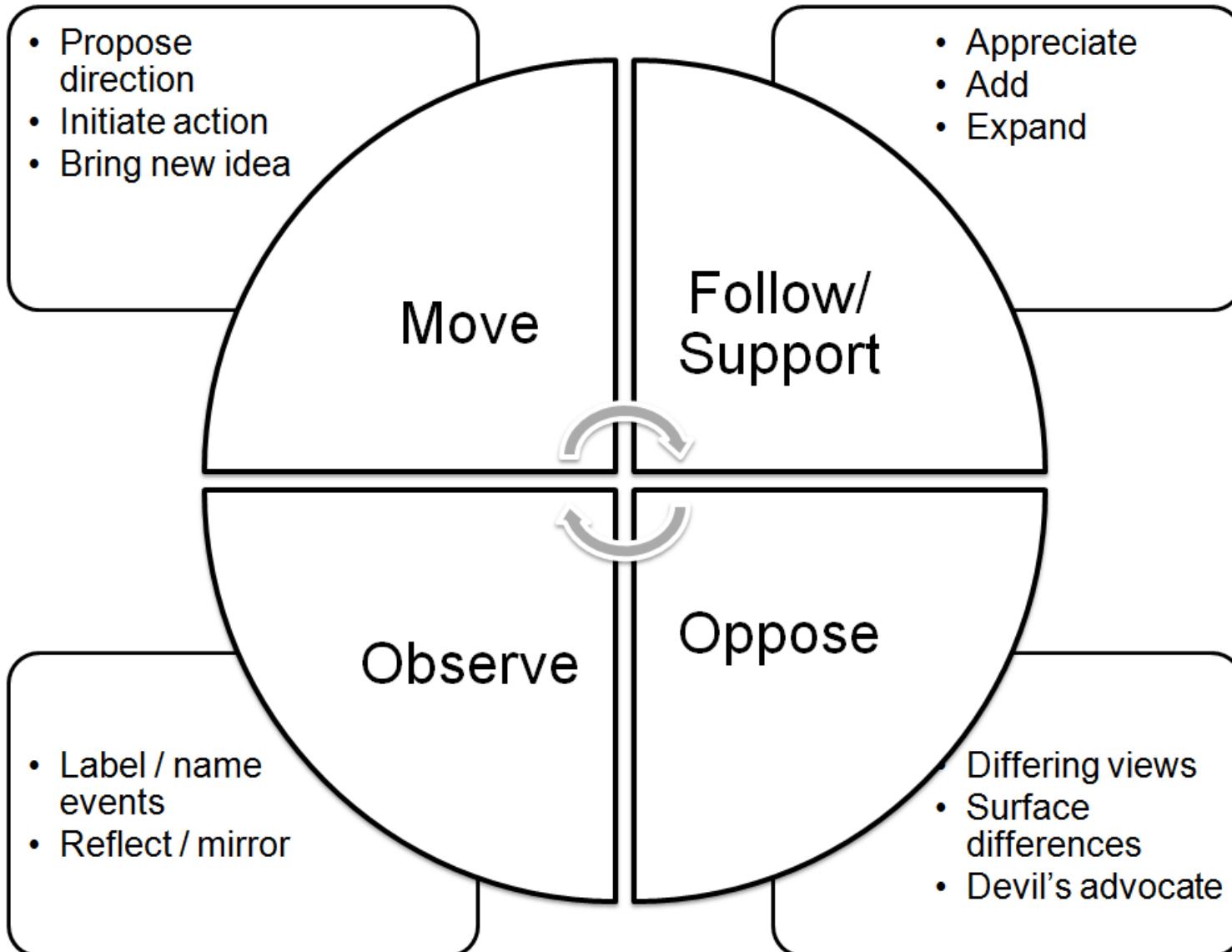
Myers Briggs

Take the Test



Outils de management

David Kantor

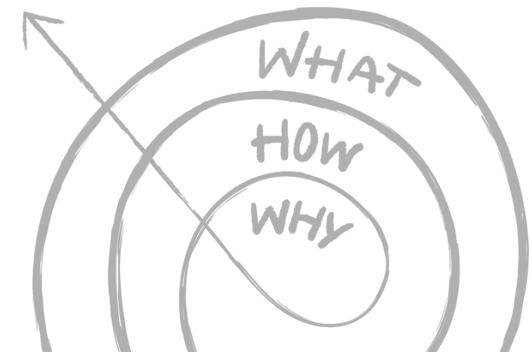


Le Jeu de rôle

Exercice : Gestionnaire de location de voiture

- Par groupe:

- 1. Déterminer un product backlog composé de 5 Epics décrivant le système
- 2. Prioriser les éléments
- 3. Estimer la charge de travail pour chaque élément du 1^{er} Sprint
- 4. Raffiner le premier élément en 3 user stories



MERCI
POUR CES 2 JOURNEES
DE PARTAGES

