

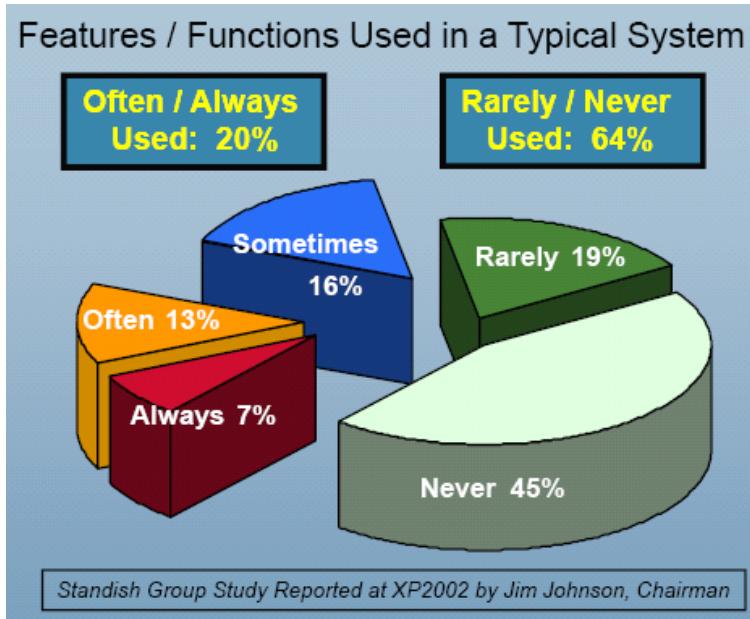
Formation Agilité

fondamentaux et pratiques avancées

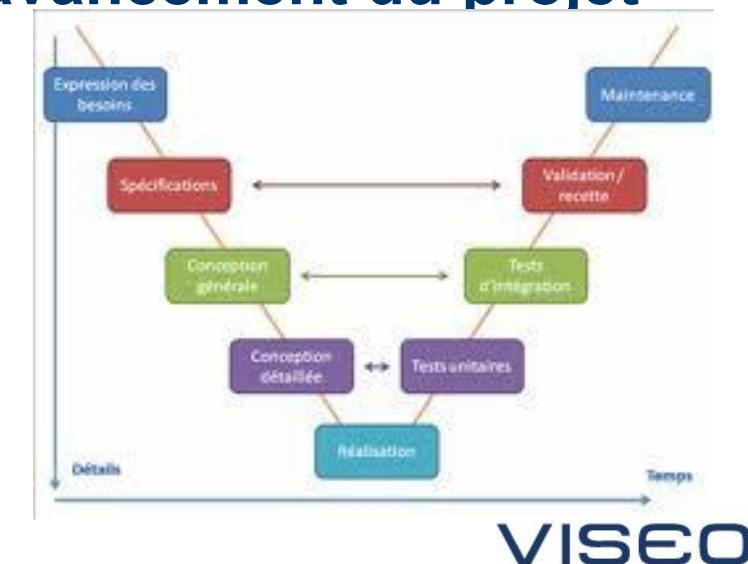
Limites et constat des méthodes prédictives



Limites des méthodes prédictives



- Seules 20% des fonctionnalités implémentées sont fréquemment utilisées.
- Près de 50 % des fonctionnalités ne sont jamais utilisées.
- Effet tunnel qui ne permet pas de mesurer l'avancement du projet par le client.
- Les anomalies ne sont constatées que lors des phases de test tardives.



Les valeurs du manifeste agile



Les valeurs du manifeste agile

Manifeste pour le développement Agile de logiciels

Nous découvrons comment mieux développer des logiciels
par la pratique et en aidant les autres à le faire.

Ces expériences nous ont amenés à valoriser :

Les individus et leurs interactions plus que les processus et les outils

Des logiciels opérationnels plus qu'une documentation exhaustive

La collaboration avec les clients plus que la négociation contractuelle

L'adaptation au changement plus que le suivi d'un plan

Nous reconnaissons la valeur des seconds éléments,
mais privilégiions les premiers.

Principes orientés client

- ❖ Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
- ❖ Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
- ❖ Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
- ❖ Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.



Principes orientés équipe

- ❖ Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
- ❖ La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
- ❖ Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.
- ❖ À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

Motivation



Principes orientés organisation

- ❖ Les processus Agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.

- ❖ Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.

- ❖ La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.

- ❖ Un logiciel opérationnel est la principale mesure d'avancement.



Scrum

3 rôles, 3 artefacts, 3 cérémonies



Les 3 rôles de Scrum



Rôle : Product Owner

❖ Le Product Owner

- Intervient et s'implique tout au long du projet.
- Représente l'ensemble des clients.
- Définit l'objectif à atteindre, la vision.
- Dictionnaire pour la planification des besoins fonctionnels.
 - Il tient à jour les plannings de haut niveau (Road map, Release plan).
 - Il priorise les besoins.
- Référent du domaine fonctionnel.
- Responsable du budget.
- Rédige l'ensemble des besoins.
- Valide les éléments livrés.
- Il est disponible.



Rôle : Équipe

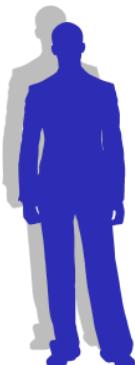
- ❖ L'équipe a la charge de développer le système.
 - Estime les besoins exprimés par le Product Owner.
 - Décompose les éléments fonctionnels en tâches techniques.
 - Auto-gérée, elle décide du plan de développement.
 - Garantit la qualité du logiciel.
 - Garantit la non régression.
 - Garante de l'architecture retenue.



Rôle : Scrum Master

✓ Le Scrum Master.

- C'est LE facilitateur.
- Animateur de l'équipe et du projet.
- Garant de l'application du processus.
- Protège l'équipe contre elle-même, et contre les éléments extérieurs.
- Collabore avec le Product Owner sur l'ensemble de ses tâches.
- Aide l'équipe à s'améliorer.
- N'a aucune relation de hiérarchie avec les membres de l'équipe.



Les 3 artéfacts Scrum



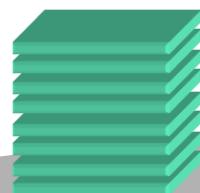
Backlog de produit et de sprint

Liste des tâches (subdivision des exigences) à réaliser durant le sprint.

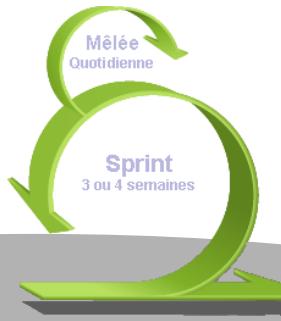


Liste priorisée des exigences produit.

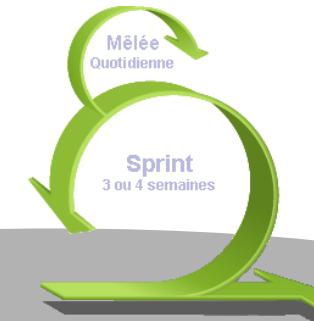
Incrément produit.



Backlog de Sprint



Backlog de Sprint



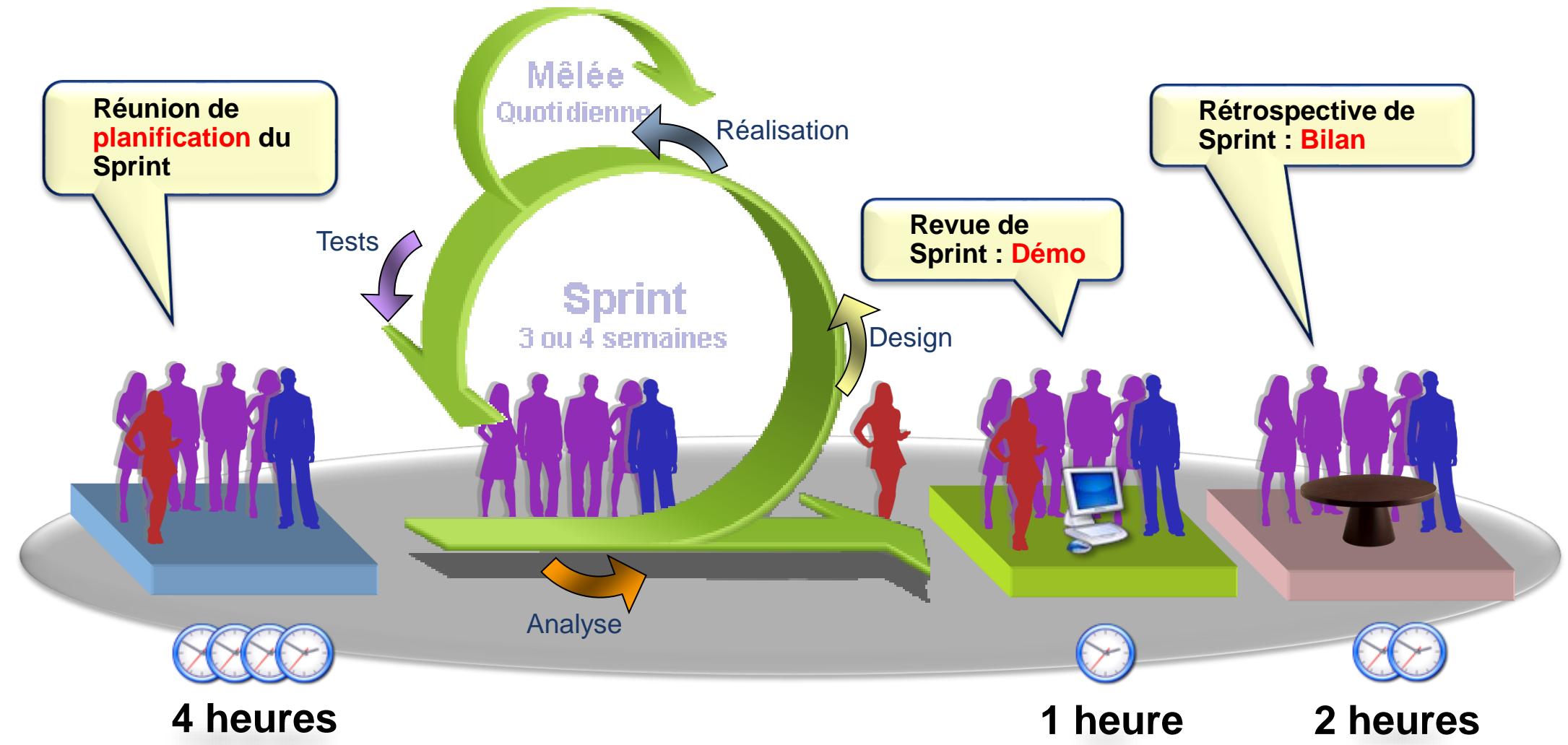
Sprint N :
2-4 semaines

Sprint N+1 :
2-4 semaines



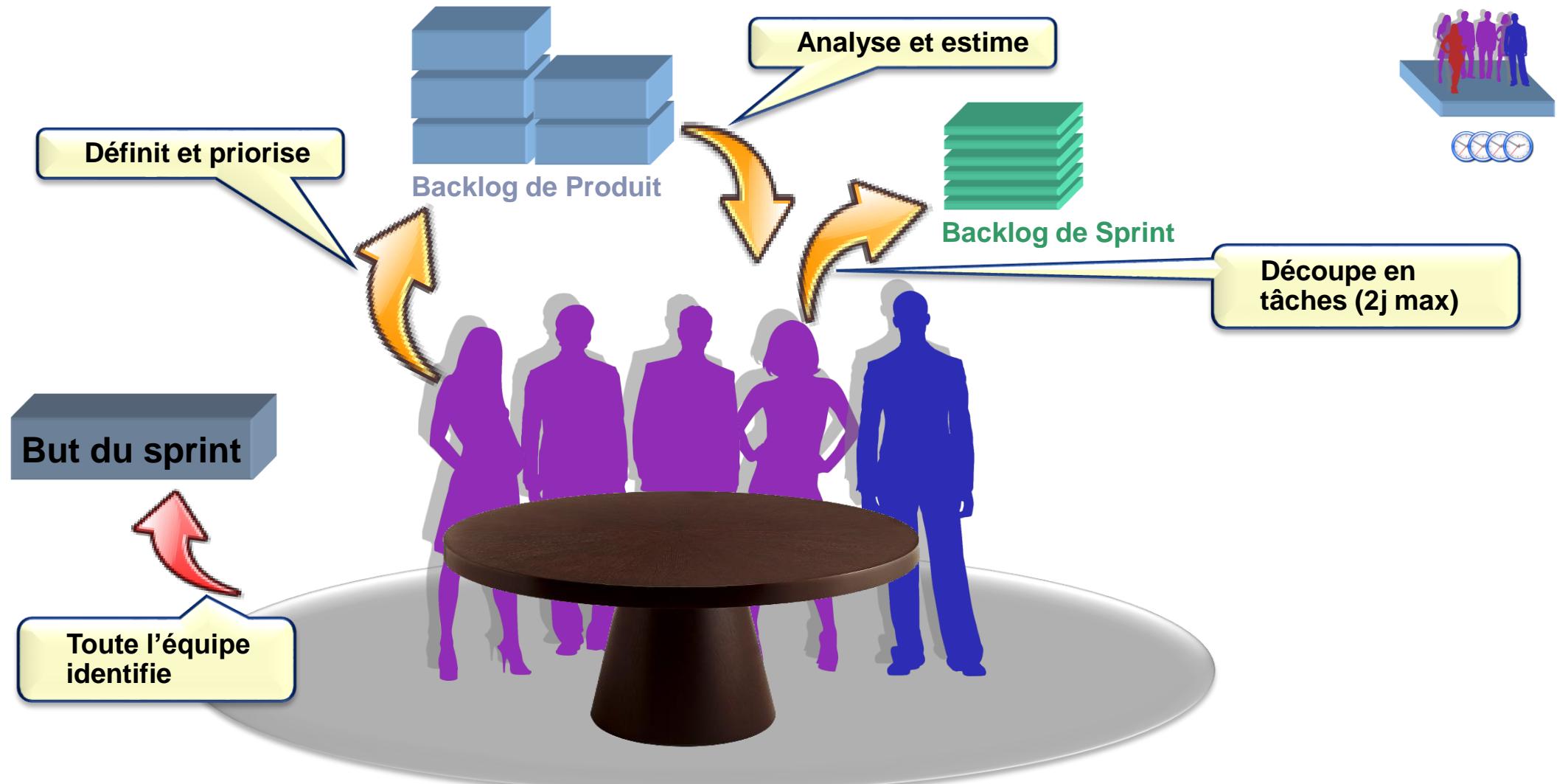
Les 3 cérémonies Scrum





Définir les objectifs et tâches du sprint : Le sprint planning





Préparation du Sprint planning meeting

- ✓ **Comme toute réunion, ça se prépare !**

- ✓ **Le Product Owner :**
 - A recensé les User Stories INVEST qui devraient être intégrées au Sprint en fonction des priorités,
 - Vérifie l'adéquation entre la vision du projet et les stories retenues pour le prochain sprint,
 - A défini les tests d'acceptance.

- ✓ **Le Scrum Master:**
 - Recense les disponibilités de l'équipe,
 - Donne au Product Owner la capacité de production de l'équipe,
 - Aide le Product Owner dans la préparation du sprint.

Déroulement d'un Sprint planning meeting

- ✓ **Introduction par le Product Owner:**
 - Définition du but du Sprint (avec l'aide de l'équipe),
 - Résumé du Backlog,
 - Présentation des User Stories qui vont être intégrées au sprint.
- ✓ **L'équipe échange avec le Product Owner pour pouvoir préparer le découpage en tâches.**

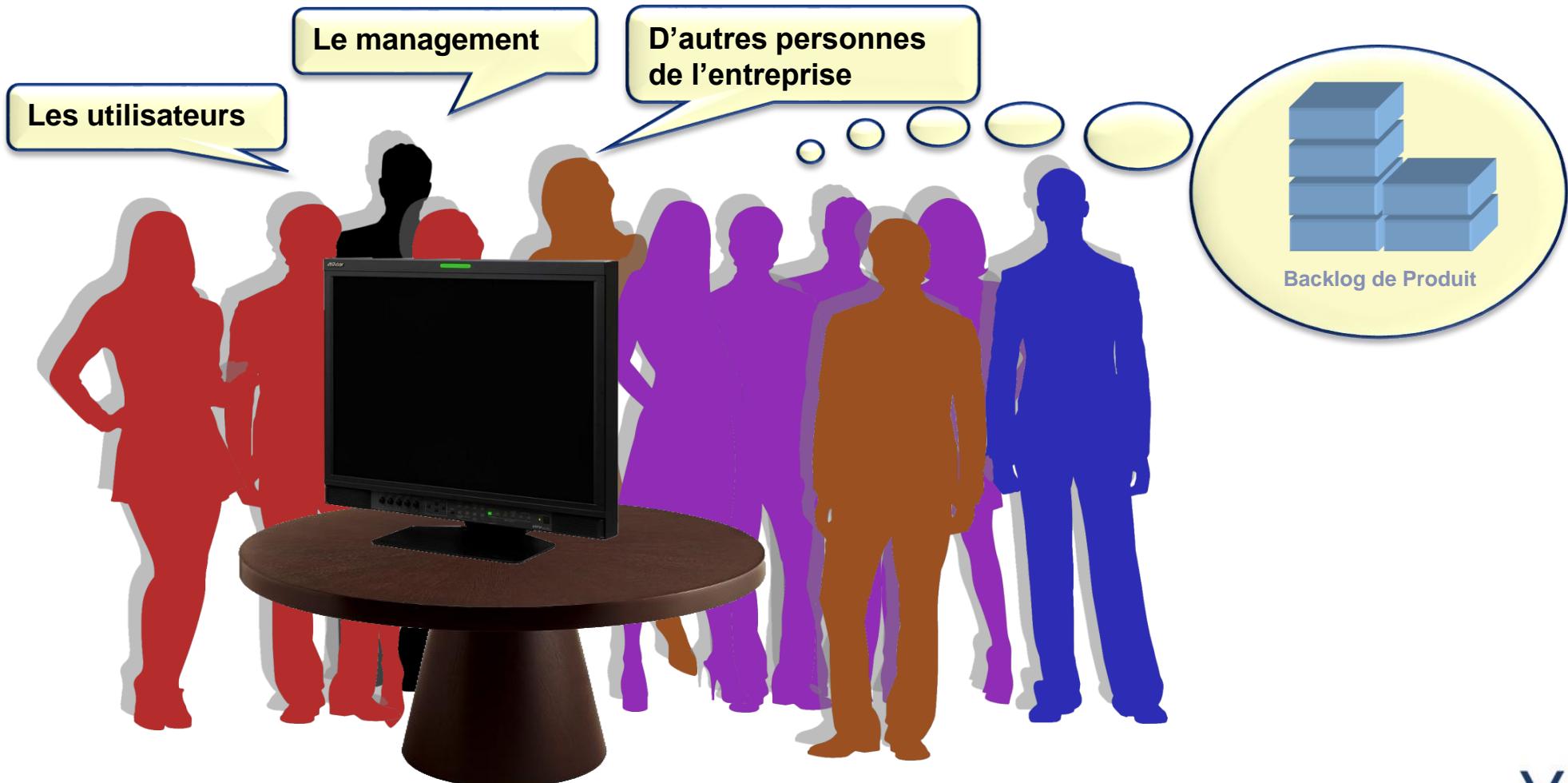


Montrer le travail réalisé : la démonstration



Revue de sprint

- ❖ **Une démo devant tout le monde.**
 - Même s'il y a peu de choses à montrer.
 - Prouver l'avancement, prouver le bon fonctionnement.



Une validation du travail effectué

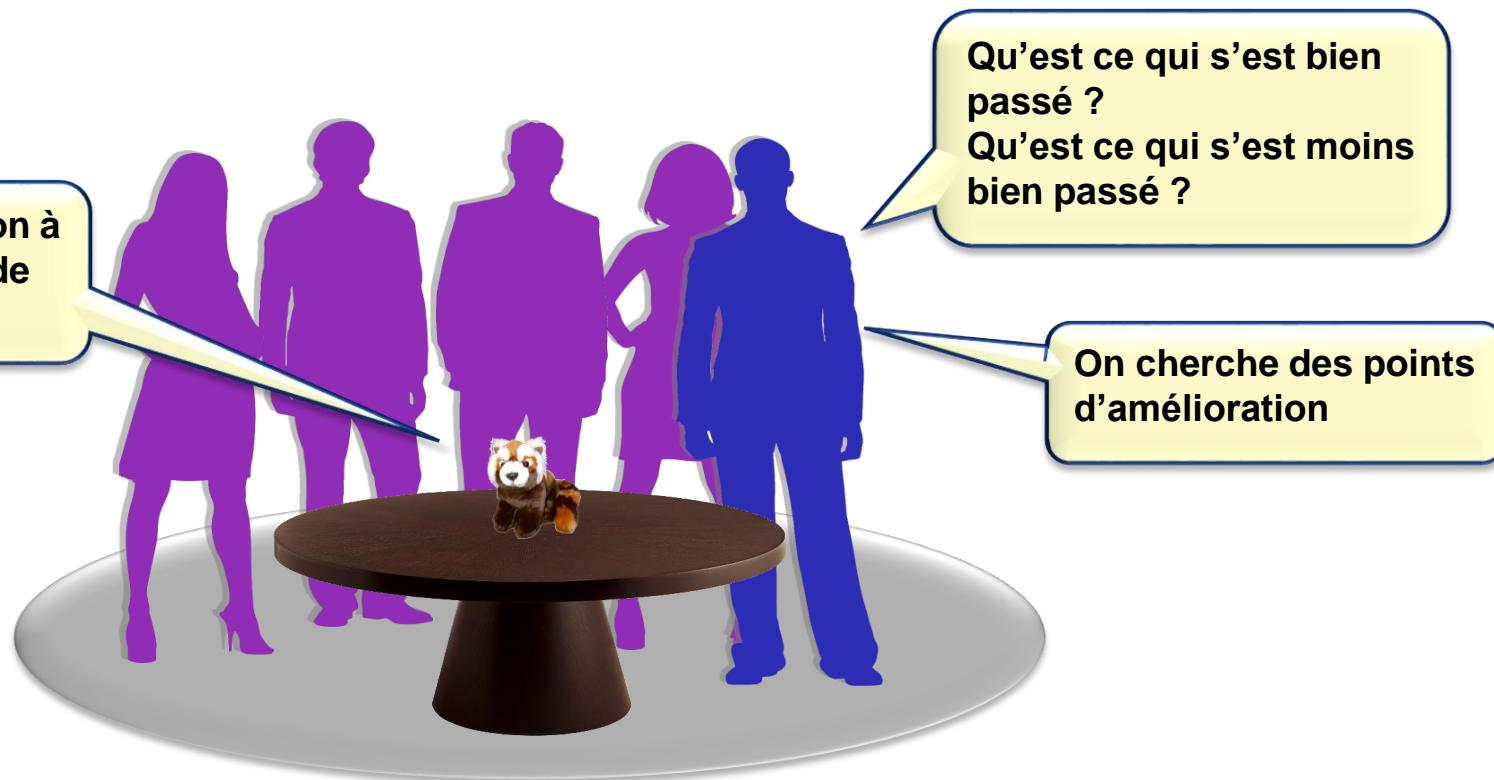
- ❖ On vérifie que les éléments d'engagement pris par l'équipe en début de sprint sont respectées. C'est la base de la responsabilisation.
- ❖ Permet la validation des items livrés sur la base des tests d'acceptance.
- ❖ Ainsi que la mise à jour du Backlog du produit :
 - Ajout de nouveaux items (US, évolutions),
 - Suppression de US obsolètes.
- ❖ Ebauche de la vision pour le prochain sprint.

Faire un bilan : la rétrospective

début



Rétrospective du sprint



Inspecter et adapter

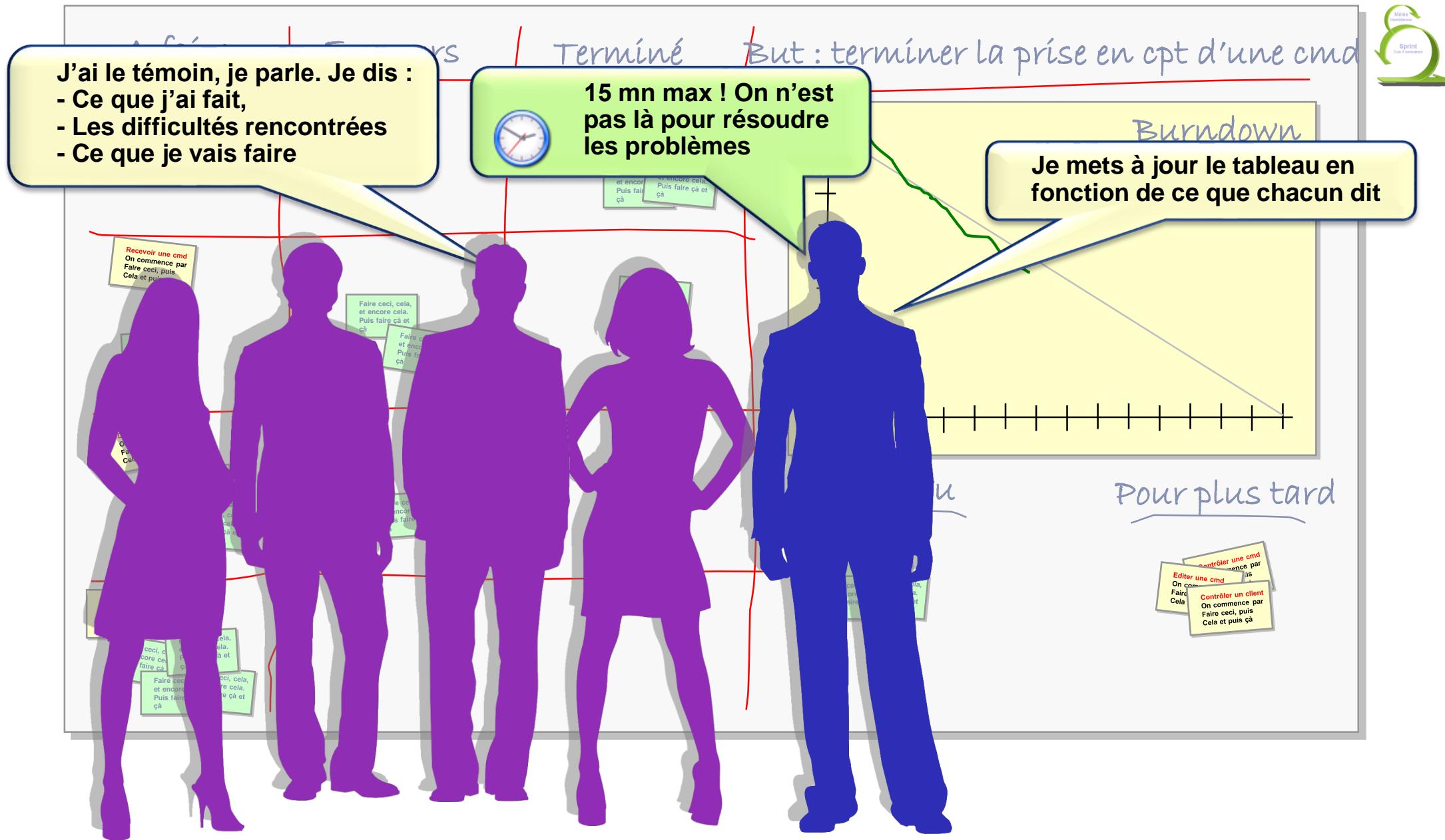
- ✓ Réunion très importante pour l'équipe :
 - Permet de prendre du recul,
 - On se pose la question de l'amélioration pour avancer plus vite,
 - Sur tous les sujets !
- ✓ Elle doit être préparée (par le Scrum Master notamment) ...
 - Attention à la monotonie.
- ✓ ... et son résultat suivi !
 - On y établit un plan d'action,
 - Qui est suivi par le SM lors du Sprint suivant.

Scrum au quotidien

Le daily meeting



Scrum au quotidien



Le daily meeting en détail

- ❖ **Le but de cette réunion :**
 - L'équipe se synchronise sur le plan des tâches à réaliser sur la journée,
 - Chaque membre informe et est informé des différents développements en cours sur le système.
- ❖ **Devant le tableau, à une heure fixe.**
 - Cette réunion n'est pas ouverte à tous (Equipe, Scrum Master).
- ❖ **Debout.**
 - Pas de stimuli extérieur.

Identifier, classer, prioriser le besoin en Scrum

Définir la cible: vision produit

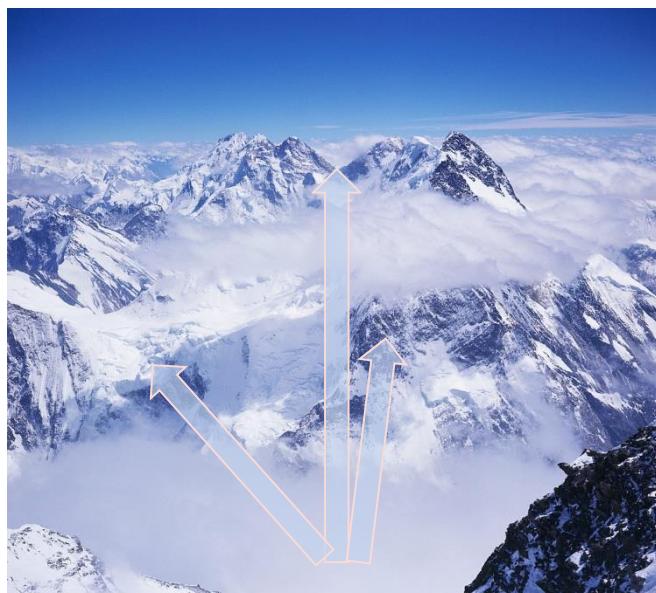


Vision produit

✓ Définition de la vision :

- L'objectif doit être clair pour tous,
- La valeur ajoutée du nouveau système est identifiée,
- Des indications temporelles sur la réalisation du système sont fournies.

Sans la vision



Nous devons atteindre ce sommet



Avec la vision



Nous allons ici



La définition du périmètre

- ✓ Définir le périmètre du système, c'est identifier les fonctionnalités devant être portées par le système.
- ✓ Chacune des ces fonctionnalités devra être détaillée jusqu'à un niveau permettant sa compréhension.



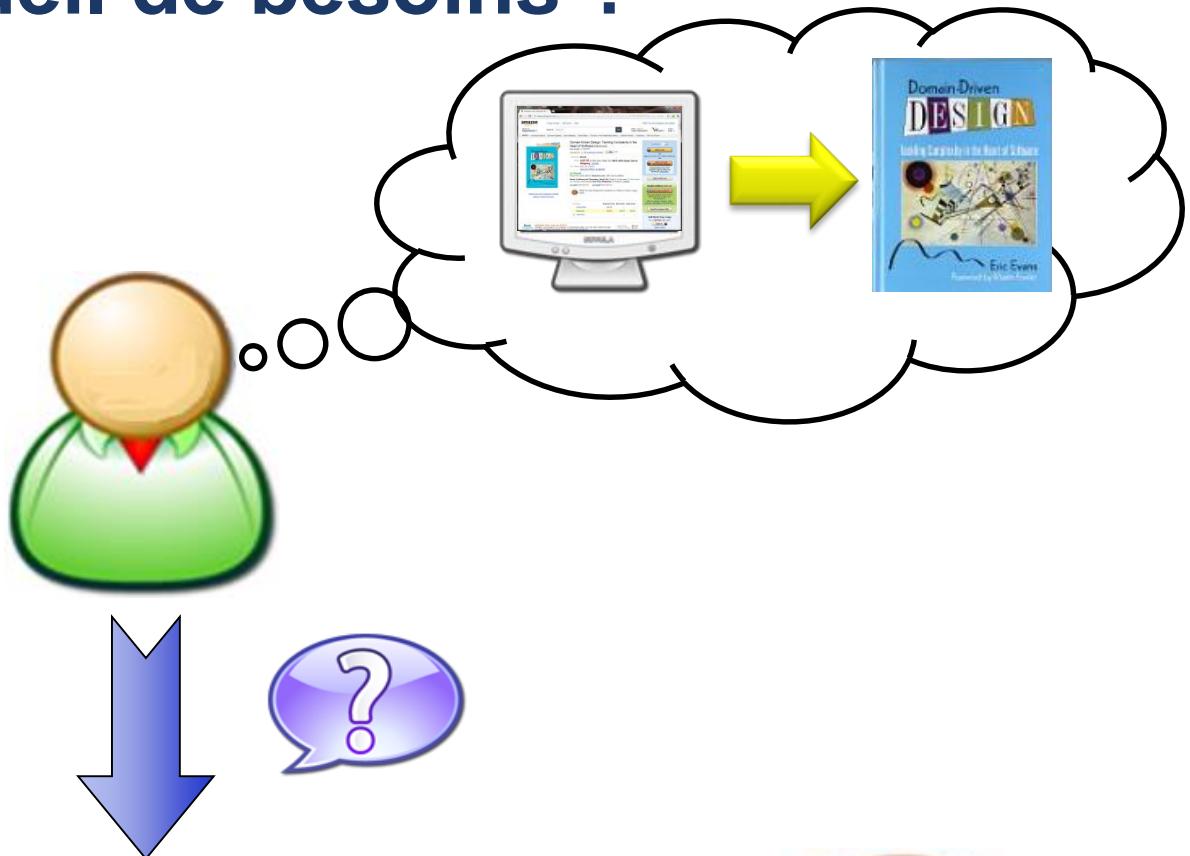
Toutes les fonctionnalités n'ont pas besoin de la même profondeur pour être comprises.

Recueillir les besoins dans les projets agiles

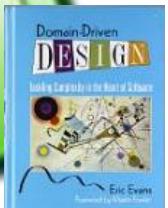
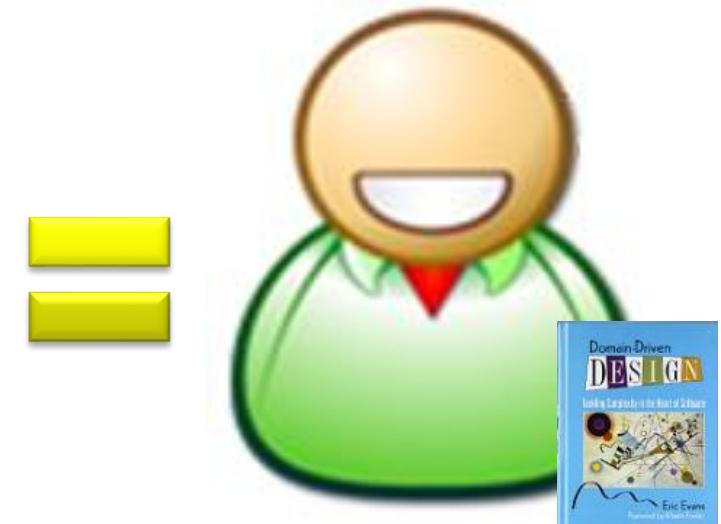
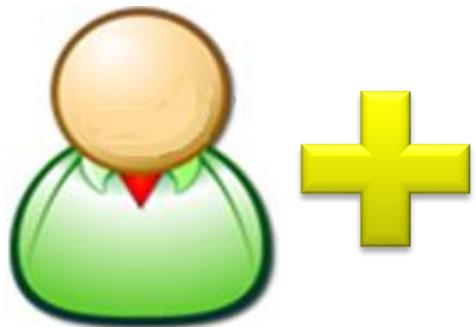


Qu'est-ce que le recueil de besoins ?

Comment passer de cela



A cela ?



Plus précisément

- ❖ Définition et classification de ce qui est nécessaire au bon fonctionnement du système à construire
- ❖ Action indispensable
 - Permet de spécifier, à larges mailles mais sans ambiguïté, les enjeux, le contexte, la complexité et le périmètre du produit
- ❖ Bénéfices
 - Apporte légitimité et aisance au consultant et à toute l'équipe
 - Facilite la réussite du projet et l'atteinte de ses objectifs
- ❖ Pièges à éviter
 - Tout définir de façon détaillée
 - Valider l'ensemble des besoins avant de développer

La valeur client

- ✓ L'expression du besoin doit être guidée par la valeur client
 - Nécessite de connaître les clients du système



- ✓ Qui sont les clients ?
 1. Ceux qui paient le système
 2. Ceux qui vont utiliser le système
 3. Ceux qui vont gérer le système
 4. Celui qui vont en retirer de la valeur : les clients du client

- ✓ La valeur dépend du type de client
 1. Intéressés par le ROI, la stratégie
 2. Intéressés par l'ergonomie, la performance, la simplicité et les services rendus dans le cadre de leur métier
 3. Intéressés par l'exploitabilité, la maintenabilité, la tolérance aux pannes, la documentation
 4. Intéressés par le rendu de service : rapidité, coût, choix, simplification, etc.

Comment découvrir la valeur client ?

- ❖ Appliquer les techniques de la psychologie appliquée au marketing
 - Personas
 - Story telling
- ❖ Appliquer les techniques du marketing
 - Innovation Games
- ❖ Appliquer les techniques du design
 - Design Thinking
- ❖ Appliquer les techniques de modélisation
 - Processus métier

Comment commencer un projet Scrum ?

Le Sprint 0

Le Sprint 0 : Comment commencer un projet



Le Sprint 0 pour commencer (1/2)

✓ Son objectif :

- Vérifier la faisabilité du projet,
- Se mettre en ordre de marche pour avancer.



Le Sprint 0 n'a pas pour objectif la réalisation du système, mais uniquement de vérifier sa faisabilité.

✓ Instanciation de la méthode projet :

- Identification des processus,
 - Livrables, quand et pour qui
- Taille des sprints,
 - Entre 2 et 4 semaines
- Définition des réunions et de leurs aspects pratiques,
 - Durée, fréquence, intervenants, lieu,...
- Mise en place des bonnes pratiques,
 - Une équipe dans un bureau, espace d'affichage, ...

Le Sprint 0 pour commencer (2/2)

✓ Et également :

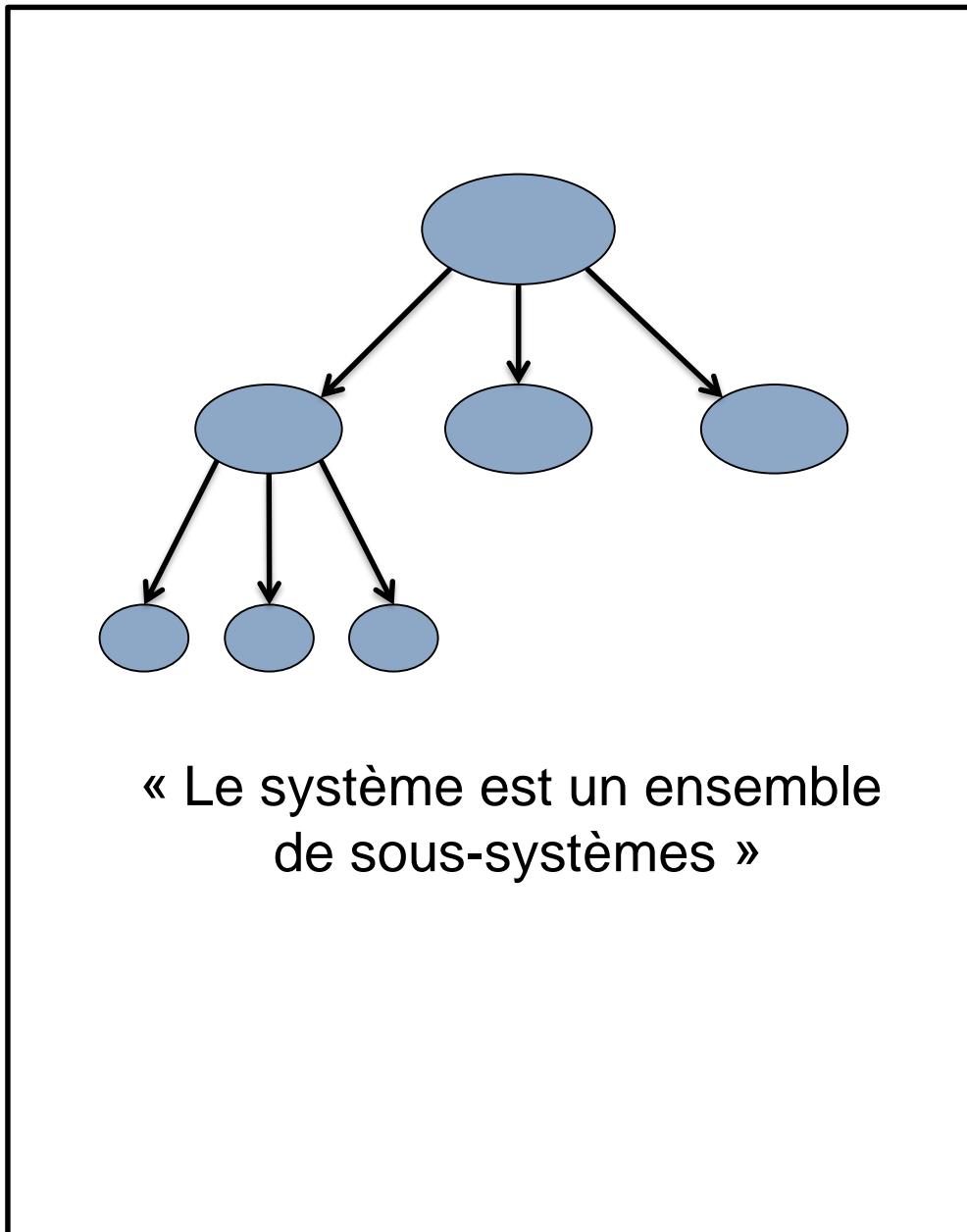
- Constitution de l'équipe et son périmètre,
- Définition de la cible,
- Cartographie de l'application et des priorités,
- Estimation et planification,
- Mise en place plateforme technique et validations techniques.

✓ Les livrables :

- Document de vision,
- Backlog du produit initialisé et priorisé,
- Liste des risques identifiés,
- Première estimation du coût du projet,
- Prototype (POC) permettant de lever les principaux risques identifiés comme bloquants,
- Définition de l'environnement technique.

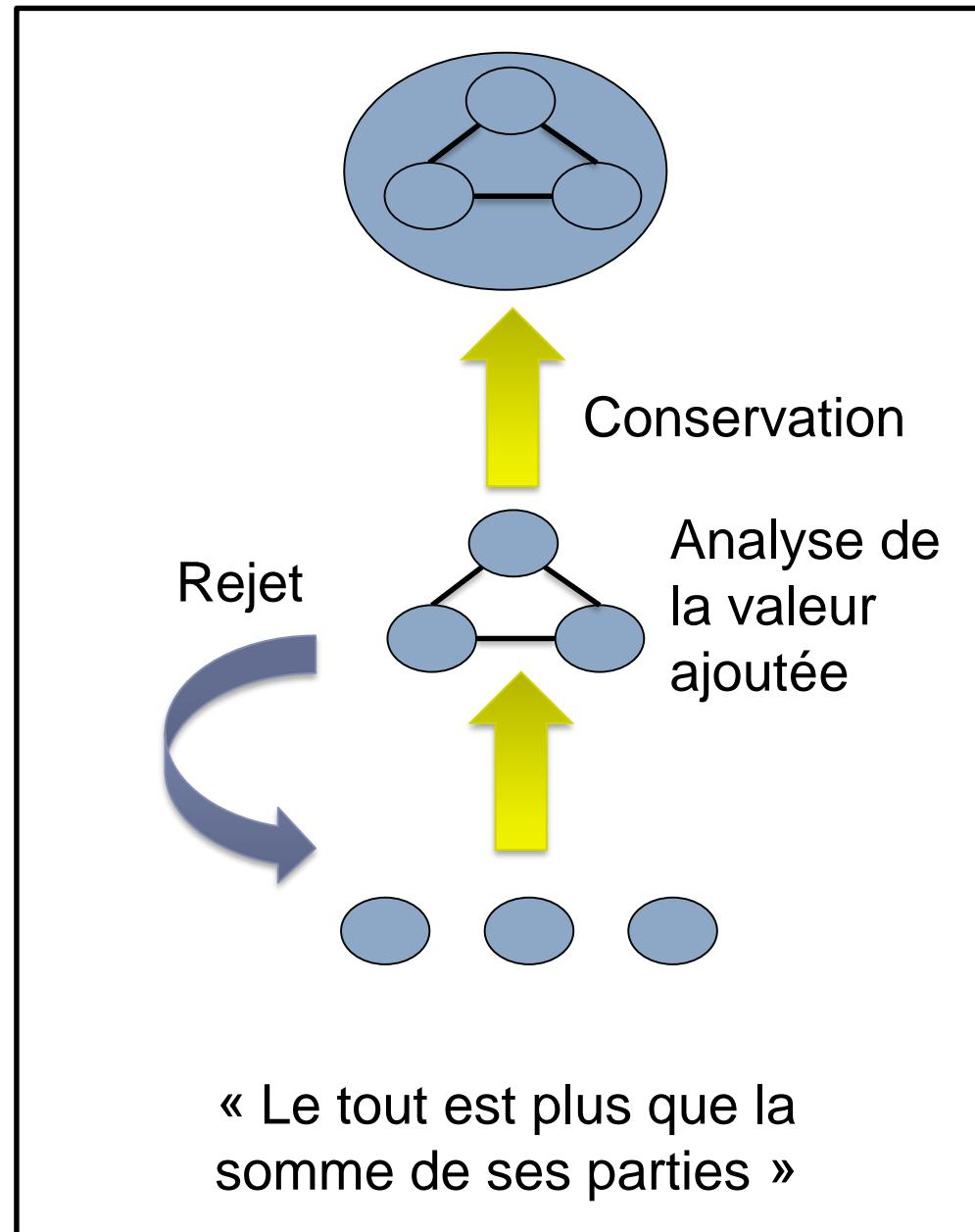
Approche réductionniste

- ❖ Définir et construire le système à partir d'une décomposition fonctionnelle
 - Décomposer chaque fonction en sous-fonction, etc.
- ❖ Nécessite de faire la bonne décomposition dès le début
 - Vision cascade
- ❖ Le besoin a une complexité de même ordre que celle du logiciel
 - Il est très difficile à formaliser
 - Il n'est pas exécutable donc quasi-impossible à vérifier



Approche Holistique

- ✓ Définir et construire le système par parties qu'on intègre au fur et à mesure
 - La complexité est atteinte par émergence
- ✓ Le système est construit de façon itérative et incrémentale
 - Vision agile
- ✓ Le besoin est simple
 - Il est guidé par la valeur utilisateur
 - C'est le système qui est vérifié et non le besoin



Etapes d'un recueil de besoin

✓ Recueillir

- Utiliser l'une ou l'autre des techniques questionnaires :interviews, observation directe, ateliers, réunions, jeux sérieux, brainstorming ...

✓ Restituer

- Par écrit, maquette, schémas
- Systématiquement et « brut de fonderie »

✓ Analyser

- Lever les ambiguïtés et incohérences, éliminer les redondances, lister les manques et surtout simplifier, initialiser et enrichir la documentation cible

✓ Montrer

- Sous la forme d'un atelier

Le leitmotiv permanent, à chaque étape : SIMPLIFIER !

Cartographier et prioriser : story map



La problématique

- ❖ Décrire le besoin du client.

Ne pas perdre les objectifs et enjeux.

- ❖ Décrire la solution.

Echanger pour définir la bonne solution.

- ❖ Estimer.

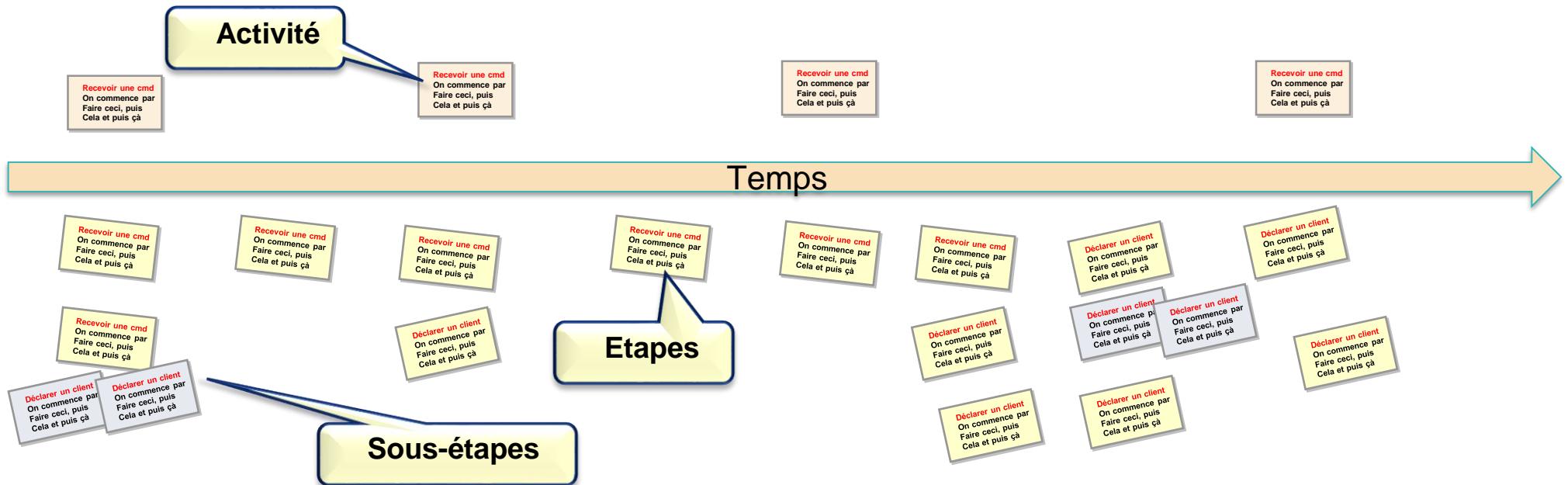
Evaluer le coût et retour sur investissement.

- ❖ Prioriser & Planifier.

Commencer par les choses à forte valeur ajoutée.

Une organisation basée sur l'expérience utilisateur

- ✓ Identifier les activités principales d'un utilisateur
 - On regroupe les étapes et les sous-étapes utilisateurs.



- ✓ Permet de visualiser le flux et la décomposition de chaque activité et des étapes associées.



Les bénéfices de la story map

- ❖ La représentation utilisée est centrée sur les besoins des utilisateurs avec un enchainement d'activités.
 - Les fonctionnalités clés sont plus rapidement identifiables.
 - Il est plus simple de visualiser la complétude du périmètre de l'application.

- ❖ La story map est une représentation visuelle des éléments fonctionnels du backlog.
 - Ne contient pas les éléments techniques.

Décrire les users stories

Décrire: les stories et le product backlog



Les stories et le Backlog

- ✓ Chacun des besoins est appelé « User story » en Scrum.
 - C'est le Product Owner qui les écrit.
- ✓ Ces stories racontent un fragment de l'application.
 - L'ensemble des stories forme le périmètre de l'application.
- ✓ Les stories sont répertoriées dans le Backlog du produit.
 - Artéfact géré par le Product Owner.
 - Référence fonctionnelle du projet.
 - Crée à l'itération 0.
 - Mise à jour tout au long du projet.
- ✓ Les User Stories permettent d'exprimer un besoin fonctionnel.

La story

Visualiser notre catalogue sans filtre

En tant que : Client

Je veux : visualiser le catalogue

Afin de : afin de pouvoir consulter les produits

Description : Le client se connecte sur le site et visualise le catalogue

Priorité : 100

Estimation : 4

Critères d'acceptance :

- *Les catégories sont affichées sur la page avec l'image d'un produit représentatif de la catégorie.*
- *En sélectionnant la catégorie l'ensemble des produits constituant cette catégorie sont affichés.*

❖ **Décrire le besoin.**

En tant que.. Je veux que... Afin de...

❖ **La story est estimée.**

Taille (relative) : chiffre ou mettre XS, S, M, L, XL

❖ **La story est priorisée.**

Chaque story a une priorité unique

❖ **La story décrit la solution avec les critères d'acceptance.**

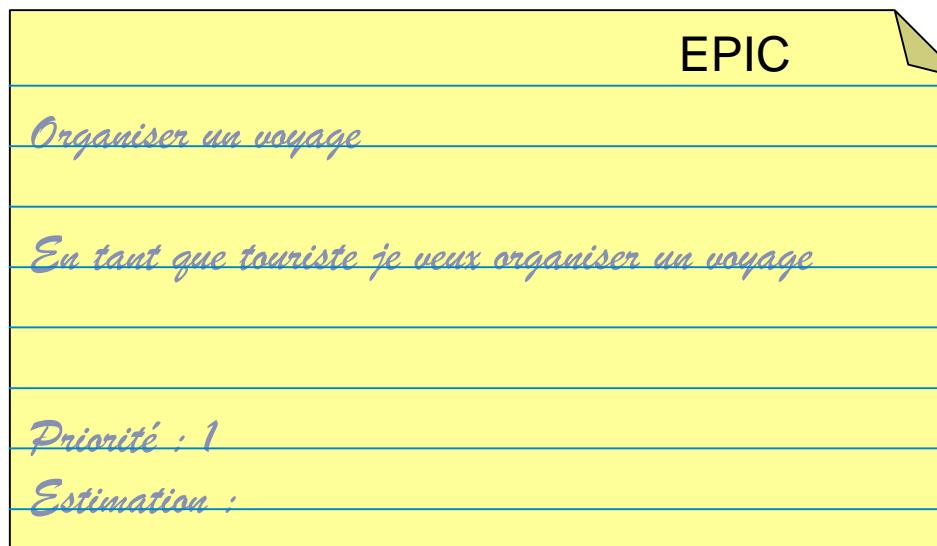
Décrire les tests qui permettent de valider la story.

Granularité

- ❖ Le découpage du projet en User Stories se fait en désagrégant petit à petit les besoins trop gros.
- ❖ Pour noter la maturité d'un besoin on utilise INVEST :
 - Indépendant (le plus possible les unes des autres),
 - Négociable (pouvant être discutées jusqu'au sprint),
 - Valorisant (apporte de la valeur ajoutée à l'utilisateur),
 - Estimable (que l'on peut chiffrer),
 - Suffisamment petit (doit tenir dans un sprint),
 - Testable (des tests ont été décrits).

Que faire si non INVEST

- ❖ Les stories non INVEST sont des EPIC.
- ❖ Elles ne peuvent pas intégrer un Sprint et doivent être retravaillées :
 - Découpage en entités fonctionnelles plus petites,
 - Découpage technique,
 - Priorités différentes au sein de la story,
 - Etc ...



Les technical Stories

- ❖ Les technical Stories représentent les besoins non fonctionnels, mais nécessaires à la réalisation du système.
- ❖ Elles doivent être intégrées au technical Backlog.
- ❖ Chaque technical Story doit être estimée en heures et non en effort et intégrée au planning.

Attention



Les technical Stories sont utilisées dès lors que la charge de travail est transverse à plusieurs Stories.

Les Spikes

- ❖ Le Spikes sont des éléments pour lesquels la solution n'a pas été trouvées :
 - Problème pour interfaçer un logiciel tiers.
 - Problème pour adresser une solution fonctionnelle
- ❖ Lorsque l'équipe a besoin de temps pour réfléchir sur un problème très précis on définit un Spike.
- ❖ Le Spike :
 - A une durée,
 - A un titre précis,
 - A un but précis,
 - Doit être planifié.

Définir le périmètre de l'équipe : signification de « terminer »



La notion de « Terminé » (DONE)

- ✓ Cette notion est importante car elle détermine l'ensemble des actions et livrables devant être réalisés pour pouvoir considérer une tâche ou une User Story comme terminée.
- ✓ La définition du « DONE » doit intervenir dès le Sprint 0. Il s'agit de se mettre d'accord sur « les règles du jeu ».
Mais ne pas hésiter à la compléter si nécessaire !
- ✓ Avoir des règles très strictes et nombreuses a un coût qui sera inclus dans l'estimation des User Stories.
- ✓ Elle doit être affichée !



Le « DONE » pour les tâches et User Stories

✓ « DONE » permet de définir le périmètre d'action de l'équipe.

✓ Pour les tâches « DONE » signifie :

- Le code est terminé,
- Le code compile,
- Le code est commenté,
- Le code est « commité »,
- Les tests unitaires passent,
- La documentation est à jour.



Ces listes ne sont que des exemples que nous rencontrons le plus souvent, mais ces critères doivent être le fruit d'un accord entre le Product Owner et l'équipe.

✓ Un besoin fonctionnel est « DONE » lorsque :

- L'ensemble des tâches associées sont « DONE ».
- Les tests d'acceptance passent.

Décrire le backlog



Le backlog de produit

- ❖ Il est constitué de l'ensemble des étapes trouvés lors des story mapping. Ces étapes sont appelées « Story ».
- ❖ C'est la référence fonctionnelle du projet.
- ❖ Il doit être priorisé et maintenu à jour en permanence.

Définir mon projet, choisir les étapes intermédiaires



La planification agile

- ✓ Très précise pour le futur très proche.
- ✓ De plus en plus imprécise dans le temps.
- ✓ On utilise notre expérience passée pour estimer le temps nécessaire pour le futur.
- ✓ On estime de façon relative et sans unité.

Gérer les risques



Identifier les risques

- ❖ **Le pilotage par les risques doit être au cœur de votre projet.**



- ❖ **Il s'agit de répertorier et d'analyser les risques pouvant affecter le déroulement du projet.**
- ❖ **Chaque risque doit être qualifié en terme de criticité.**
- ❖ **A chaque risque doit correspondre un plan d'action permettant de le supprimer ou de le réduire.**

Chiffrer, Planifier, Communiquer

Estimer: story points



Les grands principes

- ✓ **Une estimation est attribuée à chacune des User Stories.**
- ✓ **Estimer le coût du projet, estimation macro :**
 - **Les besoins n'ont pas besoin d'être détaillés,**
 - **L'ensemble des besoins exprimés sont estimés.**
- ✓ **Estimer l'effort des User Stories pour déterminer le contenu d'un sprint :**
 - **Aider le Product Owner à prioriser ses besoins.**
- ✓ **Une estimation relative :**
 - **Chaque User Story est estimée en effort nécessaire à sa réalisation,**
 - **Elle est représentée au travers des Story points.**

Nouvelle métrique pour nouvelle méthode

- ✓ Le délai ou la charge ne sont pas des indicateurs fiables.
 - Dépendent de trop de paramètres.
 - Varient d'une personne à l'autre.
- ✓ Au lieu d'utiliser la charge, on peut facilement qualifier l'effort :
 - Les gens de l'art convergent facilement sur l'effort à mener,
 - Echelle décorrélée du temps,
 - Permet de comparer deux stories facilement.
- ✓ On cherche donc une métrique pour qualifier l'effort nécessaire à la réalisation des User Stories.
 - Le story point.

Les story points

- ✓ **Les story points ne sont pas des heures, ni des jours.**
- ✓ **Les story points ne sont pas formalisés, on peut utiliser :**
 - La suite de Fibonacci 0, 1, 2, 3, 5, 8, 13, 21, etc ...,
 - La taille de T-Shirt XXS, XS, S, M, L, XL, XXL,
 - Tout référentiel parlant pour l'équipe.
- ✓ **La valorisation de l'effort estimé est propre à :**
 - Une équipe,
 - Un projet.
- ✓ **Deux stories d'efforts comparables doivent avoir le même nombre de story points.**

Le planning poker

- ✓ « Jeu » agile préconisé par Scrum.
- ✓ L'équipe utilise un jeu de cartes représentant les story points.



- ✓ Une User Story de complexité moyenne est utilisée comme étalon avec un complexité arbitrairement positionnée à 5 (ou 3, ou 8).
- ✓ Chaque membre de l'équipe évalue en story points les user stories proposées par le Product Owner.
 - En cas de désaccord, l'équipe échange entre elle pour arriver à un consensus.
 - S'il y a convergence, l'évaluation est notée sur la user story dans le Backlog, sinon on recommence.

Quand peut-on chiffrer une story ?

- ❖ Se poser la question INVEST sur chaque story avant de la chiffrer, sinon refuser la story.
- ❖ Lorsque la story a beaucoup changé par rapport à son premier chiffrage.
- ❖ Lorsque la production de précédentes stories nous a appris des éléments modifiant de manière importante la compréhension de la story.



On ne re-chiffre pas tout, tout le temps.

Validation

Planning Poker

Planifier: capacité de production et vitesse



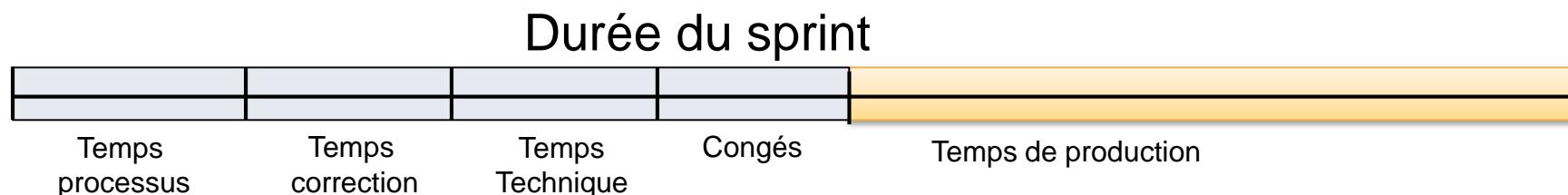
La vélocité

- ✓ La vélocité est le nombre de story points réalisés (notion de « DONE ») sur une période donnée.
 - Elle représente la production effectuée lors du sprint.
 - vélocité (SP/j) = Nombre de SP réalisés / temps de production
- ✓ Elle permet d'estimer la productivité à venir en calculant un coefficient :
 - Moyenne de l'ensemble des vélocities
 - Moyenne de la vélocité sur les 3 derniers sprints,
 - Moyenne de la vélocité sur les 5 derniers sprints moins les extrêmes.
 - Etc ...

Vélocité et productivité

La capacité de production est le nombre de story points que l'on va réaliser dans le prochain Sprint :

- Déduire la capacité de production à venir, à partir des vélocités passées est une bonne pratique,
- Elle doit pourtant être pondérée par plusieurs facteurs variables entre les sprints
 - Temps processus,
Les réunions et tout ce que la méthode implique.
 - Temps correction,
Il existe une liste de défauts qu'il va falloir intégrer au fil de l'eau (),
 - Temps technique,
Certains aspects techniques demandent un temps de réflexion
 - Disponibilité de l'équipe,
Toutes les absences planifiées.



Capacité de production

- ✓ Détermination de la capacité de production de l'équipe pour le prochain sprint.

vélocité (SP/j) = nombre de SP réalisés / temps de production

capacité de production (SP) = temps de production * vélocité moyenne

- ✓ Vélocité : c'est du constaté
- ✓ Capacité de production : c'est du planifié

Evolution de la vélocité

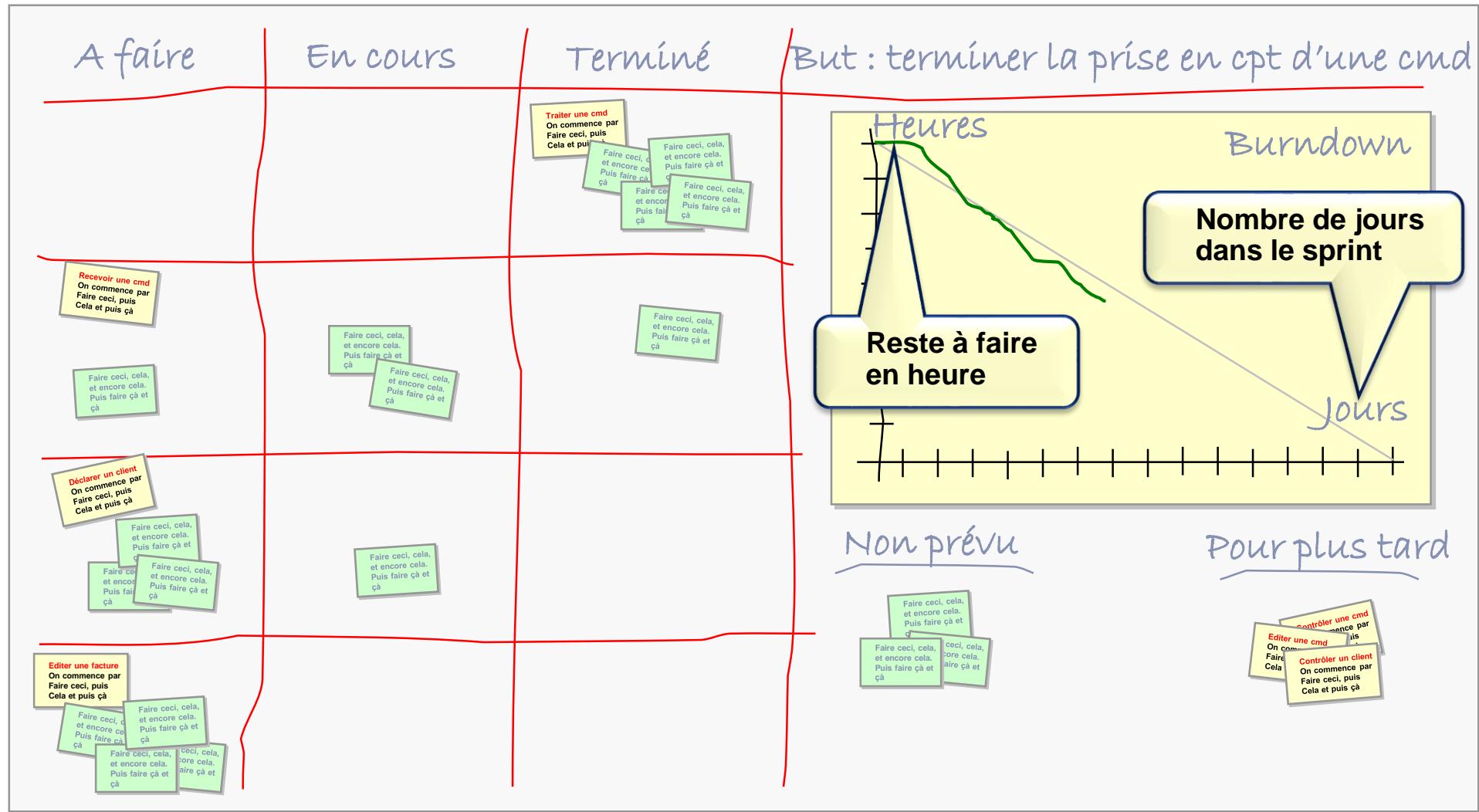
- ✓ Les chiffrages en story points ne sont pas remis en cause au cours du projet (sauf modification majeure).
- ✓ Par contre c'est la vélocité qui évolue :
 - Avec le temps, l'équipe prend confiance et donc réussit plus de stories dans le même temps donc plus de story points,
 - Avec l'expérience, l'équipe appréhende mieux les stories et donc les réalise plus rapidement,
 - L'équipe apprend à travailler collectivement.
- ✓ La variable d'adaptation est donc la vélocité.
C'est un très bon indicateur de l'avancement du projet vis-à-vis du management.
- ✓ La vélocité doit s'améliorer avec le temps et l'expérience jusqu'à une valeur idéale.

Le suivi visuel en Scrum



Un suivi de l'avancement du sprint

Le Scrum Board



Les cas particuliers

- ❖ L'équipe de développement est en avance :
 - Remboursement de dette technique, correction d'anomalies,
 - Discussion avec le Product Owner pour ajout d'une US.
- ❖ L'équipe est en retard:
 - Négociation avec le Product Owner pour supprimer une US,
 - On décale la fin du sprint.
- ❖ Le but du Sprint change :
 - Négociation avec l'équipe pour échanger quelques US.



Rappelez-vous :

" Les individus et leurs interactions plus que les processus et les outils "

Actualiser les prévisions et objectifs : vitesse, burndown et backlog



Mise à jour du suivi du projet

- ❖ On calcule la vélocité de l'équipe.
On suit son évolution et on essaie de la projeter dans le temps.
- ❖ On met à jour le release burndown ou le Burnup du projet.
- ❖ On actualise le backlog en ajoutant des éléments que l'on a pu découvrir lors de la démo.

Kanban

Pourquoi Kanban



Origine de Kanban

- Méthode en provenance de l'industrie, mise en place par la société Toyota dans les années 1960
- Fondée sur le principe de production sur besoin (Juste à temps) :
 - Limite la production du poste amont aux besoins exacts du poste aval
 - Permet de mieux gérer les stocks et d'éliminer le gaspillage
- Kanban = « Fiche » ou « Etiquette » (terme japonais) fixée sur les éléments produits
 - Chaque fiche représente un nombre d'éléments à utiliser
 - Chaque poste de production possède plusieurs fiches
 - Dès que le nombre de fiches « consommées » passe sous un seuil défini, un ordre de fabrication est déclenché pour réapprovisionnement
 - La chaîne n'est pas arrêtée, le nombre de fiches restantes est suffisant pour attendre le réapprovisionnement
- Adaptée pour le développement de logiciels

Quand utiliser la méthode Kanban

- **Kanban est une approche de gestion du changement.**
- **Ce n'est pas**
 - **Un processus de développement,**
 - **Un cycle de vie d'un logiciel,**
 - **Une méthode de gestion de projet**
- **Cette méthode est adaptée :**
 - **A l'ensemble des projets de développement de logiciel.**
 - **A la maintenance applicative.**
 - **A L'activité de support dont les priorités changent très rapidement.**

Objectif de la méthode Kanban

- **Kanban s'articule autour des deux principes fondamentaux suivants :**
 - **La visualisation de la chaîne de création de valeur.**
 - **La limitation du « Travail A Faire » (TAF).**

Kanban et les rôles

- La méthode Kanban ne prescrit aucun rôle.
 - Avec Kanban les équipes multidisciplinaires sont facultatives.
 - Le tableau peut ne pas être détenu par une équipe spécifique. Un tableau est lié à un Workflow unique, pas nécessairement à une seule équipe.
- Chaque membre de l'équipe sera affecté à la réalisation des tâches présentes dans les différentes colonnes.

Les rôles disponibles

– Le Product Owner :

- Garant et promoteur de la vision produit
- A délégation pour définir les priorités et maximiser la valeur produite
- Collabore avec l'équipe au quotidien
- Génère et collecte les feedbacks
- Décide quand et quoi livrer

– L'équipe de réalisation :

- Groupe de 7 ± 2 personnes
- Engagements au sein de l'équipe
- Auto organisée pour atteindre l'objectif
- Produit et délivre de la qualité
- Pluridisciplinarité des acteurs

– Le Concepteur du Système Kanban :

- Facilite le travail de tous
- Protège l'équipe des interruptions
- Assure l'élimination des obstacles
- Champion de la méthode
- Propose des pistes d'amélioration

La tableau Kanban



Déterminer le processus

- **Le processus est l'ensemble des étapes nécessaires à la livraison du besoin.**
- **Il faut dans un premier temps identifier les différentes étapes qui constituent le processus.**
- **Chaque projet met en place son propre processus.**

Construire le tableau

- **Le tableau Kanban doit refléter les différentes étapes du processus retenu.**
 - En colonnes les étapes du processus.
 - En ligne les différents besoins qui vont transiter par chacune des colonnes.
- **Chaque besoin va être traité par différentes personnes en fonction des compétences requises**
 - Il faut identifier les acteurs pouvant travailler sur les différentes étapes du processus
- **Chaque colonne doit faire apparaître deux sous-colonnes afin de connaître :**
 - Les éléments en-cours de réalisation.
 - les éléments terminés et donc disponible pour l'étape suivante.

Le temps de cycle

- Le temps de cycle le temps de cycle est le délai écoulé entre le moment où un besoin est pris en charge, et la satisfaction de ce besoin.
- C'est le principal indicateur de la méthode.
- Ce temps de cycle inclut les phases de réalisation propre à chaque colonne, mais aussi le temps d'attente entre 2 colonnes
 - L'objectif est de diminuer ces temps d'attentes afin de faire transiter les besoins le plus rapidement depuis la première colonne jusqu'à la dernière colonne du tableau Kanban.
- Dans cet organisation ou le temps de cycle doit être optimisé, il faut résorber très rapidement les goulots d'étranglement.

Les goulots d'étranglement

- Il arrive dès lors que les éléments d'une colonne ne sortent pas de cette colonne au même rythme que la mise à disposition des nouveaux éléments en entrée de cette même colonne.
 - Par manque d'information.
 - Un bug inexplicable.
 - Une tâche qui ne se termine pas.
- Ils ralentissent le temps de cycle.
- Un des premiers réflexes est de le mettre de côté pour plus tard.
- Il faut en faire les identifier et les traiter le plus rapidement possible.

Le TAF (Travail A Faire)

- **Le TAF est le nombre d'élément maximum qu'une colonne peut contenir.**
 - Chaque colonne possède son propre seuil.
 - Il n'est pas possible de prendre une nouvelle tâche si celle-ci fait passer le compteur de tâche au dessus du seuil.
- **La détermination du seuil de chaque colonne peut se faire de manière empirique.**

Nombre de personne travaillant dans la colonne * 2 - 1 = TAF

- **La mise en place de ce seuil permet d'identifier les goulets d'étranglement et donc de les résorber très rapidement.**
- **Les personnes affectés au colonnes amont mais bloqués par ces goulets d'étranglement viennent aider le temps de la résolution du problème et de la résorption du goulot identifié.**

Ajuster la limite Kanban (TAF)

- **Expérimenter**
 - En fonction de la taille de l'équipe
 - Pour réagir, ni trop tôt, ni trop tard, en cas de problème
 - Pour améliorer la productivité et le temps de cycle moyen
- **Exemples**
 - Limite Kanban trop petite => Les gens sont inoccupés
=> Mauvaise productivité
 - Limite Kanban trop élevée => Stories non traitées
=> Mauvais temps de cycle

Le management visuel

- L'objectif est de fournir toutes les informations nécessaires en un coup d'œil
 - Le tableau Kanban.
 - Les colonnes représentatives du processus identifié.
 - Les stories positionnées dans les différentes colonnes.
 - La personne en charge de la story
 - Le TAF de chaque colonne
 - Les règles du « Done » associé à chaque colonne
- Ces éléments permettent :
 - De connaître l'avancement des différents besoins.
 - De rapidement visualiser les goulets d'étranglement.
 - De connaître les éléments pouvant intégrer la prochaine livraison.

La gestion des flux



Flux poussé

- **Un système adoptant le flux poussé vise à produire un travail en prévision d'un éventuel besoin.**
- **Les analystes produisent des spécifications qui seront envoyées aux développeurs quelque soit leur capacité de travail.**
- **Le principe est donc de produire au maximum des capacités afin de fournir un stock pour les étapes suivantes sans se soucier de la capacité de production des étapes avales.**

Flux tiré

- **le principe du flux tiré bouleverse la façon d'appréhender la performance de la production.**
- **Les analystes produisent des spécification à la demande des développeurs en fonction de leur capacité de production à venir.**
- **L'objectif n'est plus de faire produire au maximum des possibilités, mais de faire s'écouler un flux, le plus régulièrement et le plus rapidement possible, en fonction de la demande de la ressource aval.**
- **Si les équipes sont jugées en fonction du taux d'occupation de leurs ressources, il y a manifestement une incohérence. Le temps de défilement et le niveau des travaux en attente de réalisation sont par contre des mesures qu'il faut apprendre à utiliser.**
- **Pour palier à l'inoccupation des personnes, celles-ci peuvent être affectées à différentes colonnes afin de lisser leur charge de travail.**

La gestion des besoins avec Kanban



User Stories

- L'ensemble des besoins utilisateurs sont rédigés sous forme de User Stories.
 - Ils sont positionnés en entrée du Tableau Kanban.
 - Ils doivent être priorisés d'après leur valeur métier par le Product Owner.

L'estimation des besoins

- Kanban ne préconise pas l'utilisation d'estimation de charge ou d'effort de réalisation.
- Kanban s'appuie sur le temps de cycle moyen pour pouvoir planifier et estimer les livraisons :
 - Estimer une charge de travail ne permet pas de connaître le délai de réalisation.
 - Les temps d'attente sont généralement supérieurs au temps de réalisation.
 - Le temps de cycle moyen est l'indicateur permettant de connaître la date de mise à disposition des différents besoins.

La granularité des éléments

- Afin de tirer le meilleur du système kanban, il est recommandé de découper les besoins en éléments de taille similaire.
- La taille des éléments doit respecter les critères suivants :
 - lisser la demande :
 - la taille de l'élément doit être suffisamment petite pour permettre la prise en charge rapide de demandes urgentes pouvant arriver à tout instant.
 - avoir un débit suffisant :
 - la mise à disposition des éléments doit être régulier.
 - fonctionner avec du management visuel :
 - le tableau Kanban doit être vivant.

Eléments trop petits vs trop grands

- **Les éléments trop grands :**

- monopolisent une personne durant sa prise en compte,
- favorisent l'émergence des goulots d'étranglement,
- sont vecteur de démotivation car l'impression du travail qui n'avance pas s'installe.

- **Les éléments trop petits :**

- chaque besoin peut devenir dépendant d'un autre,
- la mise a jour du tableau peut devenir fastidieuse.
- maîtriser la livraison dans le respect de la qualité et des coûts demande une certaine expérience.

La planification

- ✓ L'ensemble des besoins identifiés sont placés dans la colonne « A faire » et sont priorisés par le Product Owner.
- ✓ Chaque nouveau besoin va venir s'intercaler dans cette liste en fonction de sa propre priorité.
- ✓ Il n'est pas possible de déterminer à l'avance le contenu d'une livraison puisque les priorités changent à chaque instant.

La livraison

- Kanban n'impose aucun rythme de livraison
 - La livraison d'une nouvelle version englobe tous les éléments disponibles dans la colonne « Terminé ».
 - Le déclenchement d'une livraison dépend donc des éléments disponibles.
 - Livraison immédiate dès lors qu'une anomalie bloquante est résolue.
 - Livraison tous les mois suite à la mise à disposition d'un ensemble fonctionnel cohérent.
 - Livraison au bout de X mois pour l'ajout et la correction d'éléments de priorités mineur.
 - C'est le PO qui déclenche la livraison.
 - Elle peut intervenir :
 - Plusieurs fois par jour pour la résolution d'anomalies bloquantes
 - Tous les jours

Les critères de sortie

- *Des critères de sortie définis pour chaque activité du processus. Ce sont les conditions pour qu'un élément de travail puisse passer d'une colonne à une autre*
- *Ces critères doivent être affichés sur chacune des colonnes.*
- *Il est de la responsabilité de chacun de vérifier ces critères lorsqu'il déplace un élément sur le tableau.*

Scrum / Kanban : des solutions différentes

- ❖ **Limitation du travail en cours :**
 - Scrum : limitation par la durée : sprint,
 - Kanban : nombre de tickets par étapes.
- ❖ **Un suivi visuel :**
 - Le tableau de suivi de tâches Scrum réinitialisé à chaque sprint,
 - Le tableau kanban est persistant.
- ❖ **Amélioration continue :**
 - On cherche à optimiser le flux,
 - On prend le temps de faire des bilans (rérospective).
- ❖ **Livraison régulière :**
 - Périodique (à chaque sprint) pour Scrum,
 - En continu pour Kanban.

Scrum & Kanban



Scrum et Kanban = Scrumban

