

PRODUCT OWNER



Myriam Lajon – decembre

© Groupe SII 2021 | Usage interne SII

2022

8 rue des Pirogues de Bercy, 75012 PARIS

Intervenants



Myriam LAJON
Coach Agile



Yoan DENIS
Coach Agile

Agenda



Naissance de l'Agilité

VUCA
Manifeste Agile



Le produit Agile

Vision
Persona
Story Mapping
Creation Backlog
Pilotage par la valeur



Le Scrum Guide

Théorie
Valeurs
Scrum Team
Evenements
Artifacts



Focus Product Owner

Preparatio
n Exam

1 objectif – 2 axes d'apprentissages

Acquérir le bon mindset pour bien appréhender les responsabilités du product owner

Comprendre la théorie
(scrum guide)



Pour comprendre la mise
en pratique et les
adaptations et ses impacts



Certification



« Vraie » vie



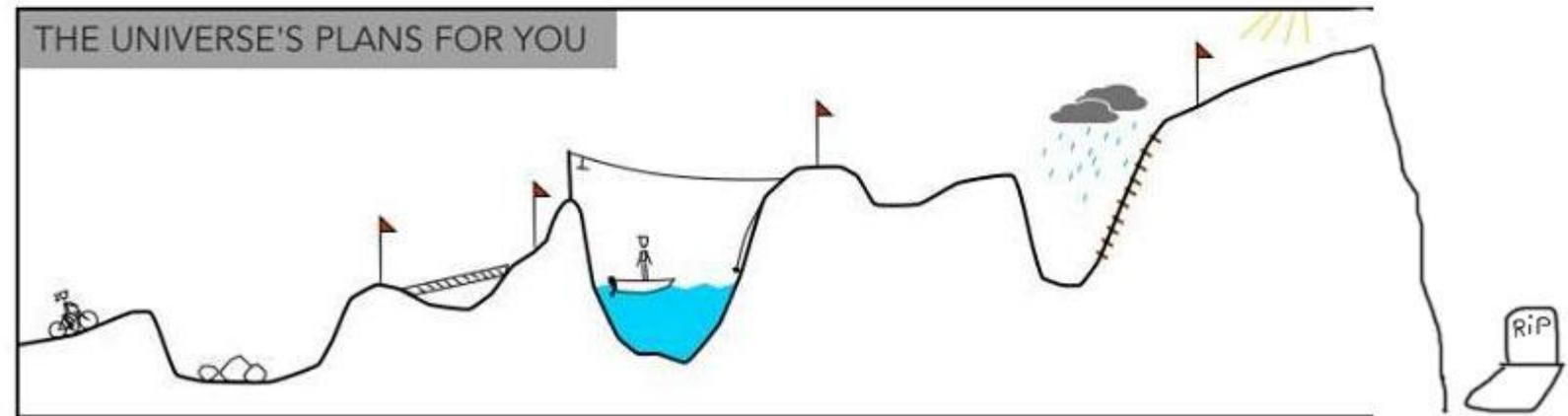
Naissance de l'Agilité

- Emergence de l'Agilité
- Le manifeste Agile
- Les Frameworks

Emergence de l'Agilité



Nous faisons des plans



DOGHOUSE DIARIES

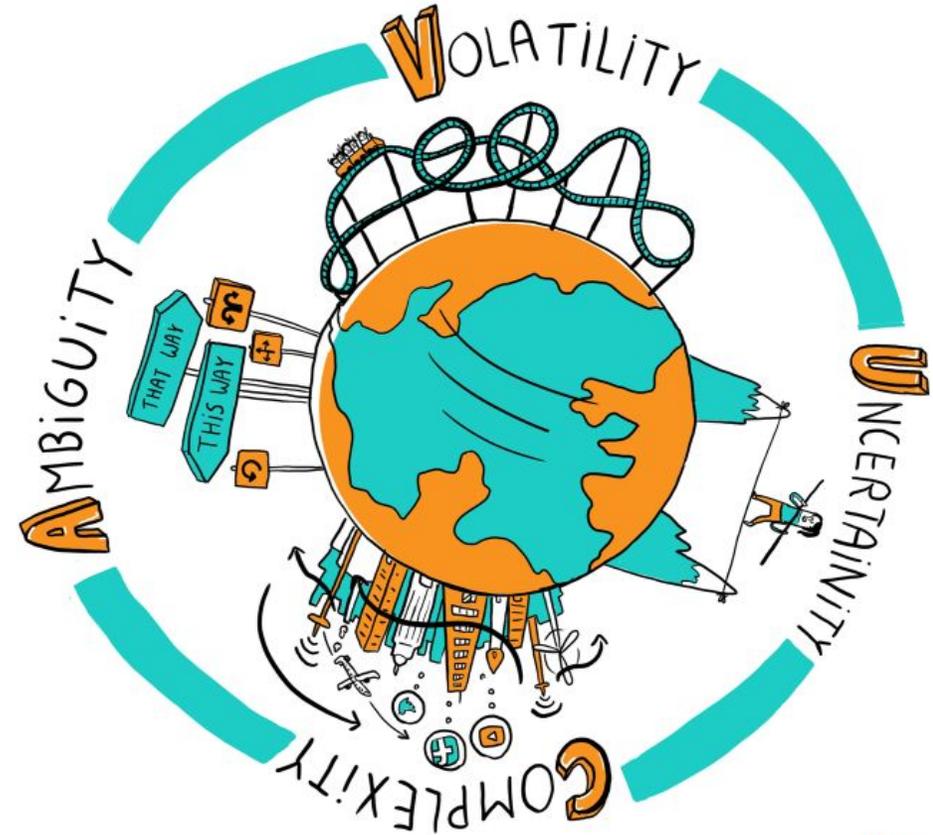
VUCA vous dites ...

Volatilité: Changements rapides, difficile à prédire et sans structure ou tendances claires.

Incertitude: Changements perturbateurs fréquents où le passé est pas un très bon prédicteur de l'avenir.

Complexité: Évolutions complexes, avec des aspects technologiques, sociétales, géopolitiques et écologiques entrelacées.

Ambiguïté: Peu de clarté sur ce qui est réel ou vrai et difficile de prédire l'impact des action ou des initiatives.



by **design**doppi

VOLATILITE	INCERTITUDE
 <p>AGILITE</p>	 <p>PROACTIVITE RELATIVISME</p>

S'adapter à un monde V.U.C.A nécessite d'adopter des pratiques collaboratives adaptées

COMPLEXITE	AMBIGUITE
 <p>INTELLIGENCE COLLECTIVE</p>	 <p>ORGANISATION APPRENANTE (TEST & LEARN)</p>

VOLATILITE	INCERTITUDE
<p>Remettre en cause les habitudes</p> <p>Etre à l'aise avec le changement</p> <p>Renforcer l'agilité</p>	<p>Etre plus authentique</p> <p>Encourager l'expression d'idées</p> <p>Développer la confiance (soi, autres)</p>

COMPLEXITE	AMBIGUITE
<p>Fonction en "Bottom-up"</p> <p>Valoriser l'intelligence collective</p> <p>Développer la transversalité</p>	<p>Composer avec les paradoxes</p> <p>Faire preuve de discernement</p> <p>Apprendre des échecs</p>

Les approches traditionnelles mises à mal par la révolution numérique

L'IT est principalement au service de la Performance Opérationnelle interne des entreprises

1990

Automatisation des tâches de gestion des processus métier, ERP, systèmes transactionnels (mainframes) ...

L'IT révolutionne l'offre de produits / services et les canaux de vente des entreprises

2000

Vente en ligne d'offre de services/produits numériques, applications mobiles, intégration d'objets connectés, de services numériques tiers...

De grosses applications complexes développées sur la base de **besoins prédictibles de stabilité, robustesse et performance**



Les approches traditionnelles fonctionnent bien

Des applications exposées à des écosystèmes turbulents (qui évoluent fréquemment) et fortement concurrentiels.

Les besoins sont instables et imprédictibles

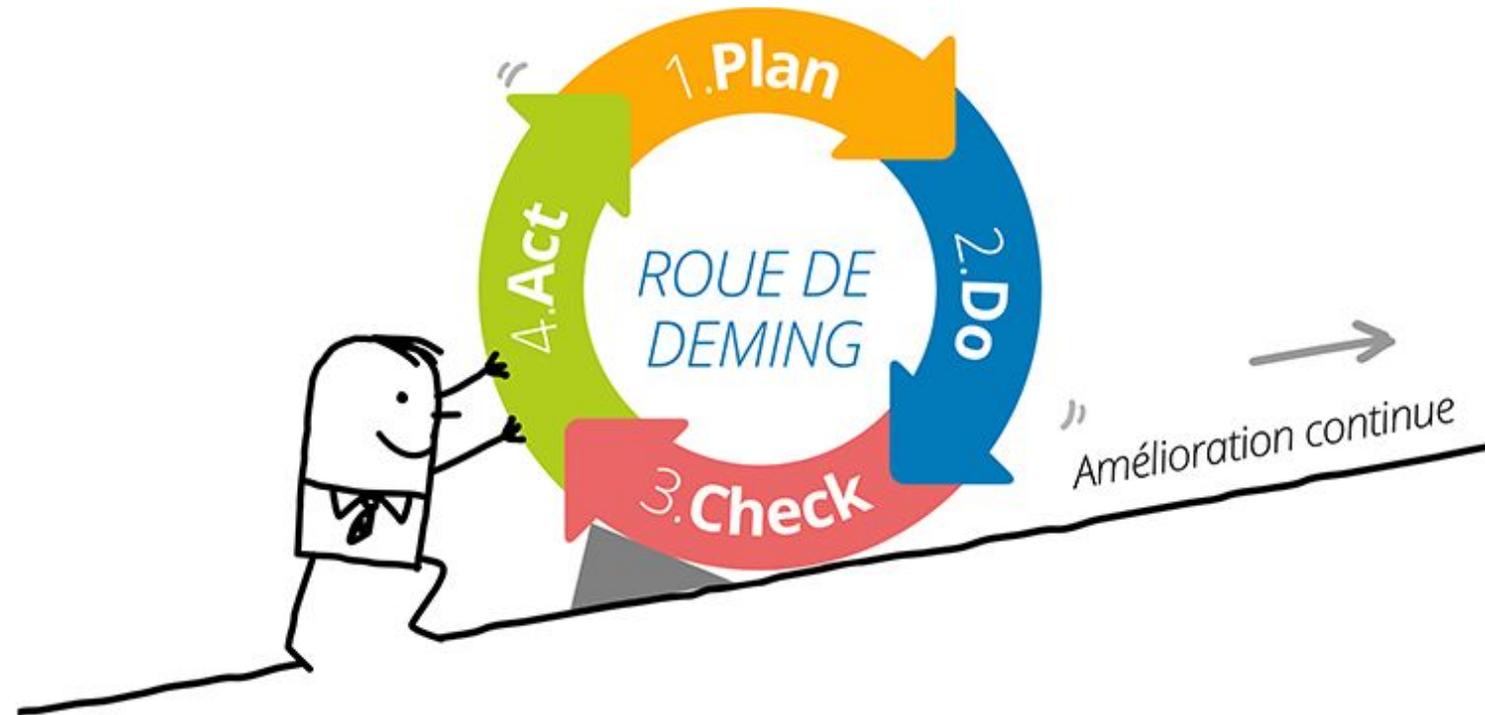


Les approches traditionnelles trouvent leurs limites

Favoriser l'apprentissage et l'expérimentation

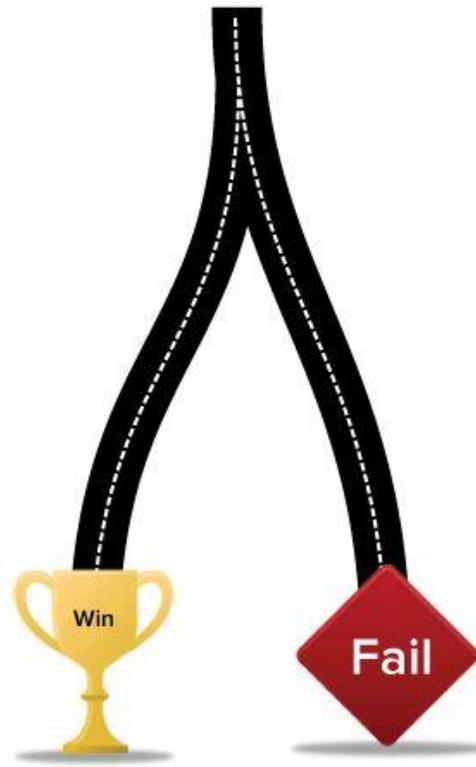
La **Roue de Deming** décrit un processus **d'apprentissage** et d'amélioration continue de la qualité, basé sur des cycles courts permettant de contrôler et d'adapter régulièrement les travaux effectués.

Ces principes sont au cœur de la culture Agile.

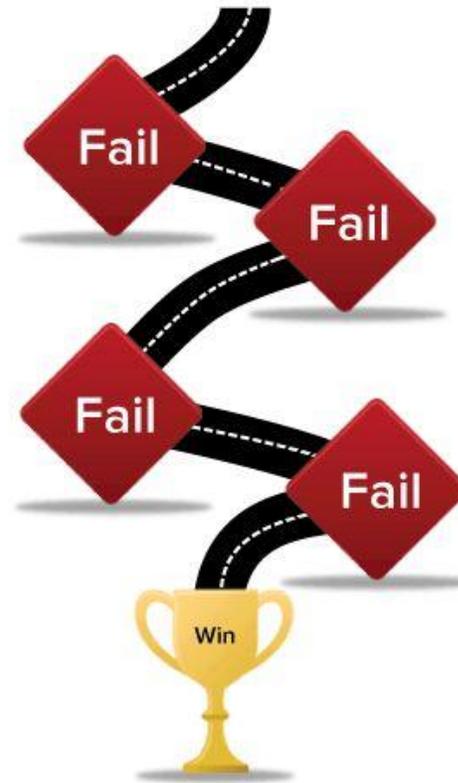


Echouer vite ! Fail Fast – Learn Fast

What Most
People Think



What Successful
People Know

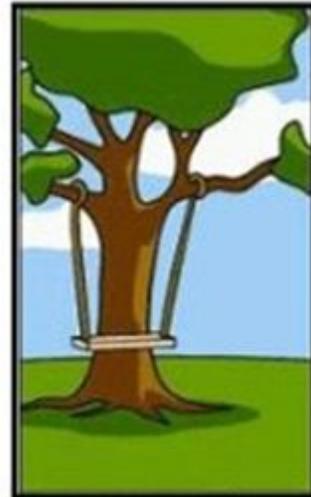


@douglaskarr

Nous avons TOUS une vision



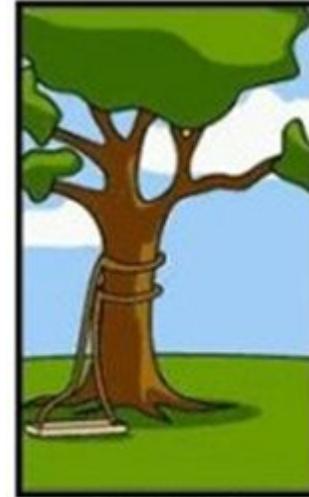
Comment le client a exprimé son besoin



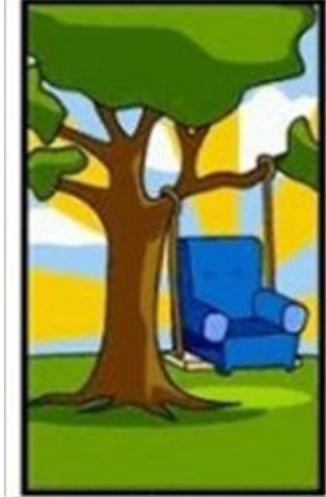
Comment le chef de projet l'a compris



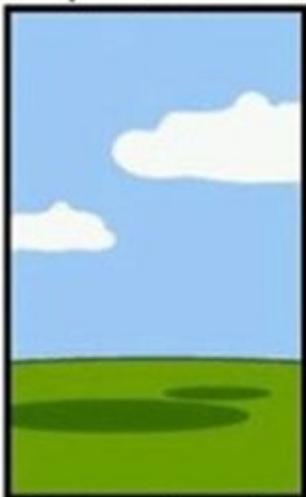
Comment l'ingénieur l'a conçu



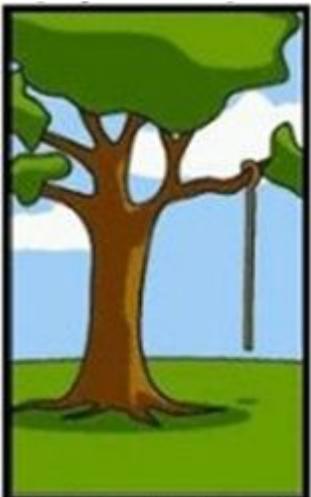
Comment le programmeur l'a écrit



Comment le responsable des ventes l'a décrit



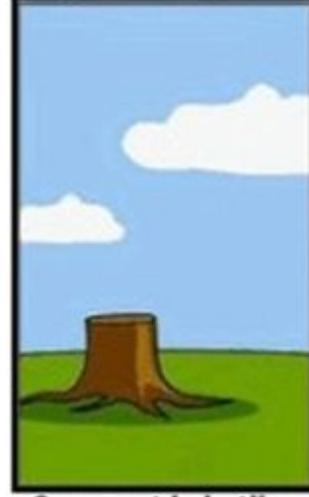
Comment le projet a été documenté



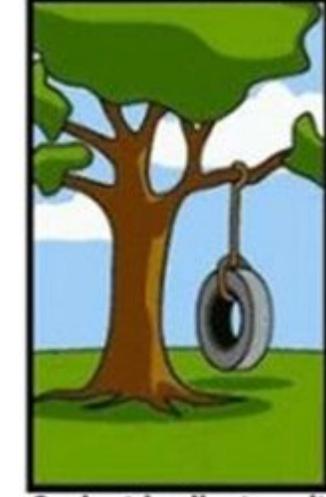
Ce qui a finalement été installé



Comment le client a été facturé



Comment la hotline répond aux demandes

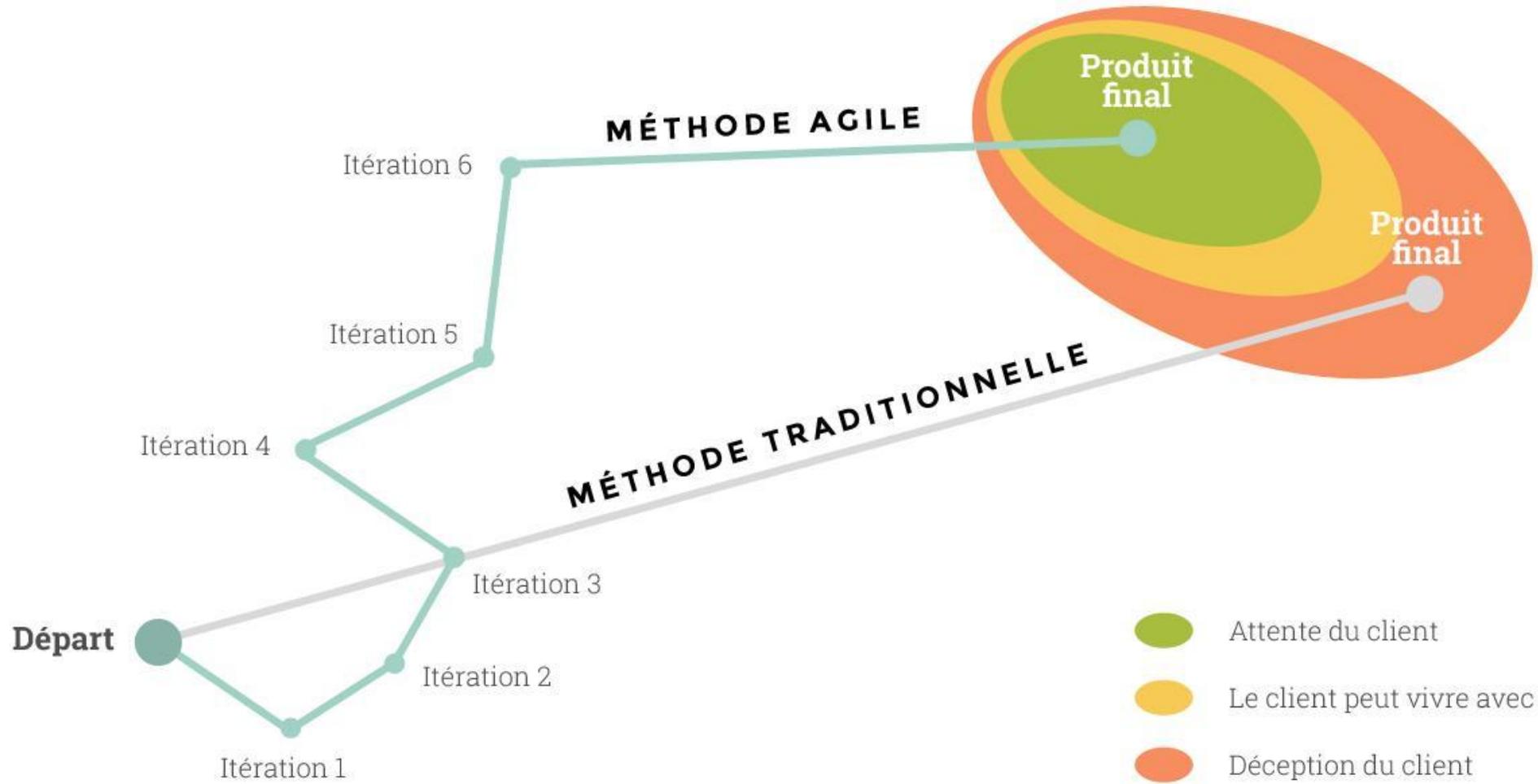


Ce dont le client avait réellement besoin

64% rarely or never
used features
Chaos Report - Standish Group

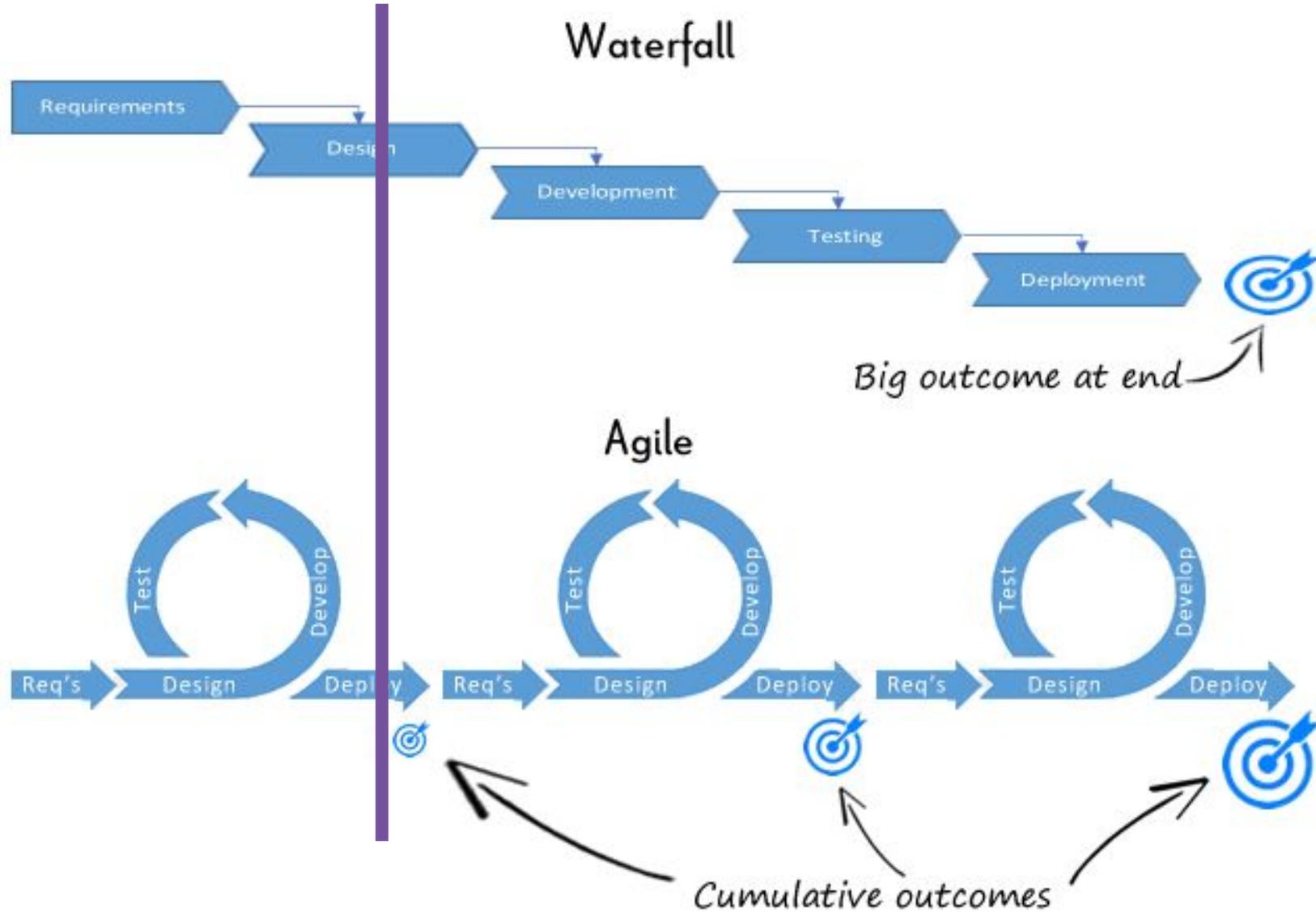
Ce qui est nécessaire VS ce que vous VOUS êtes engagé à faire

Plus une prévision est lointaine, plus elle fausse

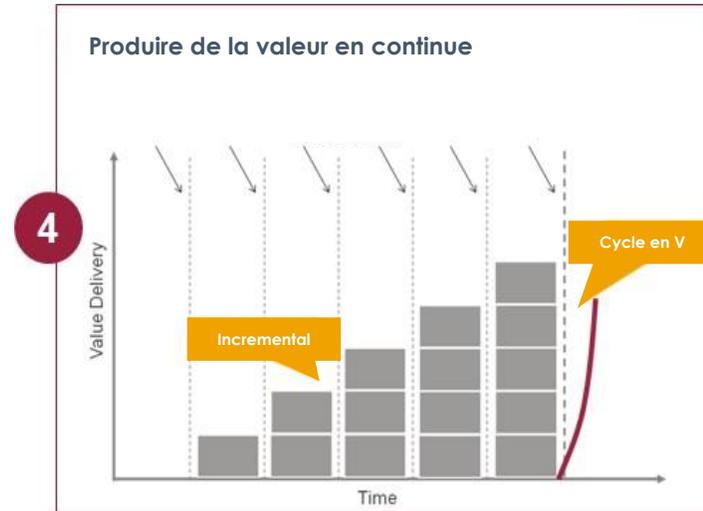
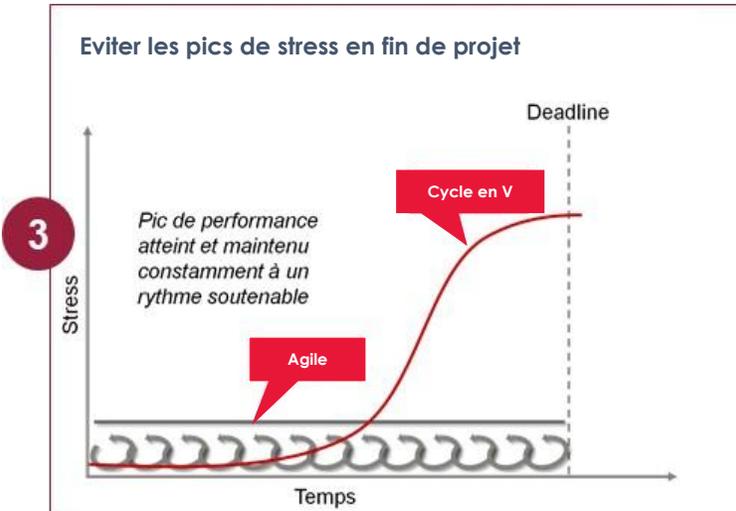
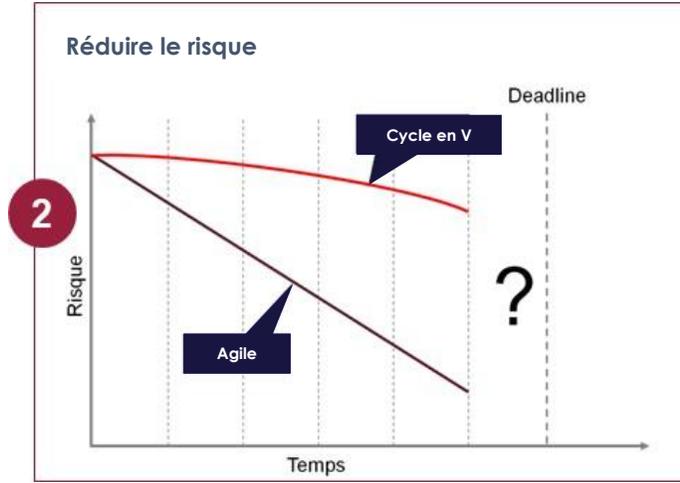
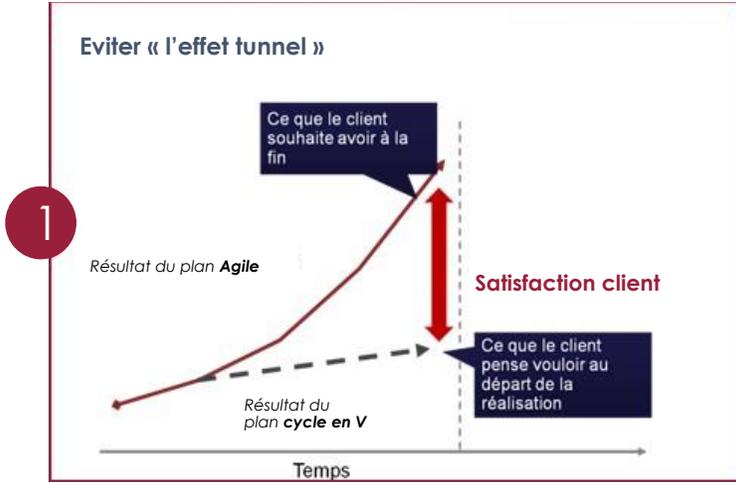


Pourquoi l'Agilité?

L'agilité délivre plus rapidement de la valeur



Délivrer en agile



Se confronter souvent aux utilisateurs et clients

Pour obtenir rapidement et fréquemment du feedback

Mesurer le gain produit / service

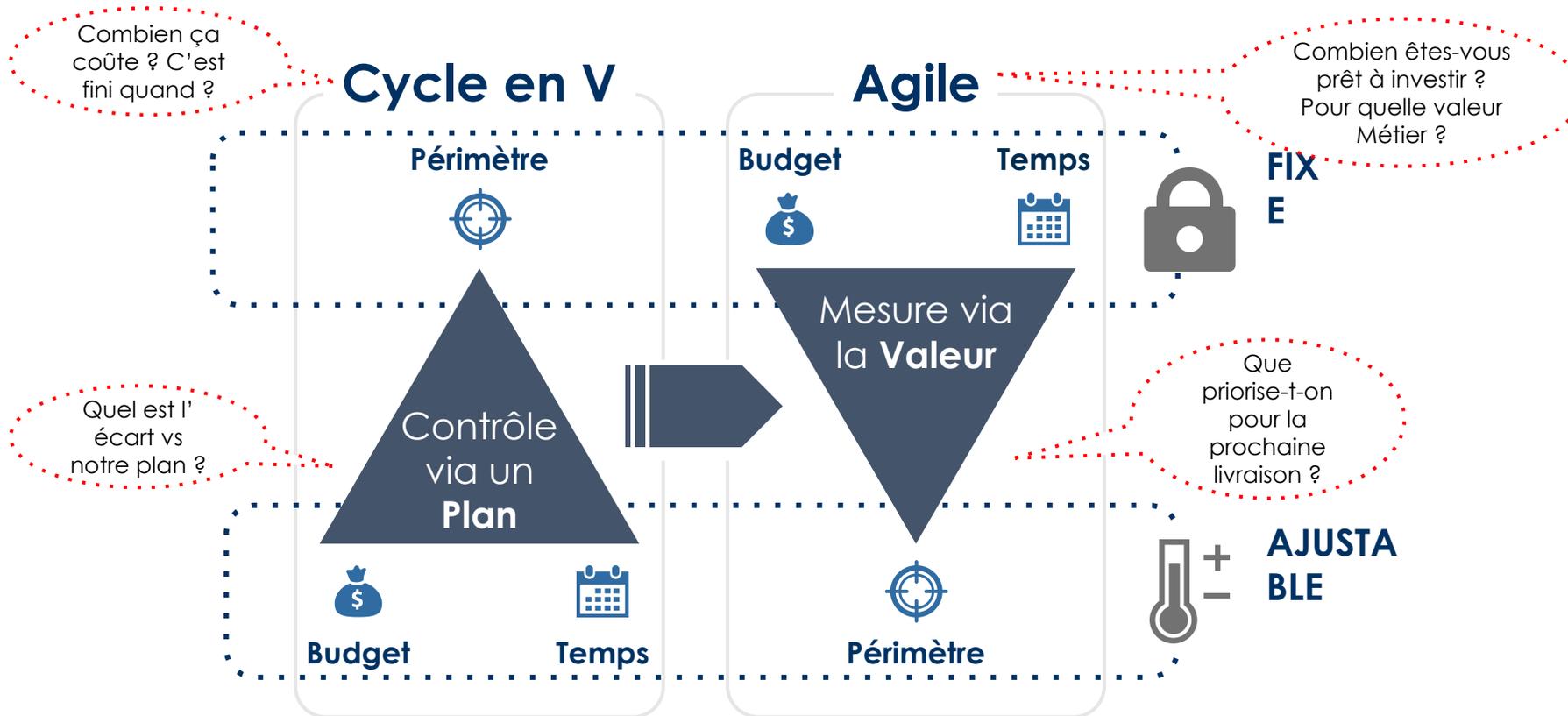
Obtenu à chaque livraison

Intégrer le changement

Adapter le produit, le processus et le plan en conséquence

Agilité et Cycle en V

En Agile, la variable d'ajustement, c'est le Périmètre Fonctionnel, et ça tombe bien car traditionnellement beaucoup de fonctionnalités développées ne sont jamais utilisées



L'Agile ne coûte pas moins cher que le cycle en V

En Agile, on livre plus tôt de la valeur

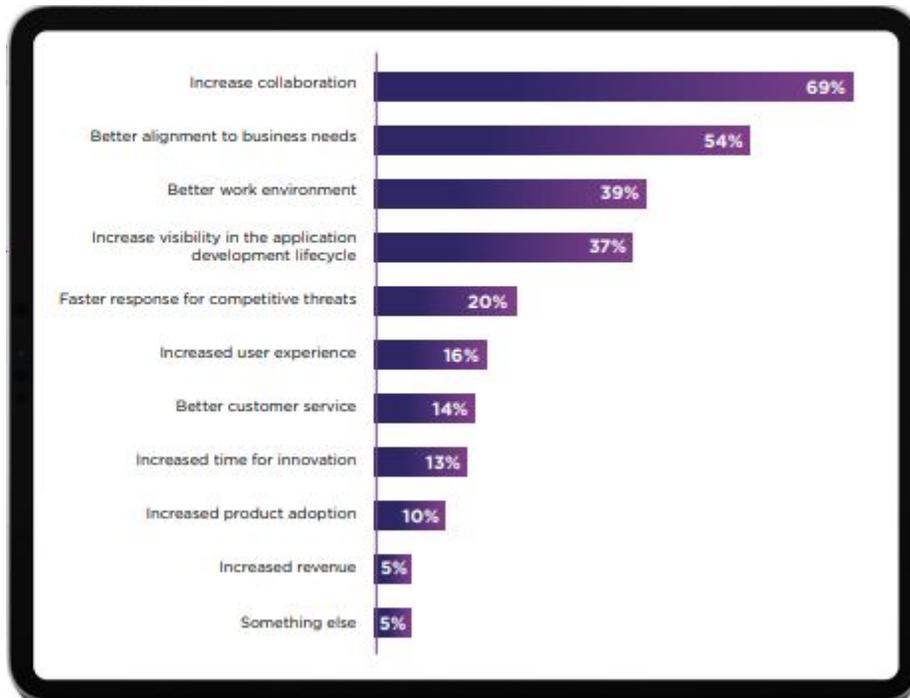
En Agile, on corrige plus vite les erreurs et on applique le principe « Build it, Run it »

Bénéfices de l'Agilité



16th annual State of Agile report

Le rapport State of Agile fournit des informations clés issues de la plus longue enquête annuelle continue sur les techniques et pratiques Agile : analyse de l'impact sur les personnes, les processus et les outils.



Les 2 plus importants bénéfiques :

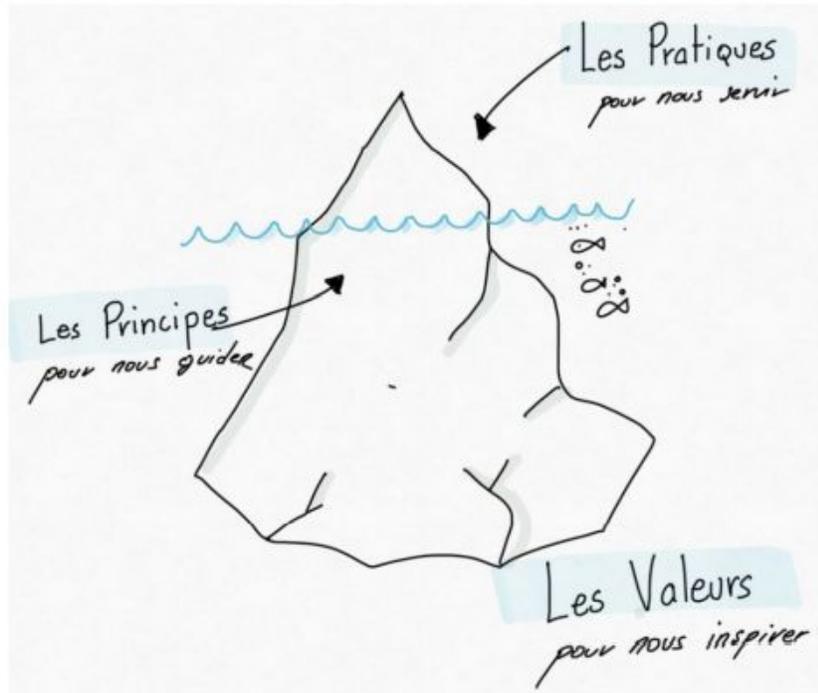
- Amélioration de la collaboration
- Meilleur alignement avec les besoins business



Le Manifeste Agile



Comment se définit l'agilité?



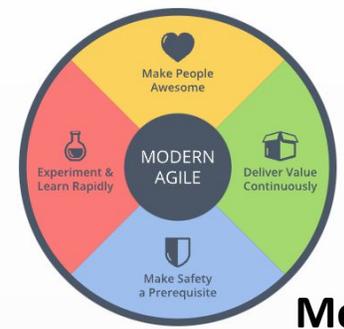
**L'agilité se définit
comme une culture**
soutenue par des valeurs
et des principes
permettant d'être
performant dans un
environnement complexe.

D'où vient cette culture?

Du 11 au 13 Février 2001, dans un hôtel dans l'Utah, dix-sept personnes se sont réunies pour essayer de définir les conditions nécessaires pour réussir le développement logiciel en entreprise.

Ce groupe a écrit le **Manifeste pour le Développement Agile de Logiciels.**

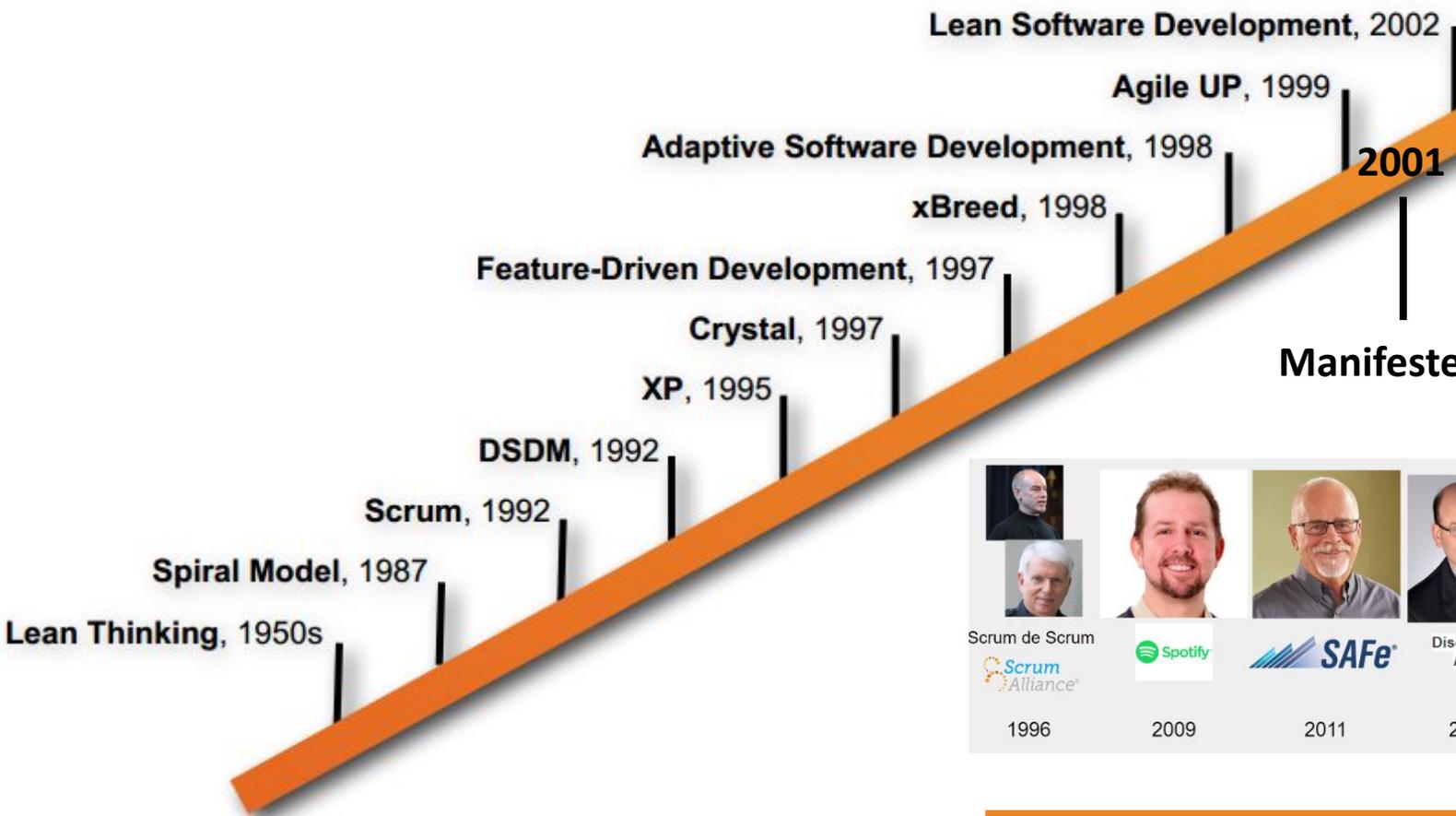
Fondation du mouvement Agile



Modern Agile
J. Kerievsky
2016

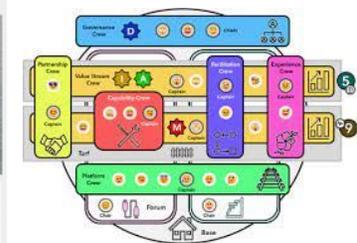


The heart of Agile
A. Cockburn
2015



Manifeste Agile

1996	2009	2011	2012	2014	2015	2018

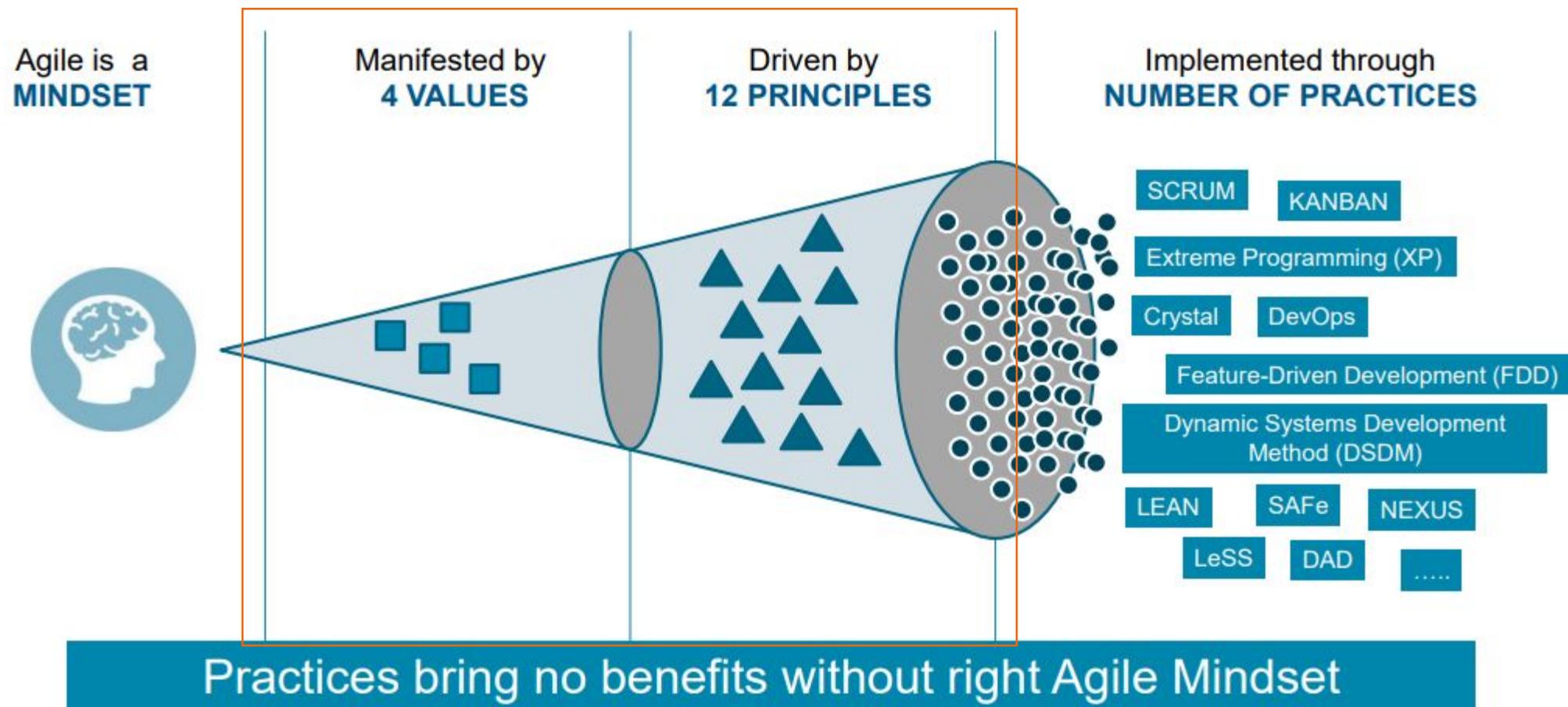


unFIX Model
J. Appelo
2022



BEING vs DOING

The real challenge is to BE Agile rather than DOING agile



Manifeste pour le développement Agile des logiciels

4 valeurs

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

... Plutôt que ...

Les processus et les outils

Des logiciels opérationnels

... Plutôt que ...

Une documentation exhaustive

La collaboration avec les clients

... Plutôt que ...

La négociation contractuelle

L'adaptation au changement

... Plutôt que ...

Le suivi d'un plan

Nous reconnaissons la valeur des seconds éléments mais **privilégions les premiers.**

Manifeste pour le développement Agile des logiciels

12 Principes



01

Satisfaction

Notre plus haute priorité est de **satisfaire le client** en livrant rapidement et régulièrement des fonctionnalités à **grande valeur ajoutée**

02

Changement

Accueillez positivement les **changements de besoins**, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client

03

Livrer

Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts

04

Ensemble

Les utilisateurs ou leurs représentants et les développeurs doivent **travailler ensemble quotidiennement** tout au long du projet

05

Motivation

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

06

Dialogue

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



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 **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

- "... **Les individus et leurs interactions** plus que les processus et les outils ..."
(**Ouais !! Je ne vais plus devoir suivre des processus stupides !**)
- "... **Des logiciels opérationnels** plus qu'une documentation exhaustive ..."
(**Youpi !! La documentation à la poubelle ! J'ADORE ce truc agile !**)
- "... **La collaboration avec les clients** plus que la négociation contractuelle ..."
(**Je finirai quand j'aurai fini et je ne dirai jamais quand !**)
- "... **L'adaptation au changement** plus que le suivi d'un plan ..."
(**Pas de plans ! Pas de chefs de projet ! Pas d'architectes !**)

On déploie la culture agile...

En :

- Responsabilisant ceux qui font;
- Donnant du sens et le droit à l'erreur;
- Redistribuant la prise de décision;
- Limitant le travail en cours;
- Établissant des règles claires;
- Partageant des feedbacks
- Travaillant sur des petits périmètres priorisés par la valeur;
- Testant, tentant, innovant, modifiant et apprenant.

la fin des idées reçues :

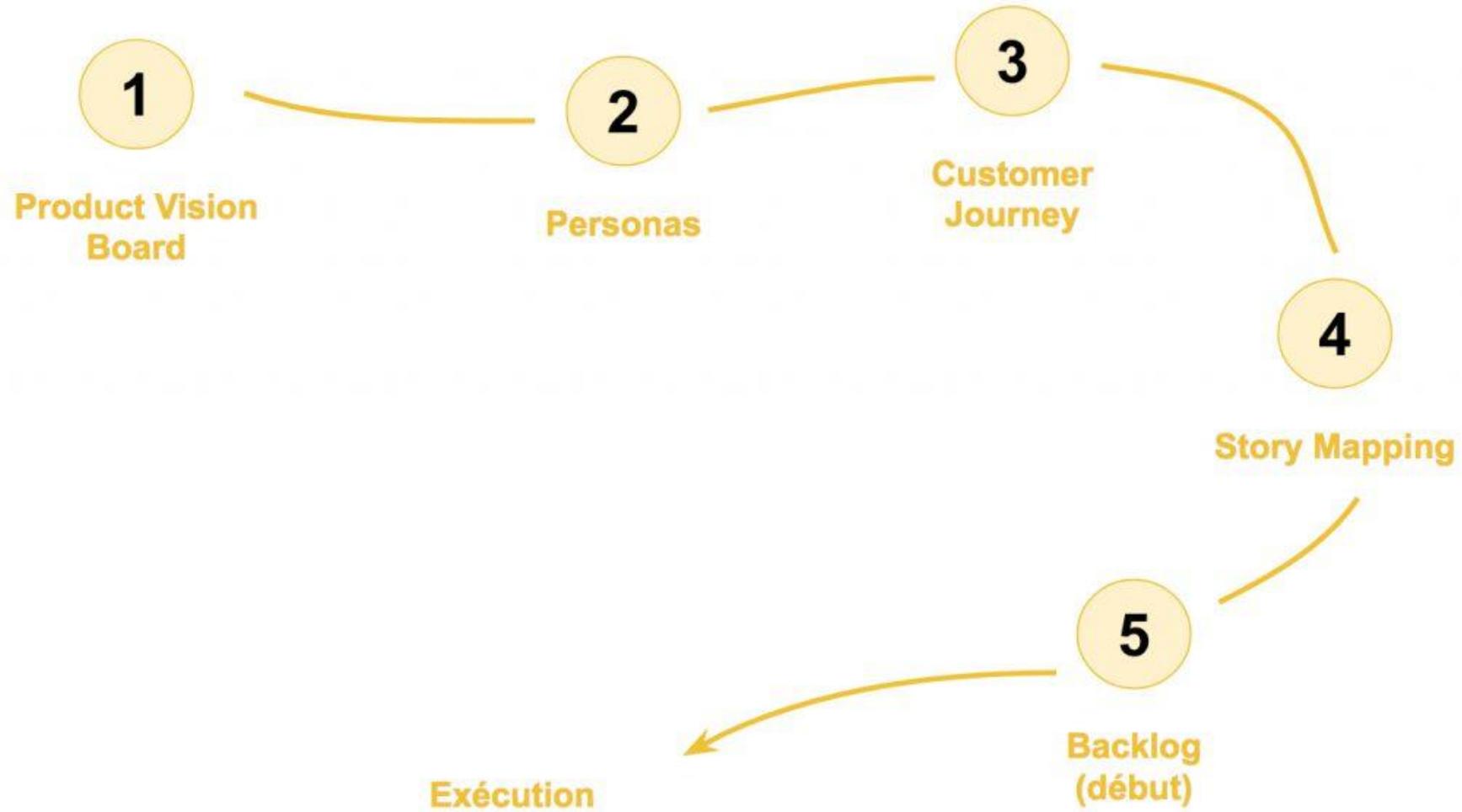
- L'agilité, c'est d'abord de la DISCIPLINE,
- L'agilité, c'est aussi du PROCESS,
- L'agilité, c'est aussi de la DOCUMENTATION.

2

Le produit Agile

- Vision
- Personas
- Story mapping
- backlog

Le produit agile



UX Design (USER EXPERIENCE)

UI

Interface Utilisateur
(Produit)



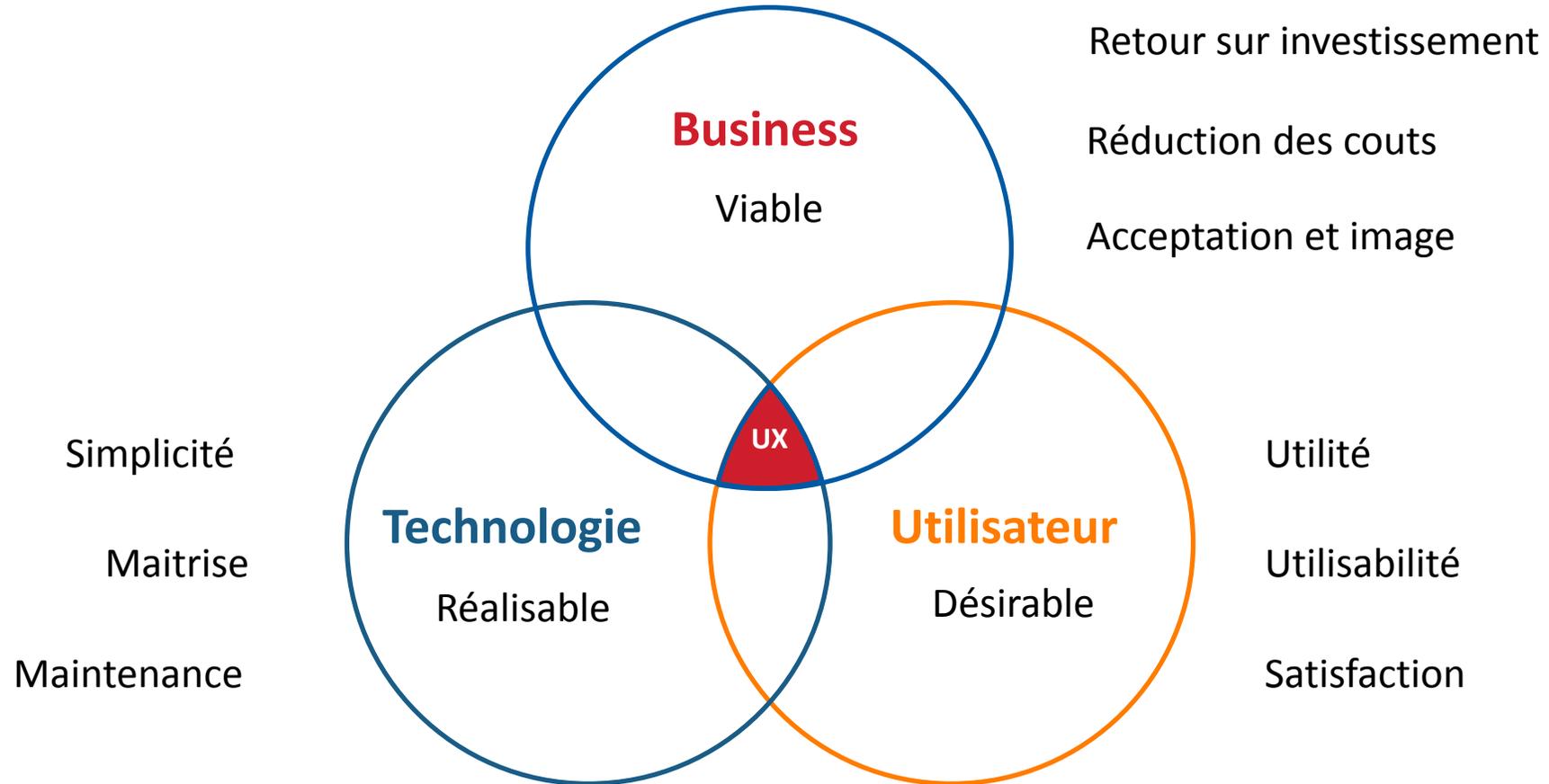
VS



UX

Expérience Utilisateur
(Produit + Utilisation)

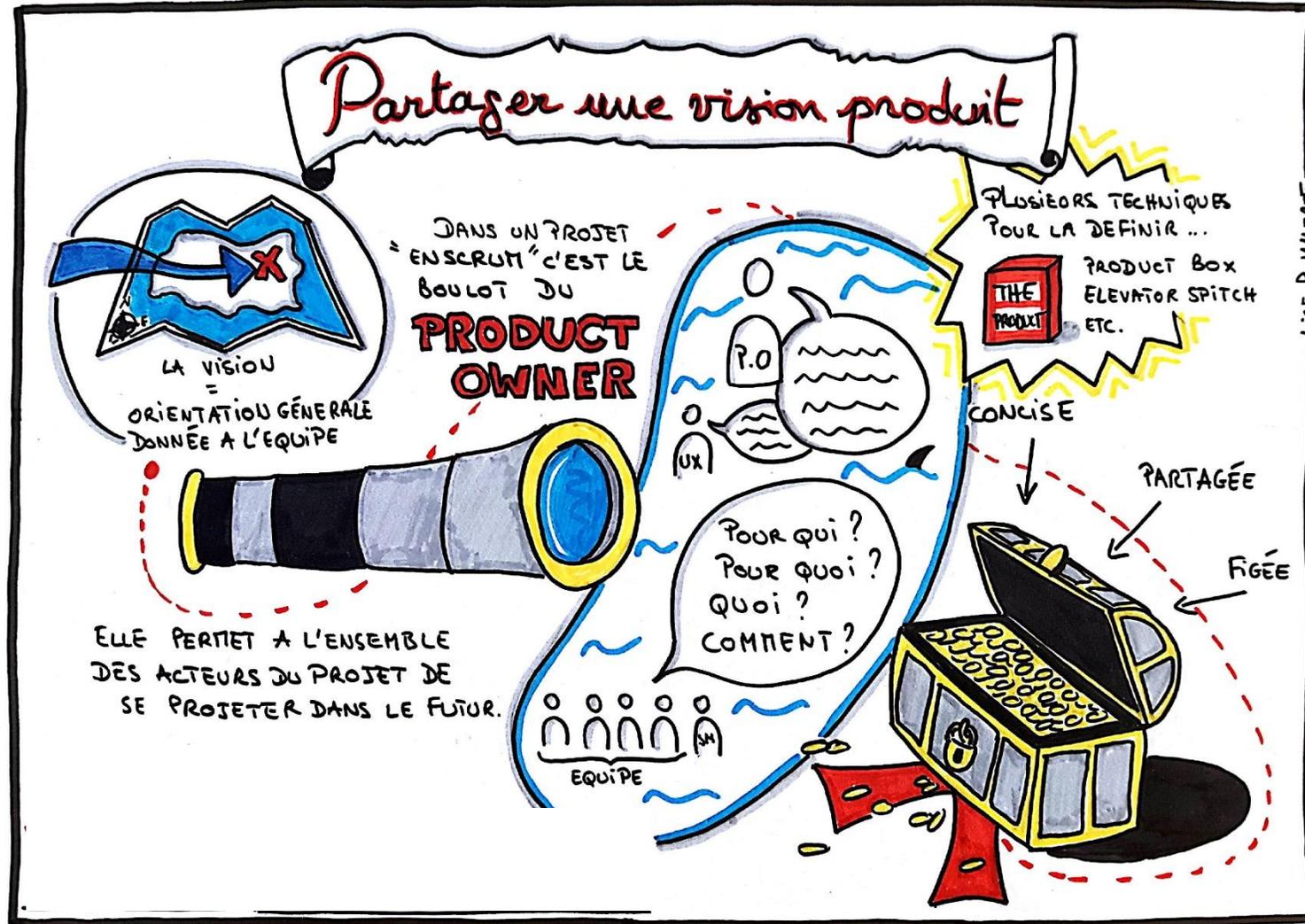
UX Design (USER EXPERIENCE)



CONCEVOIR LE « BON » PRODUIT

UNE DÉMARCHE DE CONCEPTION CENTRÉE SUR L'UTILISATEUR

Développer et partager une vision produit



LISE POMHRET

Quel est l'objectif (ou les objectifs) du produit ?

Les 5 questions clés à poser :

- Qui va utiliser le produit ?
- Quels besoins le produit va adresser ?
- Quelles caractéristiques du produits sont critiques pour satisfaire les besoins identifiés et donc pour réussir le projet ?
- Quelles sont les différences notables que le produit doit avoir avec l'existant ou par rapport à des produits de la concurrence ?
- Quels sont l'échéance et le budget cibles pour développer et lancer le produit ?



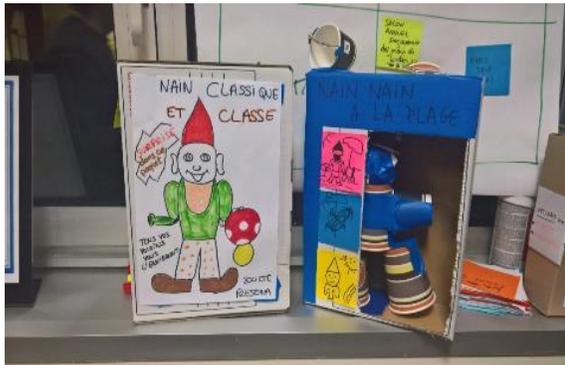
Outils pour construire la Vision Produit

VISION



Décrit **pourquoi** on crée le produit, **pour qui** il est destiné, quelles sont les **solutions** à mettre en place pour répondre au « pourquoi », quel est son « ROI ».
Elle permet un **alignement** de toutes les parties prenantes.

Product Vision Box



Remember the future



Elevator Pitch



PROBLEM What is the problem you are solving?	SOLUTION What is your solution? How does it work?	UNIQUE VALUE PROPOSITION What is your unique value proposition? How does it differ from competitors?	UNFAIR ADVANTAGE What is your unfair advantage? How does it give you a competitive edge?	CUSTOMER SEGMENTS Who are your target customers? What are their needs and pain points?
EXISTING ALTERNATIVES What are the existing alternatives? Why are they not good enough?	KEY METRICS What are the key metrics that you will use to measure success? How will you track them?	HIGH-LEVEL CONCEPT What is the high-level concept of your solution? How does it work?	CHANNELS How will you reach your target customers? What are your distribution channels?	EARLY ADOPTERS Who are your early adopters? How will you reach them?
COST STRUCTURE What are the major cost drivers? How will you manage costs?		REVENUE STREAMS How will you generate revenue? What are your revenue streams?		

Lean Canvas

Elevator Pitch

- Une courte déclaration pour positionner le produit
- Expliquer le produit à quelqu'un en 2 minutes
- Suivre cette syntaxe

Pour (clients)

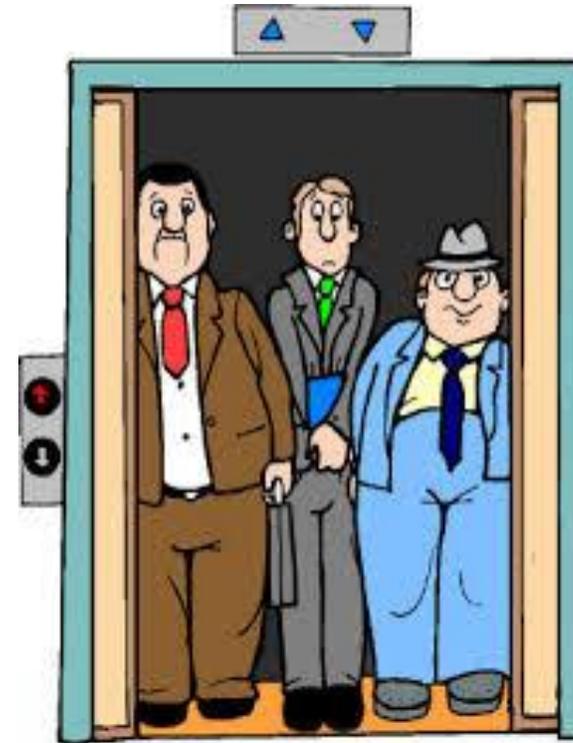
Quoi (exposé du besoin ou de l'opportunité)

Le (nom du produit) *est un* (catégorie du produit)

Qui (bénéfice clé, raison de convaincre d'acheter)

A la différence de (produit concurrent, ancien produit)

Notre produit (différenciant principal)



La product box

- **But : identifier les dispositifs les plus intéressants et les plus vendeurs**
- Demandez à vos clients qu'ils vendent votre produit dans une foire, sur un marché...

- La product box devra **obligatoirement** contenir les éléments suivants

- Sur le devant:

- Nom du produit
- Image
- Slogan - 3/4 Bullets arguments pour « vendre » le produit

- Sur le reste de la boîte:

- Vue plus détaillée contenant la liste des fonctionnalités,
- Utilisateurs
- les pré-requis ...



Lean Canvas



10 Vérifier les hypothèses et itérer sur le lean Canvas

Produit

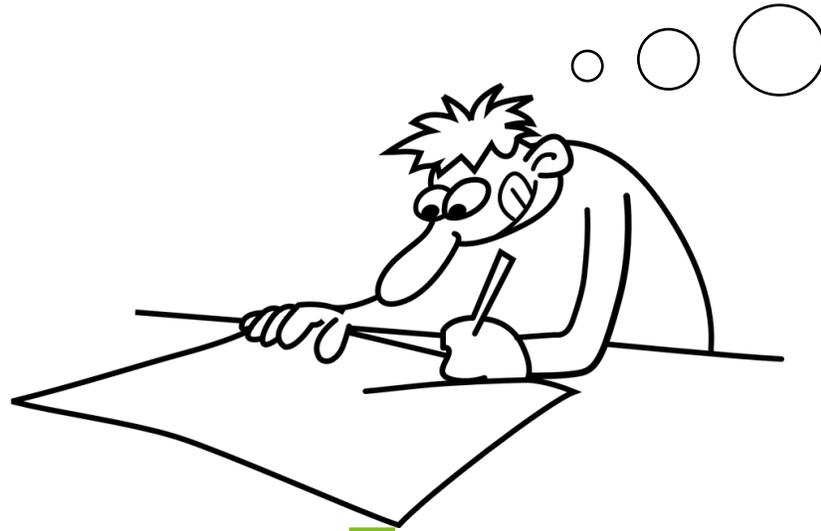
Marché

Problème 2 top 3 des problèmes à résoudre pour les early adopters supposés.	Solution 3 3 grandes fonctionnalités qui vont répondre aux problèmes de nos early adopters	Proposition de valeur unique 4 raison qui doit pousser vos prospects à devenir des utilisateurs. En quoi résolvez-vous leurs problèmes ?	Avantage concurrentiel 9 ce qui vous rend plus performant que vos concurrents pour traiter les problèmes identifiés	Segment clients 1 Qui sera la cible de notre produit ?
Alternative Alternatives actuelles avec leurs limites	Indicateurs clés 6 quelles métriques mesurer pour valider ou invalider vos hypothèses ?		Canaux 5 quels canaux gratuits et payants que vous pouvez utiliser pour atteindre vos futurs utilisateurs ?	Early-adopters Qui sont nos potentiels early adopters ?
Coûts 7 quels sont vos coûts fixes et variables ?		Revenus /gains 8 quel est le modèle économique ? Comment allez-vous gagner de l'argent ?		

L'importance de la vision



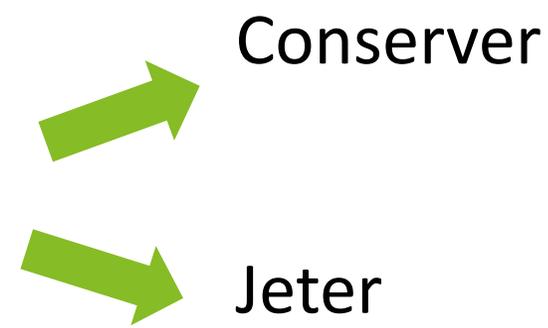
Le vrai besoin



Tout ce que vous imaginez sur le produit ne sont que des hypothèses



Vérifier toutes les avec les utilisateurs potentiels avec le minimum d'effort (MVP)



Le vrai besoin

Les remarques des utilisateurs ne sont que des inputs. Il faut remonter aux besoins profonds du client au-delà de la solution qu'il pousse spontanément, par exemple avec la méthode des "5 pourquoi". Vous pourrez alors vous focaliser sur l'important sans être esclave de l'urgent.



Nous devons garder iso-fonctionnelle les fonctions d'administration

Ce sont les paramètres des droits des utilisateurs

Surtout pour permettre à certaines personnes d'accéder à des fichiers spécifiques.

Parce qu'ils ont changé d'équipe.

Parce que le changement manuel n'a pas été fait.

Parce que les fonctions de synchronisation avec l'annuaire d'entreprise ont été dépriorisées l'année dernière.

Pourquoi ? Qu'est-ce qui est utilisé actuellement ?

Pourquoi sont-ils utilisés ?

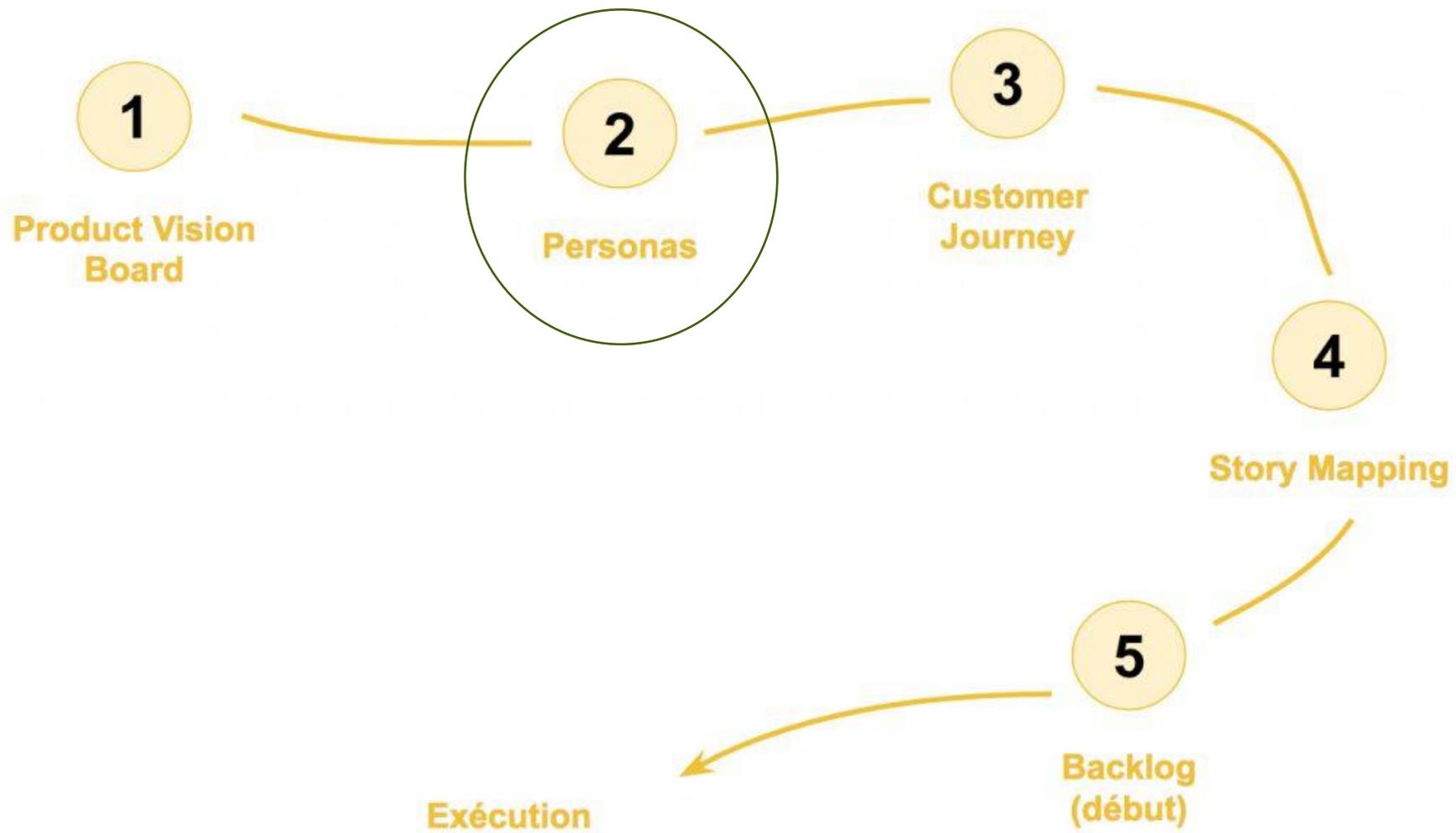
Pourquoi ne peuvent-ils pas y accéder avec la configuration standard ?

Pourquoi est-ce qu'ils sont toujours associés à leur ancienne équipe ?

Pourquoi le changement est manuel ?

Donc, la priorité est plutôt la synchro avec l'annuaire non ?

Le produit agile



Les Personae

L'empathie client

Un persona:

- C'est un utilisateur ou un usager-type, une représentation fictive de votre cible de client.
- On peut l'utiliser pour imaginer des besoins, des réactions, fixer des priorités, élaborer une communication ,...



Les caractéristiques des personas recouvrent en général les points suivants:

- Le type d'utilisateur: âge, genre, profession, catégorie socioprofessionnelle, etc. (le QUI)
- Le contexte: situation, besoins, objectifs, contraintes (le QUOI et POURQUOI)
- La personnalité: goûts, réactions, leviers de motivation (le COMMENT)

Les Utilisateurs / Nos Personas

- On peut donner une représentation aux rôles principaux en créant des personnages
- Faites une fiche décrivant leur environnement, leur mode de vie, leurs habitudes...
- Associez y des photos
- Cela permet aux équipiers de garder à l'esprit les caractères particuliers des clients pour lesquels on travaille -> développe l'empathie



Igor Vasiliev, 32 ans
« Administrateur réseau et télécom »

Son entreprise

- ▶ Igor travaille pour JTI depuis 4 ans. Il est responsable des achats réseau et télécom pour l'Asie et certains pays d'Europe

Sa relation à la technologie

- ▶ Il maîtrise parfaitement l'informatique. Il surfe chaque jour sur Internet et utilise le courrier électronique du matin au soir.

Son poste de travail

- ▶ Processeur Pentium IV 2700MHz
- ▶ 2Go RAM
- ▶ Windows XP
- ▶ Firefox 14.0

Sa personnalité

- ▶ joueur en ligne tous types de jeux
- ▶ assez exigeant n'aime pas que les choses traînent

Son rôle

- ▶ Commander et valider les commandes pour l'Asie et l'Europe
- ▶ Demander et valider des devis pour de nouveaux projets et les changements particuliers
- ▶ S'assurer du bon déroulement des commandes

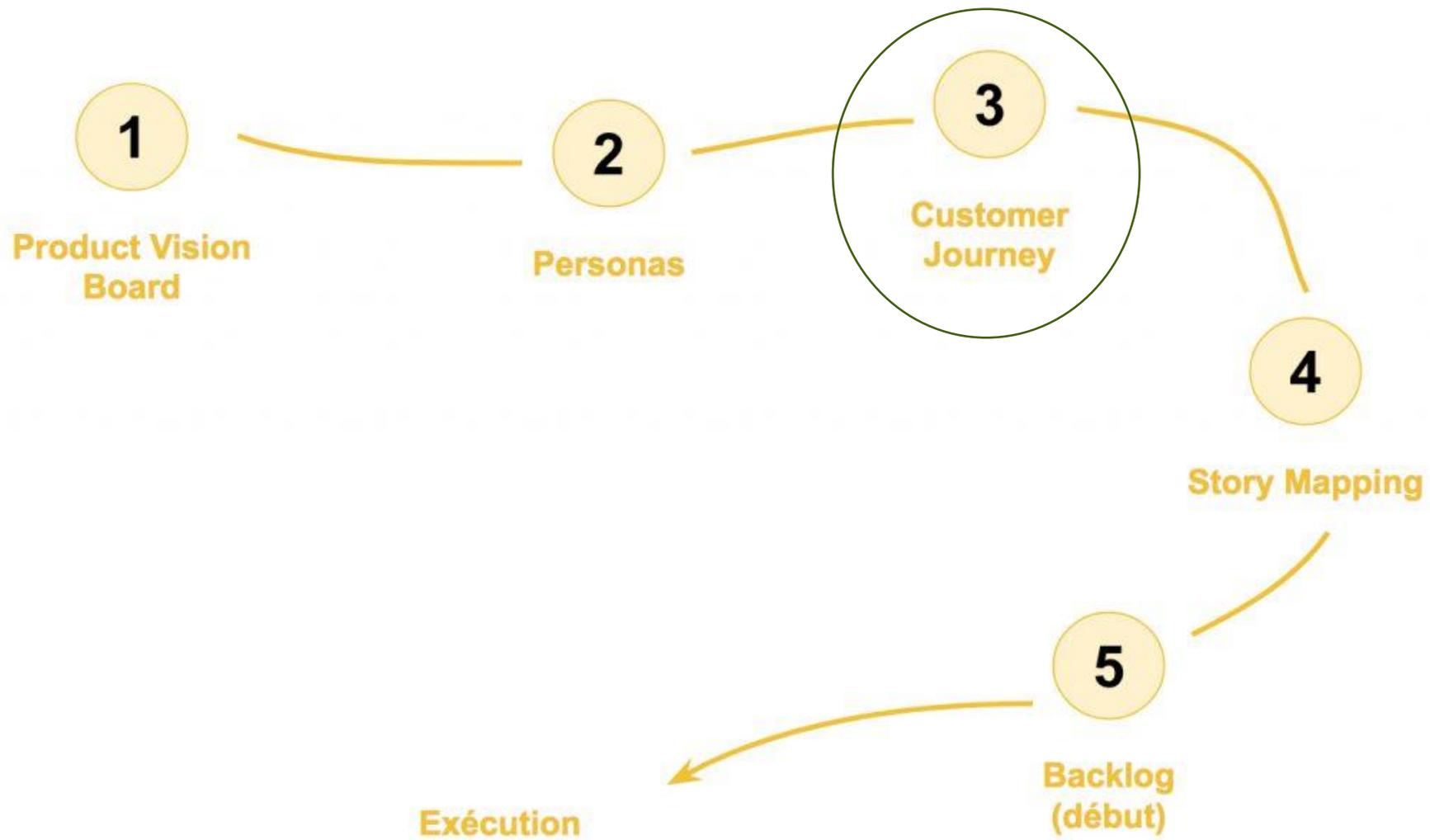
Ses besoins

- ▶ S'informer sur l'avancement des commandes
 - ▶ État de la commande
 - ▶ Date de livraison prévue
- ▶ Modifier la commande (le ou les centres de coût et la référence client de la commande)
- ▶ Rédiger rapidement et simplement des commandes, des demandes de cotation, des demandes de changements et des déplacements de site

Profil : Rédacteur / validateur
Pays : cf liste
Solutions : cf liste

Bec pays	
Armenia	
Azerbaijan	
Belarus	
Bulgaria	
Georgia	
Italy	
Kazakhstan	
Mongolia	
Romania	
Russian Federation	
Serbia	
Spain	
Ukraine	

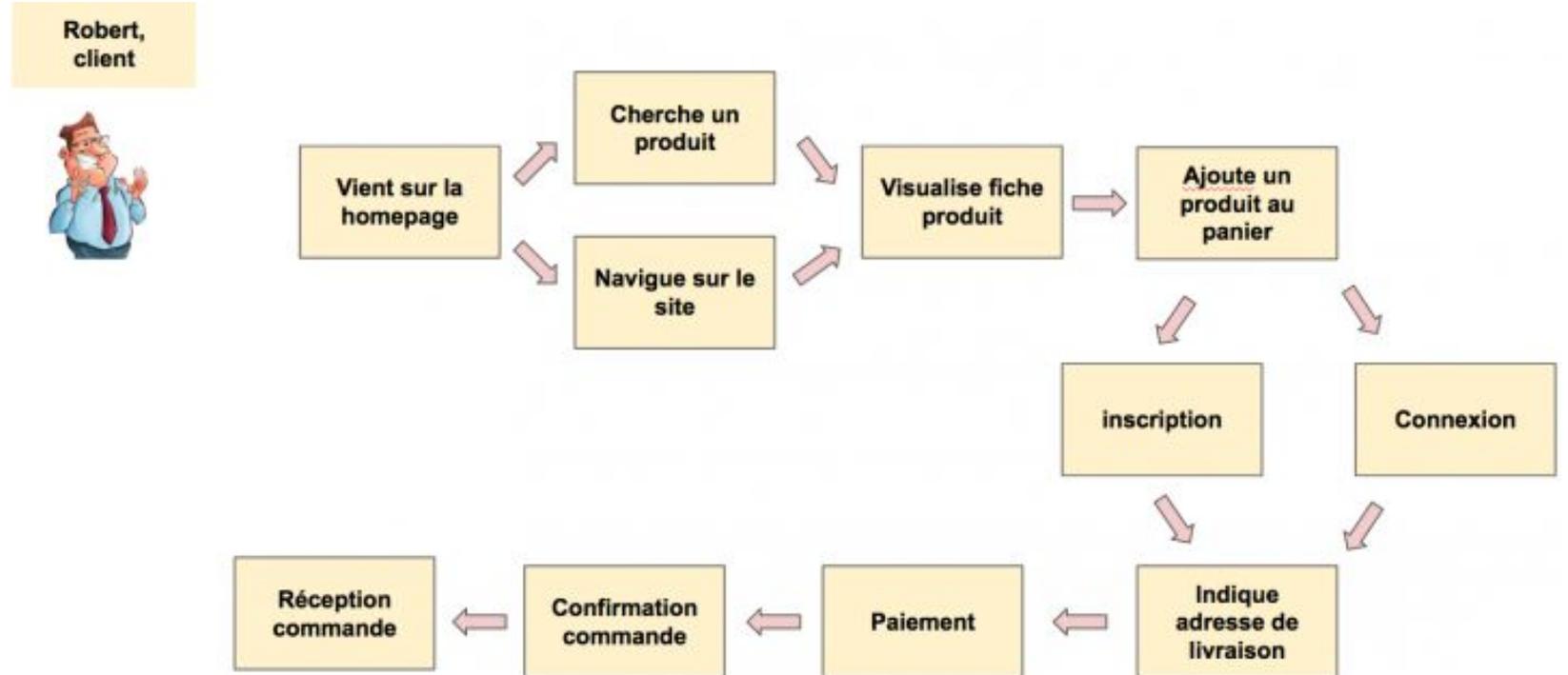
Le produit agile



USER JOURNEY

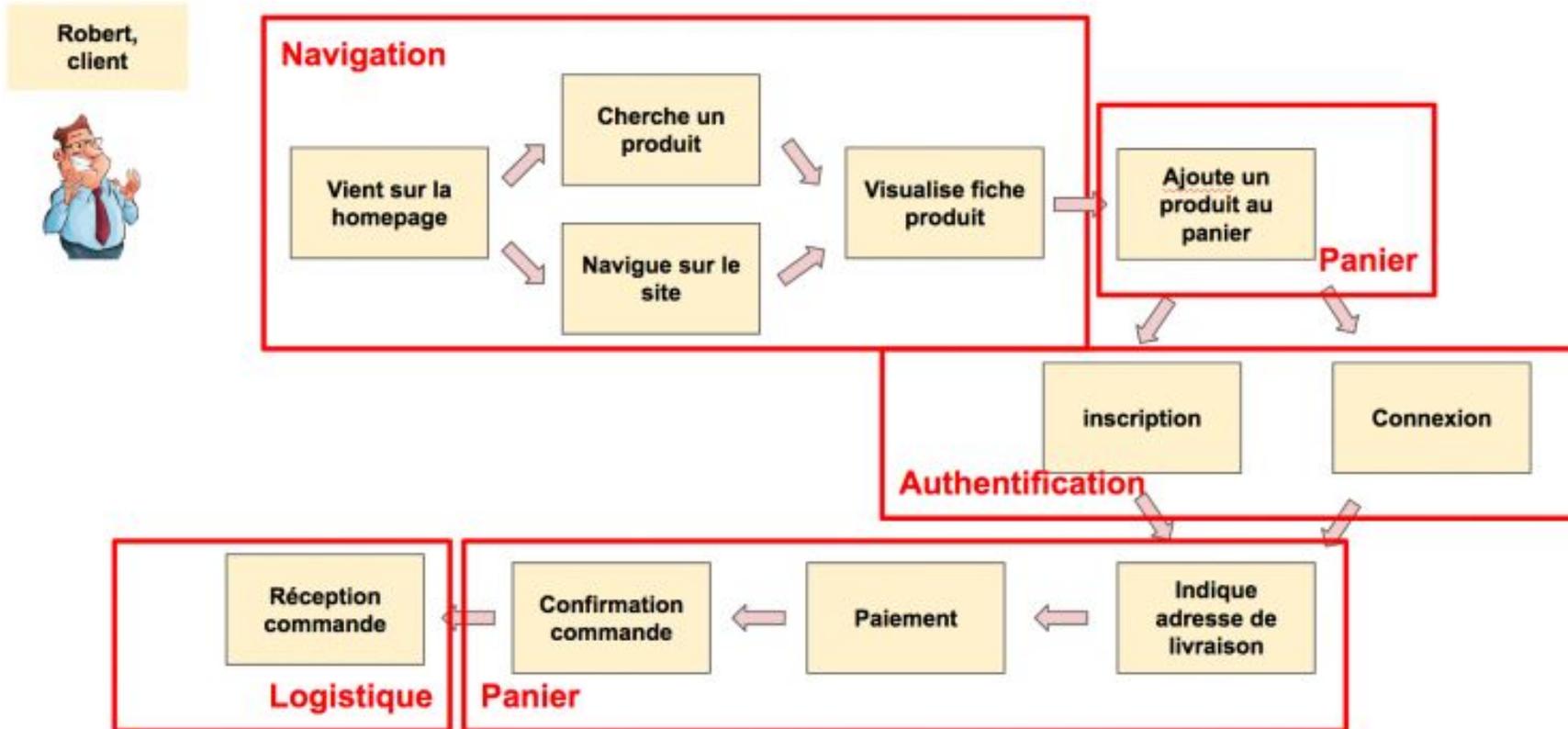
SCENARIO UTILISATEUR

- Représenter le parcours de nos utilisateurs au sein de notre application
- Facilite la vision du côté fonctionnel du produit
- Simple à mettre en place

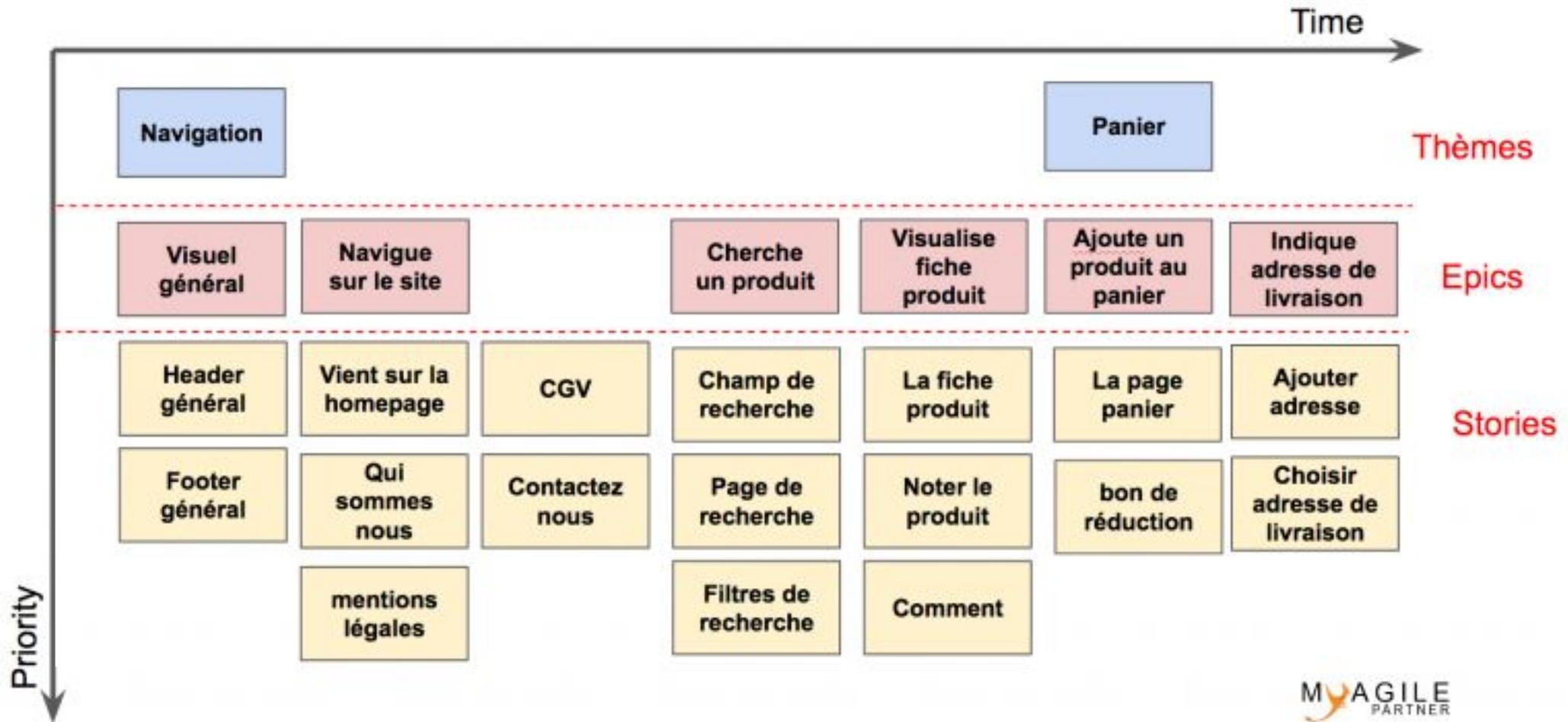


USER JOURNEY

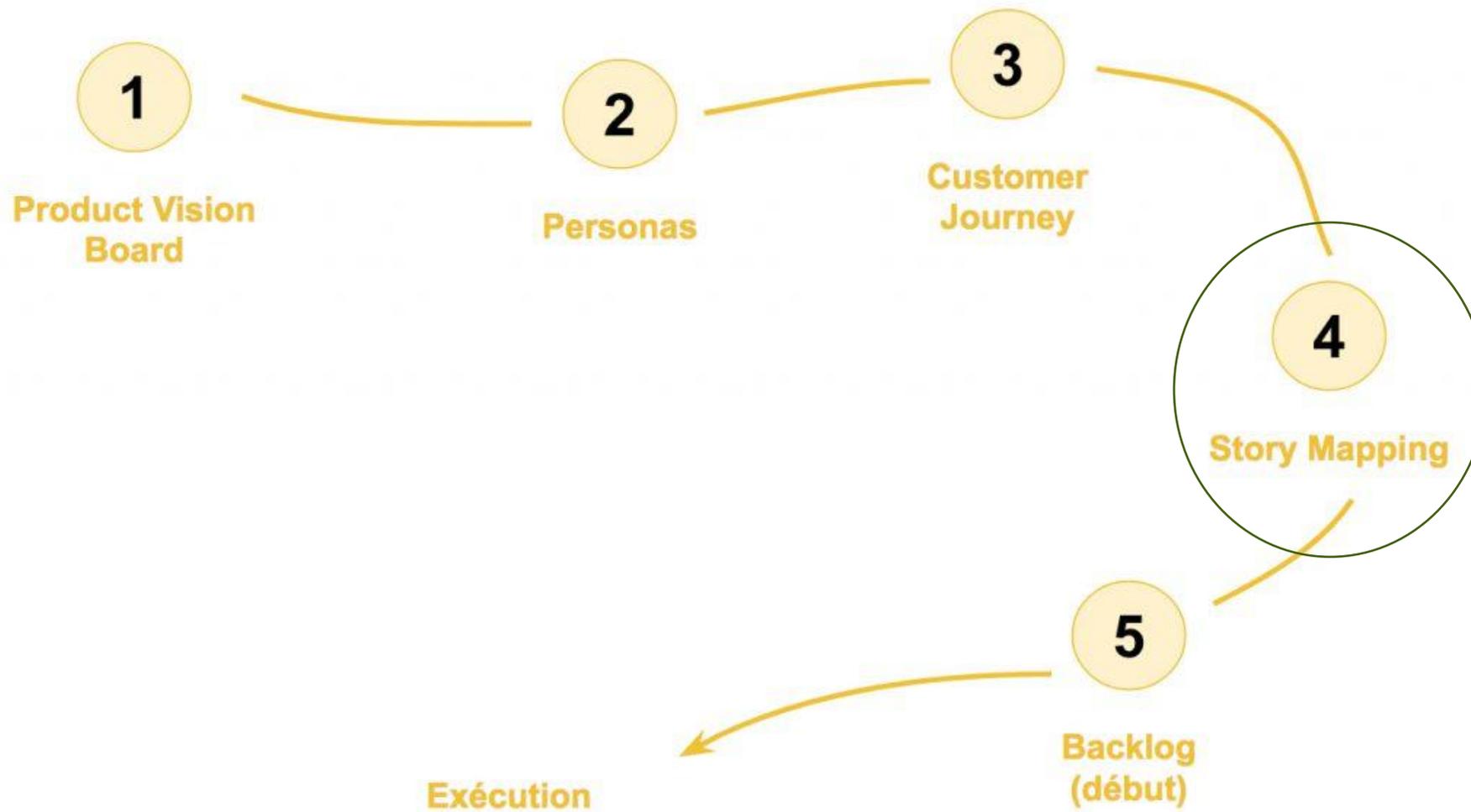
DEFINITION DES THEMES



USER JOURNEY -> STORY MAPPING

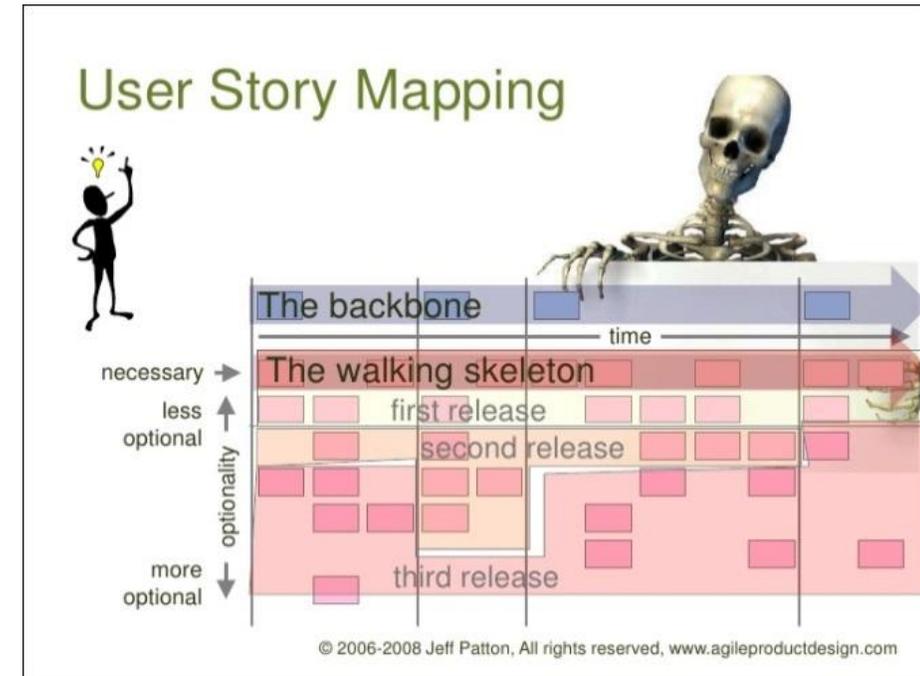
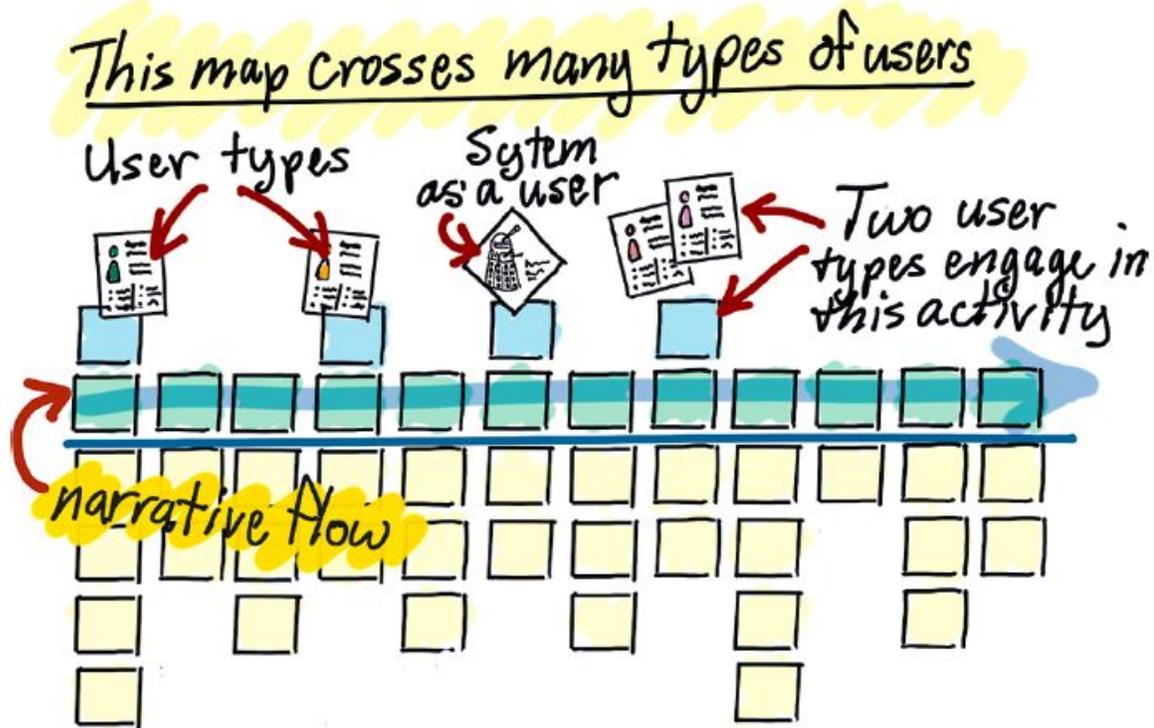
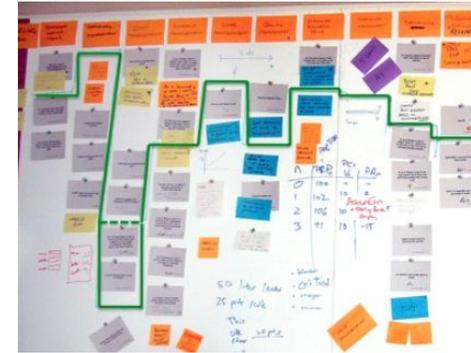


Le produit agile

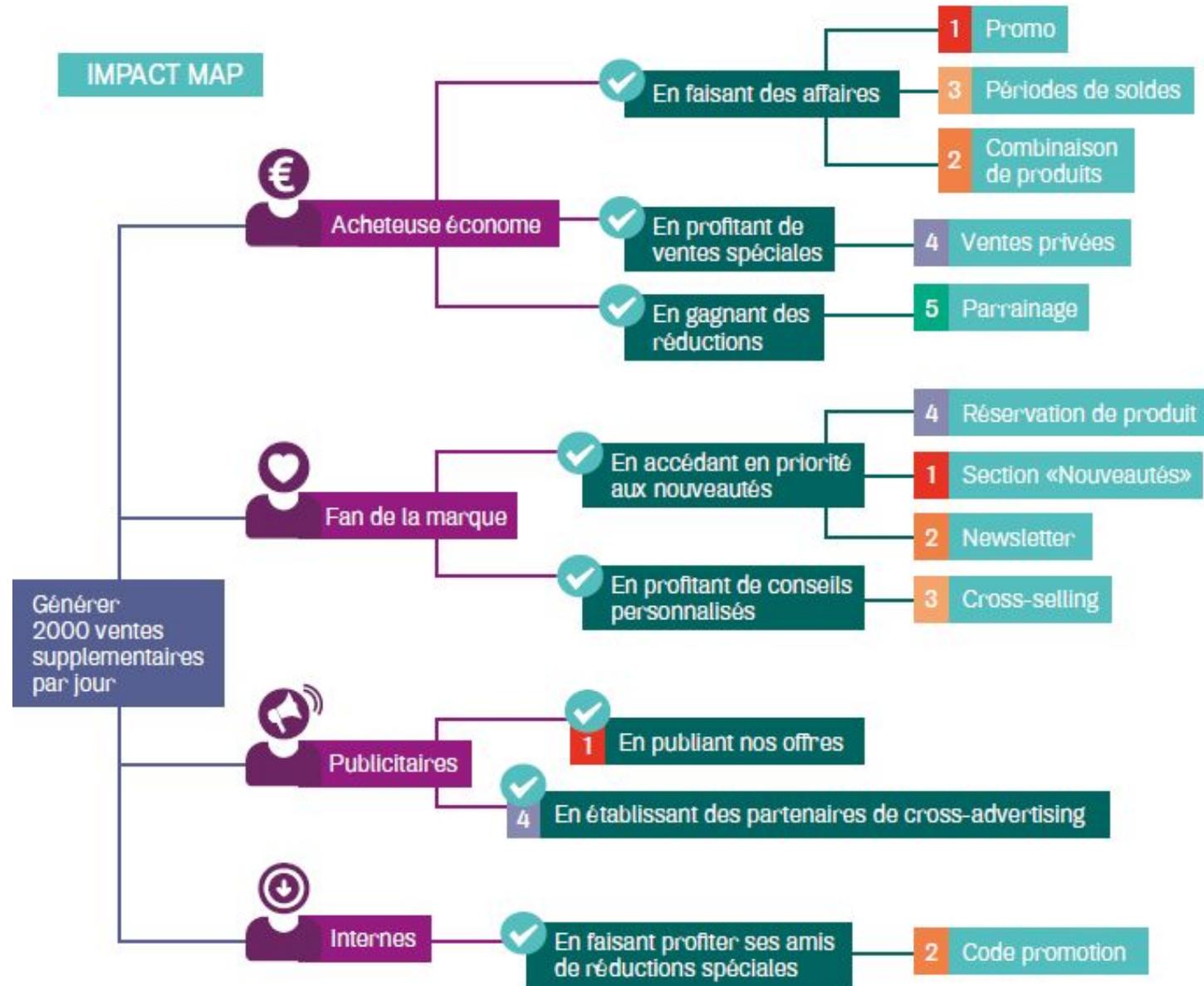


STORY MAPPING

- Une autre façon de représenter le backlog de produit
- Une pratique **collaborative**
- La définition, de manière **macroscopique**, le besoin
- L'élaboration de la roadmap de réalisation priorisée par la valeur.



IMPACT MAPPING





Valeur



MVP

Un **MVP** n'est pas une version light de votre produit mais un produit à part entière présenté sous une forme différente du produit cible afin de **permettre la validation de vos hypothèses de valeur ajoutée auprès de vos clients**

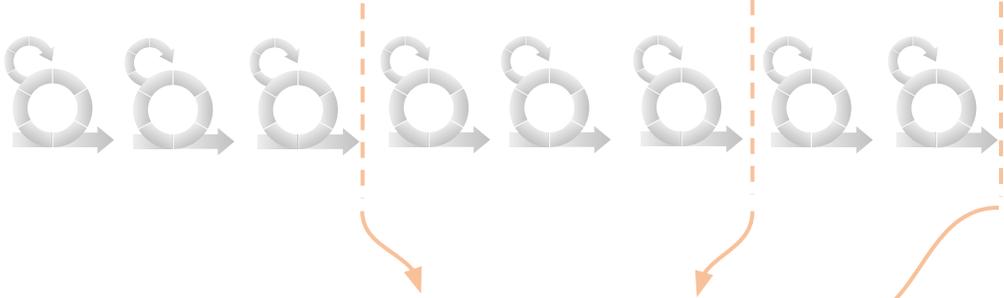


Plutôt que de réaliser un prototype dont le temps de développement aurait été coûteux et le niveau de qualité insuffisant pour convaincre, le fondateur de Dropbox Drew Houston a réalisé une vidéo de démonstration de son produit (qui n'existait pas encore).



Plutôt que d'investir dans coûteux un stock de chaussures et monter une boutique en ligne au succès imprévisible, le fondateur de Zappos Tony Hsieh a simplement conçu un semblant de site e-commerce présentant des chaussures qu'il avait photographié dans les magasins physiques proches de chez lui où il allait les acheter avant de les envoyer aux clients les ayant commandé en ligne.

MVP – Livrer régulièrement

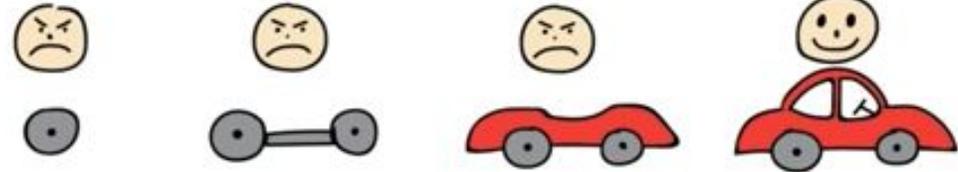


Mise en production régulière

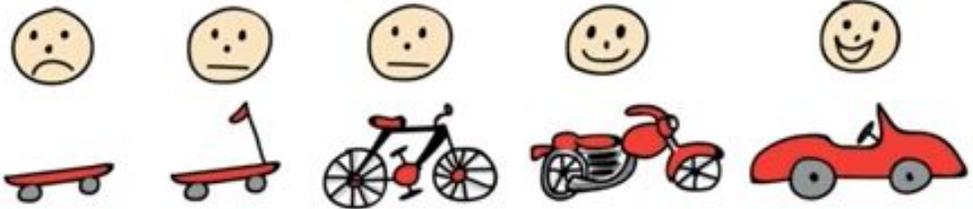
- Bénéfices utilisateurs plus tôt
- Apprentissage avec des retours terrains réels

Comment Prioriser ?

Not like this...



Like this!



Qu'est ce que la valeur?

Ce que le produit représente comme valeur aux yeux de l'utilisateur

Valeur pour l'entreprise

Amélioration de l'efficacité de l'entreprise:

- Gain de parts de marché
- Avoir un avantage concurrentiel durable
- Réduire ses coûts
- Rendre un service à sa clientèle
- Eviter des pénalités
- Réduire un risque
- Améliorer son image

....

Valeur pour l'utilisateur

Assouvir des besoins:

- Bien-être (confort matériel, environnement de travail)
- Estime (besoin de considération, reconnaissance des capacités)
- Sécurité (craintes de prise de risques individuels)
- Orgueil (réussite d'un défi, promotion)
- Intérêt (salaires, primes, avantages divers)
- Nouveauté (curiosité vis-à-vis d'un nouveau produit, d'un projet innovant)

Maximiser la valeur consiste à trouver le meilleur consensus dans les besoins exprimés

Comment définir la valeur?

Les étapes de création de valeur :

- Décider quels sont les bénéfices recherchés et clarifier les critères de valeur
- Identifier les processus métier qui créent ces valeurs et trouver des améliorations
- Identifier des User Stories/features basées sur les améliorations qui créent le plus de valeur

Priorisation

Modèle MoSCoW

Modèle de Kano

Modèle de Wigers

Buy a feature

Prune the Product Tree

WSJF



PRIORISATION : MoSCoW

Le méthode de MoSCoW permet de prioriser à moyen terme selon les critères suivants :

M – Must have : doit être réalisée.

S – Should have : devrait être réalisée si possible.

C – Could have : pourrait être réalisée s'il n'y a pas d'impact sur les autres tâches en cours.

W – Won't have : ne sera pas réalisée tout de suite mais serait souhaitable pour une version ultérieure.

PRIORISATION : KANO

- Le produit possède cette fonctionnalité, qu'en pensez-vous ? (forme fonctionnelle)
- Le produit ne possède pas cette fonctionnalité, qu'en pensez-vous ? (forme dysfonctionnelle)

Avec comme réponses possibles :

- « Aime » (J'aimerais ça)
- « Attend » (Je m'attends à ce qu'il en soit ainsi)
- « Neutre » (Cela m'est égal)
- « Vit avec » (Je l'accepte)
- « N'aime pas » (Je n'aimerais pas ça)

- O** > Obligatoire/Essentielle
- L** > Linéaire
- E** > Excitante
- C** > Contradictoire
- Q** > Questionnable
- I** > Indifférent

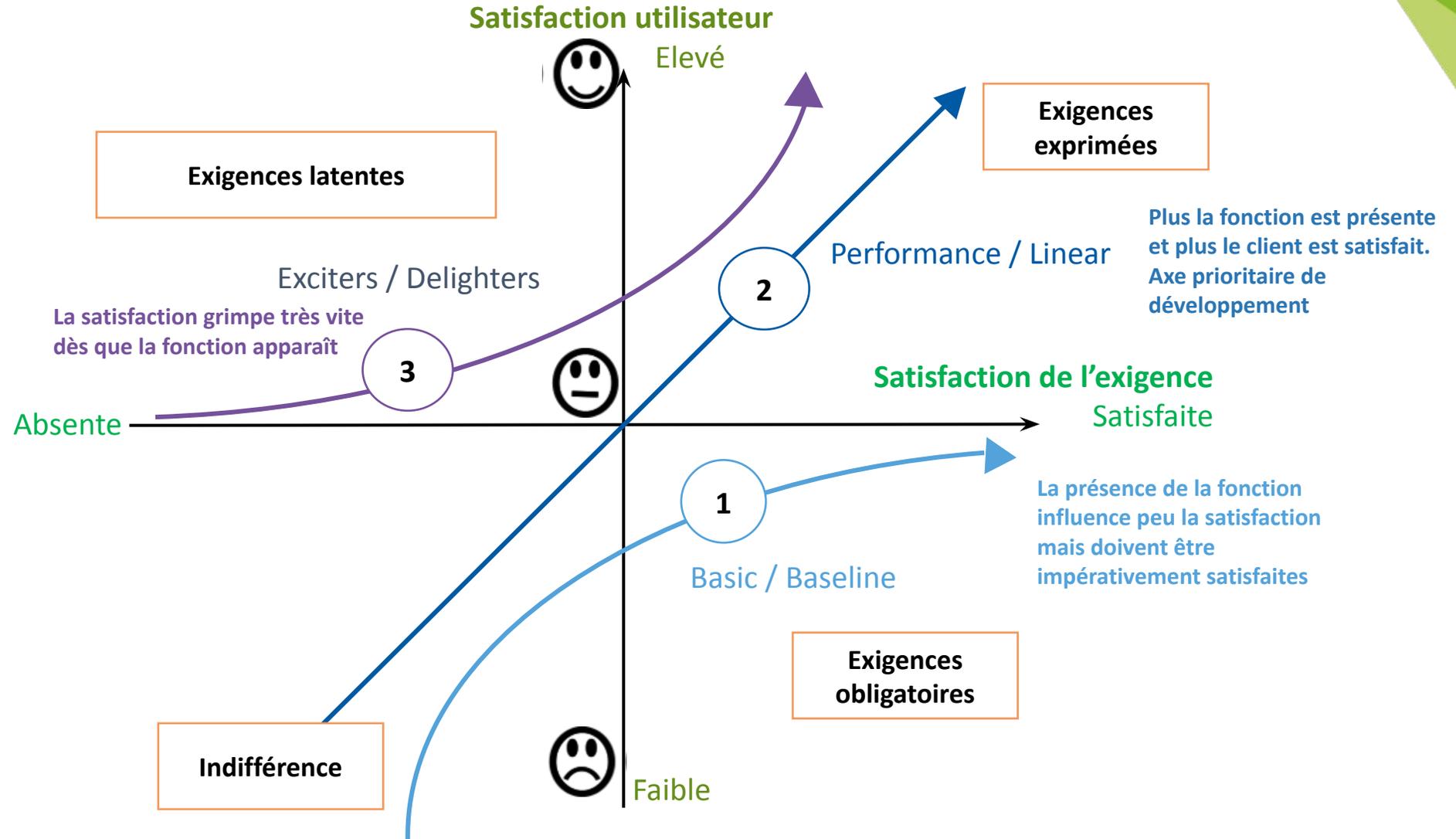
		QUESTION SOUS FORME DISFONCTIONNELLE				
		Aime	Attend	Neutre	Vit avec	N'aime pas
QUESTION SOUS FORME FONCTIONNELLE	Aime	Q	E	E	E	L
	Attend	C	I	I	I	O
	Neutre	C	I	I	I	O
	Vit avec	C	I	I	I	O
	N'aime pas	C	C	C	C	Q

PRIORISATION : KANO

Le modèle de Kano propose de classifier les exigences en trois catégories:

- Obligatoire
- Exprimée
- Latente

Et de mesurer leur corrélation avec la satisfaction du client.



PRIORISATION : KANO - Exemple

On essaie avec les fonctions de votre téléphone ?

- Téléphoner
- Suivre la chaîne parlementaire
- Accorder ma guitare
- 2 jours sans recharger



PRIORISATION : Wiegers

Priorisation de Karl Wiegers (1999) ou méthode des poids relatifs

Principes:

- Evaluation pour chaque exigence du backlog du bénéfice relatif de son implémentation et du préjudice relatif de son non implémentation.
- Utilisation d'une échelle de 1 à 9 sur le bénéfice (1 – Peu de valeur, 9 – Bénéfice maximale)
- Utilisation d'une échelle de 1 à 9 sur le préjudice (1 – Faible préjudice, 9 – Préjudice maximale)

Exigence	Bénéfice relatif	Préjudice relatif	Valeur totale	% Valeur	Cout estimé en SP	% du cout	Priorité
Item 1	5	8	13	48 %	20	60%	0.8
Item 2	6	2	8	30%	5	15%	2
Item 3	2	4	6	22%	8	24%	0.92
Total	13	14	27	100%	33	100%	

Haute prio.

PRIORISATION : Buy a feature

Principes:

- A utiliser dans un contexte multi-sources d'un product backlog.
- Chaque feature a un prix (estimé à l'avance)
- Donner un budget fini à chaque participant (100 unités par exemple).
- Tous les participants doivent « acheter » des features
- Les items qui ont le plus de voix sont réalisés en premier
- Pour les autres, ils doivent négocier entre eux et apporter des arguments constructifs

Exigence	Stakeholder 1	Stakeholder 2	Stakeholder 3	Stakeholder 4	TOTAL
Item 1	50	20	30	30	130
Item 2	30	10	70	30	140
Item 3	20	60	0	40	120
...					

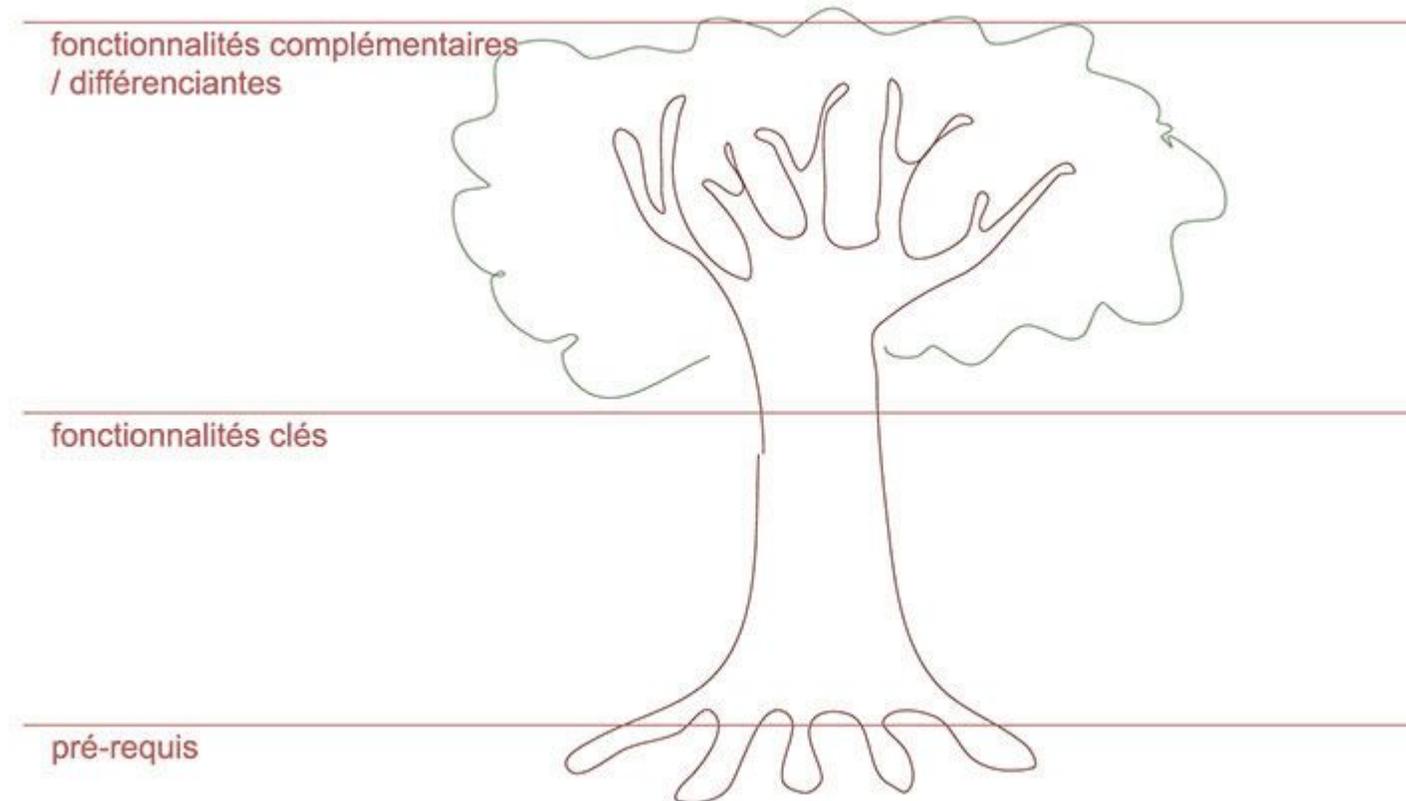
Haute prio.

PRIORISATION : Prune the product Tree

Identifier, catégoriser et prioriser les fonctionnalités en équipe

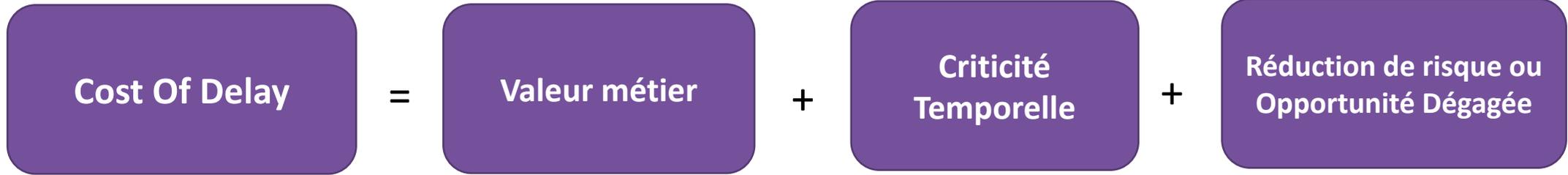
Principes:

- ❑ Les features sont sur des post-it
- ❑ Les racines pour les pré-requis du produit/service (développeur, infrastructure, prestataire...)
- ❑ Le tronc pour les fonctionnalités indispensables au produit/service (connexion utilisateur...)
- ❑ Les branches pour décrire les fonctionnalités complémentaires au produit voire différenciantes
- ❑ Les doublons seront enlevés
- ❑ Les participants voteront sur les éléments qu'ils jugent les plus importants



PRIORISATION : WSJF

Pour prendre en compte le coût du délai...



Standard de l'industrie

Origine : Reinersten / Principles of Product Development Flow

Valeur relative du bénéfice business en terme de revenus générés ou de pénalités financières

Importance relative de respecter sa mise à disposition à court termes

Capacité à réduire au plus tôt un risque business ou de créer les conditions concrétisant de nouvelles opportunités business

Feature	Valeur métier	Criticité Temporelle	Réduction Risque ou Opportunité Dégagée	CoD	Taille	WSJF

Echelle pour chaque paramètre : 1, 2, 3, 5, 8,13, 21

Traiter une colonne à la fois, prendre la plus faible valeur et lui attribuer la valeur "1".

Principe de priorisation

Avec le temps et l'évolution des marchés, la priorisation va changer.
Elle doit donc être revue régulièrement





Maximiser la valeur



Analyser le marché



- Définir des « Personae »
- Calculer l'impact d'une fonction sur les types d'utilisateurs
 - Nombre de ventes supplémentaires si elle est présente
- Analyser les fonctionnalités par rapport aux concurrents
- Différentiateur
 - caractéristique qui va faire pencher l'utilisateur vers mon produit
 - caractéristique qui rend le produit moins cher ou moins cher à exploiter
 - Prix et TCO (Total Cost of Ownership)
 - caractéristique qui fait que j'augmente la marge de mon service (produit interne)
 - caractéristique qui vient « attaquer » le différentiateur d'un concurrent
- Ne pas réaliser en priorité de fonctions dont le ROI est faible ou négatif !

Le ROI, le TCO

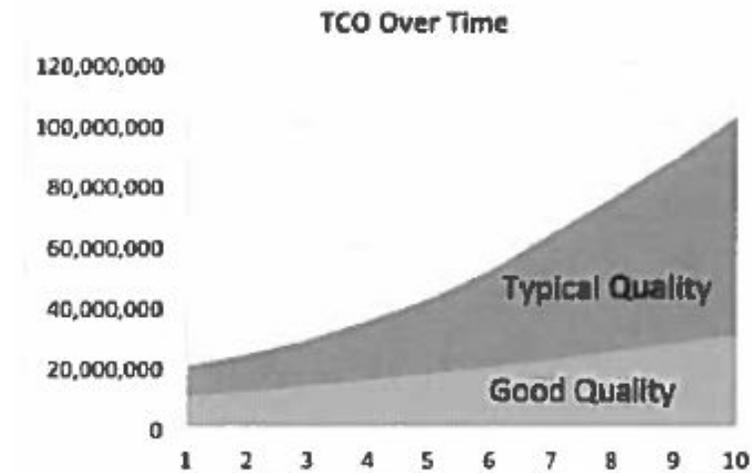
- Le ROI (Return On Investment) est égal à **Valeur créée / Total Cost of Ownership**
 - TCO = Coûts de développement + coûts d'exploitation
 - Coûts de production
 - Coûts de déploiement
 - Coûts de consommation
 - Coûts de support / Maintenance
- Le ROI dépend de plusieurs facteurs :
 - Fonctions fournissant de la valeur au client
 - Absence de fonctions inutiles
 - Haute qualité du code (facilité de maintenance et d'évolutivité)
 - Défauts trouvés et corrigés avant la livraison
 - Compétences et disponibilité des personnes
 - Prédicibilité



Le ROI long terme



- **Améliorer la valeur consiste à**
 - Optimiser le ROI
 - Nouvelle fonctionnalité qui crée de la valeur
 - Réduction des coûts d'exploitation des fonctionnalités existantes
 - Réduction des coûts de production (développement des prochaines fonctionnalités)
- Equilibrer les décisions court-long terme
 - Développement à faible cout
 - Coût de maintenance élevé
 - Attention à la dette technique



Minimiser les gaspillages, ce qui n'a pas de valeur

1. La **Surproduction** de fonctionnalités ou de documentations non nécessaires immédiatement
2. Les **Surstocks** de fonctions ou de documents non livrés aux clients car en cours de développement ou non mis à jour
 - Aussi appelé le WIP (Work In Progress)
3. Le **Transport inutile** : livraisons ou déplacements, opérations de transfert
4. Les **Opérations Inutiles** et processus qui n'ont pas de valeur
 - Se poser la question de leur utilité
5. Les **Mouvements** : temps perdu à transmettre ou rechercher des informations
 - Trop grand nombre de mails
 - Répertoires de documents partagés non structurés
 - Temps passé au téléphone pour trouver le bon interlocuteur
7. Les **Erreurs** : bugs non décelés lors du développement
 - Plus il est détecté rapidement plus un bug est rapide à corriger, moins il coute cher
8. Les **Temps d'Attente** : temps qui ralentissent le projet
 - Les retards de livraison
 - Les questions qui restent sans réponse
 - Les décisions qui tardent à venir

Eviter la dette technique



Dette technique : toute action qui entraîne des coûts supplémentaires dans le futur

- Des bugs connus « qu'on corrigera plus tard »
- Des infractions aux règles de design et de code qui créent des bugs « latents »
- Des parties du code que personne n'ose toucher, du code trop complexe
- Des tests non automatisés, qu'on ne fait pas souvent
- De la connaissance exclusive (seule une personne connaît)
- Des doublons

Dette volontaire

Choix conscient en toutes connaissances de causes

Exemple : on délivre une version sans tout tester pour tenir la date, mais on sait que pour la prochaine itération on doit d'abord finir les tests et corriger les défauts

Dette involontaire

Décision orientée court terme qui ne considère pas les impacts

Exemple : Pression implicite pour diminuer la qualité, ne jamais accepter que des tâches de refactoring soient planifiées dans un sprint

Le Product Owner est le seul gérant de cette dette

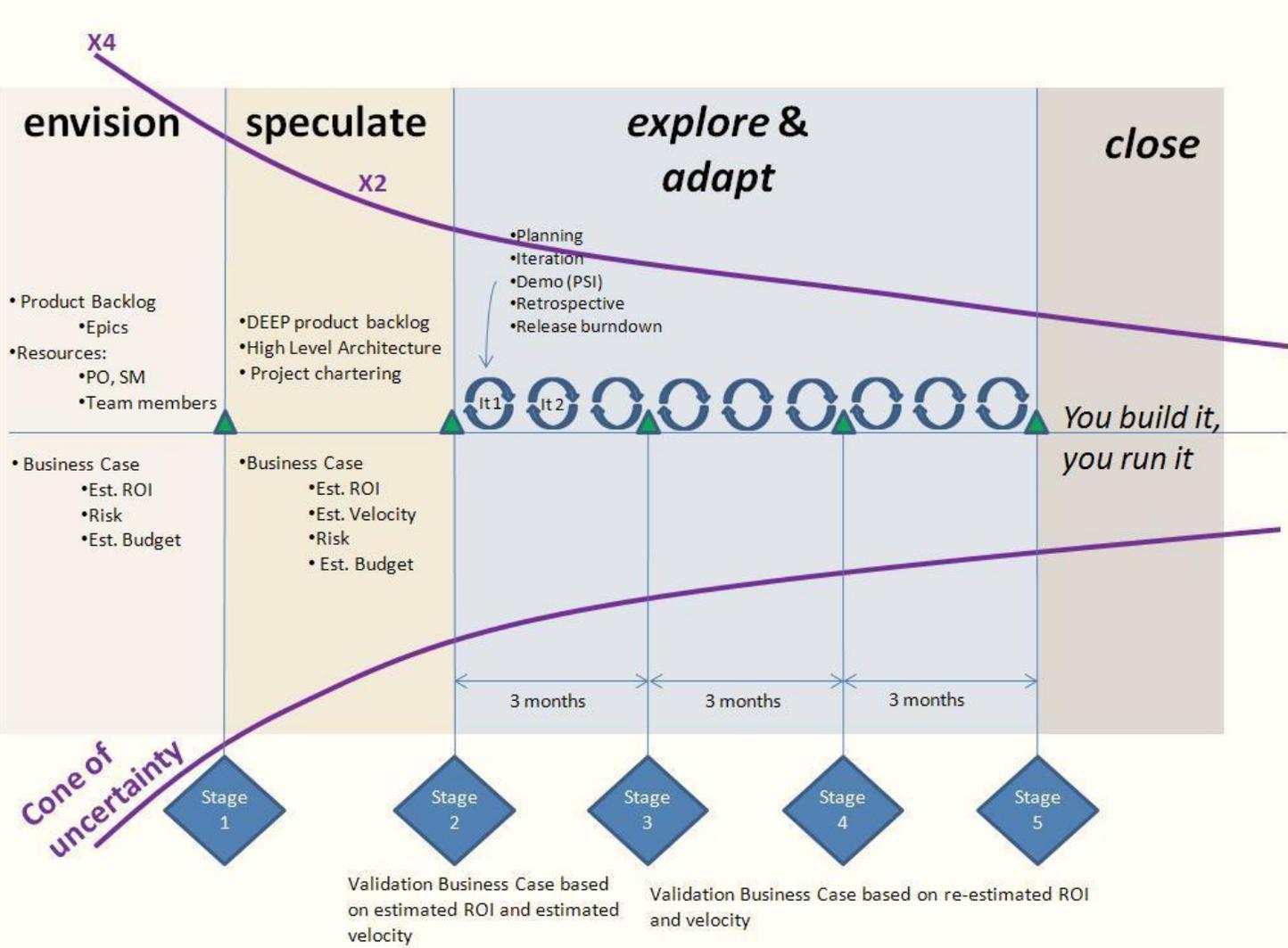
L'équipe doit expliquer et l'informer en toute transparence. La dette technique impactera la vitesse ou la valeur produite

Planification n



Le cône d'incertitude

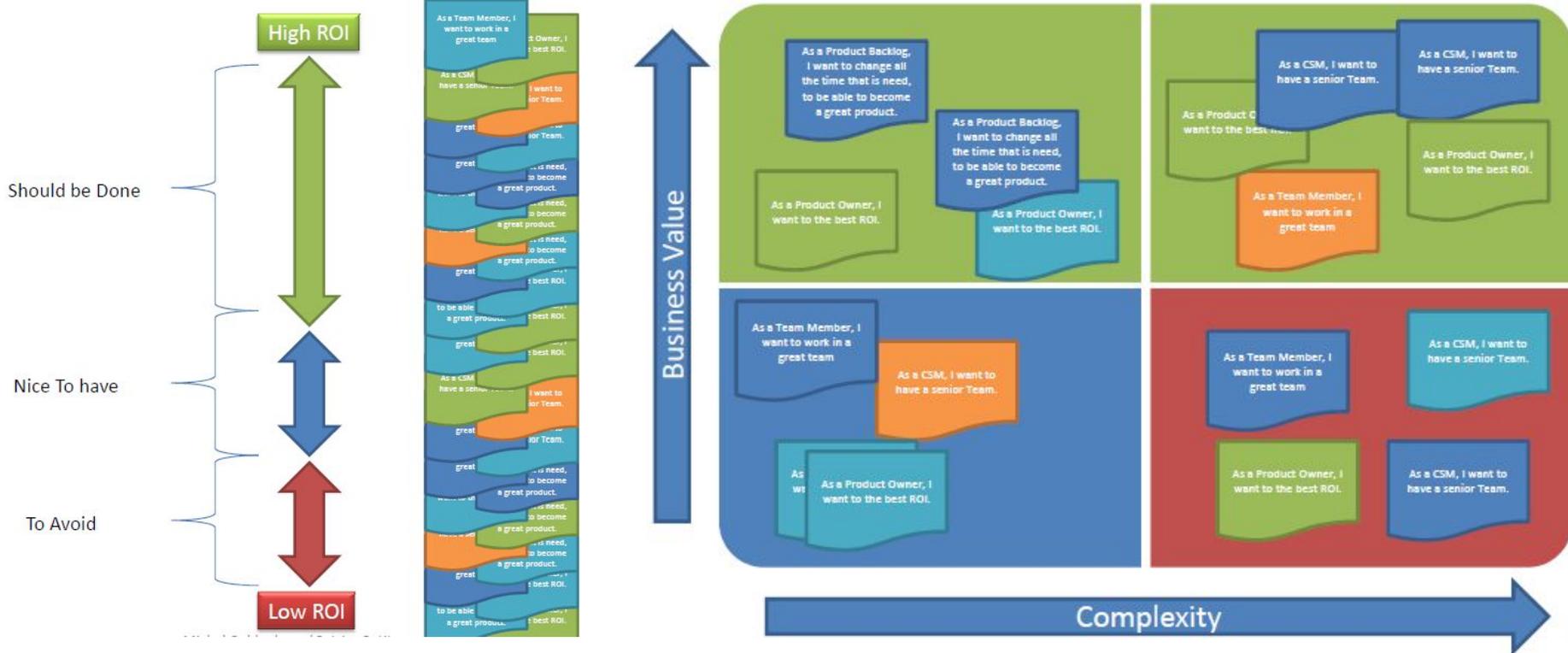
illustre qu'à mesure qu'une prévision s'allonge, elle est de moins en moins certaine



Optimiser la livraison de la valeur



Product Backlog « hiérarchisé » :



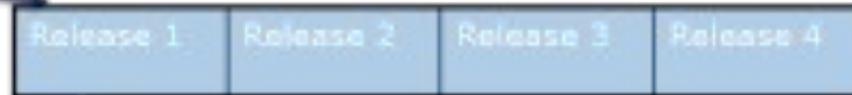
PLANIFICATION DES RELEASES

Une planification à plusieurs niveaux

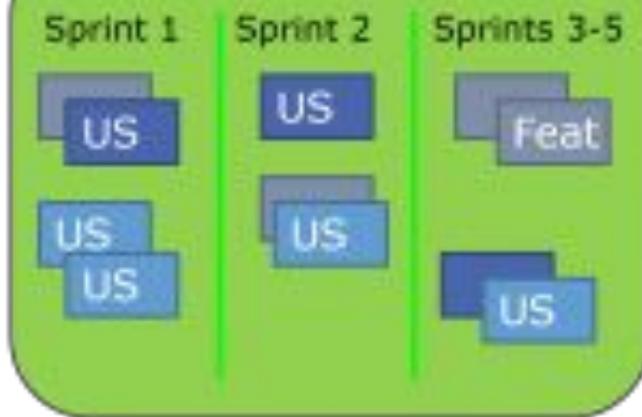
Vision

F1	P1
F2	P2
F3	P2
F4	P3

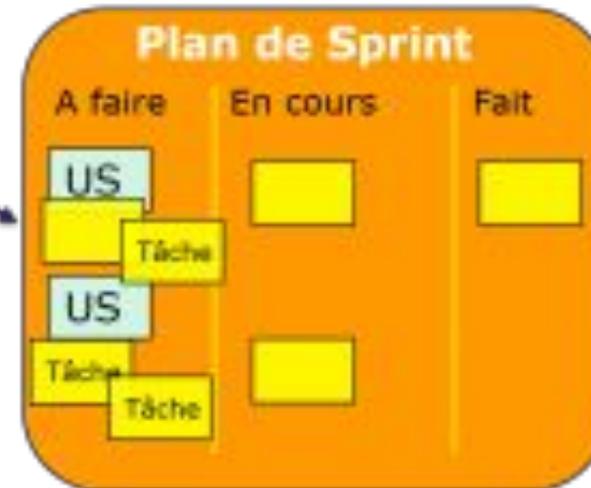
RoadMap



Plan de Release



Plan de Sprint



ROADMAP

Date	12/04	Juin	Q3	Q4	Année Prochaine
Nom release	Release 1
Objectif	Profiter des promotions en cours	Être informé des produits qui m'intéressent	Accéder à des conseils personnalisés	Première sortie privilégiée	Fidéliser
Fonctionnalités	<ul style="list-style-type: none"> ▶ Mise en avant du produit ▶ Code promo 	<ul style="list-style-type: none"> ▶ Soldes ▶ Nouveautés 	<ul style="list-style-type: none"> ▶ Promo combinées ▶ Cross-selling 	<ul style="list-style-type: none"> ▶ Ventes privées ▶ Pré-lancement ▶ Code employé 	<ul style="list-style-type: none"> ▶ Fidélité ▶ Parrainage ▶ Réservation
Métriques	<ul style="list-style-type: none"> ▶ Nombre codes promos utilisés ▶ Nouveaux comptes ▶ <u>Nombre ventes</u> 	<ul style="list-style-type: none"> ▶ Consultation nouvelles pages ▶ Nouveaux comptes ▶ <u>Nombre ventes</u> 	<ul style="list-style-type: none"> ▶ Nombre ventes en promos combinées ▶ Nombre vente en cross-selling ▶ <u>Nombre ventes</u> 	<ul style="list-style-type: none"> ▶ Participants vente-privée ▶ Ventes en vente privée ▶ Participants pré-lancement ▶ Codes utilisés ▶ <u>Nombre ventes</u> 	<ul style="list-style-type: none"> ▶ Nouveaux comptes ▶ Coupons utilisés ▶ Nombre réservations ▶ <u>Nombre ventes</u>



Mesurer la valeur



Valeur produit : Time to Market

- **Release Frequency** : Temps écoulé entre deux livraisons significatives en excluant les maintenances
- **Release Stabilization** : Temps écoulé entre la fin du codage et la livraison
- **Cycle Time** : Temps écoulé pour délivrer un incrément avec au moins une nouvelle fonctionnalité
- **Lead Time** : Temps écoulé entre la demande et la livraison de l'incrément aux mains des utilisateurs



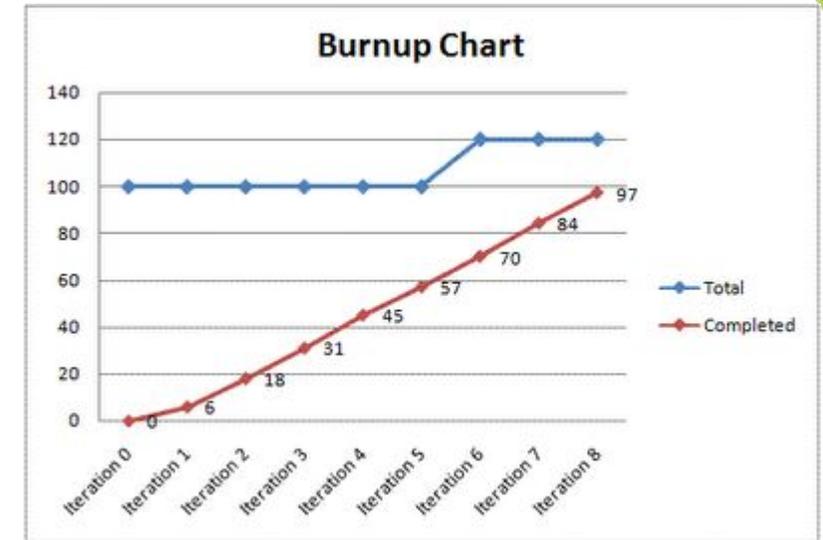
Valeur produit : Ability to Innovate

- **Innovation Rate** : Pourcentage du budget alloué aux nouvelles fonctionnalités par rapport à la maintenance de celles existantes
- **Usage Index** : Utilisation réelle des fonctions délivrées
- **Product Cost Ratio** : Total des dépenses pour le système (TCO)
- **Installed Version Index** : Pourcentage des utilisateurs utilisant la dernière version
- **Total Defects** : Nombre de défauts trouvés dans le produit



Mesure avancé du produit : BurnUp Chart

- Il permet de mettre en évidence **l'avancement du produit**.
- Il peut être consolidé à plusieurs niveaux :
 - Au niveau de Sprint : avancement en terme de finalisation des Users Stories du Sprint (à mettre à jour à chaque Daily Scrum)
 - Au niveau d'une release ou global projet : avancement global du produit final (à mettre à jour à chaque Sprint)
- 2 courbes :
 - L'avancement théorique de réalisation du produit
 - La finalisation réelle des Users Stories.
- L'avancement du produit n'est incrémenté qu'à la finalisation totale d'une User Story (quelque soit le reste à faire en terme de charges).
- A la fin d'un sprint, si toutes les Users Stories ont un taux d'avancement de 90% mais aucune n'est terminée L'avancement produit est égal à 0 car le produit n'est pas démontrable et utilisable en l'état
- **L'objectif de l'équipe est donc de finaliser le maximum de Users Stories plutôt qu'atteindre un taux d'avancement en charges maximal.**

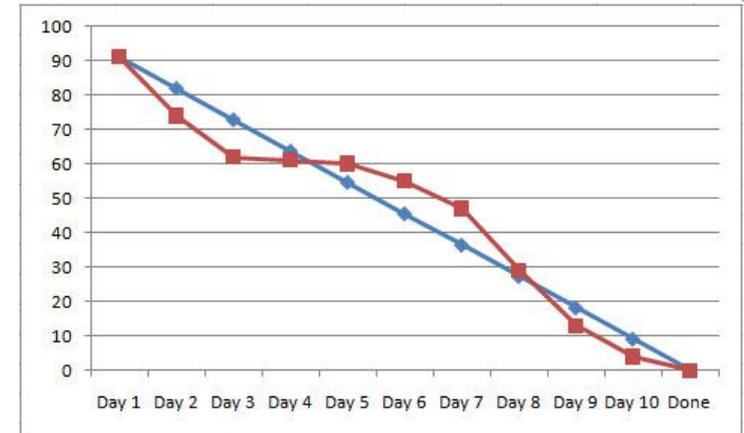


Mesure avancé du produit : Burndown Chart

- Il permet de mettre en évidence **l'avancement du sprint**.
- Il doit être mis à jour tous les jours (avant le Daily Scrum par exemple)

2 courbes :

- La capacité à faire restante de l'équipe d'ici la fin du Sprint
- Le reste à faire pour réaliser l'engagement du Sprint d'ici la fin du Sprint



- **Seuls le travail restant pour compléter une tâche et les dates sont importants.**
- **L'engagement se mesure par la capacité globale de l'équipe à réaliser le reste à faire produit de l'itération**

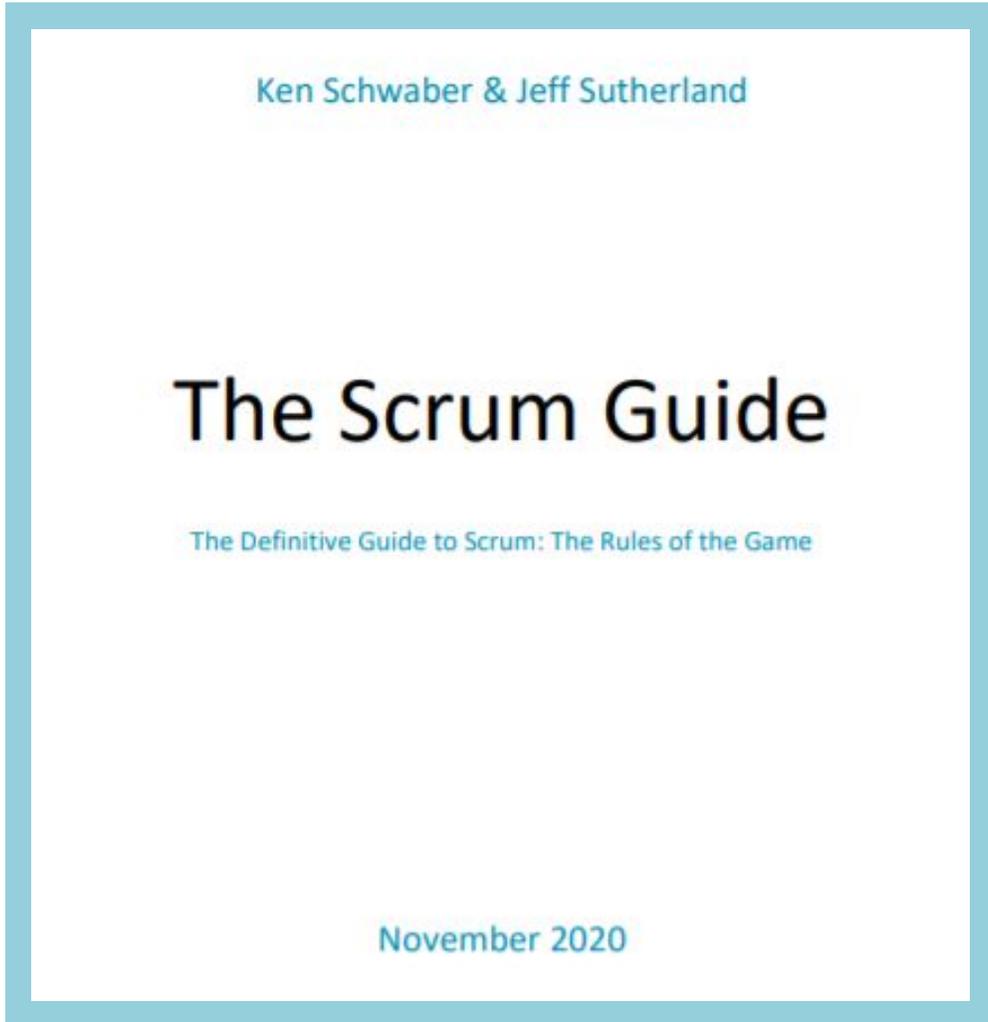
3

Lecture du scrum guide

Le scrum guide



Le Scrum Guide



Scrum : cadre de travail (framework) inventé par Jen Schwaber et Jeff Sutherland dans lequel les personnes peuvent aborder des **problèmes complexes** et adaptatifs tout en livrant de **manière efficace** et créative des produits de la **plus grande valeur possible**.

Scrum est:

- Simple
- Léger
- Difficile à maîtriser

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Définition de Scrum

Scrum est un cadre de travail léger qui aide les personnes, les équipes et les organisations à générer de la valeur grâce à des solutions adaptatives pour des problèmes complexes.

En bref, Scrum a besoin d'un Scrum Master pour favoriser un environnement où :

1. Un Product Owner ordonne le travail à faire pour résoudre un problème complexe dans le Product Backlog.
2. La Scrum Team transforme une sélection de ce travail en un Increment de valeur lors d'un Sprint.
3. La Scrum Team et ses parties prenantes inspectent les résultats et s'adaptent pour le prochain Sprint.
4. *Répéter*

Scrum est simple. Essayez-le tel qu'il est et, déterminez si sa philosophie, sa théorie et sa structure aident à atteindre les objectifs et à créer de la valeur. Le cadre de travail Scrum est volontairement incomplet, ne définissant que les parties nécessaires pour mettre en œuvre la théorie Scrum. Scrum repose sur l'intelligence collective des personnes qui l'utilisent. Plutôt que de fournir aux gens des instructions détaillées, les règles de Scrum guident leurs relations et leurs interactions.

Divers processus, techniques et méthodes peuvent être employés dans ce cadre de travail. Scrum englobe des pratiques existantes ou les rend inutiles. Scrum rend visible l'efficacité relative du management existant, de l'environnement et des techniques de travail, afin que des améliorations puissent être apportées.

EN BREF

★ 3 types de Resp

Product Owner
Scrum Master
Developers

3 Artefacts

Backlog produit
Backlog de sprint
Incrément

5 Evenements

Sprint Planning
Daily Scrum
Sprint Review
Rétrospective
Sprint

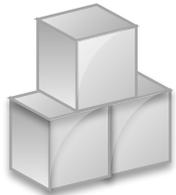
5 Valeurs

Engagement
Focus
Ouverture
Respect
Courage

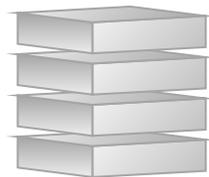
3 Piliers

Transparence
Inspection
Adaptation

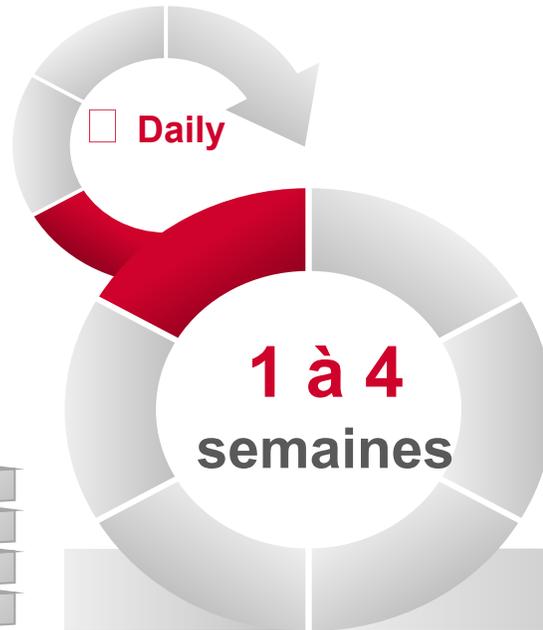
Backlog produit



□ Sprint planning



Sprint Backlog
Figé pendant le sprint



SPRINT

□ Revue de sprint

□ Rétrospective



**Version intermédiaire
potentiellement
déployable**

Théorie Scrum



Scrum est fondé sur :

❑ **l'empirisme**

- La connaissance s'acquiert par l'expérience
- La prise de décision s'appuie sur l'observation des faits

❑ **la pensée lean**

- Réduit le gaspillage
- Se focalise sur l'essentiel

Les trois piliers de Scrum :

- La transparence
- L'inspection
- L'adaptation

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Théorie Scrum

Scrum est fondé sur l'empirisme et la pensée Lean. L'empirisme affirme que la connaissance provient de l'expérience et que la prise de décision s'appuie sur l'observation de faits. La pensée Lean réduit le gaspillage et se focalise sur l'essentiel.

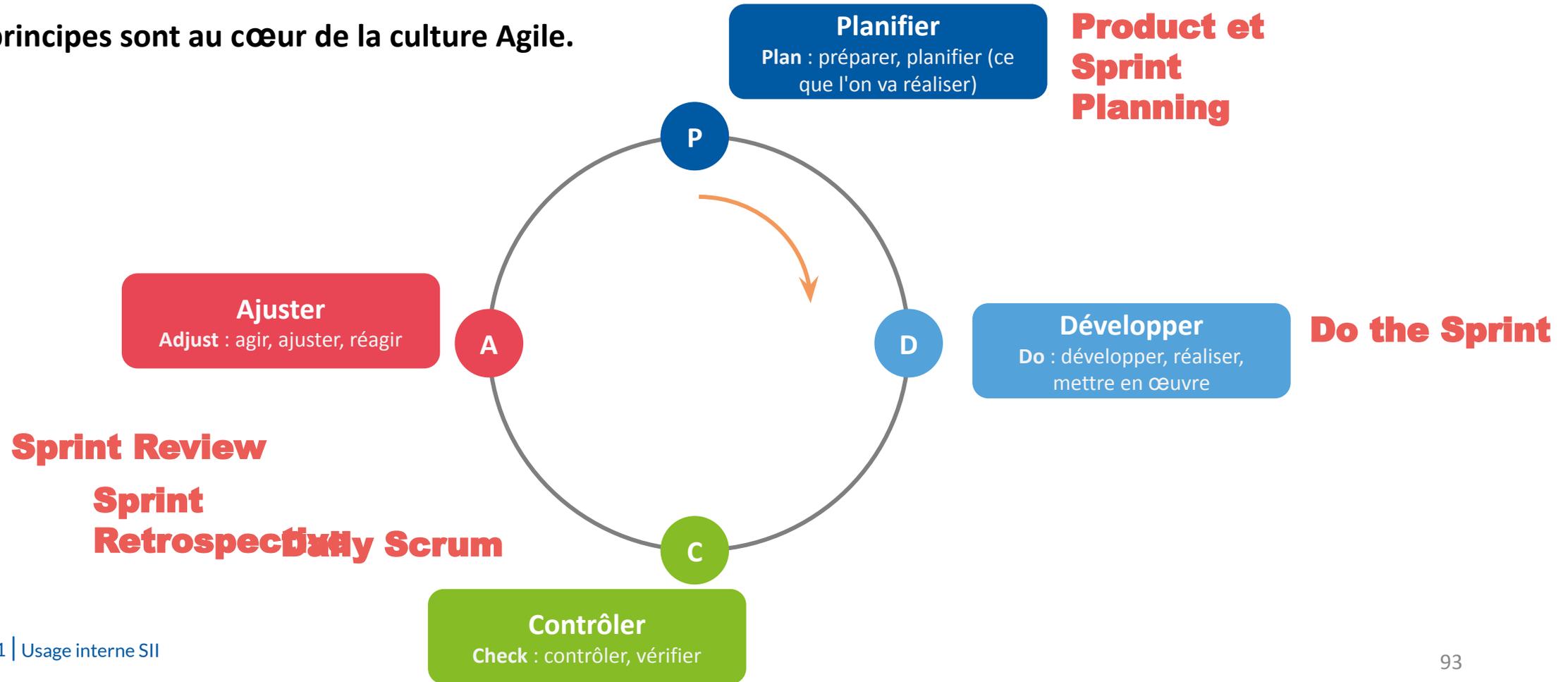
Scrum utilise une approche itérative et incrémentale pour optimiser la prédictibilité et le contrôle de risque. Scrum engage des groupes de personnes qui ont collectivement toutes les compétences et l'expertise requises pour faire le travail et partager ou acquérir de telles compétences selon les besoins.

Scrum combine quatre événements formels pour l'inspection et l'adaptation dans un événement conteneur, le Sprint. Ces événements fonctionnent parce qu'ils mettent en œuvre les piliers empiriques de Scrum de transparence, d'inspection et d'adaptation.

Favoriser l'apprentissage et l'expérimentation

La **Roue de Deming** décrit un processus d'apprentissage et d'amélioration continue de la qualité, basé sur des cycles courts permettant de contrôler et d'adapter régulièrement les travaux effectués.

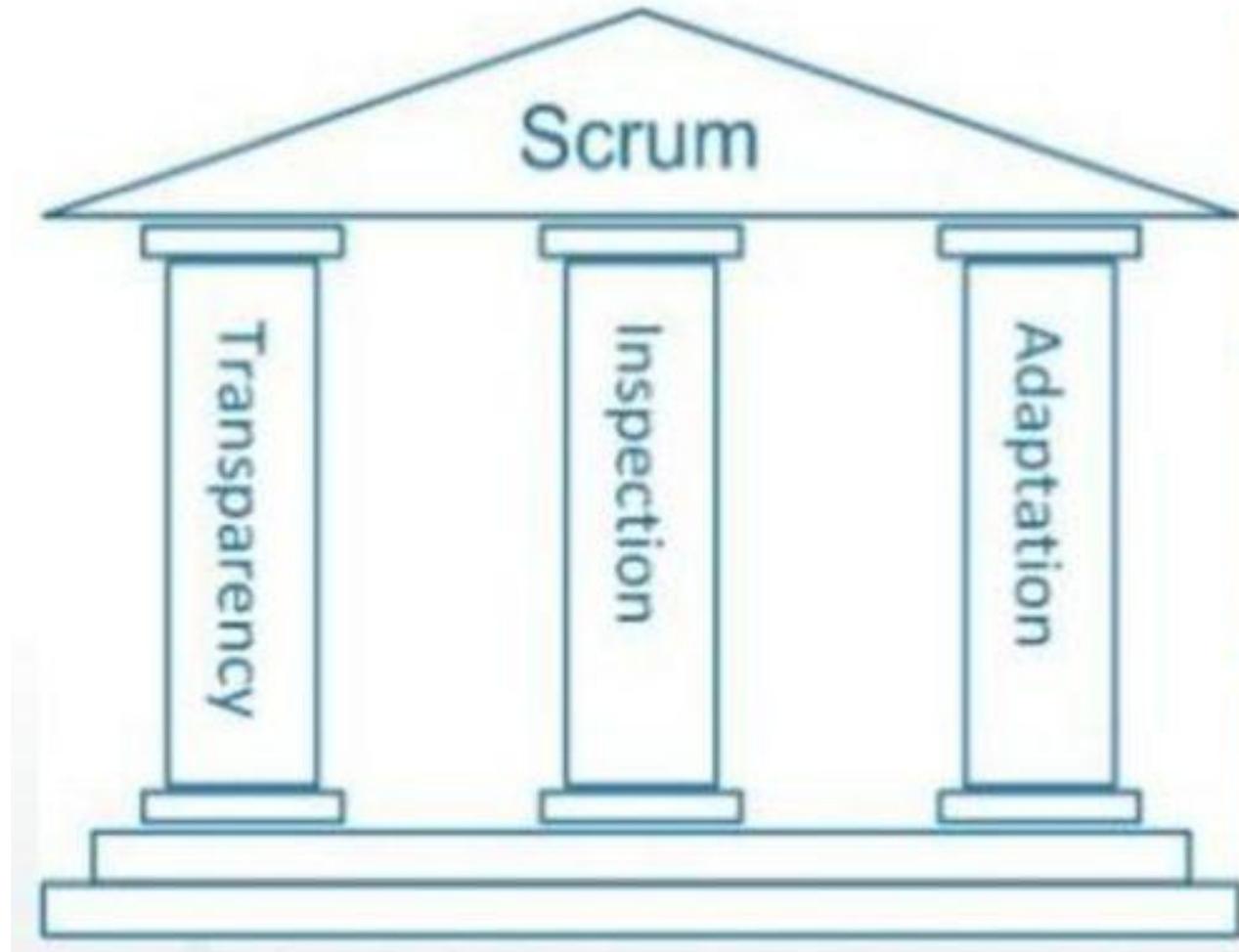
Ces principes sont au cœur de la culture Agile.



Les piliers



Les trois piliers empiriques de Scrum



Extrait du Scrum Guide



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Transparence

Le processus et le travail émergents doivent être visibles pour ceux qui effectuent le travail ainsi que pour ceux qui le reçoivent. Avec Scrum, les décisions importantes sont fondées sur l'état perçu de ses trois artefacts formels. Des artefacts peu transparents peuvent mener à des décisions qui diminuent la valeur et augmentent le risque.

La transparence permet l'inspection. Une inspection sans transparence est trompeuse et source de gaspillage.

Extrait du Scrum Guide



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Inspection

Les artefacts Scrum et les progrès vers les objectifs convenus doivent être inspectés fréquemment et avec diligence pour détecter des écarts ou des problèmes potentiellement indésirables. Pour faciliter l'inspection, Scrum fournit une cadence sous la forme de ses cinq événements.

L'inspection permet l'adaptation. Une inspection sans adaptation est considérée comme infructueuse. Les événements Scrum sont conçus pour provoquer le changement.

- Scrum prescrit quatre occasions formelles d'inspection et d'adaptation, tel que décrit dans la section Événements Scrum du Scrum Guide :
 - Sprint Planning
 - Daily Scrum
 - Sprint Review
 - Sprint Retrospective

Extrait du Scrum Guide



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Adaptation

Si certains aspects d'un processus s'écartent des limites acceptables ou si le produit résultant est inacceptable, alors le processus appliqué ou les éléments produits doivent être adaptés. L'adaptation doit être effectuée le plus rapidement possible afin de minimiser tout écart supplémentaire.

L'adaptation devient plus difficile lorsque les personnes impliquées ne sont pas en possession de tous leurs moyens ou autogérées. Une Scrum Team doit s'adapter dès lors que l'inspection révèle quelque chose de nouveau.

- Scrum prescrit quatre occasions formelles d'inspection et d'adaptation, tel que décrit dans la section Événements Scrum du Scrum Guide :
 - Sprint Planning
 - Daily Scrum
 - Sprint Review
 - Sprint Retrospective

Les valeurs



Scrum c'est 5 valeurs permettant l'émergence des 3 piliers

5 valeurs



Courage

De faire les bonnes choses et de travailler sur des problèmes difficiles



Focus

Sur le travail à faire pendant le sprint et sur les objectifs de l'équipe



Engagement

Les gens s'engagent personnellement à la réalisation des objectifs de l'équipe Scrum



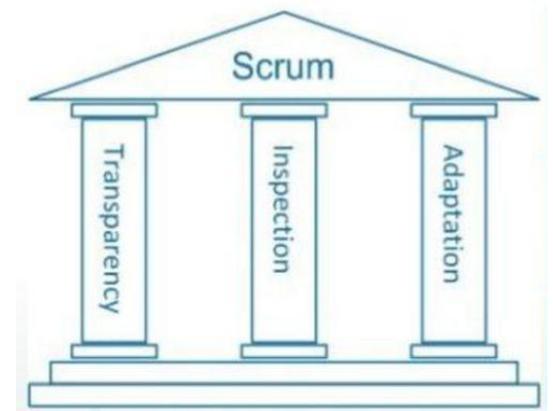
Respect

Les membres de l'équipe Scrum se respectent mutuellement en tant que personnes compétentes et indépendantes



Ouverture

De l'équipe et des parties prenantes sur tout le travail et les défis allant avec l'exécution du travail



“Because it involves to so many variables, the scrum is impossible to regulate consistently. It is inevitable that it becomes a tangle of broken rules and roguish players.” — [The Evolution of Scrum](#)

Extrait du Scrum Guide



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Valeurs Scrum

L'application réussie de Scrum dépend de la capacité des personnes à mieux vivre ces cinq valeurs :

Engagement, focus, ouverture, respect et courage

La Scrum Team s'engage à atteindre ses objectifs et à se soutenir mutuellement. Son but principal est la réalisation du Sprint pour progresser le plus possible vers ces objectifs. La Scrum Team et ses parties prenantes sont ouvertes sur le travail et les défis à relever. Les membres de la Scrum Team se respectent mutuellement pour être des personnes compétentes et indépendantes et, sont respectés en tant que tels par les personnes avec lesquelles ils travaillent. Les membres de la Scrum Team ont le courage de mener les bonnes actions et de travailler sur des problèmes difficiles.

Ces valeurs orientent le travail, les actions et l'attitude de la Scrum Team. Les décisions et les mesures prises, ainsi que la façon dont Scrum est utilisé, doivent renforcer ces valeurs, ne pas les diminuer ni les saper. Les membres de la Scrum Team apprennent et explorent ces valeurs tout en travaillant avec les événements et les artefacts Scrum. Lorsque ces valeurs sont incarnées par la Scrum Team et les personnes avec lesquelles elle travaille, alors les piliers empiriques Scrum de transparence, d'inspection et d'adaptation émergent en consolidant la confiance.

Les artéfacts



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Les Artefacts de Scrum

Les artefacts de Scrum représentent un travail ou une valeur. Ils sont conçus pour maximiser la transparence des informations clés. Ainsi, tous ceux qui les inspectent ont la même base d'adaptation.

Chaque artefact contient un engagement qui apporte l'information nécessaire à la transparence et au focus rendant possible la mesure de la progression :

- Pour le Product Backlog, il s'agit de l'Objectif de Produit.
- Pour le Sprint Backlog, c'est l'Objectif de Sprint.
- Pour l'Increment, c'est la Definition of Done (définition de Fini).

Ces engagements existent pour renforcer l'empirisme et les valeurs Scrum au sein de la Scrum Team et ses parties prenantes.

Scrum – Artefacts pour maximiser la transparence

Product Backlog

C'est ce qui est nécessaire pour le produit à un instant T

- Engagement via le **Product Goal**
- Liste ordonnée des éléments d'un produit (fonctionnel et non fonctionnel)
- Constitue le **référentiel** pour tout changement/évolution à apporter à un produit (build et run, fonctionnel et technique). Unique source d'exigences
- Il est de la responsabilité du **Product Owner** (contenu, disponibilité et priorisation).
- Ces items sont composés d'une **description, d'une valeur et d'une estimation**.
- Les **premiers** items sont généralement plus **détaillés** que les suivants.
- Il n'est jamais complet, **évolue** au fur et à mesure et existe tant que le produit existe.

Sprint Backlog

C'est le périmètre que l'équipe réalise pour le produit dans un sprint

- Engagement via le **Sprint Goal**.
- Constitue un **engagement** de qui sera présent dans le prochain incrément et le **travail nécessaire** pour livrer cette fonctionnalité dans un incrément « fini ».
- Il rend **visible** tout le travail que l'équipe de réalisation identifie comme nécessaire pour **atteindre l'objectif de l'itération**.
- Il peut également comprendre une **amélioration prioritaire** identifiée lors de la précédente rétrospective.
- Il est **suffisamment détaillé** que pour que la **progression** soit **compréhensible** lors du daily.
- **Seuls** les développeurs peuvent **changer son contenu** pendant une itération.

Incrément

C'est le **produit démontré** qui peut potentiellement aller en production

- Engagement via la **Definition of Done**
- Est constitué des **items du backlog produit « finis »** pendant l'itération ainsi que la **valeur cumulative** des incréments livrés dans les itérations précédentes.
- A la fin de l'itération, l'incrément **doit être « fini »**, ce qui implique qu'il doit être dans un état « **shippable** ».

Définitions des ENGAGEMENTS

	Product Goal	Sprint Goal	Definition of Done
Qu'est-ce que c'est ?	Etat futur du produit; Objectif à long terme de la Scrum Team	But unique du Sprint	Liste de critères qui satisfait les mesures de qualité requises pour le produit
Quand est-il défini ?	Au démarrage d'un produit, ou celui précédemment défini est atteint ou devient obsolète	Pendant le Sprint Planning	Au démarrage de la Scrum Team et mis à jour régulièrement
Où le trouve-t-on ?	Dans le Product Backlog (1 à la fois)	Dans le Sprint Backlog (1 par Sprint)	Fait partie de l'Incrément
Qui le définit ?	Le Product Owner	Scrum Team	Scrum Team (ou partagée par toutes les scrum Team si standard organisation ou si même produit)
Quelle temporalité ?	Tant que l'objectif n'est pas atteint, ou non-obsolète	La durée du Sprint	Tant que la Scrum Team travaille sur le produit
Pourquoi ?	Sert de cible à la Scrum Team pour planifier	Permet de créer coh2rence et focus	Apporte de la transparence sur la compréhension commune du travail

Définition du produit

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

« Un produit est un vecteur de valeur. Il a une limite claire, des parties prenantes connues, des utilisateurs ou des clients bien définis. Un produit peut être un service, un produit physique ou quelque chose plus abstrait. »

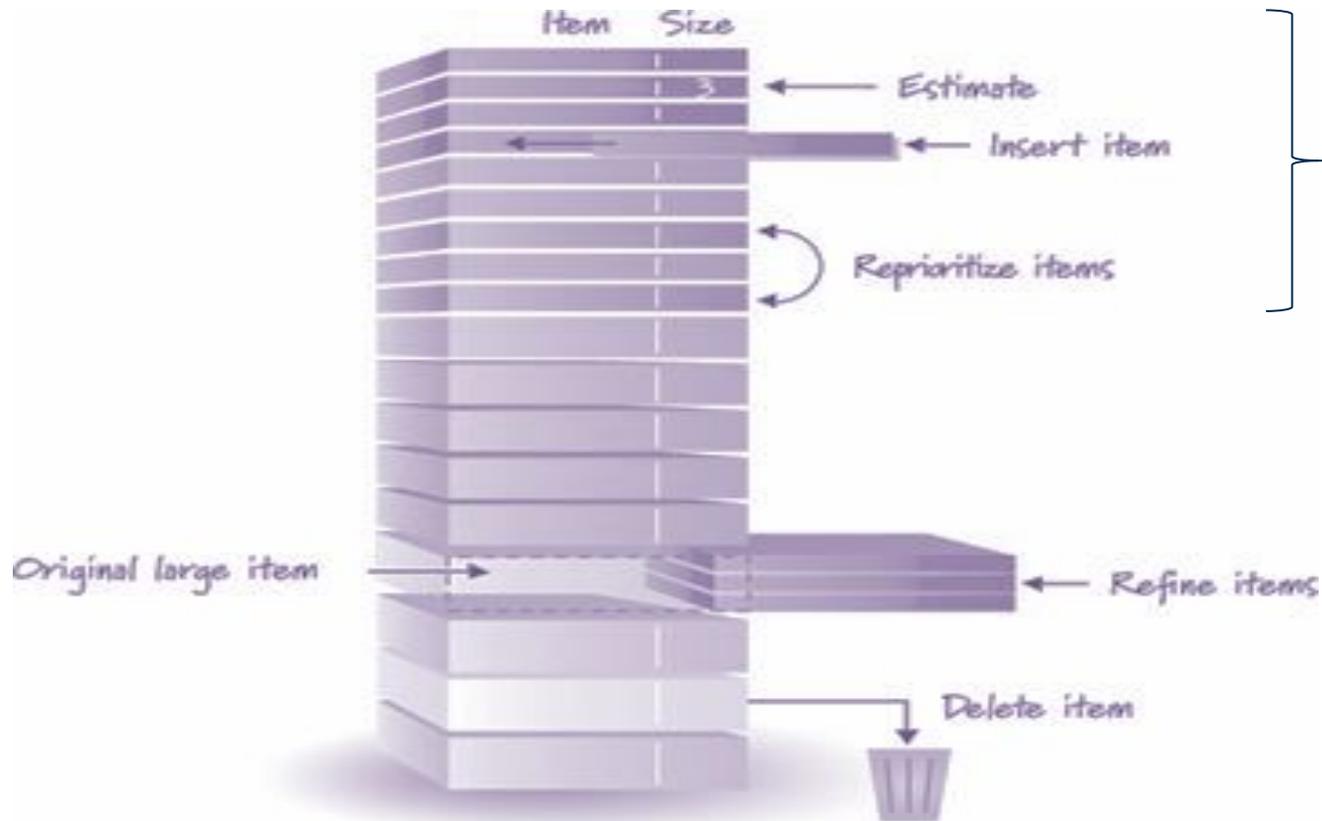
En quoi consiste le Product Backlog?



Objectif produit + les éléments qui permettent d'atteindre cet objectif



Objectif produit : état futur du produit



Les éléments les plus prioritaires sont placés **en haut du Product Backlog**

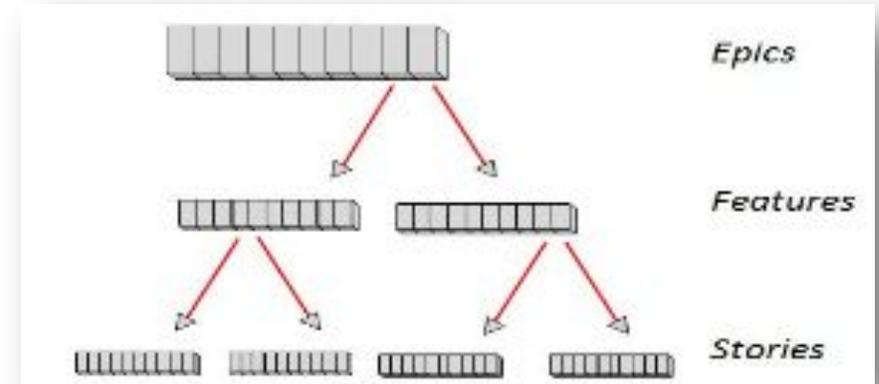
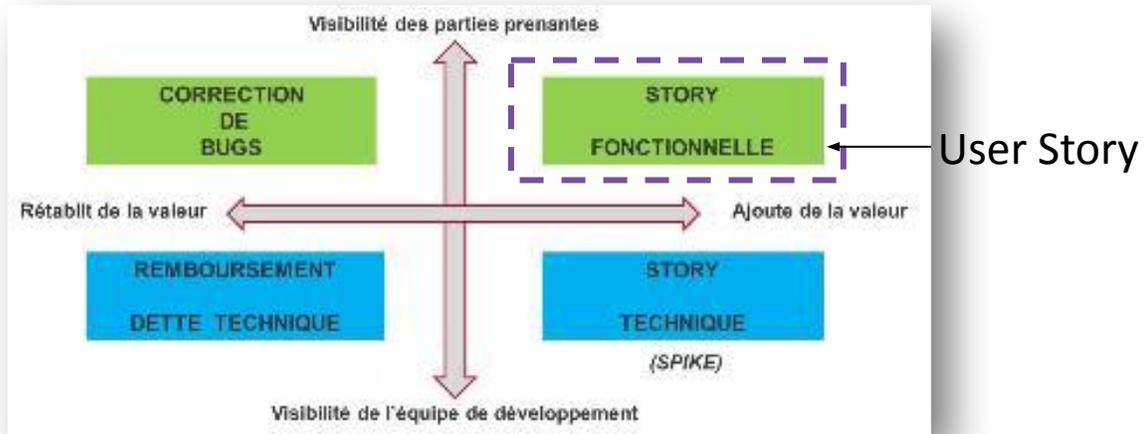
Un bon Product Backlog doit être **D.E.E.P.** c'est-à-dire :

- D**étaillé (juste ce qu'il faut quand il le faut)
- E**stimé (au-delà d'une liste c'est un outil de planning)
- E**mergent (évolutif tout au long du projet)
- P**riorisé (avec ce qui a le plus de valeur en tête de liste)



- ✓ Le Product Backlog est en perpétuelle évolution.
- ✓ Ses items doivent être affinés juste à temps.

Les différents items composant le Product Backlog



- Le Backlog est alimenté par 4 axes :
 - Le besoin
 - La technique
 - La correction d'anomalies
 - Le refactoring

- Souvent, on décompose ces éléments en 2 ou 3 niveaux de granularité
 - Epics
 - Features
 - Stories

Différents niveaux de granularité dans un Product Backlog

- **EPIC** : correspond à un pan applicatif regroupant plusieurs fonctionnalités
- **Features** : correspond à une fonctionnalité ou à un écran applicatif
- **User story** : correspond à un besoin utilisateur unitaire et démontrable
- **Correctifs / bug** : anomalies à corriger dans le sprint en cours ou prochain sprint
- **Technical story** : correspond à un besoin technique, migration technique, ...

Chaque élément doit être valorisé, estimé et priorisé

Gestion des exigences : User Story

Une user story représente un usage d'un utilisateur du produit, et s'écrit:

- **En tant que...**
- **Je veux / peux...**
- **Afin de ...**

Les Users Stories permettent de s'assurer que les besoins sont bien compris

La première étape d'écriture des Users Stories est d'identifier les différents rôles/utilisateurs intervenants.

Syntaxe d'une US

<Identifiant> – <TITRE> -- <Complexité>

En tant que <rôle utilisateur>

Permet de préciser le profil utilisateur et de savoir à qui bénéficie l'histoire utilisateur

Je peux <action>

Description de la fonctionnalité

Afin de <bénéfice>

Explicitation du bénéfice apporté

Description : tout élément permettant de comprendre ce qu'il y a à réaliser (règles de gestion métier, maquettes, impacts techniques etc.)

Cas d'acceptation

Un scénario démontrant le fonctionnement nominal de l'histoire

En cas d'une activité technique / refonte infra ou architecture

Qui veut ce système ? <> Et pourquoi ? <>

Prenant en compte <Etat du système>

Etant donné <Condition>

Lorsque <Rôle + Action>

Alors <Résultat Attendu>

Une bonne user story est **INVEST** (Modèle Invest (Bill WAKE 2003))

- **Independent / Indépendante** Indépendance des autres stories
- **Negotiable / Négociable** La story peut évoluer en fonction des remarques de l'équipe
- **Valuable / Source de valeur** La story apporte de la valeur au client
- **Estimable** La complexité de la story doit pouvoir être estimée par l'équipe
- **Small / Courte** Une story doit être assez petite pour en faire tenir plusieurs sur une itération
- **Testable** La story est explicite et permet de préparer des cas de tests

HOW TO SPLIT A USER STORY

1 PREPARE THE INPUT STORY



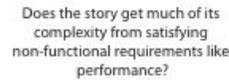
* INVEST - Stories should be:
Independent
Negotiable
Valuable
Estimable
Small
Testable

WORKFLOW STEPS



DEFER PERFORMANCE

Could you split the story to just make it work first and then enhance it to satisfy the non-functional requirement?



OPERATIONS

Can you split the operations into separate stories?

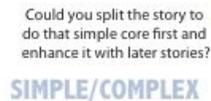
Does the story include multiple operations? (e.g. is it about "managing" or "configuring" something?)

BUSINESS RULE VARIATIONS

Can you split the story so you do a subset of the rules first and enhance with additional rules later?

Does the story have a variety of business rules? (e.g. is there a domain term in the story like "flexible dates" that suggests several variations?)

2 APPLY THE SPLITTING PATTERNS



SIMPLE/COMPLEX

When you apply the obvious split, is whichever story you do first the most difficult?

Could you group the later stories and defer the decision about which story comes first?

MAJOR EFFORT

Does the story get the same kind of data via multiple interfaces?

Can you split the story to handle data from one interface first and enhance with the others later?

INTERFACE VARIATIONS

Does the story have a complex interface?

Is there a simple version you could do first?

VARIATIONS IN DATA

Can you split the story to process one kind of data first and enhance with the other kinds later?

BREAK OUT A SPIKE

Are you still baffled about how to split the story?

Can you find a small piece you understand well enough to start?

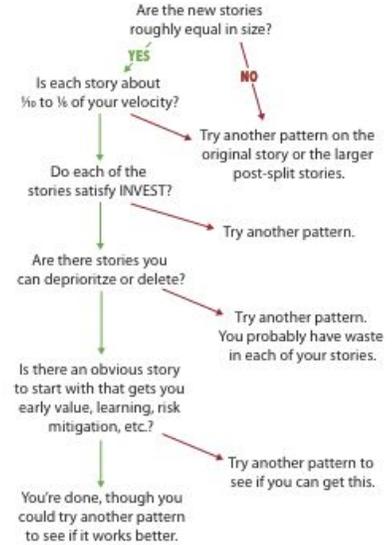
Write that story first, build it, and start again at the top of this process.

Can you define the 1-3 questions most holding you back?

Write a spike with those questions, do the minimum to answer them, and start again at the top of this process.

Take a break and try again.

3 EVALUATE THE SPLIT



Template US

La story A4

Mais aussi technique
ou anomalie

#ID	Titre	Type : Fonctionnel
Description <i>En tant que...</i> <i>Je veux...</i> <i>Afin de...</i>		
Règles métier • Règle 1 • Règle 2 • Règle 3		
Tests d'acceptance <i>Jeu de données</i> Nom du scénario 1 <i>Etant donné que ...</i> <i>Lorsque ...</i> <i>Alors ...</i> Nom du scénario 2 <i>Etant donné que ...</i> <i>Lorsque ...</i> <i>Alors ...</i> Nom du scénario 3 <i>Etant donné que ...</i> <i>Lorsque ...</i> <i>Alors ...</i>		

Le mot de passe doit être composé
de 8 caractères avec
au moins 1 chiffre et 1 majuscule

Etant donné que je suis sur la page de connexion
Lorsque je saisis le mot de passe **Ygst45mp**
Alors la page d'accueil apparaît

BDD

**LANGUAGE NATUREL
PAR L'EXEMPLE**

#03	Dashboard	Type : Fonctionnel
Description As a User I want to see my dashboard So that I can select the mission I want		
Rules To display : <ul style="list-style-type: none">• Header ("Tableau de bord")• Mission cards		
B.D.D. Scenario : Looking at the Dashboard Given I'm logged in the app When I'm on the Dashboard Then I see the Header And the Mission cards		

Extrait du Scrum Guide



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Product Backlog

Le Product Backlog est une liste ordonnée et émergente de ce qui est nécessaire pour améliorer le produit. C'est l'unique source du travail entrepris par la Scrum Team.

Les éléments du Product Backlog qui sont susceptibles d'être réalisés dans un seul Sprint par la Scrum Team sont considérés comme prêts à être traités en Sprint Planning. Ils acquièrent généralement ce degré de transparence après avoir été affinés. L'affinement du Product Backlog consiste à décomposer et à définir davantage les éléments du Backlog en éléments plus fins et plus précis. Il s'agit d'une activité continue visant à ajouter des détails, tels qu'une description, un ordre et une taille. Les attributs varient souvent en fonction du domaine d'activité.

Les Developers qui effectueront le travail sont responsables du dimensionnement. Le Product Owner peut influencer les Developers en clarifiant ses explications et les aider à trouver des compromis.

Engagement : Objectif de Produit

L'Objectif de Produit décrit un état futur du produit qui peut servir de cible à la Scrum Team pour planifier. L'Objectif de Produit est dans le Product Backlog. Le reste du Product Backlog émerge pour définir « ce qui » permettra d'atteindre l'Objectif de Produit.

Un produit est un vecteur de valeur. Il a une limite claire, des parties prenantes connues, des utilisateurs ou des clients bien définis. Un produit peut être un service, un produit physique ou quelque chose plus abstrait.

L'Objectif de Produit est l'objectif à long terme de la Scrum Team. Ils doivent atteindre (ou abandonner) un objectif avant de s'attaquer au suivant.

Backlog Refinement

Ce n'est pas un évènement, mais une activité



L'objectif du Backlog Refinement est de s'assurer que le Backlog soit prêt pour le prochain Sprint Planning

Le Product Owner doit faciliter cette activité en collectant en avance de phase les besoins des clients

Les developers discutent avec le Product Owner sur la clarté du besoin, les critères d'acceptation, les dépendances, etc..., tout en augmentant leur connaissance du produit

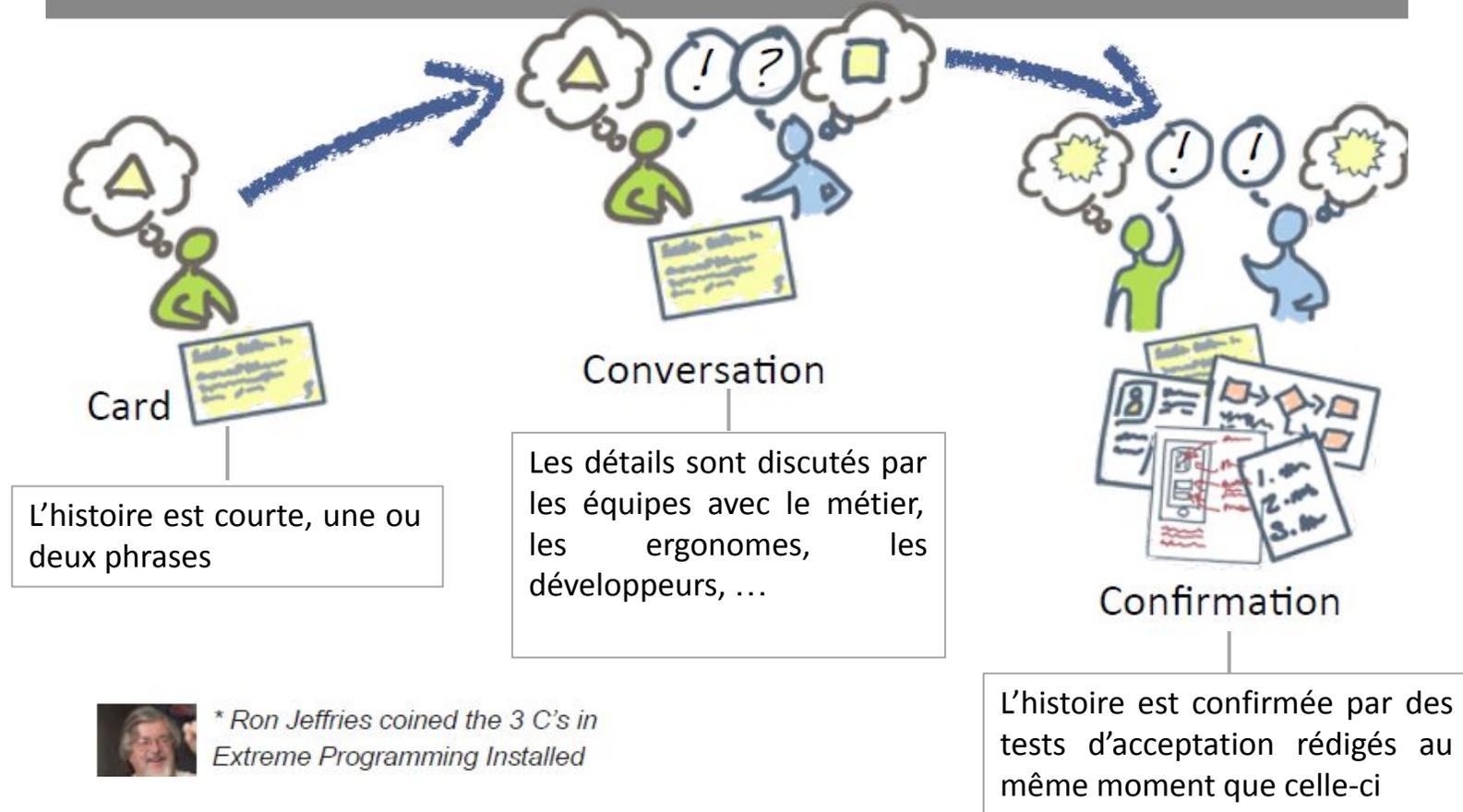
C'est un vrai échange de clarification du besoin.



Scrum team
A lieu régulièrement
Permet d'avoir un sprint planning plus court

USER STORY : 3Cs

Stories have a simple lifecycle



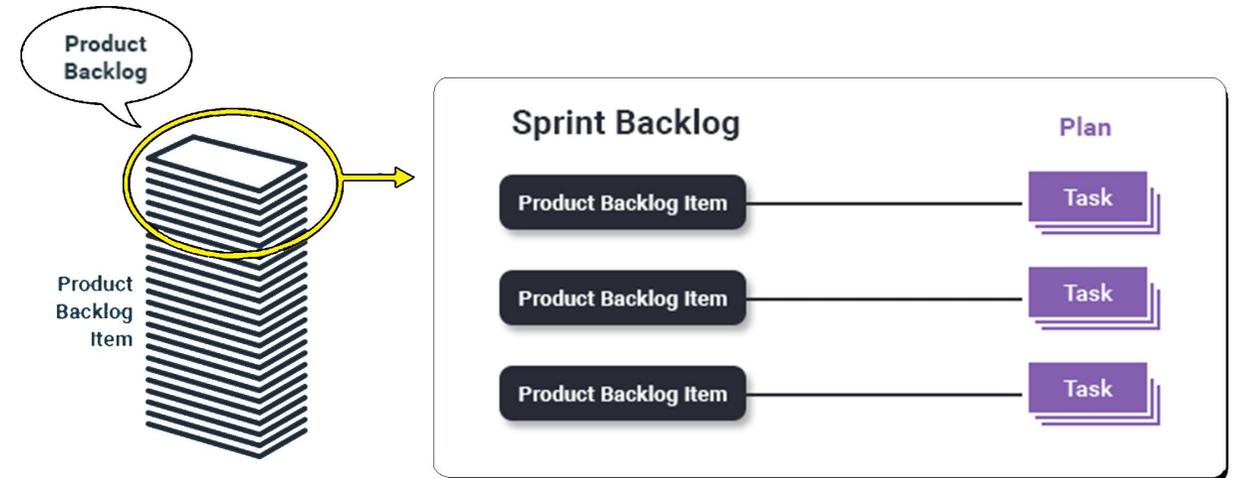
Sprint Backlog

Le Backlog du Sprint est : **l'ensemble des items sélectionnés pour le Sprint + l'objectif de sprint + le plan pour y arriver.**

Le Backlog du Sprint est une prévision que les développeurs ont fait sur les fonctionnalités qui seront présentes dans le prochain incrément et le travail nécessaire pour livrer cette fonctionnalité dans un incrément «Fini».

Visibilité en temps réel du travail accompli par les développeurs

Suivi progression: Burn-Down



L'Objectif de Sprint est l'unique but du Sprint. Bien que l'Objectif de Sprint soit un engagement fait par les Développeurs, il offre une certaine flexibilité en termes de travail nécessaire pour atteindre cet objectif.

Extrait du Scrum Guide

Sprint Backlog

Le Sprint Backlog est composé de l'Objectif de Sprint (le « pourquoi »), de l'ensemble des éléments du Product Backlog choisis pour le Sprint (le « quoi »), ainsi que d'un plan d'action pour la réalisation de l'Increment (le « comment »).

11

Le Sprint Backlog est un plan élaboré par et pour les Developers. Il s'agit d'une image très visible et en temps réel du travail que les Developers prévoient d'accomplir durant le Sprint afin d'atteindre l'Objectif de Sprint. Par conséquent, le Sprint Backlog est mis à jour tout au long du Sprint selon ce qu'on en apprend. Il devrait être suffisamment détaillé pour que les Developers puissent inspecter leur progression durant le Daily Scrum.

Engagement : Objectif de Sprint

L'Objectif de Sprint est l'unique but du Sprint. Bien que l'Objectif de Sprint soit un engagement fait par les Developers, il offre une certaine flexibilité en termes de travail nécessaire pour atteindre cet objectif. L'Objectif de Sprint crée également de la cohérence et du focus, tout en encourageant la Scrum Team à travailler ensemble plutôt que sur des initiatives séparées.

L'Objectif de Sprint est créé pendant l'événement de Sprint Planning, puis ajouté au Sprint Backlog. Lorsque les Developers travaillent pendant le Sprint, ils gardent l'Objectif de Sprint à l'esprit. Si le travail s'avère être différent de ce à quoi ils s'attendaient, ils collaborent avec le Product Owner pour négocier le périmètre du Sprint Backlog dans le cadre de ce Sprint, sans que cela n'affecte l'Objectif de Sprint.

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020



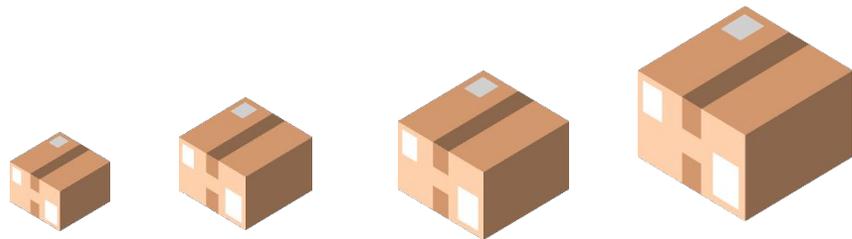
Increment

Un Increment est une première étape concrète vers l'Objectif de Produit

C'est la 'Livraison' de fin de Sprint, répondant à la « Definition of Done »

Un travail qui ne remplirait pas les conditions de la Definition of Done ne peut pas être considéré comme un Increment.

Doit toujours être dans un état publiable



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Increment

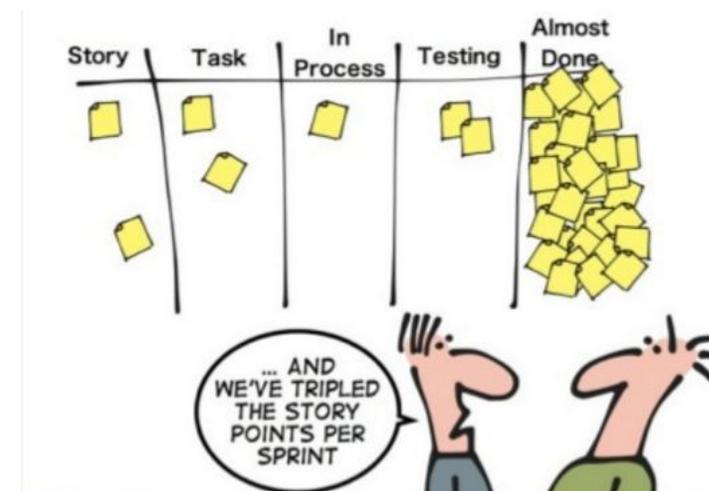
Un Increment est une première étape concrète vers l'Objectif de Produit. Chaque Increment s'ajoute à tous les Increments précédents et fait l'objet d'une vérification approfondie, ce qui garantit que tous les Increments fonctionnent ensemble. Afin de fournir une valeur, l'Increment doit être utilisable.

Plusieurs Increments peuvent être créés durant un Sprint. La somme des Increments est présentée lors de la Sprint Review, ce qui permet de démontrer l'utilité de l'empirisme. Toutefois, un Increment peut être livré aux parties prenantes avant la fin du Sprint. La Sprint Review ne doit jamais être considérée comme le seul moment pour délivrer de la valeur.

Un travail qui ne remplirait pas les conditions de la Definition of Done ne peut pas être considéré comme un Increment.

Une définition de Done permet une définition commune des critères de « done », tout le monde comprend ce que « Fini » signifie

- Liste de critères qui satisfait les **mesures de qualité requises pour le produit**
- Objectif : **Transparence**, Créer une définition commune des critères de « Done »
 - « Done » pour un item de Backlog
 - « Done » pour un Sprint
 - « Done » pour une Release
- Important pour avoir :
 - Une qualité élevée
 - La connaissance du travail à faire pour estimer
 - Une transparence envers le Product Owner qui sait ce qui a été fait quand un incrément est livré
- Si la Definition of Done pour un Increment fait partie des **standards de l'organisation**, toutes les Scrum
- Teams doivent la suivre au minimum. Si cela ne fait pas partie des standards de l'organisation, la Scrum Team doit créer sa propre Definition of Done qui soit appropriée pour le produit.
- Si **plusieurs équipes** travaillent sur un **même produit** alors elles doivent partager la **même Definition of Done**
- L'équipe fait évoluer cette définition, plus elle devient mature, plus elle y ajoute des critères rigoureux de qualité



Extrait du Scrum Guide

Engagement : Definition of Done (Définition de Fini)

La Definition of Done (Définition de Fini) est une description formelle de l'état de l'Increment lorsqu'il satisfait les mesures de qualité requises pour le produit.

Dès qu'un élément du Product Backlog satisfait à la Definition of Done, il se transforme en Increment.

La Definition of Done apporte de la transparence en permettant à chacun une compréhension commune du travail Fini dans le cadre de l'Increment. Si un élément du Product Backlog n'est pas conforme à la Definition of Done, il ne peut pas être publié ni même présenté lors de la Sprint Review. Il est alors renvoyé au Product Backlog pour être pris en compte ultérieurement.

12

Ken Schwaber & Jeff Sutherland

The Scrum Guide

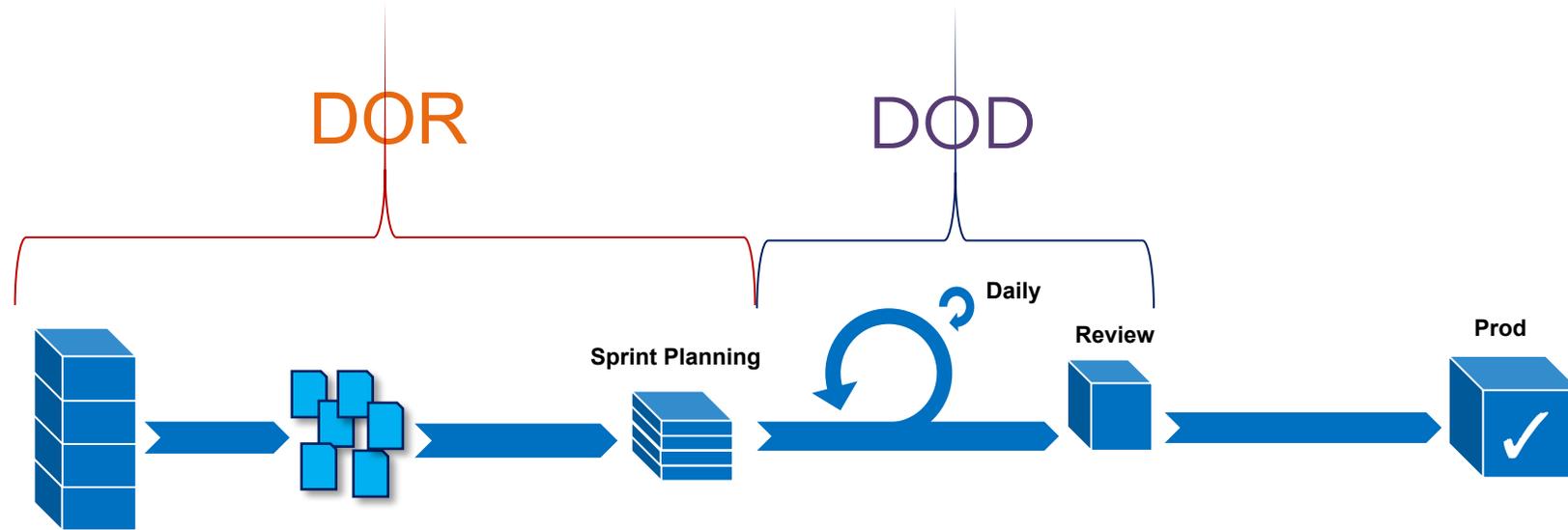
The Definitive Guide to Scrum: The Rules of the Game

November 2020

Si la Definition of Done pour un Increment fait partie des standards de l'organisation, toutes les Scrum Teams doivent la suivre au minimum. Si cela ne fait pas partie des standards de l'organisation, la Scrum Team doit créer sa propre Definition of Done qui soit appropriée pour le produit.

Les Developers sont tenus de se conformer à la Definition of Done. Si plusieurs Scrum Teams travaillent ensemble sur un même produit, elles doivent la définir ensemble et s'y conformer.

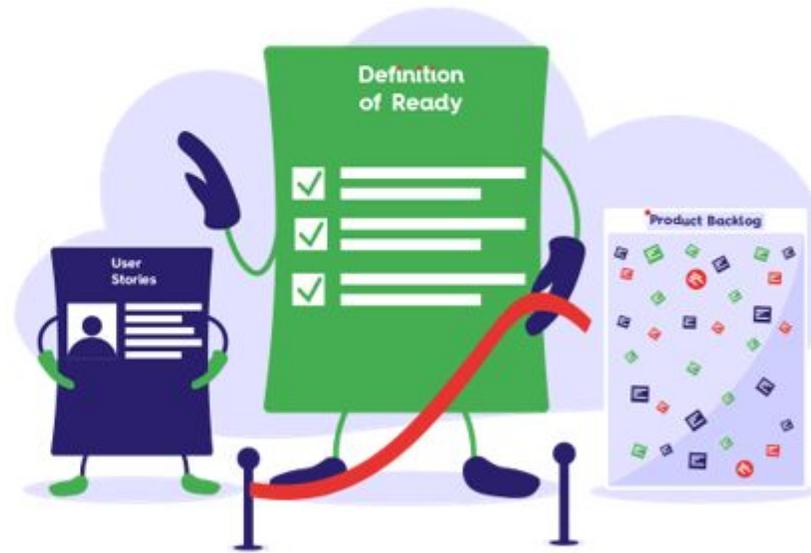
Definition of Ready VS Definition of Done



DoR : Definition Of Ready

Définition

- *La Definition Of Ready (DOR) est la liste (définie par l'équipe) d'éléments attendus qu'une user story doit rassembler pour être candidate au développement.*



Exemple DoD

- Code source écrit
- Tests TU + TI sur le build du jour J-1 validés
- Code source revu par un pair
- Respect des critères de qualimétrie
- Manuel Utilisateur mis à jour
- Doc architecture mis à jour
- Mise à jour des tableaux de bord effectués
- ...

Team "Done" List

...With a Story

- All Code (Test and Mainline) Checked in
- All Unit Tests Passing
- All Acceptance Tests Identified, Written & Passing
- Help File Auto Generated
- Functional Tests Passing

...With a Sprint

- All Story Criteria, Plus...
- Product Backup Updated
- Performance Testing
- Package, Class & Architecture Diagrams Updated
- All Bugs Closed or Postponed
- Code Coverage for all Unit Tests at 80% +

...Release to INT

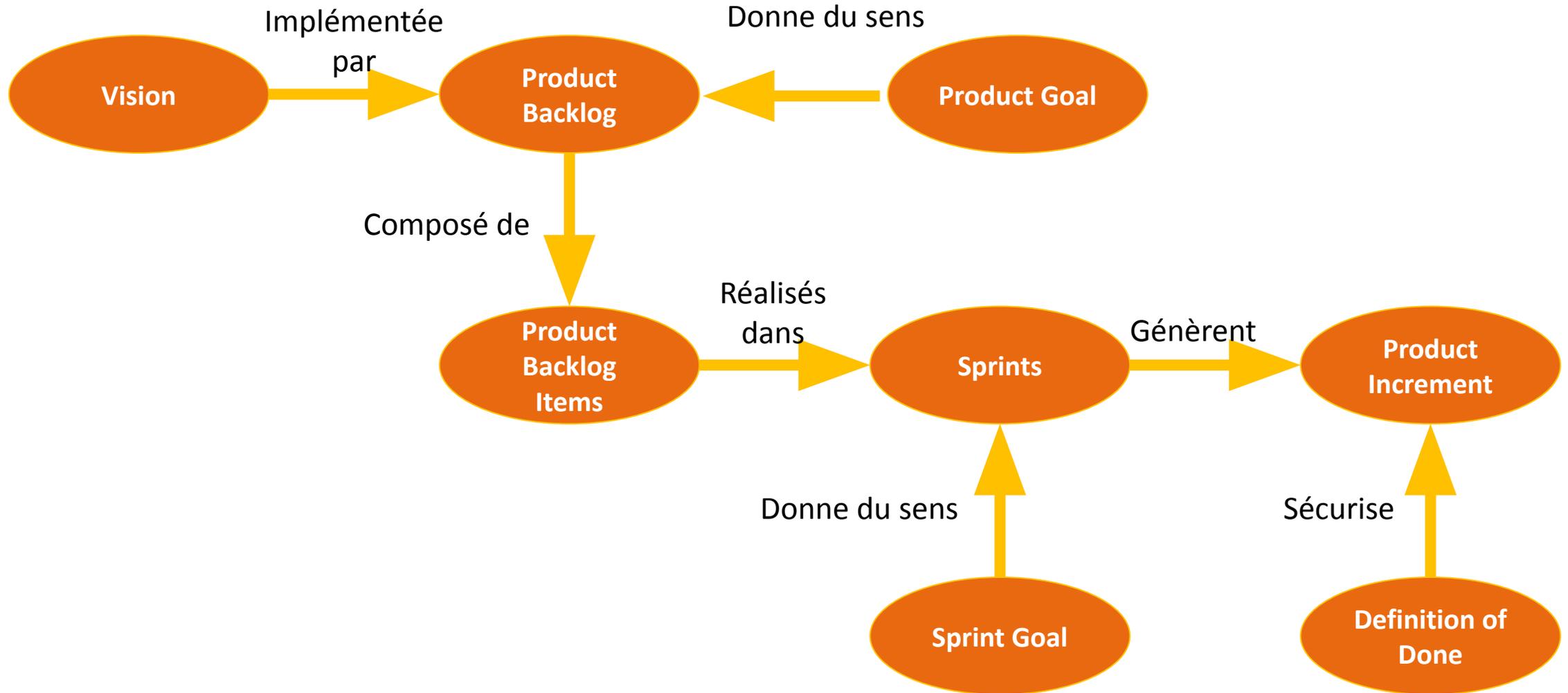
- All Sprint Criteria, Plus...
- Installation Packages Created
- MOM Packages Created
- Operations Guide Updated
- Troubleshooting Guides Updated
- Disaster Recovery Plan Updated
- All Test Suites Passing

...Release to Prod

- All INT Criteria, Plus...
- Stress Testing
- Performance Tuning
- Network Diagram Updated
- Security Pass Validated
- Threat Modeling Pass Validated
- Disaster Recovery Plan Tested



De la vision produit aux Product Increments





Les responsabilités



Composition de la scrum team

*Version 2020: plus de rôle,
mais des responsabilités, et
plus de sous équipe*



Product Owner (PO)

Maximise la
valeur apportée
par le produit

Scrum Team

10 personnes ou moins

Auto ~~Organisée~~ gérée

Pluri disciplinaire

Responsable de la création d'un
Increment qui ait de la valeur et qui
soit utile, à chaque Sprint.



Scrum Master (SM)

Garant de la méthode
Facilitateur,
Responsable de
l'efficacité de l'équipe



Developers



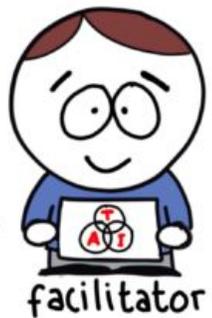
Délivre un produit potentiellement
exploitable à chaque itération



Scrum team

- ❑ **Pluridisciplinaire** avec toutes les compétences nécessaires dans l'équipe pour produire un logiciel qui fonctionne
- ❑ **Auto-gérée**, personne (ni même le scrum master) n'indique à l'équipe de réalisation comment on doit transformer les items du backlog produits en incréments
- ❑ Structurée et habilitée par l'organisation à s'organiser et gérer son propre travail
- ❑ Sa taille ne doit pas varier d'un sprint à l'autre, l'équipe doit être stable
- ❑ **Fournit un incrément fini potentiellement publiable à la fin de chaque sprint**

Scrum Team



SCRUM MASTER!

Extrait du Scrum Guide



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Scrum Team

L'unité fondamentale de Scrum est une petite équipe, la Scrum Team. La Scrum Team se compose d'un Scrum Master, d'un Product Owner et de Developers. Il n'y a pas d'équipe dans l'équipe ni de hiérarchies. Il s'agit d'une seule et même unité stable, composée de professionnels focalisés sur un seul objectif à la fois, l'Objectif de Produit.

Les Scrum Teams sont pluridisciplinaires, leurs membres ont toutes les compétences nécessaires pour créer de la valeur à chaque Sprint. Elles sont également autogérées, elles décident en interne qui fait quoi, quand et comment.

La Scrum Team doit être suffisamment petite pour rester réactive et assez grande pour accomplir un travail significatif durant le Sprint, habituellement dix personnes au plus. En général, nous avons constaté que les petites équipes communiquent mieux et sont plus productives. Si les Scrum Teams deviennent trop grandes, elles devraient envisager de se réorganiser en plusieurs Scrum Teams cohérentes, chacune axée sur le même produit. Par conséquent, elles doivent partager le même Objectif de Produit, le même Product Backlog et le même Product Owner.

La Scrum Team est responsable de toutes les activités liées au produit : collaboration des parties prenantes, vérification, maintenance, exploitation, expérimentation, recherche et développement, ainsi que tout ce qui pourrait être nécessaire. Elles sont structurées et habilitées par l'organisation à gérer leur propre travail. Travailler sur des Sprints à un rythme soutenable améliore le focus et la cohérence de la Scrum Team.

Toute la Scrum Team est responsable de la création d'un Increment qui ait de la valeur et qui soit utile, à chaque Sprint. Scrum définit trois responsabilités spécifiques au sein de la Scrum Team : les Developers, le Product Owner et le Scrum Master.

Scrum Master

- ❑ **Responsable de la bonnes applications du cadre Scrum et des valeurs Agiles** en aidant tout le monde à comprendre la théorie, les pratiques, les règles et les valeurs de scrum
- ❑ Responsable **efficacité** de l'équipe
- ❑ Coach et forme les parties prenantes et les personnes en contact à adapter leurs **interactions** avec l'équipe Scrum pour maximiser la valeur créer par cette equipe
- ❑ Il ~~est au~~ **rend service au PO**, en l'aidant à trouver des techniques pour gérer efficacement le Product Backlog, pour l'aider à organiser et maximiser la valeur de son produit
- ❑ Il ~~est au~~ **rend service à la scrum team**, en supprimant les obstacles à la progression, en les coachants pour qu'ils soient auto organisé et pluridisciplinaire
- ❑ Il ~~est au~~ **rend service à l'organisation** en aidant les parties prenantes à comprendre et adopter Scrum ainsi que le développement empirique de produits et contribue à lever les obstacles



Scrum Master

Responsabilités	Activités
<ul style="list-style-type: none"> • Garant du cadre scrum • Garant de l'efficacité de l'équipe en se focalisant sur les interactions et les impediments • Accompagne le Product Owner dans la gestion du produit Agile • S'assure que l'équipe est fonctionnelle et productive 	<ul style="list-style-type: none"> • Facilite les discussions, les échanges • S'assure que l'équipe identifie les obstacle rencontrés et prennent des actions • Apprend à l'équipe à comprendre le but de chaque cérémonie et à être efficace, et les animent dans un premier temps • Supporte l'équipe dans la mise en place des mesures pour suivre les progrès • Protège l'équipe des perturbations extérieures en accompagnant les parties prenante à comprendre Scrum • S'assure que le Product Owner et les developpers travaillent en collaboration • Est garant des livrables effectués, au même titre que n'importe quel autre membre de l'équipe

- **Ses compétences**

- Bonne connaissance et expérience de Scrum
- Aptitude à comprendre les aspects techniques
- Facilité à communiquer – Talent de médiateur
- Capacité à guider
- Ténacité – Inclinaison à la transparence
- Facilitateur, générateur intelligence collective
- Empathique, à l'écoute

- **La bonne personne**

- Ce n'est pas un chef de projet ...
- Il ne commande pas
- Il n'impose pas
- Il ne contraint pas
- Il n'est pas la seule interface avec le client

Extrait du Scrum Guide



Scrum Master

Le Scrum Master est redevable de la mise en place de Scrum tel que défini dans le Guide Scrum. Pour ce faire, il se doit d'aider tout un chacun, l'équipe et l'organisation à comprendre la théorie et la pratique Scrum.

Le Scrum Master est redevable de l'efficacité de la Scrum Team. Il permet à la Scrum Team d'améliorer ses pratiques en suivant le cadre de travail Scrum.

Le Scrum Master est, avant tout, un véritable leader au service de la Scrum Team et de l'ensemble de l'organisation.



Le Scrum Master rend service à la Scrum Team de plusieurs façons :

- Accompagner les membres de l'équipe en matière d'autogestion et de pluridisciplinarité ;
- Aider la Scrum Team à se focaliser sur la création d'Increments de grande valeur qui répondent à la Definition of Done ;
- Faire en sorte qu'il n'y ait pas d'obstacles pouvant entraver la progression de la Scrum Team ; et
- S'assurer que tous les événements Scrum ont bien lieu et sont efficaces, productifs et respectent bien les temps impartis (timeboxés : gardés dans une boîte de temps).

Le Scrum Master rend service au Product Owner de plusieurs façons :

- L'aider à trouver des techniques pour définir efficacement l'Objectif de Produit et gérer efficacement du Product Backlog ;
- Sensibiliser la Scrum Team à la nécessité de bien comprendre le besoin d'avoir des éléments du Product Backlog clairs et concis ;
- Encourager l'application de la planification produit empirique dans un environnement complexe ; et
- Faciliter la collaboration des parties prenantes, selon les demandes ou les besoins.

Le Scrum Master rend service à l'organisation de plusieurs façons :

- Accompagner, former et encadrer l'organisation dans son adoption de Scrum ;
- Planifier et apporter conseils sur les implémentations de Scrum au sein de l'organisation ;
- Faciliter la compréhension de l'approche empirique en environnement complexe des employés et des parties prenantes ; et
- Contribuer à lever les obstacles qui peuvent se dresser entre les parties prenantes et les Scrum Teams.

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Product Owner

- Représente les besoins du client et/ou des utilisateurs
- Responsable de **maximiser la valeur du produit** résultant du travail de l'équipe scrum
- Responsable sur retour sur investissement
- Exprime le besoin clairement** dans le Product Backlog, la **vision**
- Donne les **priorités fonctionnelles**
- Seul responsable de la gestion du Backlog Produit
- Toute l'organisation doit respecter ses décisions



Product Owner



Responsabilités	Activités
<ul style="list-style-type: none">• Succès métier du projet• Définit la Roadmap/Contenu/Budget• Acteur clé du recueil des besoins• Représente les utilisateurs du produit• Responsable du ROI	<ul style="list-style-type: none">• Il définit les fonctionnalités du produit (et objectif du produit), et s'assure que le Product Backlog est transparent, visible, compris (équipe et parties prenantes)• Prépare le sprint avec l'équipe : priorise et ordonnance les fonctionnalités qui seront développées et répond aux questions de l'équipe• Ajuste les besoins et les (re) priorise• Implique les utilisateurs et les parties prenantes, notamment lors de la revue de sprint mais à tout moment, pour s'assurer que le produit répond bien au besoin• Gère les enjeux du produit et la stratégie

• Ses compétences

- Bonne connaissance du domaine métier
- Maîtrise des techniques de définition de produit
- Capacité à prendre des décisions rapidement
- Capacité à détailler au bon moment
- Esprit ouvert au changement
- Aptitude à la négociation, à l'écoute

• La bonne personne

- Motivée - impliquée
- Participe au cérémonial scrum
- Implication au jour le jour

Extrait du Scrum Guide



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Product Owner

Le Product Owner est redevable de maximiser la valeur du produit résultant du travail de la Scrum Team. La manière de procéder peut varier considérablement selon les organisations, les Scrum Teams et les individus.

Le Product Owner est également redevable de la gestion efficace du Product Backlog. Ce qui inclut :

- Formuler et communiquer explicitement l'Objectif de Produit ;

6

-
- Créer et communiquer clairement les éléments du Product Backlog ;
 - Ordonner les éléments dans le Product Backlog ; et
 - S'assurer que le Product Backlog est transparent, visible et compris.

Le Product Owner peut effectuer le travail ci-dessus ou peut déléguer ce travail à d'autres. Quoi qu'il en soit, le Product Owner en demeure redevable.

Pour que les Product Owners réussissent, toute l'organisation doit respecter leurs décisions. Ces décisions sont visibles dans le contenu et dans l'ordre du Product Backlog et, via un Increment inspectable lors de la Sprint Review.

Le Product Owner est une personne et non un comité. Le Product Owner peut représenter les besoins de nombreuses parties prenantes dans le Product Backlog. Ceux qui souhaitent modifier le Product Backlog peuvent le faire en essayant de convaincre le Product Owner.

Developers

- S'engagent à traiter** tout ou partie utile d'un increment à chaque sprint
- Créer un plan de Sprint, un Sprint Backlog
- Adapter leur plan chaque jour par rapport à l'Objectif de Sprint
- Se tenir mutuellement responsables en tant que professionnels.



Developers



Responsabilités	Activités
<ul style="list-style-type: none">• S'engagent sur l'atteinte de l'objectif du sprint• Garantissent la qualité produit•	<ul style="list-style-type: none">• Délivrent de la valeur métier en produisant un produit/service de haute qualité• Découpent les PBI en tâches• Font des activités d'Architecture, Conception, Analyste fonctionnelle, Stratégie de développement, Qualité du code, Recette etc... tout ce qui est nécessaire pour réaliser un produit de qualité• Prennent des décisions ensemble, notamment sur la manière de travailler et peuvent utiliser les pratiques d'ingénierie (XP, pair-programming, capitalisation, bonnes pratiques, ...) pour atteindre leur but• Font de la veille technologique

- **Ses compétences**

- Mature et autonome
- Compétences généralistes
- Expérimentée dans l'environnement technologique
- Sait s'auto-gérer

- **L'organisation**

- Ne comporte pas de rôles prédéfinis pour ses membres
- Il n'y a pas non plus de notion de hiérarchie interne : Toutes les décisions sont prises ensemble.
- Personne ne donne d'ordre à l'équipe sur sa façon de procéder
- L'équipe s'adresse directement au Product Owner.
- La composition de l'équipe doit rester stable et dédiée.

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Developers

Les Developers sont les membres de la Scrum Team qui s'engagent à traiter tout ou partie utile d'un Increment à chaque Sprint.

Les compétences spécifiques requises pour les Developers sont souvent larges et varient selon le domaine d'activité. Toutefois, les Developers sont toujours redevables de :

- Créer un plan de Sprint, un Sprint Backlog ;
- Inculquer la notion de qualité en adhérant à une Definition of Done ;
- Adapter leur plan chaque jour par rapport à l'Objectif de Sprint ; et
- Se tenir mutuellement responsables en tant que professionnels.



Les événements



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Événements Scrum

Le Sprint est un conteneur pour tous les autres événements. Chaque événement dans Scrum est une occasion formelle pour inspecter et adapter les artefacts Scrum. Ces événements sont spécifiquement conçus pour permettre la transparence requise. L'incapacité d'organiser les événements conformément à leur prescription est une occasion perdue pour inspecter et s'adapter. Les événements sont utilisés dans Scrum dans le but de créer de la régularité, minimisant le besoin d'avoir d'autres réunions non définies par Scrum.

Idéalement, tous les événements se tiennent à la même heure et au même lieu pour réduire la complexité.

Extrait du Scrum Guide



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Sprint

Les Sprints sont au cœur de Scrum, où les idées sont transformées en valeur.

Ce sont des événements d'une durée fixe, d'un mois ou moins, pour créer une cohérence. Un nouveau Sprint commence immédiatement après la fin du précédent.

Tout le travail nécessaire pour atteindre l'Objectif de Produit, y compris le Sprint Planning, les Daily Scrums, la Sprint Review et la Sprint Retrospective, se fait dans le cadre des Sprints.

Durant le Sprint :

- Aucun changement n'est permis, qui pourrait remettre en cause l'Objectif de Sprint ;
- Les objectifs de qualité ne sont jamais revus à la baisse ;
- Le Product Backlog est affiné si nécessaire ; et
- Le périmètre peut être clarifié et renégocié avec le Product Owner selon ce qu'on en apprend.

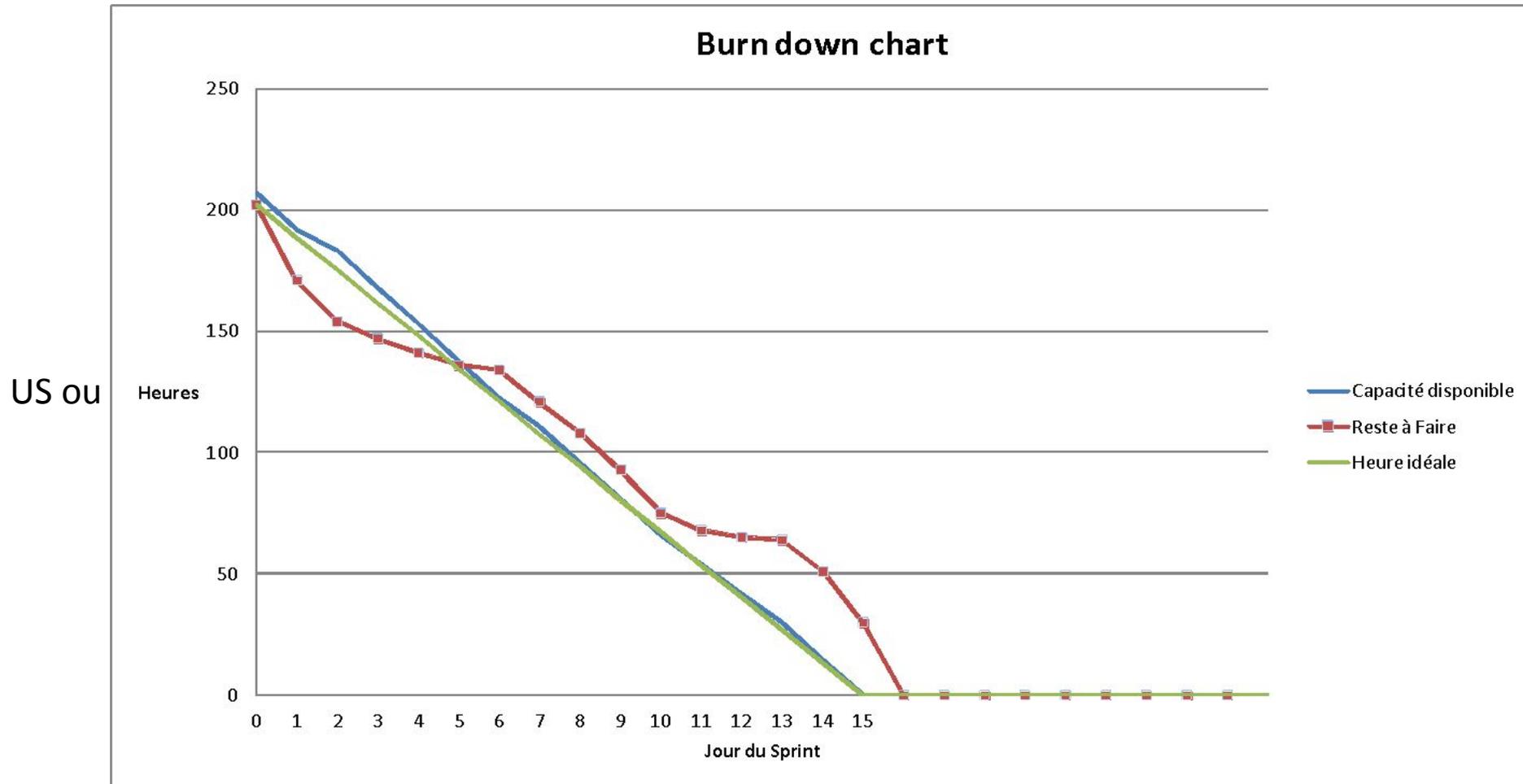
Les Sprints permettent la prédictibilité en assurant l'inspection et l'adaptation de la progression vers l'Objectif de Produit, une fois par mois calendaire au moins. Lorsque l'horizon d'un Sprint est trop lointain, l'Objectif de Sprint risque de ne plus être le bon, la complexité augmente et, avec elle, le risque. Les Sprints plus courts raccourcissent le cycle de l'apprentissage, limitant ainsi les risques liés aux coûts et à l'effort. Chaque Sprint peut être considéré comme un projet court.

Diverses pratiques existent pour évaluer la progression, telles que les courbes du « burn-down », ou celles du « burn-up » ou les diagrammes de flux cumulatifs. Bien que leur utilité soit prouvée, ces courbes ne remplacent pas l'importance de l'empirisme. Dans des environnements complexes, une grande part est laissée à l'inconnu. Seul ce qui s'est déjà passé peut être utilisé pour une prise de décision à venir.

Un Sprint peut être annulé si l'Objectif de Sprint devient obsolète. Seul le Product Owner a le pouvoir d'annuler le Sprint.

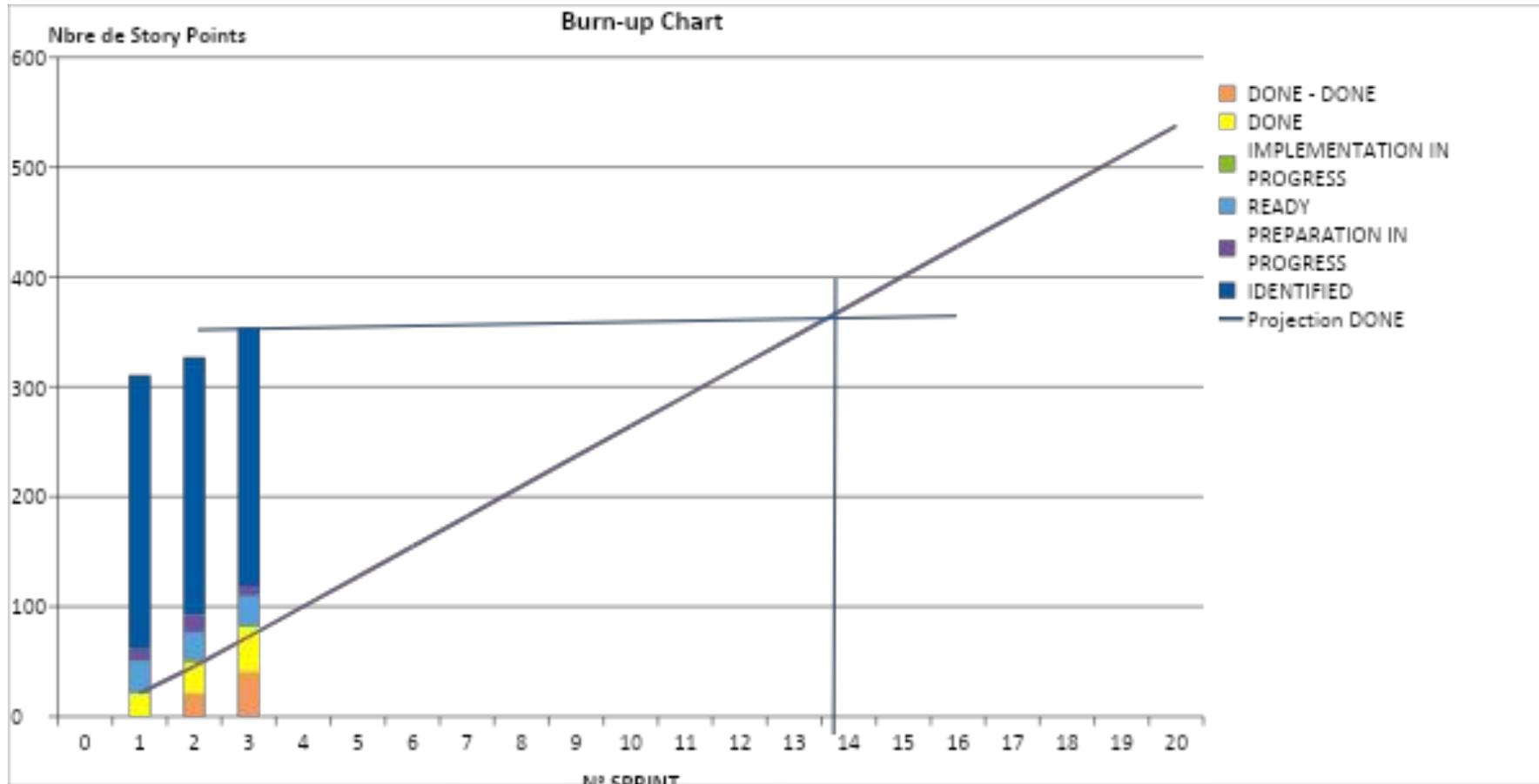
Le burn-down chart

Objectif: doit permettre à tout moment de visualiser l'avancement de l'itération.



Le burn-up chart

Objectif: doit permettre à tout moment de visualiser l'avancement de réalisation de l'objectif du produit



Nombre de points

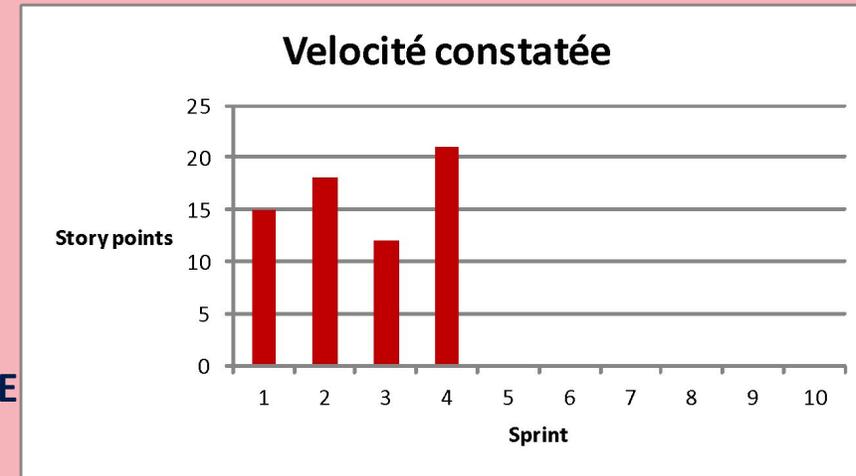
LA VÉLOCITÉ DE L'ÉQUIPE:

NOMBRE DE POINTS D'EFFORT

RÉALISÉS PAR ITÉRATION.

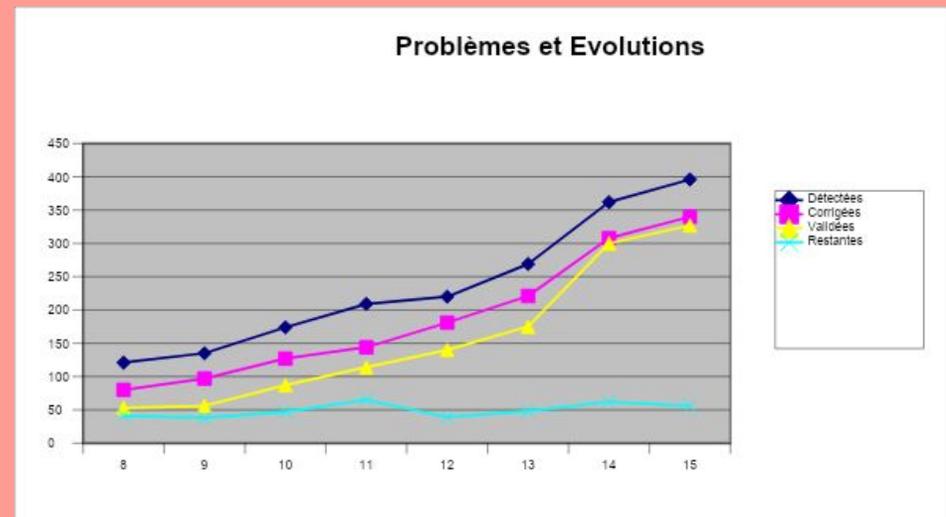
DÉDUCTION D'UNE VÉLOCITÉ MOYENNE

PERMETTANT UNE PROJECTION DU TRAVAIL DE L'ÉQUIPE SPRINT APRÈS SPRINT.



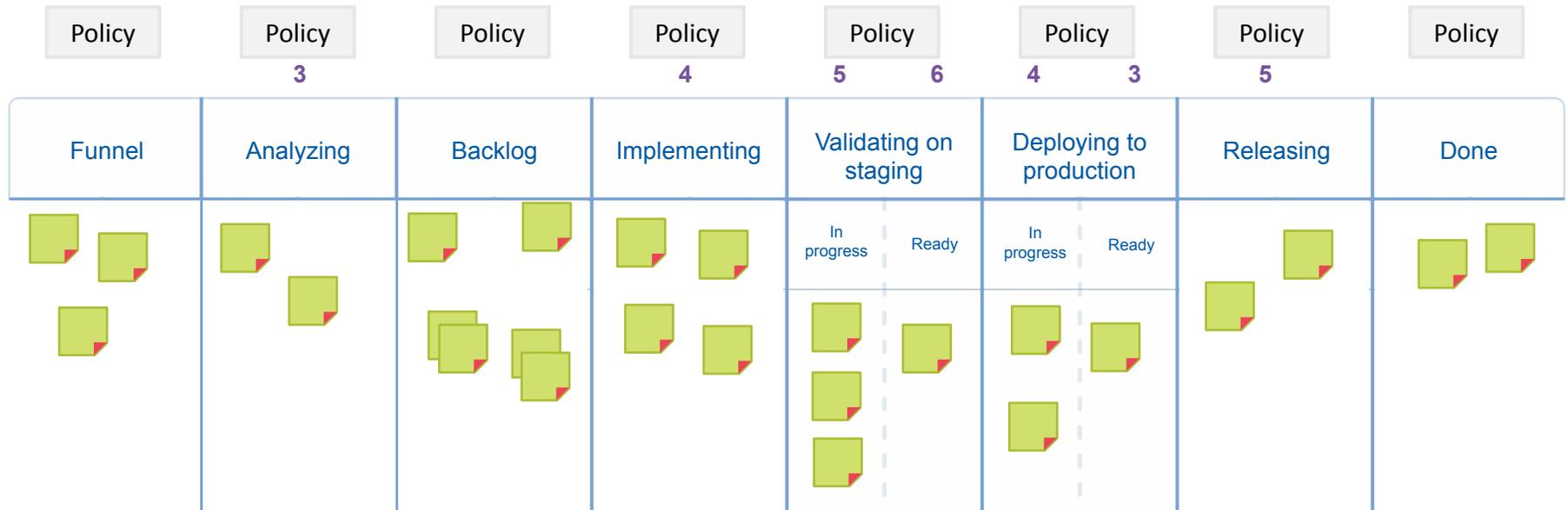
L'état de la dette:

La dette est le reste à faire d'une user story validée par le Product-owner et livrée. (Bug ou évolution ou documentation)

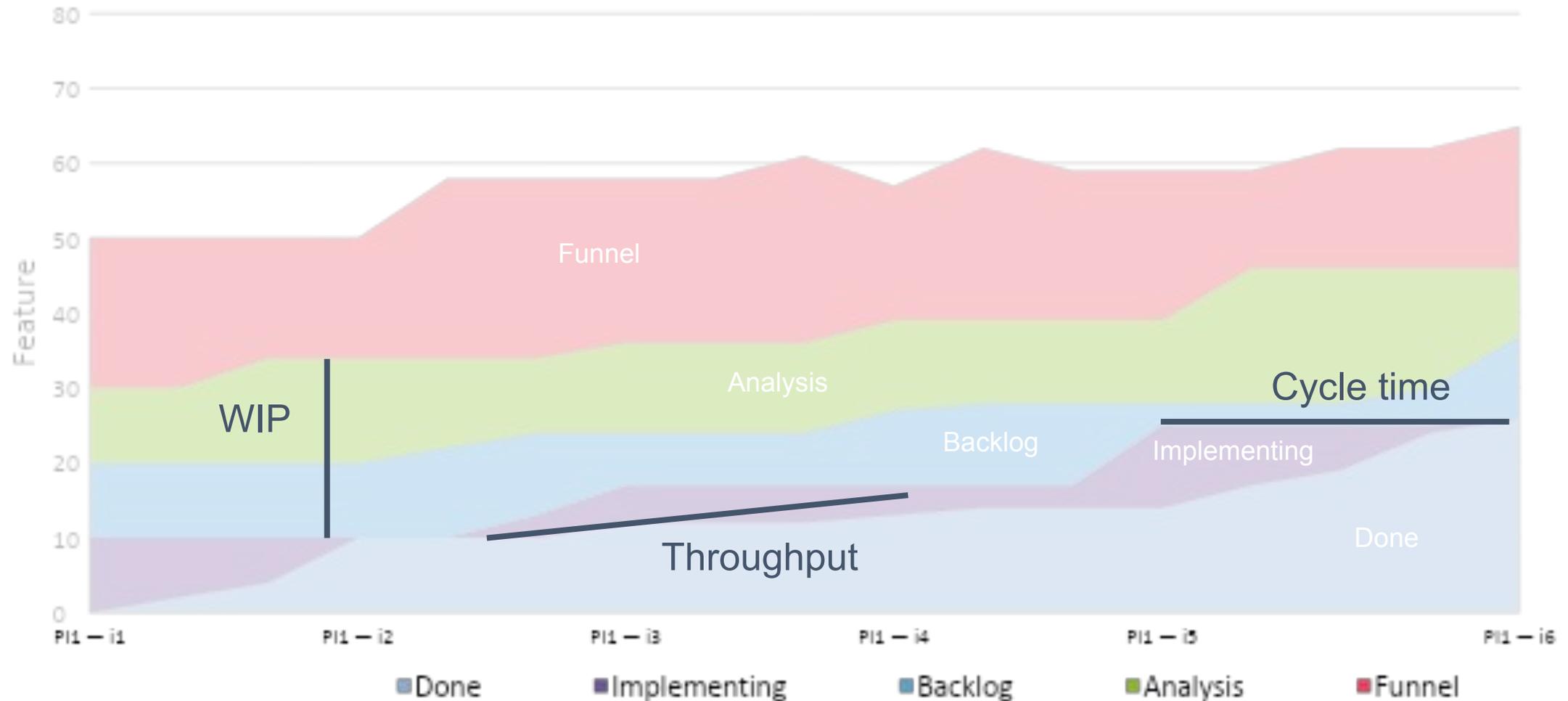


Visualisation du flow de travail

- Tout le travail est visualisé et la progression de chaque élément est suivie en permanence
- Fonctionne en Flux tiré
- Les colonnes définissent les états par lesquels passe chaque élément du flow
- Policy : décrit de façon explicite les choses qui doivent être faites pour que la carte passe à la colonne suivante
- Les limites de travail en cours (WIP) pour les États sélectionnés sont là pour améliorer le flux (ajuste la demande à la capacité)
- Le flux à travers le tableau Kanban est mesuré, ce qui informe sur les améliorations futures...

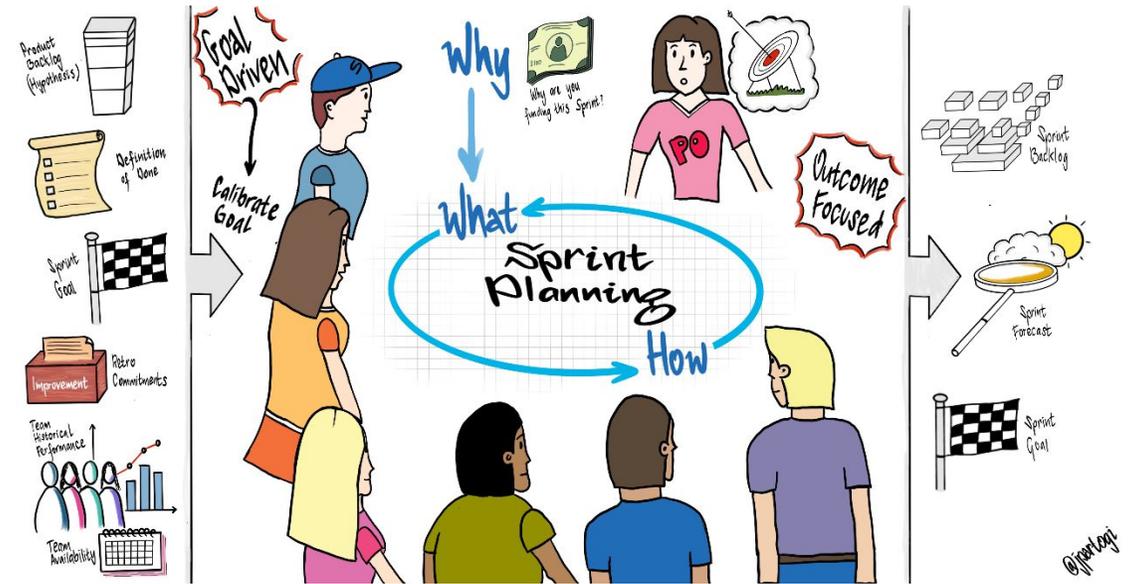


Cumulative Flow Diagram



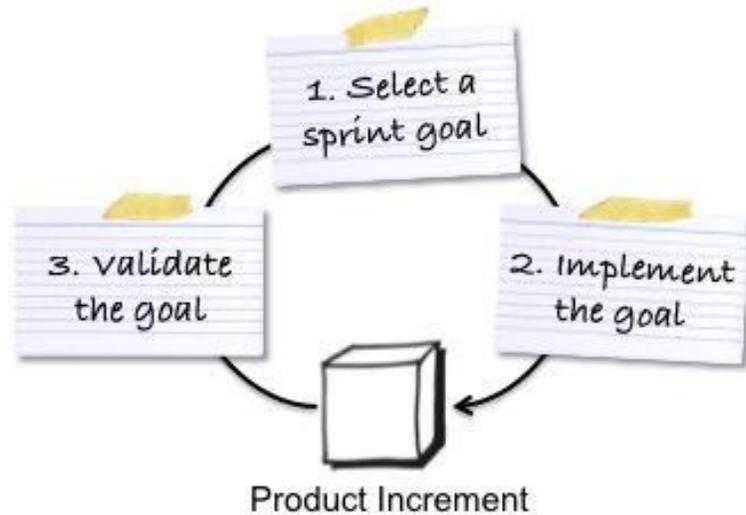
Sprint Planning

- ❑ Le sprint planning répond aux questions suivantes :
 - ❑ Pourquoi ce Sprint est-il important ?
 - ❑ Que peut-on réaliser durant ce sprint ?
 - ❑ Comment termine-t-on le travail choisi ?
- ❑ La sortie de cette cérémonie est :
 - ❑ Le **Sprint Backlog** (=objectif de sprint + éléments du product backlog + plan pour les livrer).



Tous les membres de l'équipe SCRUM participent collaborativement
Timebox: durée MAX 8h pour 1 sprint d'un mois

Sprint Planning



Pourquoi ce Sprint est-il important ?

Le PO expose l'intention, en lien avec l'objectif du produit
Le Product Owner propose comment améliorer le produit, de l'intention du sprint
La Scrum Team définit le Sprint Goal

Que peut-on faire dans ce Sprint?

En discutant avec le Product Owner, les Developers choisissent les éléments du Product Backlog à inclure dans le Sprint en tenant compte de leurs capacités

Comment le travail choisi sera-t-il terminé?

Pour chaque élément choisi du Product Backlog, les Developers planifient le travail nécessaire pour créer un Increment qui répond à la Definition of Done
Les Developers initient le plan du Sprint

Le scrum master veille à ce que l'évènement ait lieu et que les participants en comprennent le but

Durant le sprint



Si le travail s'avère être différent de ce à quoi les développeurs s'attendaient, ils collaborent avec le Product Owner pour négocier le périmètre du Sprint Backlog dans le cadre de ce Sprint, sans que cela n'affecte l'Objectif de Sprint.

Extrait du Scrum Guide



Sprint Planning

Le Sprint Planning lance le Sprint en présentant le travail à effectuer durant le Sprint. Le plan qui en résulte est créé par le travail collaboratif de toute la Scrum Team.

Le Product Owner veille à ce que les participants soient prêts à discuter les éléments les plus importants du Product Backlog et de comment ces éléments représentent l'Objectif de Produit. La Scrum Team peut également inviter d'autres personnes à participer au Sprint Planning pour donner des conseils.

Le Sprint Planning aborde les thèmes suivants :



Thème 1 : Pourquoi ce Sprint est-il important ?

Le Product Owner explique comment augmenter la valeur du produit et son utilité pour le Sprint en cours. L'ensemble de la Scrum Team collabore ensuite à définir un Objectif de Sprint qui énonce clairement aux parties prenantes l'utilité du Sprint. L'Objectif de Sprint doit être finalisé avant la fin du Sprint Planning.

Thème 2 : Que peut-on faire durant ce Sprint ?

En discutant avec le Product Owner, les Developers sélectionnent les éléments du Product Backlog à inclure dans le Sprint en cours. Au fur et à mesure de la discussion, la Scrum Team affine ces éléments, améliorant ainsi leur compréhension et leur confiance dans leur capacité à les développer.

Devoir sélectionner ce qui peut ou ne peut pas être accompli durant un Sprint est une tâche difficile. Plus les Developers connaissent leurs performances passées, leur capacité à venir et leur Definition of Done, mieux ils sont à même de faire de prévisions pour le Sprint en cours.

Thème 3 : Comment le travail choisi sera-t-il réalisé ?

Pour chaque élément sélectionné du Product Backlog, les Developers planifient le travail nécessaire pour créer un Increment qui réponde à la Definition of Done. Cela se fait souvent en décomposant les éléments du Product Backlog en éléments de travail d'une journée ou moins. La façon de procéder est laissée à la seule discrétion des Developers. Personne d'autre ne leur dit comment transformer les éléments du Product Backlog en Increments de valeur.

L'Objectif de Sprint, les éléments du Product Backlog sélectionnés pour le Sprint, ainsi que le plan pour les livrer, correspondent à un ensemble appelé le Sprint Backlog.

Le Sprint Planning est limité dans le temps à un maximum de huit heures pour un Sprint d'un mois. Pour les Sprints plus courts, l'événement est généralement plus court.

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

TIPS



Réunion collaborative, échanges, alignement

Le planning s'arrête quand la capacité de l'équipe est atteinte ou que le temps est dépassé

On termine la session en exprimant son niveau de confiance dans l'atteinte des objectifs



Ce n'est pas l'endroit pour détailler tout jusqu'à la fin de l'itération

Story point c'est quoi?

Systeme

Les story points sont des unités de mesure qui permettent **d'estimer l'effort global** nécessaire pour implémenter intégralement une tâche du backlog produit ou autre.

Les équipes assignent des story points en fonction :

- **de la complexité**
- **du volume du travail**
- **du risque ou de l'incertitude**

L'estimation horaire peut rassurer mais elle s'affranchit de trop de paramètres liés à l'équipe, le contexte et engageant sur des paramètres individuels.

ESTIMATION

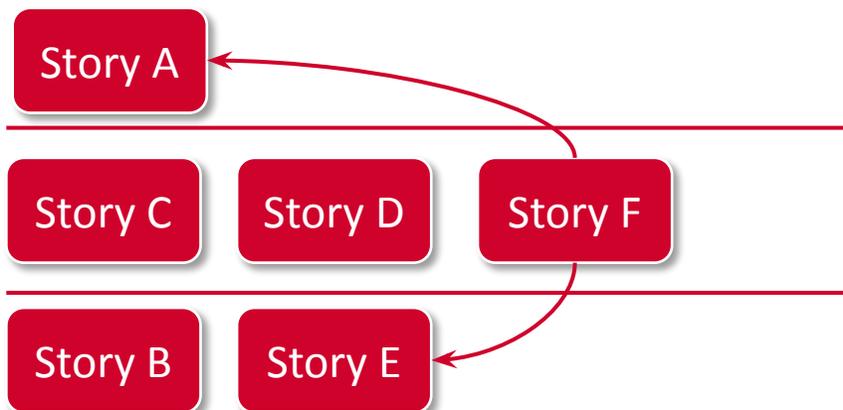
- Les Story Points s'évaluent en relatif



L'abstraction est utile puisqu'elle pousse l'équipe à prendre des décisions plus radicales en ce qui concerne la difficulté du travail.

- Estimer par analogie

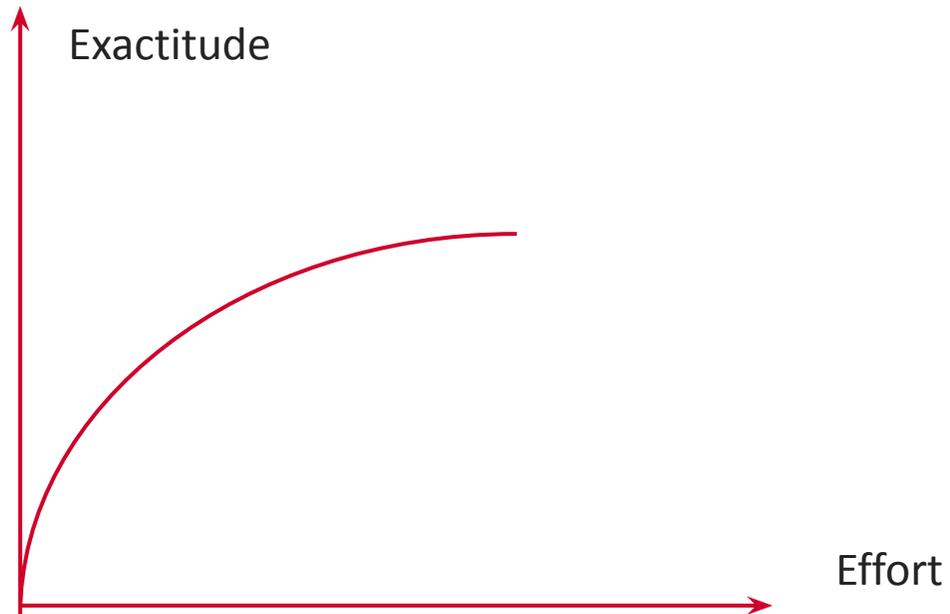
- Confirmer les estimations en comparant la Story pas rapport à plusieurs autres
- Grouper les Stories par tailles comparables



L'estimation relative permet de facilement trouver l'estimation juste en se référant aux estimations antérieures (comparaison sur les efforts nécessaires).

ESTIMATION : quel effort y consacrer?

- Un petit effort aide beaucoup
- Beaucoup d'effort aide un petit peu plus



Il vaut mieux être
approximativement juste que
précisément faux

John Maynard Keynes

Story point c'est quoi?

Système



Complexité

- Difficulté de réalisation
- Difficulté de compréhension (clarté des objectifs)
- Expérience

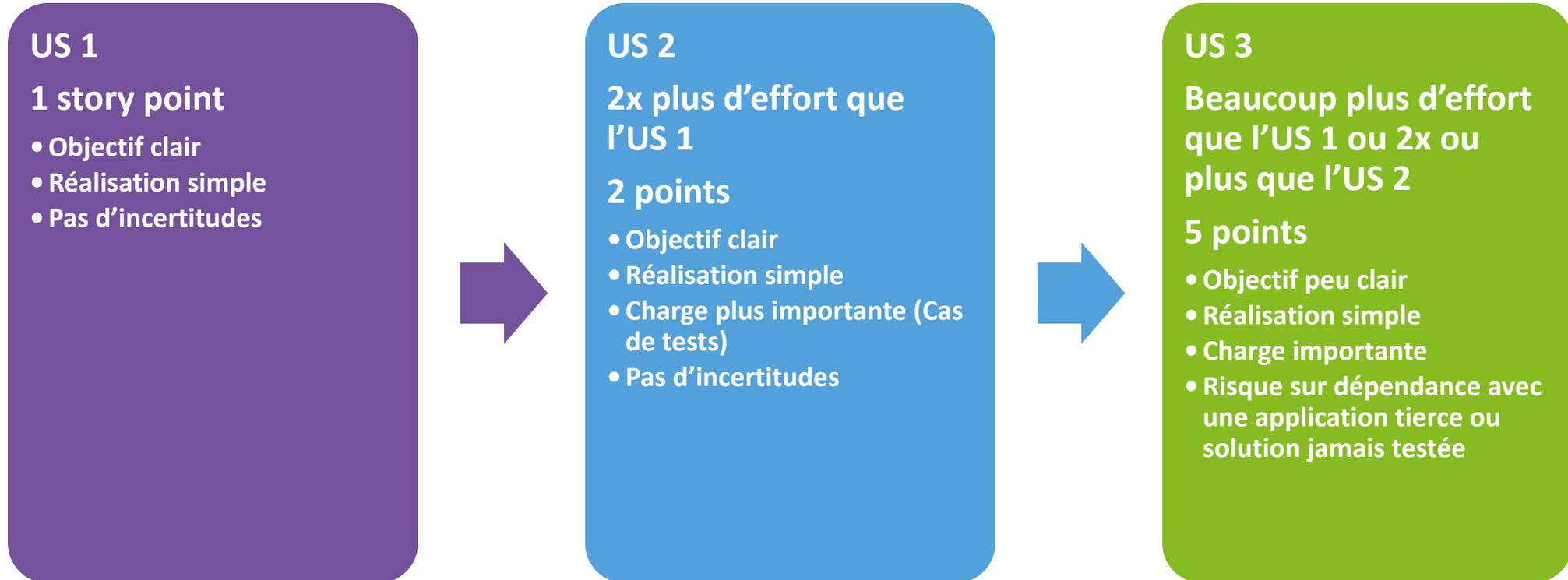
Volume de travail

- Quantité de tâches à accomplir
- Comparaison vis à vis de tâches similaires

Risque / Incertitude

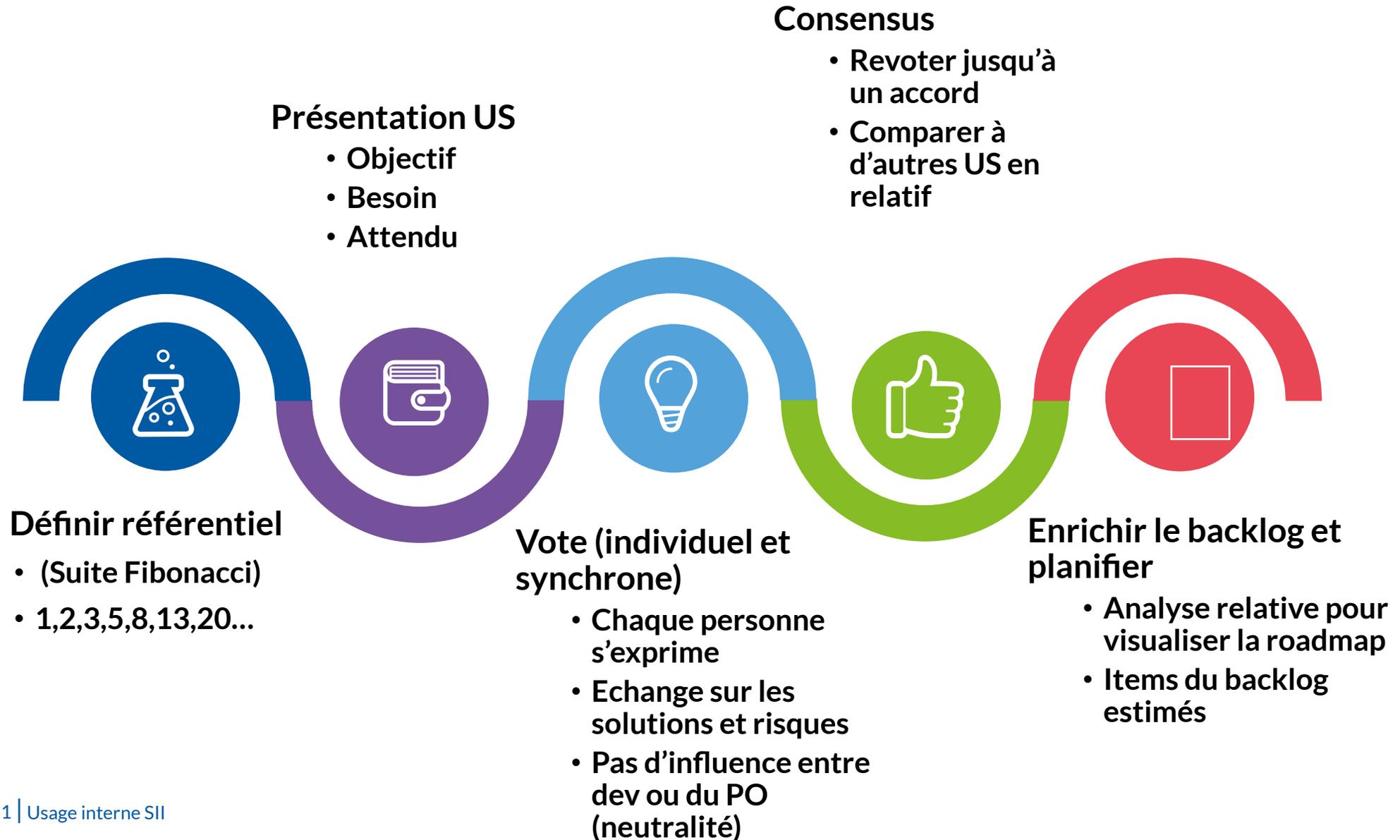
- Inconnues
- Dépendances (internes ou externes)
- Incertitude de résultat
- Manques (fonctionnels ou techniques)

Exemple pratique



Poker planning

Exemple



Daily Scrum

Le but de **Daily Scrum** est :

- D'inspecter la progression vers l'objectif Sprint



Il va permettre aux développeurs de se focaliser sur la progression vers l'Objectif de Sprint et de :

- se synchroniser
- améliorer la communication
- identifier les obstacles
- organiser le plan de bataille de la journée

NB: il n'est pas nécessaire de répondre aux 3 questions (version 2020)

- *Qu'est-ce que j'ai fait hier qui a aidé l'équipe de développement à atteindre l'objectif du Sprint?*
- *Que ferai-je aujourd'hui pour aider l'équipe de développement à atteindre l'objectif du Sprint?*
- *Est-ce que je vois tout obstacle qui m'empêche ou empêche l'équipe de développement de respecter l'objectif du Sprint?*



Developers (SM et PO « optionnels »)

Timebox: 15 minutes

Même lieu, même heure, tous les jours
(la constance réduit la complexité)

Les Developers sont responsables de son déroulement

C'est un point clé pour l'INSPECTION et
l'ADAPTATION



**Meet
after**

Extrait du Scrum Guide



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Daily Scrum

L'objectif du Daily Scrum est d'inspecter la progression vers l'Objectif de Sprint et d'adapter le Sprint Backlog si nécessaire, en ajustant les futurs travaux planifiés.

9

Le Daily Scrum est un événement de 15 minutes, pour les Developers de la Scrum Team. Pour réduire la complexité, il est tenu à la même heure et au même lieu, chaque jour ouvré du Sprint. Si le Product Owner et / ou le Scrum Master travaillent activement sur des éléments du Sprint Backlog, ils participent en tant que Developers.

Les Developers peuvent choisir la structure et les techniques qu'ils souhaitent, à condition que leur Daily Scrum se focalise sur la progression vers l'Objectif de Sprint et produise un plan d'action pour la prochaine journée de travail. Cela leur permet de se focaliser et d'améliorer l'autogestion.

Les Daily Scrums améliorent la communication, aident à identifier les obstacles, favorisent la prise de décision rapide et, par conséquent, éliminent la nécessité de faire d'autres réunions.

Le Daily Scrum n'est pas le seul moment où les Developers sont autorisés à ajuster leur plan. Ils se réunissent souvent tout au long de la journée pour des discussions plus détaillées sur l'adaptation ou la re-planification du reste du travail du Sprint.

TIPS



C'est 15 min max

C'est une synchronisation, échanges pour s'assurer que l'équipe va atteindre les objectifs de sprints

C'est le moment de demander du support si on rencontre un problème

L'équipe trouve le format qui lui convient le mieux

Le moment de voir si il faut adapter le plan pour atteindre les objectifs du sprint

Traiter ce qui doit l'être en dehors du Daily Scrum.

Ce n'est pas du reporting au SM ou PO.

C'est n'est pas guidé par le SM, il peut le faciliter depuis le « fond de la salle »

Ce n'est pas un endroit pour dire en détails tout ce qu'on a fait et tout ce que l'on va faire

Ne pas rentrer dans le détail : identifier les problèmes sans les résoudre.

Sprint Review

Le but du **sprint review** est :

- D'inspecter l'incrément : l'objectif est-il atteint?
 - D'apprendre sur le produit : quels ont été les problèmes rencontrés? Retour des utilisateurs suite à la démonstration des fonctionnalités terminées
- Adaptation/mise à jour du product backlog par le PO suivant les retours
 - C'est une session de travail

Un bonne revue de sprint doit pouvoir permettre de conduire à des changements dans le Backlog afin de pouvoir prendre en compte des adaptations du besoin client ou de nouvelles idées ou de changements de l'environnement.



Equipe scrum et stakeholders
Timebox: 4h max pour un sprint de 4 semaines

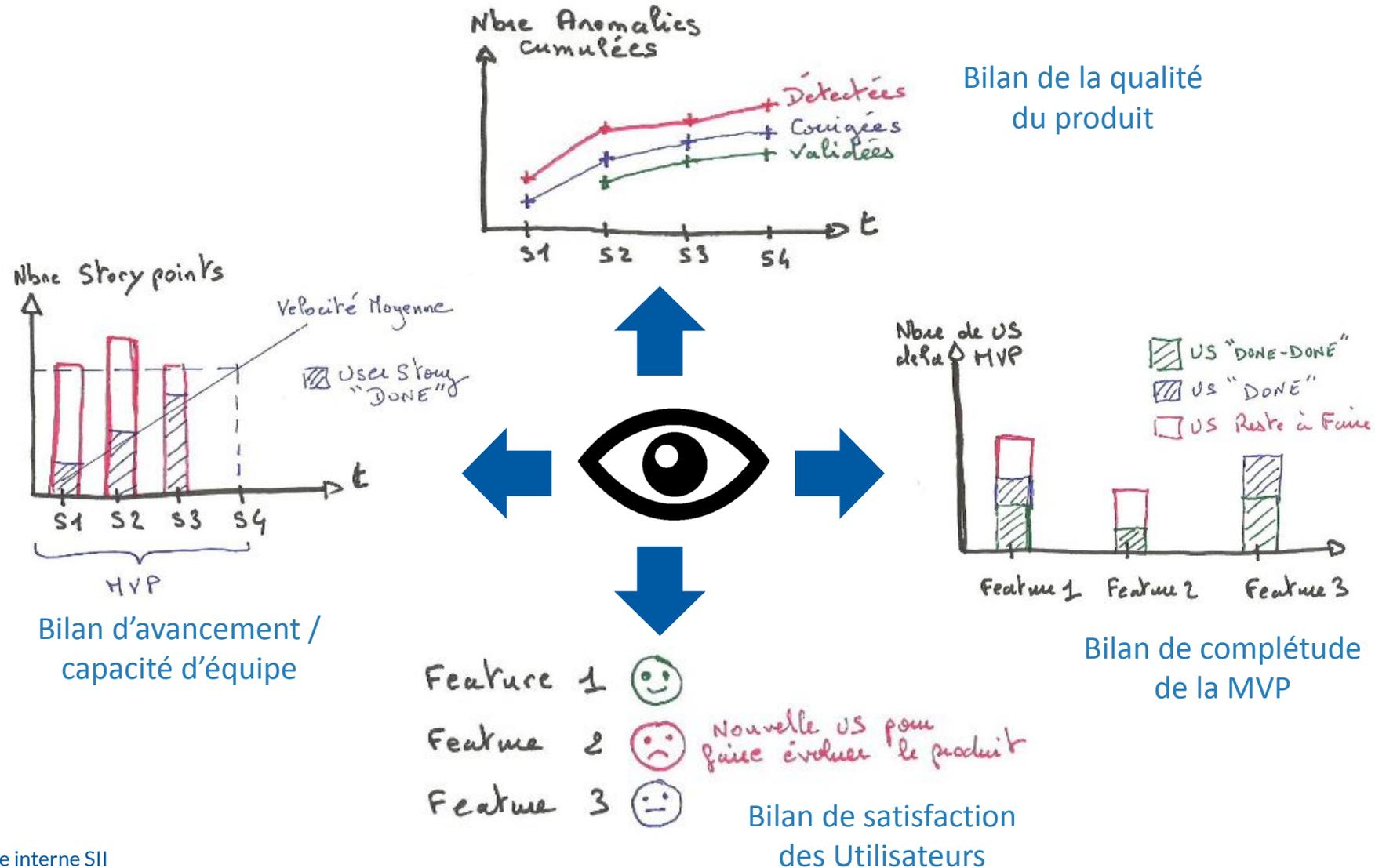
C'est un point clé pour l'INSPECTION et l'ADAPTATION sur l'incrément

Sprint Review : que fait-on?

L'équipe Scrum et les parties prenantes échangent sur ce qui a été fait pendant le sprint

- La Scrum Team présente le résultat de son travail effectué durant le Sprint (bilan du sprint + démonstration) et ce qui a changé dans son environnement.
- Les parties prenantes posent leurs questions et émettent des retours sur le produit (feedback) (Attention ça ne doit pas être un tribunal; c'est un moment d'échange)
- Les développeurs discutent de ce qui s'est bien déroulé durant le Sprint, quels problèmes ont été rencontrés, et comment ces problèmes ont été résolus
- L'ensemble du groupe convient de ce qu'il faut faire pour la suite, de sorte que la revue du Sprint fournisse une contribution précieuse aux prochaines réunions de Planification du Sprint (inspection et adaptation)
- Le Product Backlog peut également être ajusté pour répondre à de nouvelles opportunités.

Le Bilan de Sprint du point de vue du Product Owner



Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Sprint Review

L'objectif de la Sprint Review est d'inspecter le résultat du Sprint et de déterminer les adaptations futures. La Scrum Team présente les résultats de son travail aux principales parties prenantes et les progressions vers l'Objectif de Produit sont discutées.

Pendant l'événement, la Scrum Team et les parties prenantes passent en revue ce qui a été accompli durant le Sprint et ce qui a changé dans leur environnement. Sur la base de ces informations, les participants collaborent sur la marche à suivre et sur les décisions à prendre. Le Product Backlog peut également être ajusté pour répondre à de nouvelles opportunités. La Sprint Review est une session de travail et la Scrum Team doit éviter de la limiter à une session de présentation.

La Sprint Review est l'avant-dernier événement du Sprint et se limite dans le temps à un maximum de quatre heures pour un Sprint d'un mois. Pour les Sprints plus courts, l'événement est généralement plus court.

TIPS



C'est le moment de discuter de ce qui a été « done » et ainsi prendre des retours pour une mise à jour du product backlog

Une démonstration des éléments finis orientés end users et BO

Une revue des outcomes du sprint

Un moment pour récupérer des feedback sur le produit

Une vue globale de l'avancé et des problèmes principaux rencontrés

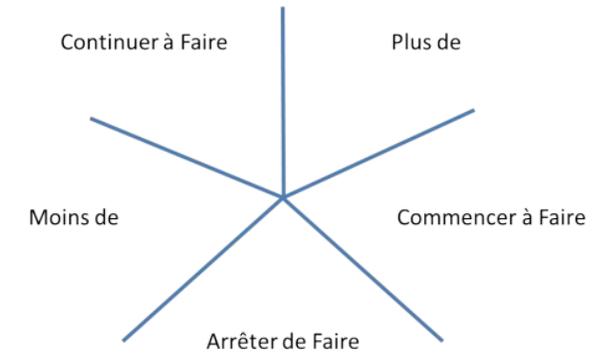
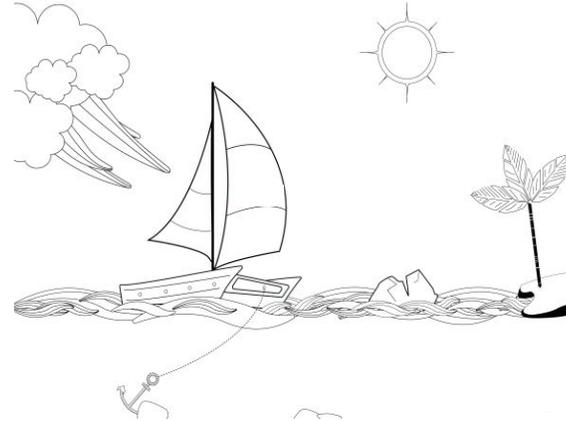


Ce n'est pas un report détaillé de ce qui a été fait

Sprint Retrospective

L'objectif de la **rétrospective** du sprint est

- ❑ D'inspecter la manière dont le sprint s'est déroulé en ce qui concerne la **qualité** et **l'efficacité**.
- ❑ Et d'identifier les améliorations à mettre en œuvre par la suite



Scrum team

Timebox: 3h pour un sprint de 4 semaines
A lieu apres la sprint review, termine le sprint

C'est un point clé pour l'INSPECTION et l'ADAPTATION

Le Scrum Master facilite souvent cet évènement

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Sprint Retrospective

L'objectif de la Sprint Retrospective consiste à réfléchir à des pistes pour améliorer la qualité et l'efficacité.

La Scrum Team inspecte le déroulement du dernier Sprint en ce qui concerne les individus, les interactions, les processus, les outils et leur Definition of Done. Les éléments inspectés varient souvent selon le domaine d'activité. Les hypothèses qui les ont fait dévier sont identifiées et leurs origines explorées. La Scrum Team discute de ce qui s'est bien passé durant le Sprint, des problèmes rencontrés et de la manière dont ces problèmes ont été (ou n'ont pas été) résolus.

La Scrum Team identifie les changements les plus utiles pour améliorer son efficacité. Les améliorations ayant le plus d'impact sont abordées dès que possible. Elles peuvent même être ajoutées au Sprint Backlog pour le prochain Sprint.

La Sprint Retrospective conclut le Sprint. Elle est limitée dans le temps à un maximum de trois heures pour un Sprint d'un mois. Pour les Sprints plus courts, l'événement est généralement plus court.

TIPS



C'est le moment de discuter des personnes, des relations/interactions, des processus et des outils

Une reunion collaborative, Faire preuve de transparence et de bienveillance

Une revue qualitative et quantitative

Un moment pour s'inspecter et definir des axes d'amélioration par des actions concrètes

Mettre à jour la Definition of Done si nécessaire.



Ce n'est pas un endroit pour se plaindre

On n'est pas à la recherche de coupables

Ne pas essayer de régler tous les problèmes à la fois

Ne pas inviter le Management / les Stakeholders.

Le Product Owner

- Zoom sur ses responsabilités et activités
- Prise de recul entre la théorie et la mise en pratique

What's the difference between Product Management and Product Ownership? Can both of them be run without Scrum, or is Product Ownership specifically just for Scrum?



Jeff Sutherland

Inventor and Co-Creator of Scrum · Author has **110** answers and **746.3K** answer views · Updated 1y

When I created the Product Owner role, I gave it more responsibility for product strategy and revenue generation than a Product Manager. I specifically pulled the best Product Manager Easel Corporation had out of Product Marketing and retrained him. It also has more responsibility for directly working with engineering 50% of the time to assure that the product fits customer needs. So it is a broader role in that sense but usually does not include Product Marketing (sales collateral, shows) or long term Competitive Analysis although it needs to support those efforts.

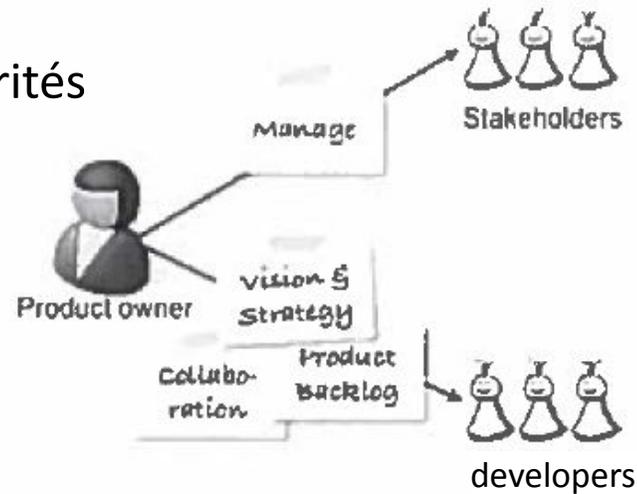
The goal was to eliminate the common Product Manager failing of throwing requirements over the wall only to have the customer receive something that they didn't want. It was also to replicate the Chief Engineer role at Toyota which requires much more responsibility than a Product Manager. So much, that I split the role by giving the Scrum Master responsibility for the team.

The basis for these comments is the Pragmatic Marketing framework: [Pragmatic Framework](#) . There have been some great Product Managers I have seen that are models for the Product Owner but they weren't following the framework.

Si on fait une synthèse des activités d'un PO...

La semaine type d'un Product Owner

- Gestion du backlog
 - Ajouter des items
 - Détailler des stories
 - Mettre à jour les priorités
 - Priorisation



- Discuter avec les utilisateurs pour « Débriefing » du dernier Sprint Review, questionner sur le produit
 - Ce qui a plu ou pas
 - Les idées nouvelles
 - Les changements de priorités
- Interagir avec les développeurs
 - Rappeler les objectifs et priorités
 - Répondre aux questions
 - Evaluer un incrément

Vis ma vie de Product Owner

Collaboration avec les utilisateurs

- Ateliers
- Echanges d'informations
- Aller sur le terrain

Gestion du Product Backlog

- Rédaction des Product Backlog Items
- Partage avec les Stakeholders
- Suppression des Items obsolètes

Priorisation du Product Backlog

- Mise en place d'un système de valeur
- Evaluation des Product Backlog Items
- Echanges avec l'équipe

Participation aux rituels Agile

- Sprint Planning
- Daily Scrum (observateur)
- Sprint Review
- Sprint Retrospective

Vis ma vie de Product Owner

Echanges avec les développeurs

- Réponses aux questions fonctionnelles
- Comprendre les problématiques techniques

Optimisation du travail de l'équipe

- Identification de quick wins
- Négociation avec les Stakeholders et l'équipe
- Arbitrages et renoncements

Acceptation des travaux réalisés

- Recette des Product Backlog Items livrés
- Ouverture de tickets d'incident
- Vérification des corrections
- Complétion du Product Backlog

Le Product Owner et les cérémonies

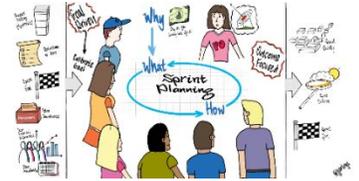
- Pendant le Sprint Planning
 - Il explicite les stories, s'assure que l'équipe les comprend bien
 - Il écoute les estimations et pose des questions
 - Il participe à la définition du Sprint Goal
- Pendant le Sprint
 - Il est disponible pour répondre aux questions
 - Il garde un œil sur ce qui est « fini » (**Done**)
 - Il prépare le Product Backlog pour les prochains Sprints
- Au Daily Scrum
 - Il est disponible pour un meet after si il y a besoin de clarification
- Pendant la Sprint Review
 - Il accepte ou non les stories « officiellement »
 - Il écoute les retours utilisateurs et envisage avec eux la suite
- A la Sprint Retrospective
 - Il est un membre de la Scrum Team comme un autre

La gestion du backlog/produit et les parties prenantes



La préparation des US

Expliciter et raffiner les US (critères d'acceptation, cas de tests) pour les prochains sprints.



La vision produit et la priorisation

Revoir la vision produit,
Revoir les objectifs métier,
Nettoyer le backlog,
Planifier les US.



La préparation du sprint

Prendre en compte la DoR (Definition Of Ready).
Prioriser le backlog en lien avec l'objectif produit

La vérification de la valeur produit



Satisfaction utilisateurs,
Qualité logicielle,
Couverture fonctionnelle.

La Revue du Sprint, évaluation d'un logiciel opérationnel

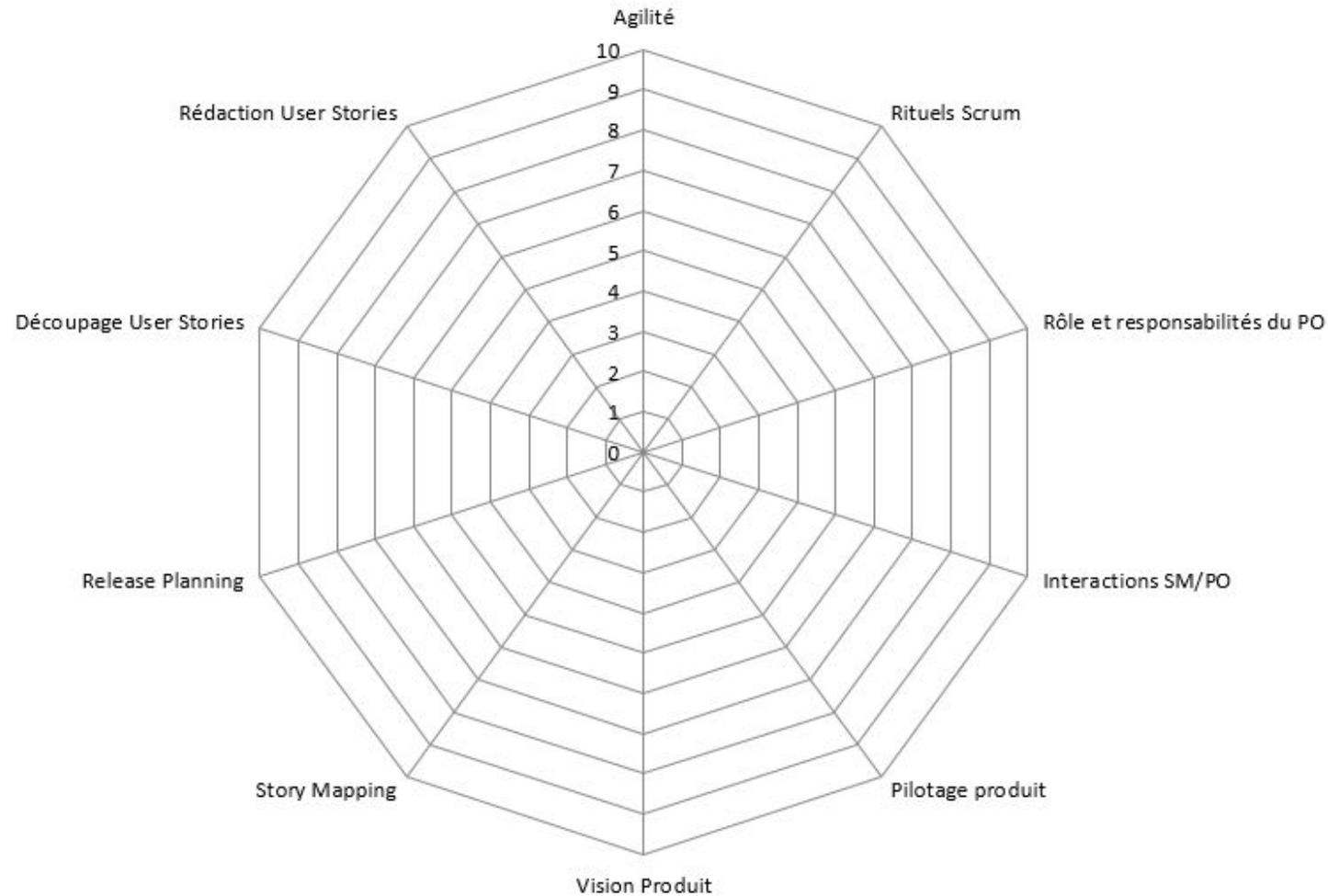
Montrer aux utilisateurs les fonctionnalités réalisées,
Recueillir leurs feedbacks et alimenter le Product backlog en conséquence,
Reprioriser le backlog



Mon radar PO

Comment est-ce que je me situe par rapport aux éléments suivants ?

(0 = je n'ai aucune connaissance, 10 = je suis Maître Jedi)



Dans la pratique, quelles difficultés rencontrez-vous?

Quelques problèmes que l'on rencontre

Product Owner pas assez disponible

- Attente d'informations
- Délais

Product Owner sans vision ou légitimité

- Décisions pas ou mal prises
- Délais
- Reworks nécessaires, beaucoup de changement de priorités

Product Owner que de nom

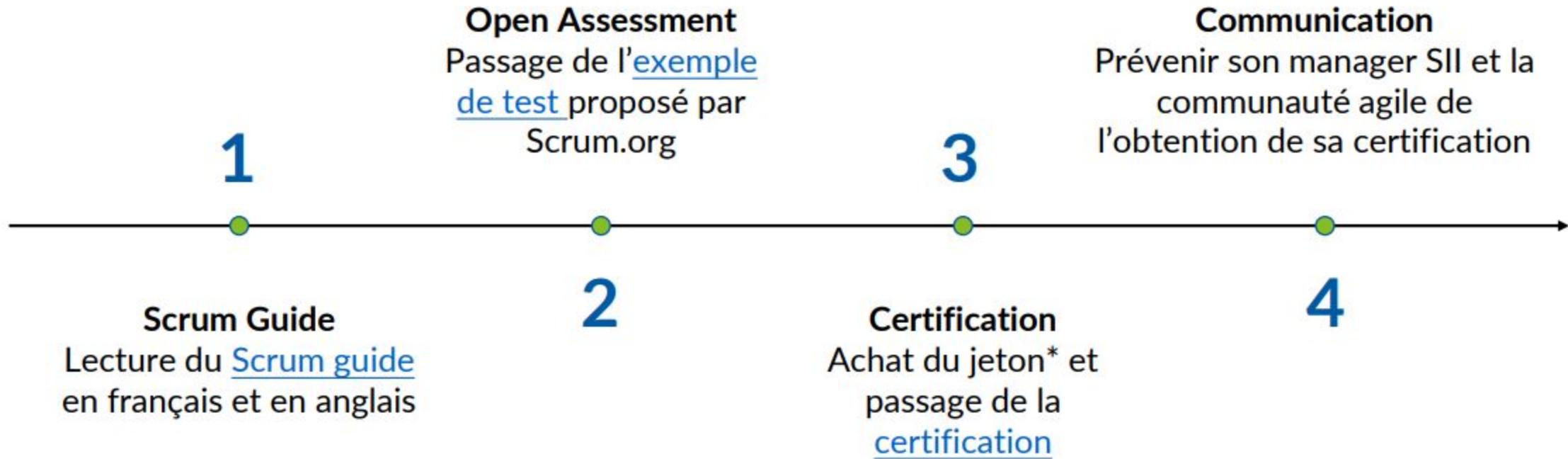
- Backlog pas ou mal formalisé
- niveau solution à implémenter
- Acceptation pas ou mal faite
- Délais
- Pas de vision

Comment y répondre en restant Agile?



PSPO 1

Plusieurs étapes



*Le passage de la certification est pris en charge par SII. Vous pouvez soit :

- demander une avance de frais pour financer le jeton (faire quand même une note de frais après le passage avec la facture)
- régler le jeton et faire une note de frais après le passage avec la facture pour le remboursement

Informations concernant l'examen PSPO 1



85%

Le score pour réussir est de **85% de bonnes réponses**



80 questions

60 minutes

Vous devez répondre à **80 questions en 60 minutes**.
Vous devez faire moins de 12 erreurs.

45 secondes par question (y compris la lecture)

- Comprendre dès la première lecture
- Repérer les mots clés dans les questions et les réponses
- Penser au mode de réponse par élimination



Informations concernant l'examen PSPO 1

Conseils:

5 familles de questions:

- Scrum Framework,
- Scrum Theory and Principles,
- Cross-functional, self-organising Teams,
- Maximizing Value
- Product Backlog Management.

- Ayez avec vous la version papier du SCRUM guide,
- Notez sur les diapositives ou le SCRUM guide quelques infos ou repères, afin de vous y retrouver rapidement,
- Passer l'examen blanc plusieurs fois, jusqu'à ne plus faire de fautes plusieurs fois de suite
- Possibilité de consulter le cours, internet!
- 50% de questions communes avec l'examen PSM I
- Certaines questions nécessitent plus d'une réponse, elles sont repérables par le symbole au lieu de , parfois c'est clairement demandé,
- Prêtez attention aux « should », « must », « could »...
- Méfiez-vous des faux amis (anglais)
- Vous avez la possibilité de mettre des questions de côté (**Bookmark**)
 - MAIS : y répondre quand même pour éviter le sans réponse
 - A la dernière question : attention à ne pas valider le questionnaire !

Informations concernant l'examen PSPO 1



Comment accéder au questionnaire ?

<https://www.scrum.org/professional-scrum-product-owner-certifications>

Sélectionnez:

PSPO I

Il va falloir d'abord créer un compte et acheter le token. Le **token n'expire pas**, l'achat peut être effectué à l'avance quelque soit la date de passage de la certification. Il est annoncé que le token sera disponible sous 24h, mais souvent il est disponible de suite.

Cliquer sur « Start », entrer le **token jeton** que vous avez acheté au préalable.

A l'issue de la validation du questionnaire, vous recevrez un diplôme de Product Owner certifié.

Shopping Cart



PSPO I Assessment

Fee for one attempt at the PSPO I assessment. Please note that all Scrum.org assessments are in the English language only. Assessment passwords do not expire and remain valid until used. **Your password will be emailed within 1 business day.** When attempting, the assessment will have a 60-minute time-box.



Lexique



- **Affinage du Backlog ou Product Backlog Refinement** : cérémonie durant laquelle on présente à son équipe les nouveaux récits utilisateurs. L'idée première d'un affinage du backlog est de pouvoir **valider la faisabilité technique et établir une estimation de réalisation des user stories**.
- **BurnDown Chart** : Graphique généralement utilisé pour suivre l'avancement du *Sprint* (horizon court terme). Il permet de suivre l'évolution du travail restant en fonction du temps. Le but consiste donc à atteindre le niveau zéro le plus tôt possible, d'où le terme « Down ».
- **BurnUp Chart** : Graphique d'avancement généralement utilisé pour suivre l'avancement d'une Release (horizon moyen ou long terme). Il permet de suivre l'évolution de la quantité valeur métier créée en fonction du temps. Le but consiste donc à atteindre la cible (haut du graphique) le plus tôt possible, d'où le terme « Up ».
- **Business Value / Valeur métier** : Valeur apportée au produit par le développement d'un item du backlog. L'échelle de valeur est généralement arbitraire.
- **Capacité** : Capacité à faire d'une équipe pour un Sprint à venir. Souvent exprimée en nombre de Story Points. Elle tient compte des congés de l'équipe
- **Definition of Done** : Liste de critère sur lesquelles l'équipe s'est mise d'accord pour qu'une User Story soit considérée comme finie
- **Definition of Ready** : Liste de critère sur lesquelles l'équipe s'est mise d'accord pour qu'une User Story soit considérée comme prête à être embarquée sur un Sprint. On parle souvent d'une US « sèche ».
- **Mêlée Quotidienne / Daily Meeting / Stand-Up Meeting** : Réunion quotidienne de moins de 15 minutes permettant à l'équipe Scrum de se synchroniser, identifier les obstacles éventuels et mesurer son avancement sur le Sprint en cours.
- **Persona** : Fiche biographique détaillée d'un utilisateur fictif de son produit
- **Planning Poker** : Technique d'estimation collective. Chaque élément du Product Backlog est estimé collectivement en se basant généralement sur l'unité appelée « Story Point ». Cette estimation va notamment aider le Product Owner à prioriser son Product Backlog.
- **Product Backlog / Backlog** : Liste ordonnée (priorisée) des besoins (généralement formulés sous forme de User story) du projet.

- **Release / Version** : Une release est une nouvelle version du produit, livrée aux utilisateurs. Elle est le fruit de 1 ou plusieurs incréments.
- **Scrum board / kanban** : tableau des tâches utilisé pour suivre visuellement l'activité du Sprint ou d'une release.
- **Sprint / Itération** : Intervalle de temps court (1 – 4 semaines, souvent appelé itération), pendant lequel l'équipe va concevoir, réaliser et tester de nouvelles fonctionnalités.
- **Sprint Backlog** : Périmètre fonctionnel d'un sprint. Liste de User Stories + objectif du sprint + plan pour y arriver
- **Story Point** : Unité d'estimation couramment utilisée sur les projets agiles. Permettant de faciliter l'estimation en soi ainsi que la planification des Releases ou Sprints. Tout en évitant de tomber dans l'illusion de la précision des estimations ou l'amalgame entre les jour/hommes et le délai de réalisation.
- **Storytelling** : Communication narrative. La technique du storytelling doit normalement permettre de capter l'attention et de susciter l'émotion.
- **Test d'acceptation** : Test écrit sur une User Story afin de valider son bon fonctionnement. Chaque test exprime une règle de gestion.
- **User Story / Story** : Technique permettant de formaliser synthétiquement les besoins sans perdre de vue l'essentiel : le besoin concerne QUI, en QUOI il consiste et dans quel BUT.
- **Vélocité** : Nombre de points réalisés sur le Sprint précédent



Merci