

L'AGILITÉ AU QUOTIDIEN

Formation 2 jours – Juillet 2019

1

INTRODUCTION

➤ LES OBJECTIFS DE LA FORMATION



Découvrir les valeurs et pratiques de l'Agilité

Prendre en compte la dimension collective des projets Agile

Appréhender au mieux les différents rôles et rituels

Se projeter dans la réalisation d'un projet Agile.

30% théorie / 70% pratique

LES RÈGLES



LOGISTIQUE

➤ HORAIRES PROPOSÉS

Jour 1 : 9:30 – 17:30

Jour 2 : 9:30 – 17:30

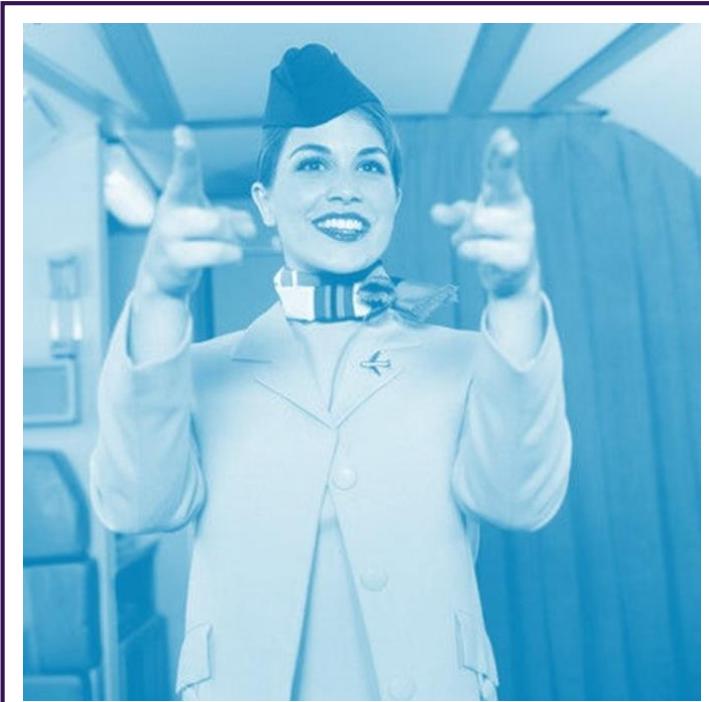
➤ DÉJEUNER

12:00 – 13:00

➤ AUTRES INTERRUPTIONS

Une pause matin et après-midi

➤ CONTRAINTES PARTICULIÈRES ?

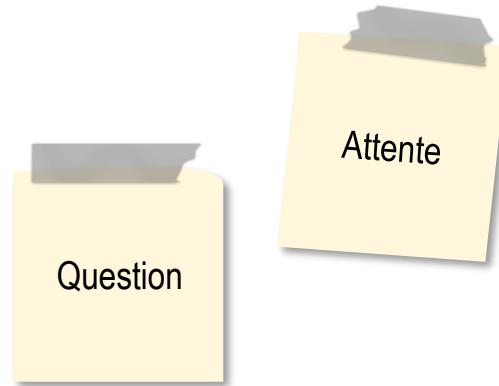


INTRODUCTION



➤ TOUR DE TABLE

➤ L'AGILITÉ ET VOUS



0 : heu... c'est quoi ça ?

+1 : j'ai suivi un MOOC

1 : je vois à peu près...

+10 : j'ai déjà expérimenté

2 : m'suis documenté !

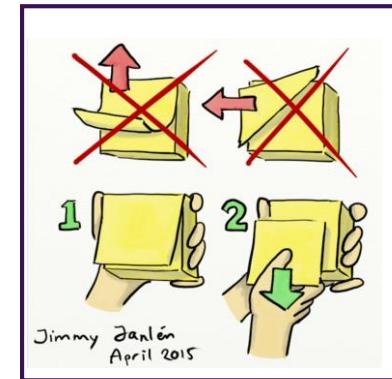
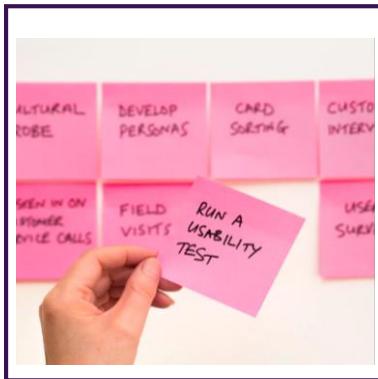
+100 : je l'ai inventé !

LA BONNE UTILISATION DES POST-IT

S Q L I 2 0 1 9



LA BONNE UTILISATION DES POST IT



➤ Marqueur = Post-it

➤ Simplicité =
transmettre une idée
sans l'expliquer

➤ Si ça ne colle pas, ça
ne compte pas !

SOMMAIRE

➤ Jour 1

- Pourquoi l'agilité
- Le Manifeste Agile
- Changer de paradigme
- Pilotage par la valeur
- La galaxie Agile
- Scrum
- Les rôles Scrum
- La gestion du backlog produit
- L'item du backlog
- La communication au sein de l'équipe Agile

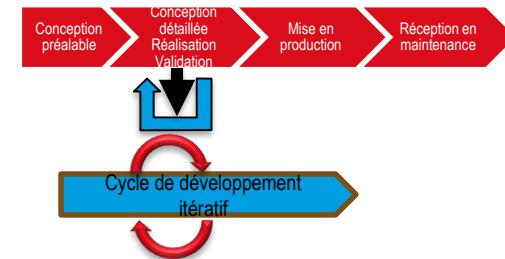
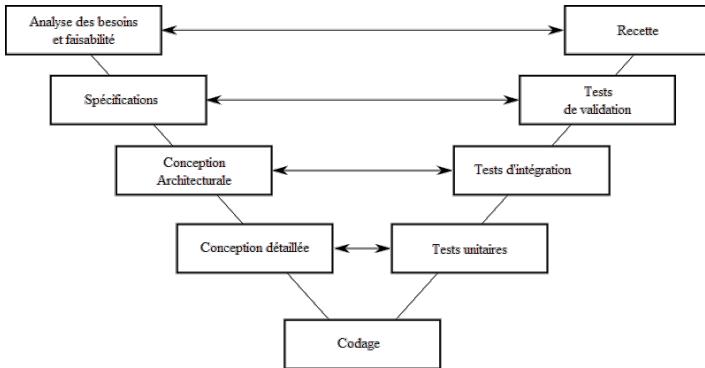
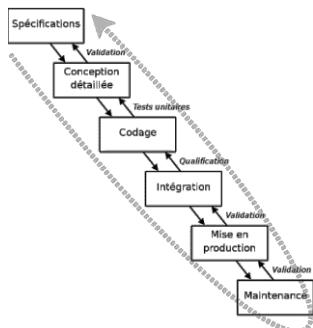
➤ Jour 2

- Retour sur la 1^{ère} journée
- Gérer les priorités
- La planification de sprint
- L'estimation
- L'itération Scrum
- Le daily Scrum
- La revue de sprint
- La rétrospective
- Management visuel & indicateurs
- Vivre Scrum !
- Kanban
- XP
- La dette technique
- Pour conclure
- Annexes

01

POURQUOI L'AGILITÉ ?

CASCADE, CYCLE EN « V » OU ITÉRATIF



- **Concept**

Passage d'une phase à l'autre après revue-validation complète
Pas de retour-arrière possible
 - **Points forts**

Distingue clairement les phases projet
Modèle adapté dans le cas où les besoins sont clairement identifiés et stables
 - **Points faibles**

La vérification du bon fonctionnement du système est trop tardive (effet tunnel)
Un changement peut coûter cher.
 - **Concept**

Le modèle du cycle en V a été imaginé pour palier au problème de réactivité du modèle en cascade.
 - **Points forts**

Anticiper dans les étapes descendantes les attendus des étapes montantes (phases de tests)
 - **Points faibles**

Vérification trop tardive du bon fonctionnement du système.
Nécessité de documenter de façon détaillée tous les éléments du projet
 - **Concept**

Seule la phase de construction est itérative / incrémentale.
 - **Points forts**

Adéquation totale du cycle de développement itératif à une approche classique
Impacte uniquement la phase de construction (qui reste interne)
 - **Points faibles**

Nécessite toujours une expression complète et « parfaite »

LES PRÉSUPPOSÉS

➤ LE CLIENT SAIT PARFAITEMENT CE QU'IL VEUT

Et il sait l'exprimer

➤ LES DÉVELOPPEURS

Comprendnent les besoins immédiatement

Maitrisent le système quelque soit sa complexité

Font peu d'erreurs

Sont interchangeables facilement

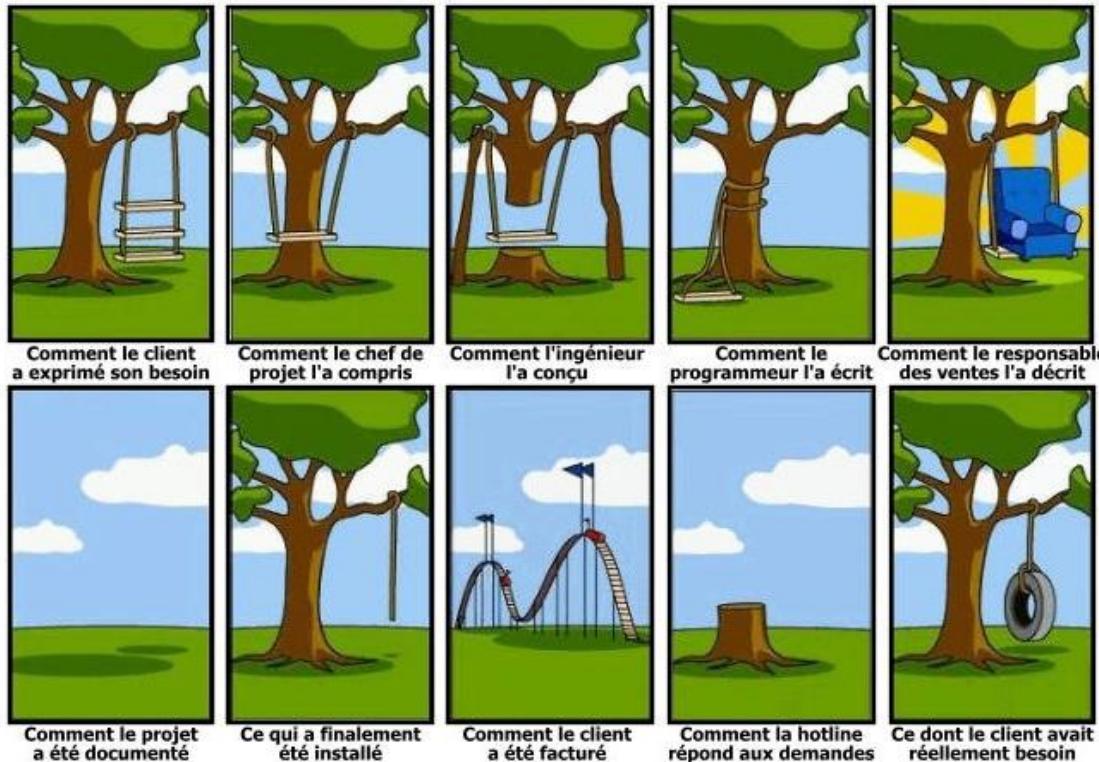
...

Bref, sont motivés !

➤ LES LIVRABLES EXTERNES ARRIVENT À TEMPS ET EN BON ÉTAT

UN PROJET PRESQUE CLASSIQUE

La vie d'un projet informatique 'bien' mené



CYCLE EN V : LES LIMITES !

Spécifications complètes & immuables

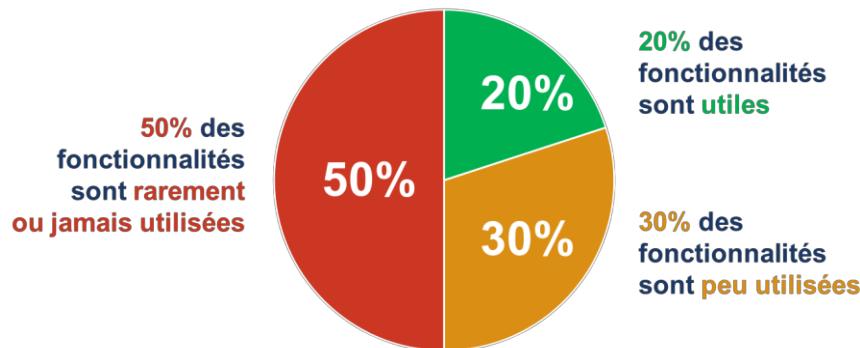
Design parfait avant la réalisation
Tout doit être documenté

Rigidité du contrat
Changement = coût

Traitement discontinu de chaque fonctionnalité

Manque de visibilité côté client
« *Effet tunnel* »

Valeur visible à la fin du projet uniquement



* The Standish group CHAOS Manifesto 2013

Taux de réussite des projets en 2012

Successful	39%
Failed (Abandonnés)	18%
Challenged (Dépassement de charges et/ou délais)	43%

Source : Standish Group – Chaos manifesto 2013

EFFET TUNNEL



Lors du développement d'un logiciel, les futurs utilisateurs **définissent leurs besoins**, puis **réceptionnent** des mois plus tard le logiciel terminé. Entre ces deux étapes, ils n'ont **rien vu**, ils n'ont été **informés de rien**, et bien évidemment, le résultat est souvent très **loin de leurs besoins**.

POURQUOI L'AGILITÉ ?



Source : State of Agile Survey – VersionOne 2019

<https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508>

LE MANIFESTE AGILE



UN PEU D'HISTOIRE

➤ 1986 :

Modèle de développement itératif et incrémental de Barry W. Boehm

➤ 1991 :

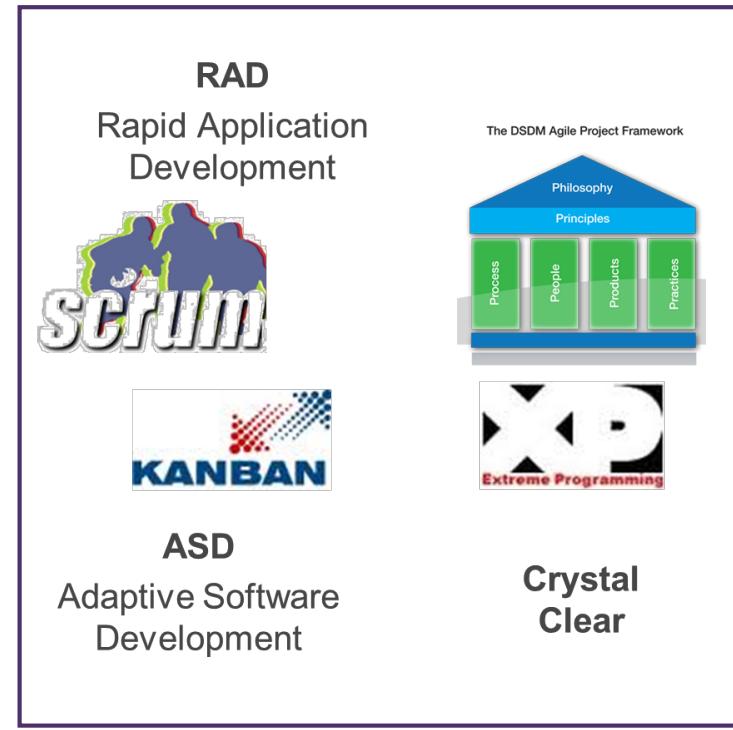
RAD (Développement rapide d'applications) de James Martin

➤ A PARTIR DE 1995 :

Développement des méthodes Agiles (scrum, XP, ...)

➤ 2001 :

Consensus et formalisation par les auteurs de ces méthodes ➔ Le **manifeste Agile**



ET MAINTENANT ...

➤ UNE TENDANCE FORTE, AU-DELÀ D'UN EFFET DE MODE

L'Agilité sort de l'informatique et s'étend efficacement à d'autres types de projets

"Agile is not only used in several of the world's leading companies now but is being applied in areas beyond software development. (.../...) some companies are applying Agile to areas within the organization such as, portfolio management, project management, vendor management, contracting, etc."

(Etude de l'institut Forrester, 2013)

➤ LE DOMAINE IMPORTE PEU (INDUSTRIE LOURDE, RECHERCHE, JOURNALISME...)

➤ IL FAUT FAIRE L'EFFORT DE S'Y METTRE MAIS ON NE LE REGRETTE PAS

LE MANIFESTE AGILE

Manifeste pour le développement Agile de logiciels

Nous découvrons comment mieux développer des logiciels
par la pratique et en aidant les autres à le faire.
Ces expériences nous ont amenés à valoriser :

- Les individus et leurs interactions plus que les processus et les outils
- Des logiciels opérationnels plus qu'une documentation exhaustive
- La collaboration avec les clients plus que la négociation contractuelle
- L'adaptation au changement plus que le suivi d'un plan

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

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

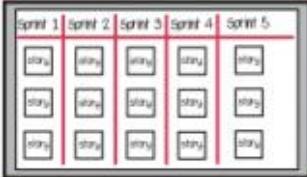
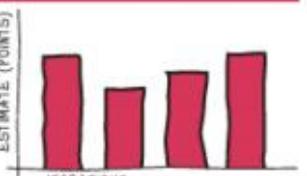
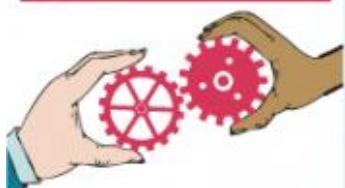
 © 2001, the above authors
 this declaration may be freely copied in any form,
 but only in its entirety through this notice.



12 PRINCIPLES

Knowledge TRAIN ●●●

*The 12 agile principles**

1 Satisfy the customer 	Welcome change 	Deliver frequently 	4 Work together 
5 Trust and support 	Face-to-face conversation 	Working software 	8 Sustainable development 
9 Continuous attention 	10 Maintain simplicity 	11 Self-organizing teams 	12 Reflect and adjust 

Copyright © 2016 Knowledge Train Limited. www.knowledgetrain.co.uk. *Copied from www.agilemanifesto.org

12 PRINCIPES

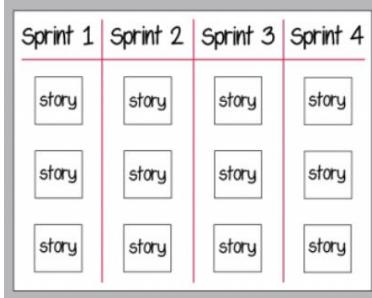


Notre plus haute priorité est de **satisfaire le client** en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.



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

12 PRINCIPES



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



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

12 PRINCIPES



Réalisez les projets avec des **personnes motivées**. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur **confiance** pour atteindre les objectifs fixés.

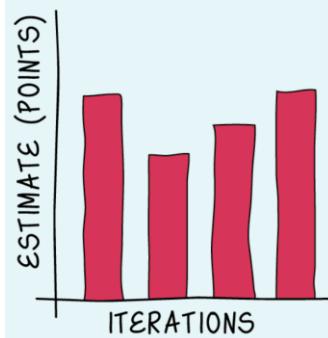


La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en **face à face**.

12 PRINCIPES

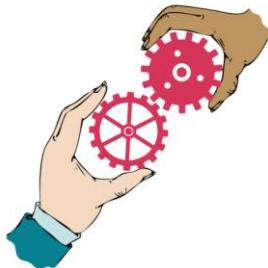


Un logiciel **opérationnel** est la principale mesure d'avancement.



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.

12 PRINCIPES



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.

12 PRINCIPES



Les meilleures architectures, spécifications et conceptions émergent d'**équipes auto-organisées**.

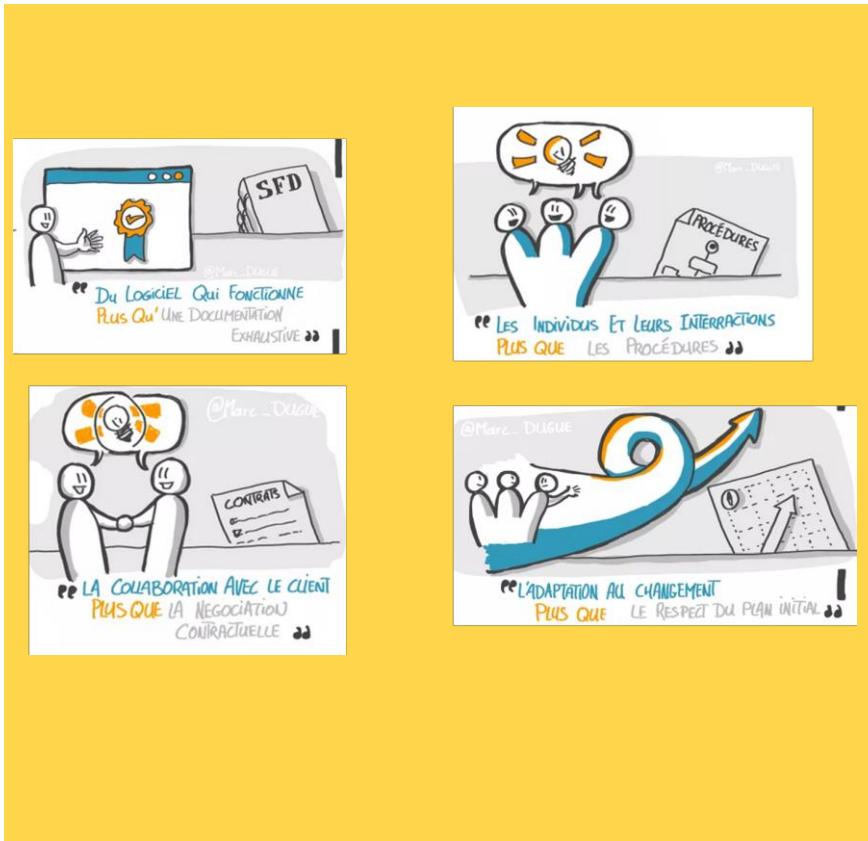


À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son **comportement en conséquence**.



ASSOCIEZ LES PRINCIPES AUX VALEURS

S Q L I 2 0 1 9



Satisfaire le client	Considérer comme naturel les changements d'exigences	Livrer fréquemment un application fonctionnel
Assez le projet autour de personnes motivées	L'échange d'information le plus efficace est en face à face	Un logiciel fonctionnel est la meilleure façon de mesurer l'avancement du projet
Simplicité - l'art de maximiser la quantité de travail à ne pas faire - est essentielle	Architectures, spécifications et conceptions issues d'équipes auto-organisées	Vérifier en continu l'excellence des pratiques techniques
Fonctionnels et développeurs travaillent ensemble	Le rythme de développement doit être soutenable indéfiniment	Régulièrement, réflexion l'équipe pour être plus efficace !



L'AGILITÉ C'EST ...

- Donner la priorité à ce qui est important
- Une équipe unie et motivée
- Un rythme de travail durable
- Plus de réactivité, d'adaptation
- Délais de mise à disposition plus courts



**Il devient nécessaire de trouver de nouvelles solutions
EN CONSERVANT
rigueur et adéquation avec les attendus
méthodologiques.**

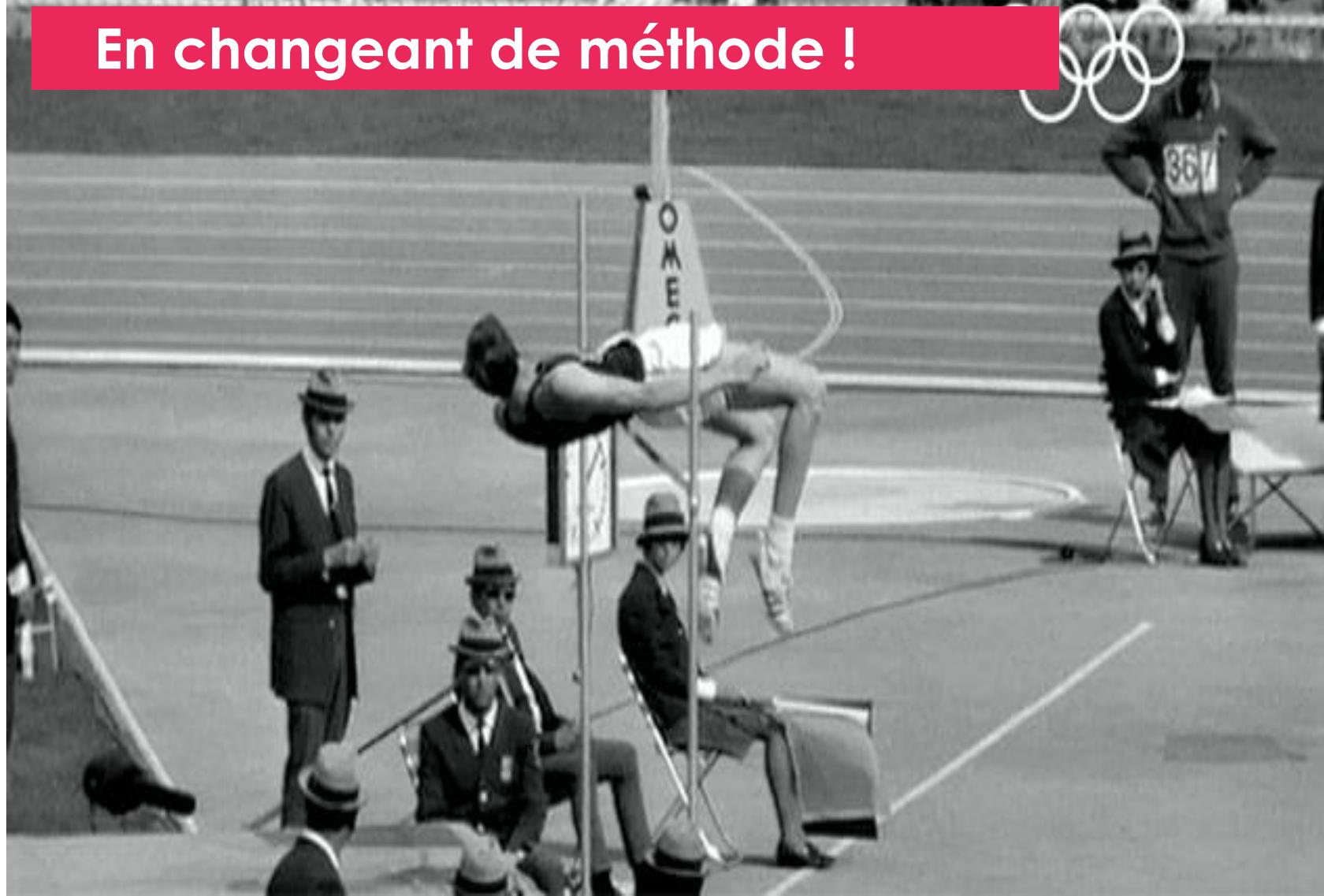
03

OK, MAIS COMMENT ?

Comment peut-on aller plus haut ?

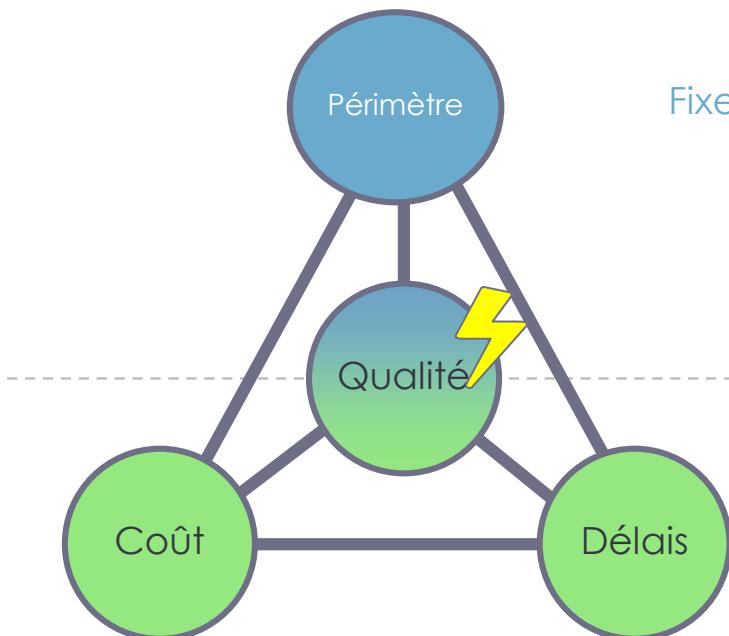


En changeant de méthode !



CHANGER DE PARADIGME

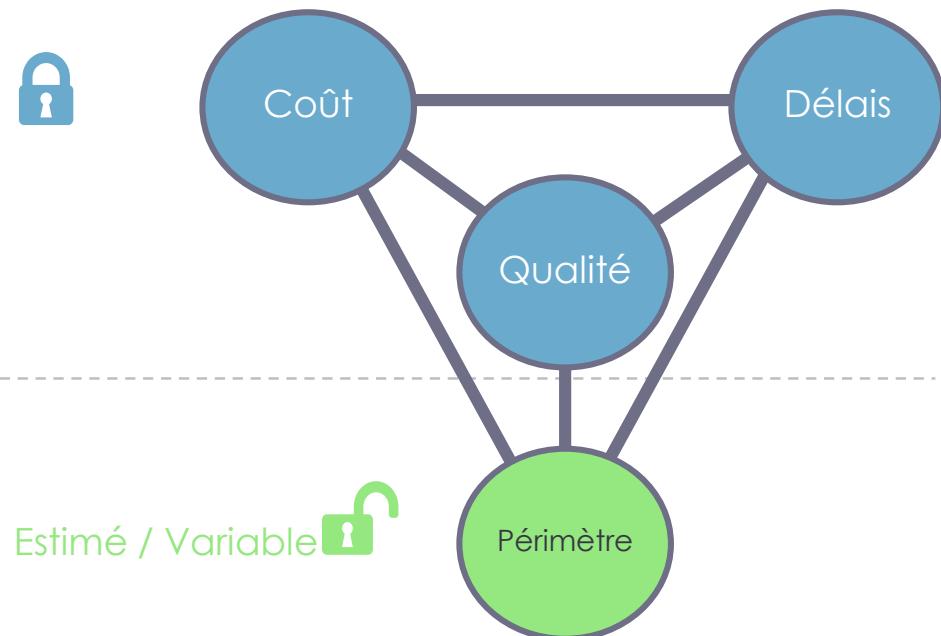
Traditionnel



Fixe



Agile



Estimé / Variable



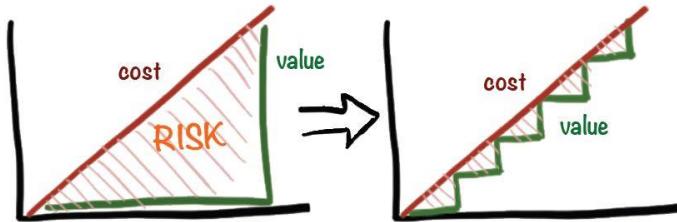
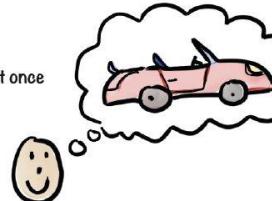
ITÉRATIF ET INCRÉMENTAL

SQLI 2019

Agile = Iterative + Incremental

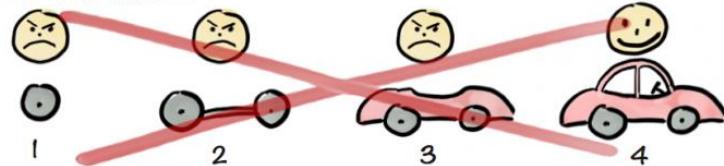
Don't try to get it all right
from the beginning

Don't build it all at once

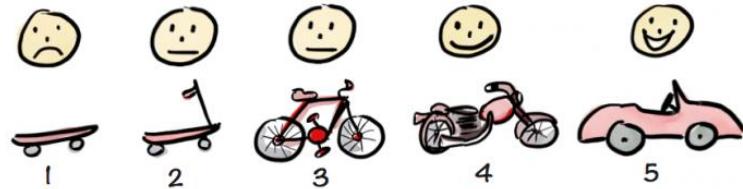


Henrik Kniberg

Not like this....



Like this!



MVP

INTÉGRER LA SOUPLESSE

➤ PRIORITÉ À CE QUI EST IMPORTANT

- Chacune des itérations de développement porte sur les fonctionnalités qui apportent le plus de valeur au métier.

➤ ITÉRATIVE ET INCRÉMENTALE

- Permet d'avoir un résultat très tôt et très vite
 - ...potentiellement exploitable
 - ...visible => feedback.

➤ ADAPTATIVE : PRISE EN COMPTE QUE DERRIÈRE UN PROJET IL Y A

- Des hommes
- Des organisations,
- les contraintes du marché ... qui sont impactés tout au long du projet.

ET SIMPLIFIER

➤ SIMPLICITÉ DU CADRE

- Peu de rôles
- Les responsabilités correspondent aux compétences.

➤ SUPPRESSION DES ACTIVITÉS NON DIRECTEMENT PRODUCTIVES DE VALEUR

- Communication directe, et visuelle
- Activité en phase avec compétence.

➤ DIMENSIONNEMENT TEMPOREL ADAPTÉ

➤ MISE EN VISIBILITÉ DE L'AVANCEMENT QUOTIDIEN ET COLLECTIF.

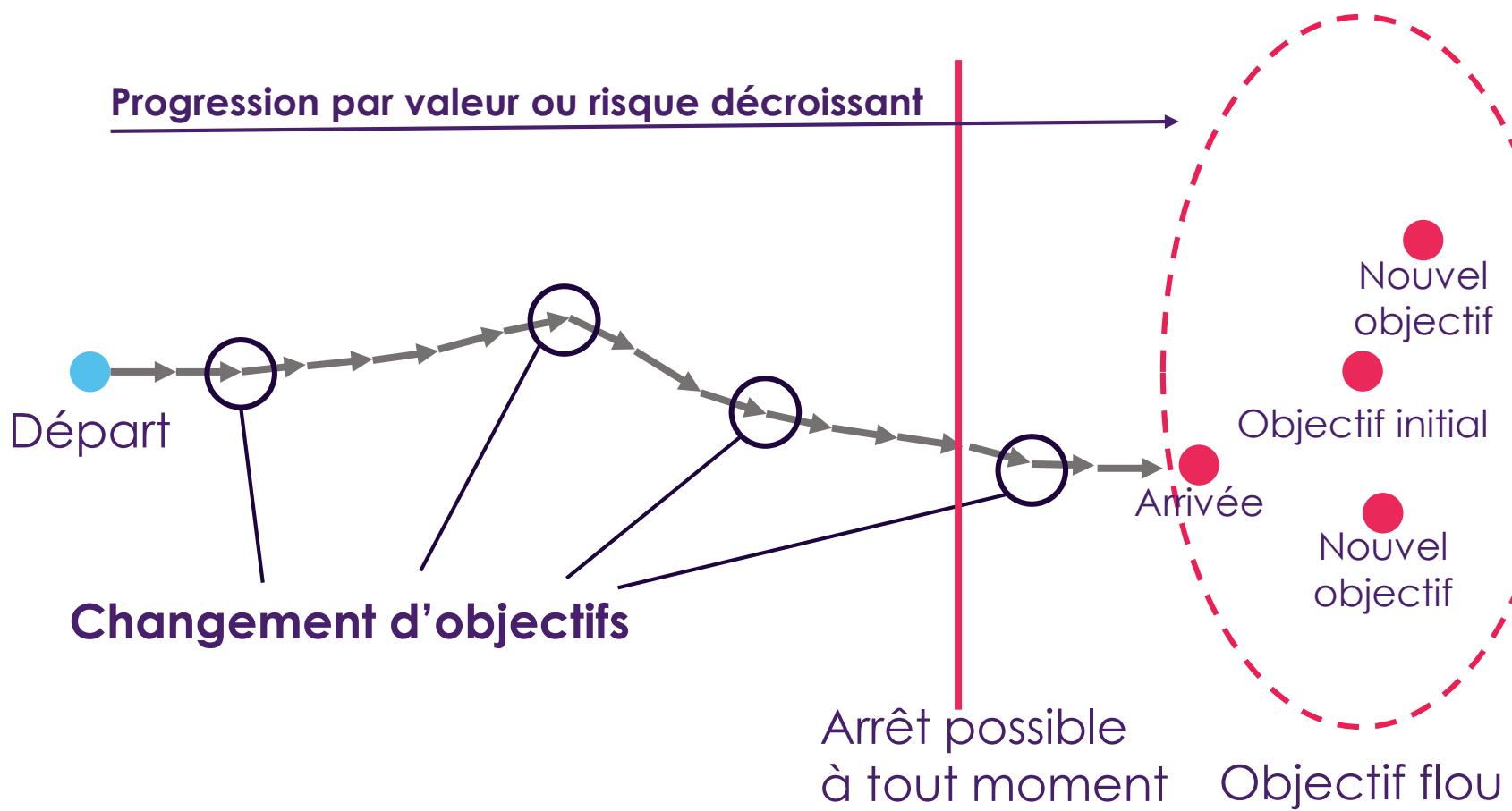
Pause



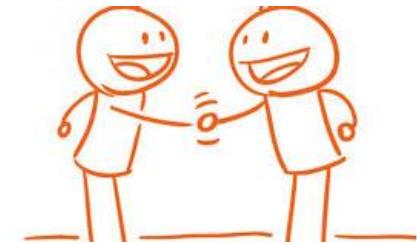
PILOTAGE PAR LA VALEUR

UN OBJECTIF QU'ON DÉCOUVRE PROGRESSIVEMENT

SQLI 2019

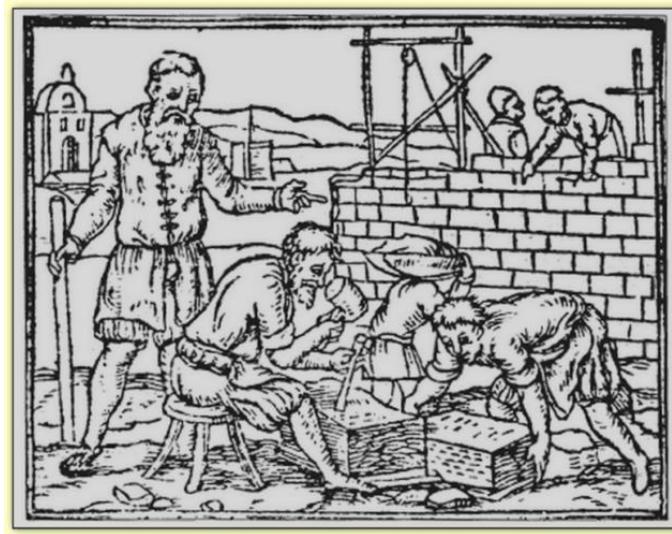


UN PARTENARIAT MÉTIER / IT

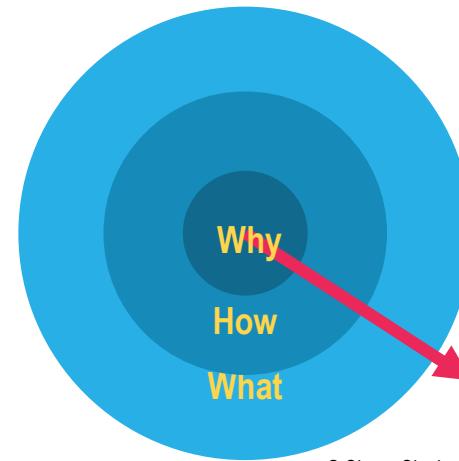


- **LE CLIENT DÉFINIT SON BESOIN ET LE PRIORISE**
 - Selon la **valeur business** et/ou le but utilisateur
- **AVANT CHAQUE ITÉRATION, LE CLIENT ET L'ÉQUIPE DÉFINISSENT ENSEMBLE CE QUI SERA DÉVELOPPÉ DURANT L'ITÉRATION SELON LES PRIORITÉS FIXÉES**
- **LE BESOIN ET LES PRIORITÉS PEUVENT ÉVOLUER ENTRE CHAQUE ITÉRATION**
- **LA VALEUR ACQUISE DU LOGICIEL EST D'AUTANT PLUS IMPORTANTE SUR LES PREMIÈRES ITÉRATIONS**
 - Le client peut arrêter le projet en cours s'il le désire (les fonctionnalités les plus importantes seront déjà implémentées)
 - Les fonctionnalités superflues peuvent être abandonnées
 - De nouveaux besoins pourront s'ajouter au long du projet suivant le même principe de valeur

DONNER DE LA VISION



Commencer par le **pourquoi**



© Simon Sinek



LE PITCH DE L'ASCENSEUR

SQLI 2019



- Vous êtes membre d'une équipe projet. Vous prenez l'ascenseur pendant 90 secondes avec le PDG de votre entreprise. Décrivez lui le besoin de changer et la vision de la situation future (« pourquoi faisons-nous ce projet? »)

- Pour (clients, utilisateurs)
Qui (font quelque chose → i.e. opportunité du projet)
Le (nom du projet/produit) est un (catégorie du produit)
Qui (bénéfice premier, raison primordiale de l'utiliser).
Contrairement à (version précédente, produit concurrent)
Notre projet/produit (argument choc de différenciation)

ACTIVITÉS SÉQUENTIELLES VS PARALLÈLES

SQLI 2019

Exigences

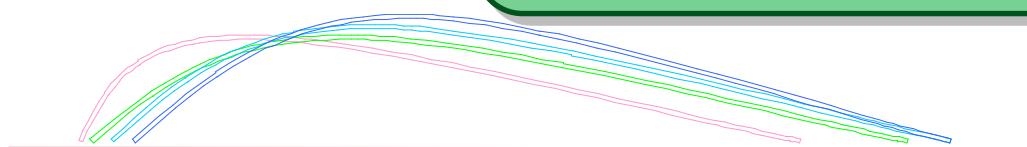
Conception

Code

Test

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

...Les équipes agiles font un peu de tout, tout le temps



Source : "The New New Product Development Game" par Takeuchi et Nonaka. Harvard Business Review, Janvier 1986.

ITÉRATIF & INCRÉMENTAL

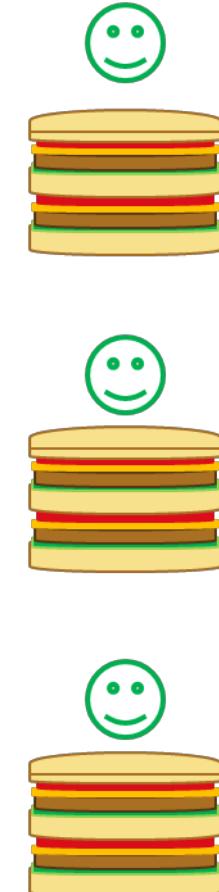
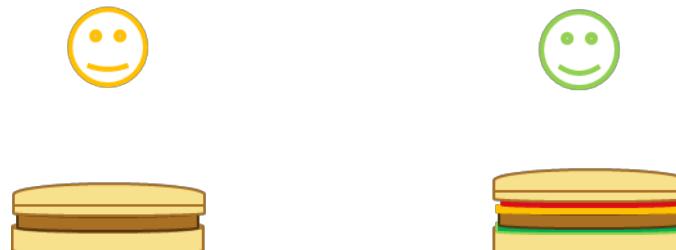
➤ ITÉRATIF



➤ INCRÉMENTAL



➤ ITÉRATIF &
INCRÉMENTAL





VOYAGE SUR LA LUNE

S Q L I 2 0 1 9



- Vous faites partie de l'équipage d'un vaisseau spatial programmé à l'origine pour rejoindre une fusée mère sur la face éclairée de la lune.
- A la suite d'ennuis mécaniques, vous avez dû alunir à 320 kms environ du rendez-vous fixé. Au cours de l'alunissage, la plupart des équipements de bord ont été endommagés, à l'exclusion des 15 objets ci-dessous.
- Il est vital pour votre équipage de rejoindre la fusée mère et vous devez choisir l'équipement indispensable pour ce long voyage.



JOONS
ENSEMBLE

VOYAGE SUR LA LUNE

UN EMETTEUR-RECEPTEUR FONCTIONNANT SUR ENERGIE SOLAIRE (FREQUENCE MOYENNE)



DES SIGNALS LUMINEUX



UN COMPAS MAGNETIQUE



DEUX PISTOLETS CALIBRE 45



UNE CARTE CELESTE DES CONSTELLATIONS LUNAIRES



UNE TRUSSÉ MEDICALE ET DES SERINGUES HYPODERMIQUES



25 LITRES D'EAU



UNE BOITE D'ALLUMETTES



DES ALIMENTS CONCENTRES



50 METRES DE CORDE EN NYLON



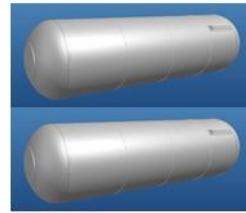
UN APPAREIL DE CHAUFFAGE FONCTIONNANT SUR ENERGIE SOLAIRE



UN CANOT DE SAUVETAGE AUTO-GONFLABLE



DEUX RESERVOIRS DE 50 KG D'OXYGENE CHACUN



UNE CAISSE DE LAIT EN POUDRE





ICEBREAKER



« 2 VÉRITÉS & 1 MENSONGE »



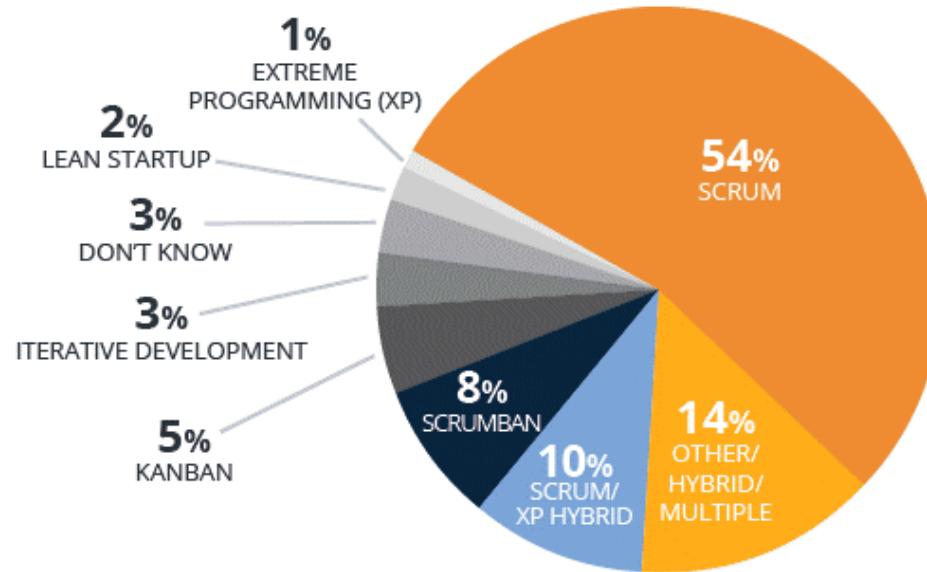
L'AGILITÉ CONCRÈTEMENT ...

LA GALAXIE AGILE

AGILE METHODS AND PRACTICES

Agile Methodologies Used

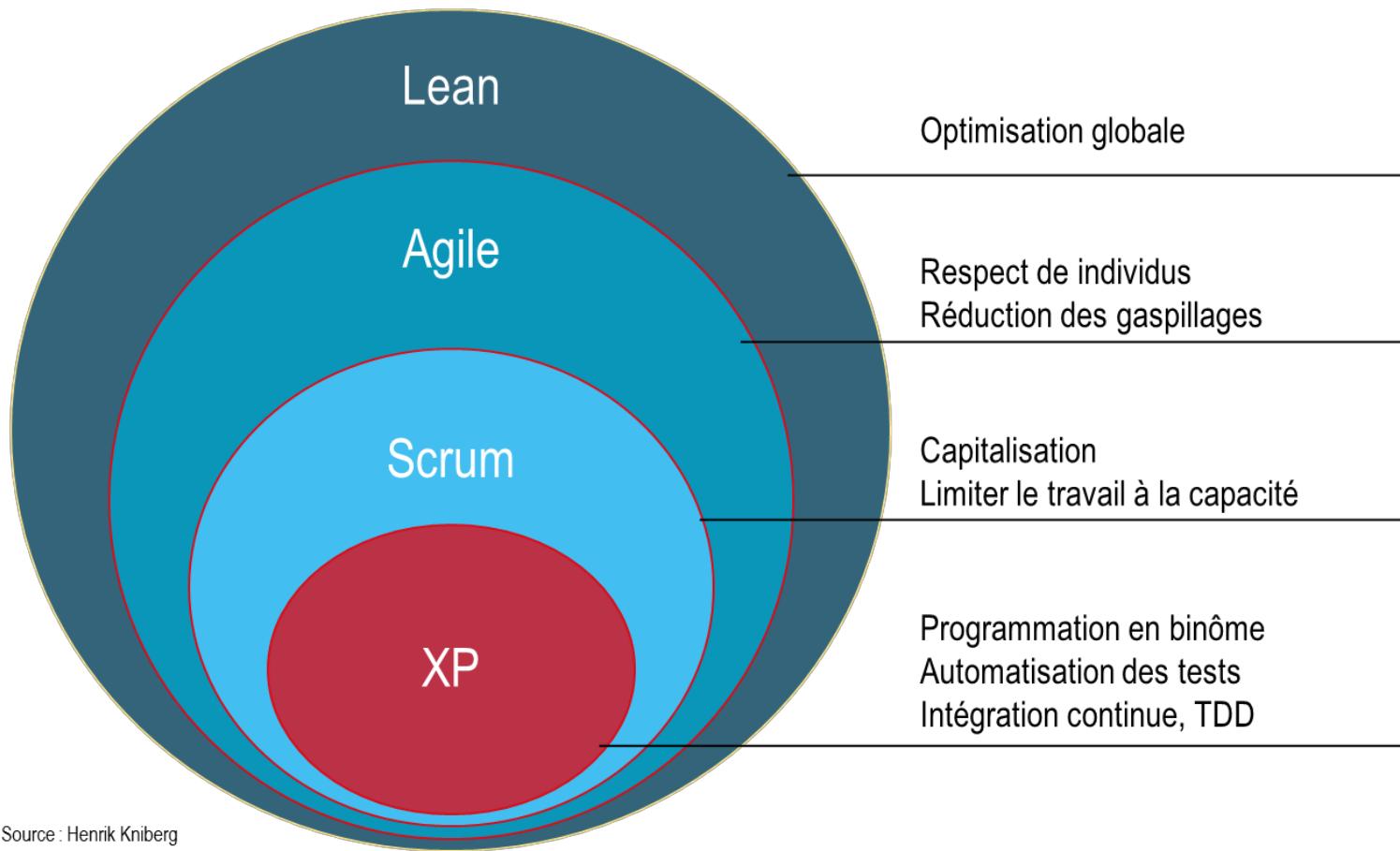
Scrum and Scrum/XP Hybrid (64%) continue to be the most common agile methodologies used by respondents' organizations.



Source : State of Agile Survey – VersionOne 2019

<https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508>

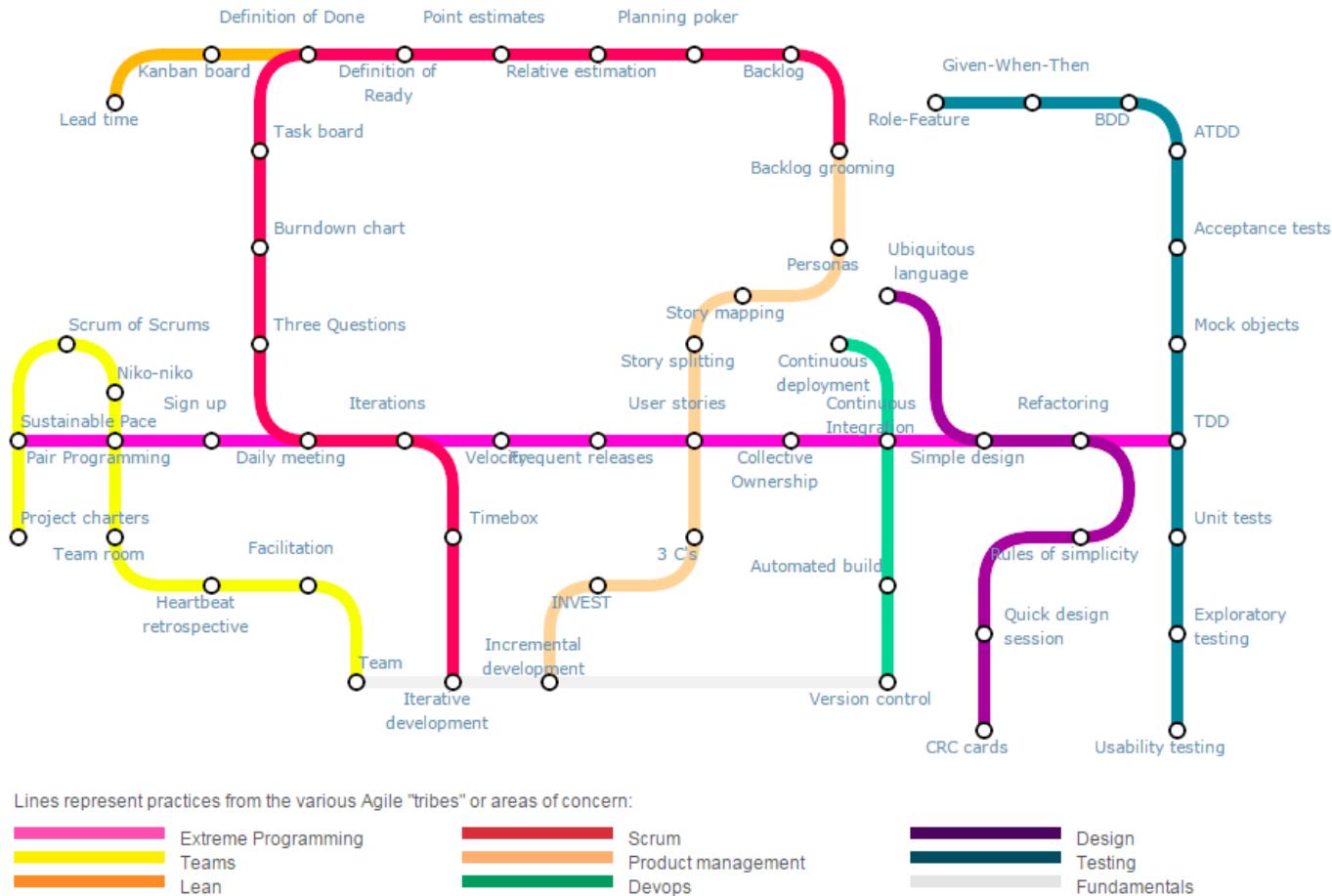
VUE D'ENSEMBLE



Source : Henrik Kniberg

DE NOMBREUSES PRATIQUES

SQLI 2019



[HTTPS://WWW.AGILEALLIANCE.ORG/AGILE101/SUBWAY-MAP-TO-AGILE-PRACTICES/](https://www.agilealliance.org/agile101/subway-map-to-agile-practices/)



AGILE BURGER QUIZZ

S Q L I 2 0 1 9

LAQUELLE
DE CES
MÉTHODES
AGILE
N'EXISTE PAS
?

SCRUM ASD

RUP DAD

FDD XP

DSDM

LA MÉTHODE AGILE QUI
N'EXISTE PAS



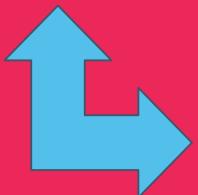
➤ [HTTPS://COACH-AGILE.COM/SERIOUS-GAME-AGILE-BURGER-QUIZ/](https://coach-agile.com/serious-game-agile-burger-quizz/)



L'AGILITÉ

► DES MÉTHODES ÉPROUVÉES

Valeurs



Principes



- S'appuyant sur un tronc commun de pratiques
- Définies pour répondre aux 4 valeurs du manifeste Agile

Méthodes

XP, SCRUM,

DevOps, Kanban...

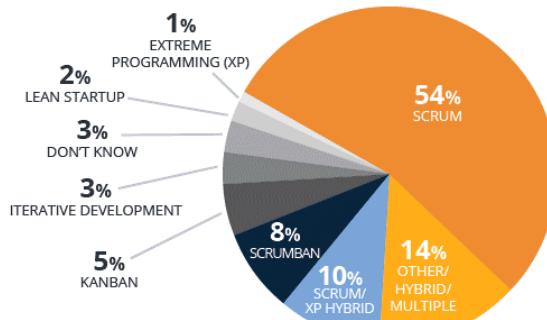


SCRUM

AGILE METHODS AND PRACTICES

Agile Methodologies Used

Scrum and Scrum/XP Hybrid (64%) continue to be the most common agile methodologies used by respondents' organizations.



LES FONDATEURS

- A L'ORIGINE, SCRUM A ÉTÉ FORMALISÉ EN 1995 PAR DEUX EXPERTS :



Jeff Sutherland

<http://jeffsutherland.com>

Ken Schwaber

<http://kenschwaber.wordpress.com>

- ILS ONT CONTRIBUÉ À DÉFINIR LE MANIFESTE AGILE (2001) ET RÉDIGÉ LE SCRUM GUIDE (2011)

RAPPEL HISTORIQUE

➤ ORIGINE DU NOM

Le terme « **Scrum** » est emprunté au rugby et signifie **mêlée**. Ce processus s'articule en effet autour d'une **équipe soudée**, qui cherche à atteindre un **but**, comme c'est le cas en rugby pour avancer avec le ballon pendant une mêlée.

➤ 2011

Scrum Guide, qui décrit les principes de la (Jeff Sutherland et Ken Schwaber)

➤ NOV. 2017:

Dernière version du Scrum Guide

➤ 2018

Scrum est toujours la méthode Agile la plus populaire



<https://www.scrum.org/resources/scrum-guide>

FONDAMENTAUX



> TROIS PILIERS

Transparence, Inspection, Adaptation

> DES VALEURS

Focus, Ouverture, Engagement,
Respect, Courage

> IDÉES CLÉS :

Le client au cœur du produit
Esprit d'équipe
La communication est la clé
Simplicité, efficacité et qualité
Flexibilité aux changements
Avancement basé sur le concret

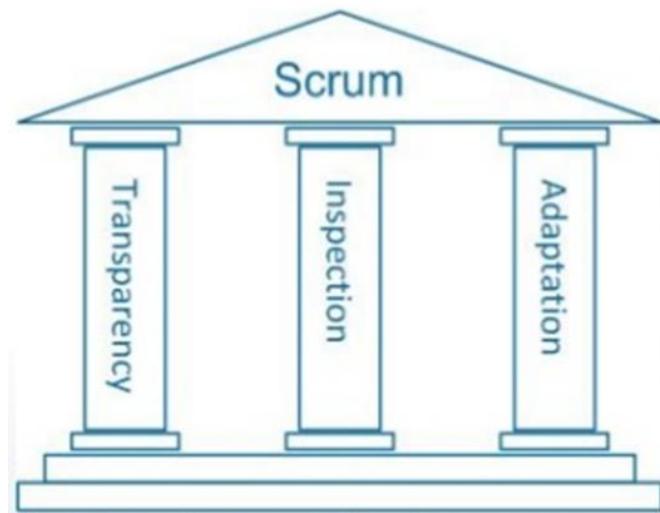
Conçu pour la gestion de développements logiciels, donc complexes

Framework adaptable à n'importe quel contexte ou à un groupe de personnes qui travaillent ensemble pour atteindre un but commun

Pas spécifique au logiciel
Dans l'industrie logicielle, Scrum est souvent complété par des pratiques d'ingénierie adaptées (comme XP par exemple)



LES 3 PILIERS DE SCRUM



➤ LA TRANSPARENCE

Scrum met l'accent sur le fait d'**avoir un langage commun** entre l'équipe et le management. Ce langage commun doit permettre à tout observateur d'obtenir rapidement une **bonne compréhension** du projet.

➤ L'INSPECTION

À intervalle régulier, Scrum propose de **faire le point** sur les différents **artéfacts** produits, afin de détecter toute variation indésirable.

Ces inspections ne doivent pas être faites trop fréquemment, ou par un inspecteur mal formé : cela nuirait à l'avancement du projet.

➤ L'ADAPTATION

Si une **dérive** est constatée pendant l'inspection, le processus doit alors être **adapté**.

Scrum fournit des **4 rituels formels**, durant lesquels cette **adaptation** est possible.

Il s'agit de la **réunion de planification de sprint**, de la **mêlée quotidienne**, de la **revue de sprint** ainsi que de la **rétrospective de sprint**.

LES 5 VALEURS DE SCRUM



COURAGE

Les membres de l'équipe Scrum ont le courage de faire ce qui s'impose et de s'attaquer aux problèmes difficiles

FOCUS

Tout le monde se concentre sur le travail du sprint et les objectifs de l'équipe Scrum

ENGAGEMENT

Les gens s'engagent personnellement à atteindre les objectifs de l'équipe Scrum

RESPECT

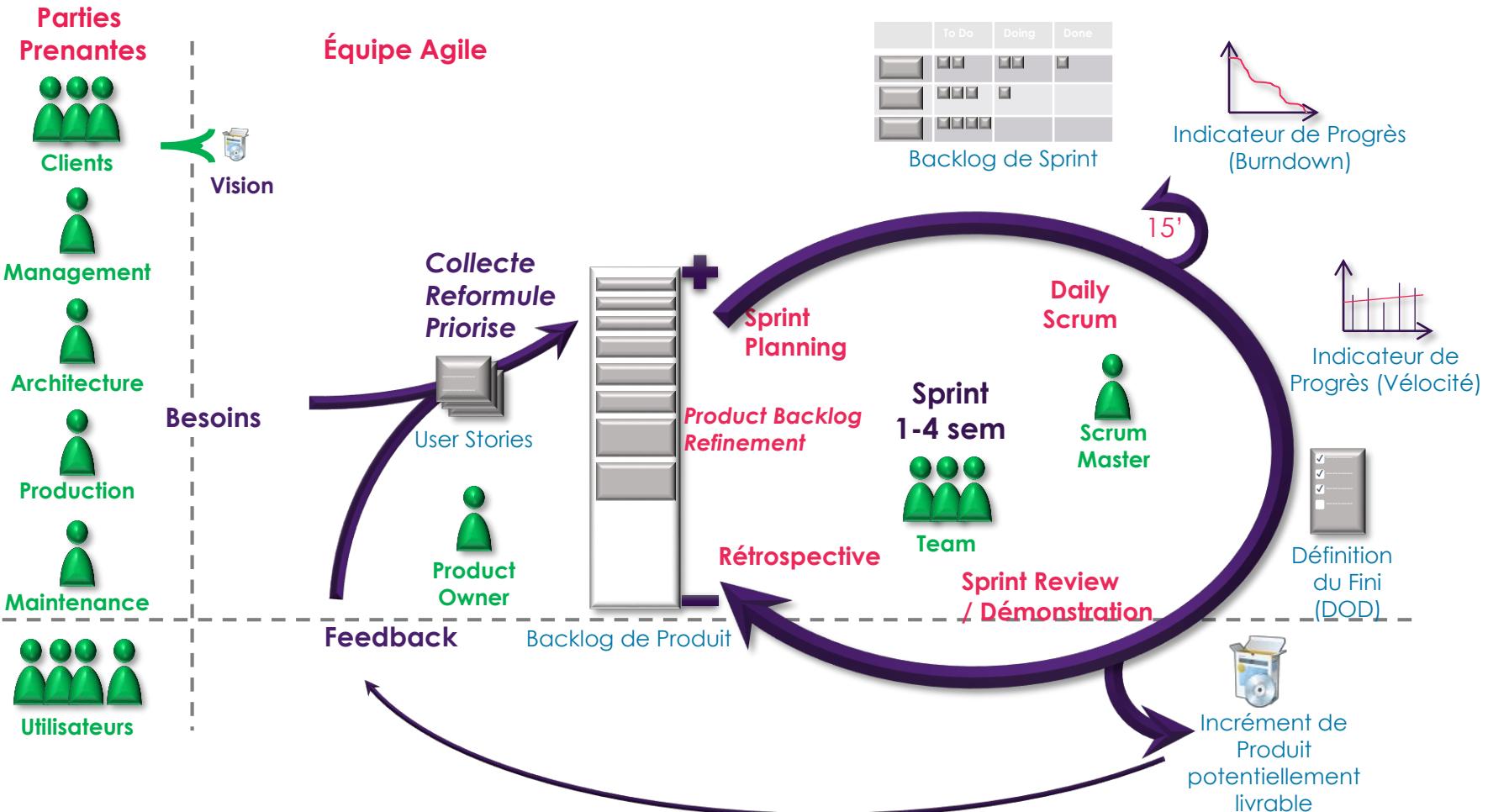
Les membres de l'équipe Scrum se respectent pour être des personnes capables et indépendantes

OUVERTURE

L'équipe Scrum et ses parties prenantes acceptent d'être ouvertes sur tout le travail et les défis liés à l'exécution de ce travail.

TOUT SCRUM EN 1 PAGE

SQLI 2019



SCRUM EN UNE AUTRE PAGE

➤ SCRUM: UN FRAMEWORK POUR DÉLIVRER UN PRODUIT AU COURS D'ITÉRATION EN DÉFINISSANT DES RÔLES, DES ÉVÉNEMENTS FORMATÉS, DES ARTEFACTS ET DES RÈGLES ASSOCIEES



Rôles	Définition / Mission
Parties Prenantes (Stakeholders)	Tout ceux qui ont des vues/souhaits sur le produit
Product Owner	Responsable de transformer les souhaits des parties prenantes dans un Produit, il/elle arbitre pour délivrer en premier les éléments ayant la plus forte valeur.
Scrum Master	S'assure que l'Equipe possède tous les moyens pour délivrer le produit. Facilite et supprime les obstacles
Membre d'Equipe (Development Team)	Toute personne qui construit le produit



Les rôles reprennent certaines attributions habituelles du Chef de Projet, mais pas toutes !



Événements	Définition
Planification d'itération (Sprint Planning)	Définition du contenu de l'itération Organisation des travaux ➔ Le 'Comment'
Mêlée quotidienne (Daily Scrum)	Inspection et adaptation quotidienne de l'avancement Identification et partage des obstacles à atteindre l'objectif
Démonstration (Sprint Review)	Démonstration et validation des réalisations comme incrément de produit Les participants donnent du feedback
Rétrospective	Inspection & adaptation de la façon de travailler en vue de s'améliorer



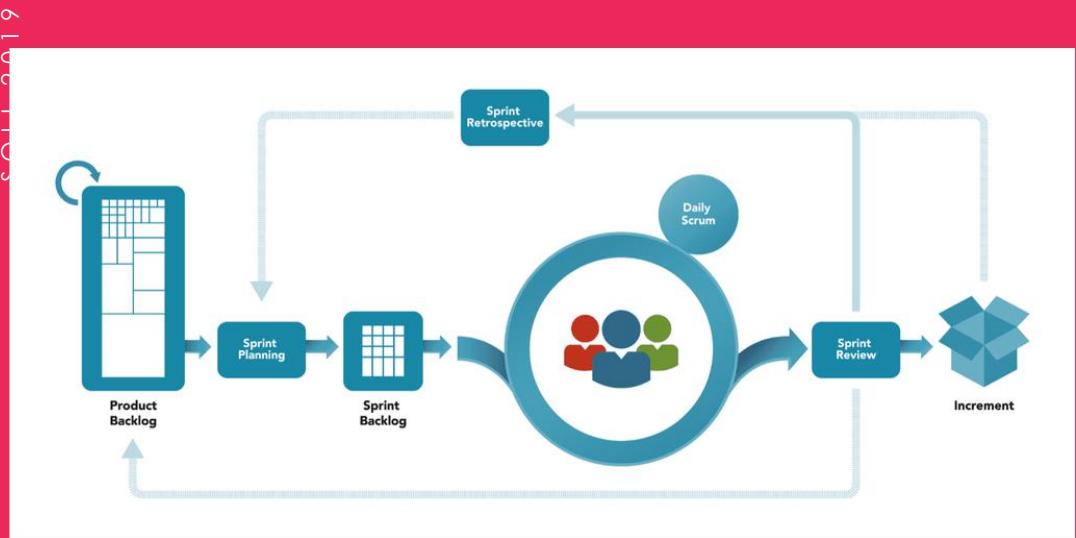
Artefacts	Définition
Backlog Produit (Product Backlog)	Liste d'exigences pour le produit priorisée selon la valeur. Taillées selon Thème > Epic > Fonctionnalités > Story
Backlog d'itération (Sprint Backlog)	Liste d'éléments du Backlog Produit à délivrer en fin d'itération. La façon de travailler (Tâches) est définie collaborativement
Incrément de Produit (Product Increment)	Résultat livré à la fin de l'itération Potentiellement livrable



Supports	Définition / Mission
Management Visuel (Remaining work & Transparency)	Panneaux partagés pour visualiser les activités en cours
Définition du Fini (DoD)	Règles convenues pour considérer qu'un élément est achevé
Courbes (Burndown & Burnup)	Indicateurs pour inspecter et adapter continuellement



SCRUM EN SYNTHÈSE



➤ 3 RÔLES DANS L'ÉQUIPE

Product Owner
Scrum Master
Membre d'équipe (dev team)

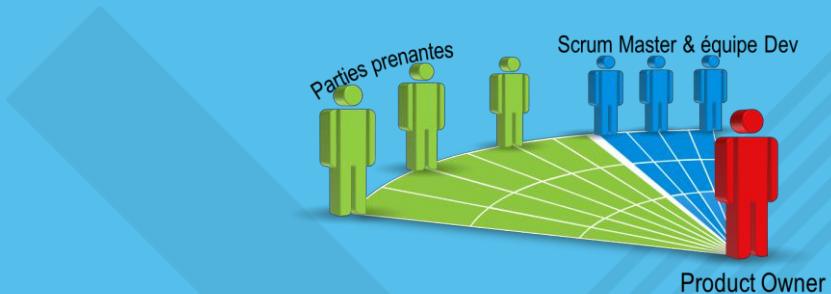
➤ 4 ÉVÈNEMENTS

- Afin de créer de la régularité et de minimiser la nécessité de réunions qui ne sont pas définies dans Scrum.
- Evènements à durée déterminée (**Timebox**)

➤ 3 ARTEFACTS :

- Représentent soit du **travail** soit de la **valeur fournissant** ainsi de la **transparence** et des **opportunités** pour l'**inspection** et l'**adaptation**
- Spécialement conçus pour maximiser la transparence d'informations essentielles afin que tous en aient la même compréhension. »

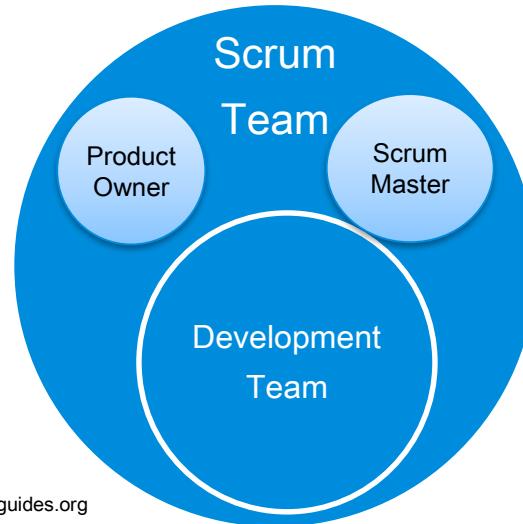
LES RÔLES DE SCRUM



LES RÔLES DE L'ÉQUIPE SCRUM

> L'ÉQUIPE SCRUM COMPREND :

- Le Product Owner
- La Development Team
- Le Scrum Master



Source : scrumguides.org



LE PRODUCT OWNER



> SON RÔLE

Porter la vision du produit à réaliser

Gérer le **backlog** de produit : contenu et priorités (maximisation de la valeur)

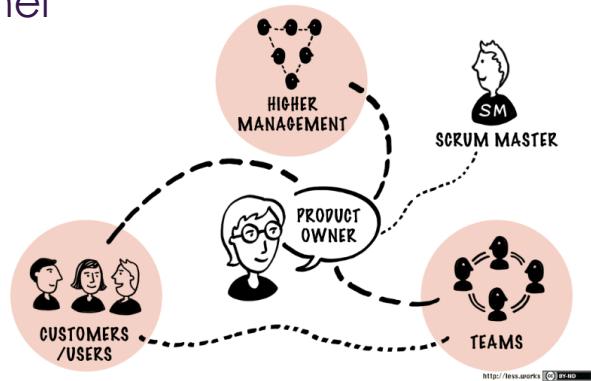
Vu comme le « représentant du client et utilisateurs » (parties prenantes)

Pas un rôle hiérarchique sur l'équipe au sens traditionnel

> SES DROITS

Suivre l'évolution de la somme de travail restant

Décider de **ce qui est fait ou non** dans une version du produit





LE PRODUCT OWNER

➤ ÉTRE AU CONTACT RÉGULIER DES UTILISATEURS

- Les « **fédérer** » autour d'une vision du produit
- Coordonner les recettes si besoin

➤ PARTICIPER AUX ÉVÉNEMENTS

- **Préparation des itérations** (contenu, scénarios de tests/d'acceptation)
- Présent au **Sprint Planning** (vision, priorisation, valeur métier des exigences)
- Présent au **Sprint Review**
- Présent au **Sprint Retrospective**

➤ S'ASSURER QUE

- Le Product Backlog est **visible**, **transparent**, et **clair**
- La 'Development Team' comprend les items du Product Backlog.

➤ ÉTRE DISPONIBLE DURANT TOUTE L'ITÉRATION

- Proche physiquement de l'équipe
- Réactif pour répondre aux questions de l'équipe

➤ DÉCIDER OPÉRATIONNELLEMENT

- Prendre les **décisions métier** (le Quoi), mais pas les décisions techniques (le Comment)
- **Fixer les priorités** métier (capacité et pouvoir de prioriser, de repousser, d'ajouter...)
- **Faire vivre** le Product Backlog (liste des user stories)
- Pouvoir changer d'avis d'une itération à l'autre (en acceptant les conséquences)
- **Valider les éléments** qui lui sont présentés.

L'ÉQUIPE DE DÉVELOPPEMENT



Membre de
l'équipe

➤ COMPOSITION

Tous ceux qui réalisent l'incrément

Entre 3 et 9 personnes

Pluridisciplinaire

Toutes les compétences nécessaires (Développeurs, Testeurs, Analystes, Architectes...)

Multifonctionnelle

Pas de rôles prédéfinis (Tests, Architecte, ...)

Établie par le management

Stable dans le temps

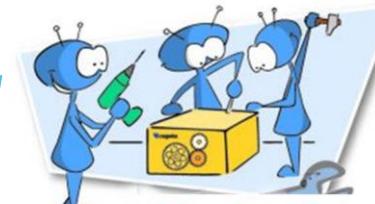
Colocalisée, si possible

➤ FONCTIONNEMENT

Autoorganisée

Autonome

L'ÉQUIPE DE DÉVELOPPEMENT



Membre de
l'équipe

➤ SES DROITS

Être protégée des interventions extérieures pendant l'itération

Définir **comment** réaliser le produit

Refuser des développements trop lourds ou pas assez bien décrits

Faire le nécessaire pour **respecter les engagements**

➤ SES DEVOIRS

Estimer collectivement les tâches avant l'itération

S'engager collectivement sur son estimation et gérer ces problèmes

S'investir sur le métier et **communiquer** avec le PO en toute transparence

Remonter avec courage les problèmes, idées d'amélioration et rétrospectives

Être **sincère** sur son avancement lors de la mêlée quotidienne

Produire en respectant ses engagements



Les 6 qualités des membres d'une équipe agile:

<https://www.stickyminds.com/article/six-behaviors-consider-agile-team>



LE SCRUM MASTER

➤ SON RÔLE

Garant de la **bonne application** de la méthode

Servant Leader, il facilite la réalisation du produit (pour maximiser le ROI et atteindre les objectifs)

Rôle de management mais pas hiérarchique

➤ SES DROITS

Parler à la hiérarchie des interférences empêchant l'équipe d'avancer sereinement

LE SCRUM MASTER



➤ SES DEVOIRS

Animer l'équipe, maintenir un rythme soutenable

Aider l'équipe à résoudre les problèmes

S'assurer que l'équipe Scrum adhère aux valeurs, pratiques et règles de Scrum

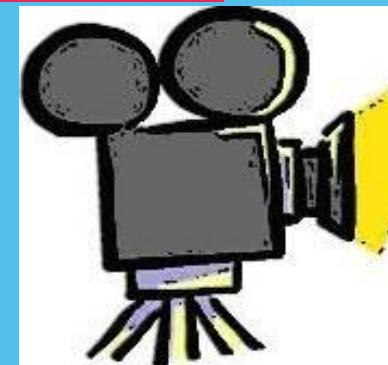
Protéger l'équipe des éléments perturbateurs

Autres équipes, intervenants externes, managers ...

Animer les rétrospectives et s'assurer qu'une démarche d'amélioration continue est suivie et faire respecter les règles

Surveiller l'état du Product Backlog et aider le Product Owner

UNE ÉQUIPE PERFORMANTE



LES PARTIES PRENANTES

➤ LEURS RÔLES

Tous ceux qui ont une **vue** sur le produit sans réellement s'investir dans sa réalisation

MOA, Utilisateurs, Architectes techniques, Managers, Partenaires, autres équipes
Interlocuteurs privilégiés du **Product Owner**

➤ LEURS DROITS

Avoir accès à une vue honnête de l'**avancement** du produit

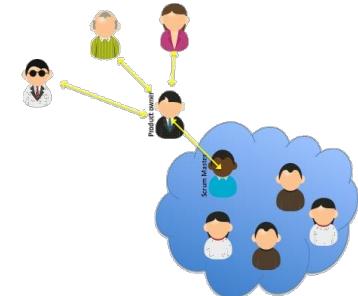
Assister aux **Sprints Review** (démonstration)

➤ LEURS DEVOIRS

Faire des retours (**feedback**)

Fournir les **moyens** à la réalisation

Faire **confiance** à l'équipe et à son mode de fonctionnement



LE SCRUM MASTER

➤ IL SERT LE PRODUCT OWNER

En comprenant la planification à long terme du produit dans un environnement empirique

En enseignant au Product Owner comment créer des éléments Backlog clairs et concis

En trouvant des techniques pour la gestion efficace du Product Backlog

En communiquant clairement la vision, les objectifs et les éléments du Product Backlog à l'équipe de développement

➤ IL SERT L'ÉQUIPE DE DÉVELOPPEMENT

En aidant l'équipe de développement à apprendre comment s'auto-organiser et développer sa transversalité

En enseignant et en menant l'équipe de développement à livrer des produits de haute valeur

En supprimant les obstacles nuisant au progrès de l'équipe de développement

En facilitant des événements Scrum à la demande ou lorsque c'est nécessaire

En accompagnant l'équipe de développement dans les environnements organisationnels dans lesquels Scrum n'est pas encore complètement adopté et compris.

➤ IL SERT LE MANAGEMENT ET LES ÉQUIPES EXTÉRIEURES

En menant et en accompagnant l'organisation dans ses efforts d'adoption de Scrum

En planifiant des mises en œuvre de Scrum au sein de l'organisation

En aidant le personnel et les parties prenantes à comprendre et à adopter Scrum ainsi que le développement empirique de produits

En provoquant le changement qui est susceptible d'accroître la productivité de l'équipe Scrum

En travaillant avec d'autres Scrum Master dans le but d'accroître l'efficacité de l'application de Scrum au sein de l'organisation





AGILE BURGER QUIZZ

S Q L I 2 0 1 9

AGILE BURGER QUIZ

SCRUM MASTER, PRODUCT OWNER OU LES DEUX ?

- 01** IL PARTICIPE FORCEMENT AU DAILY SCRUM MEETING
- 02** IL EST HABILITE A PRIORISER LES US
- 03** IL AIDE A GERER LES OBSTACLES
- 04** IL ANIME LA RETROSPECTIVE
- 05** SOUVENT ON DIT QU'IL NE SERT A RIEN
- 06** S'IL PAYE DES COUPS A L'EQUIPE C'EST MIEUX

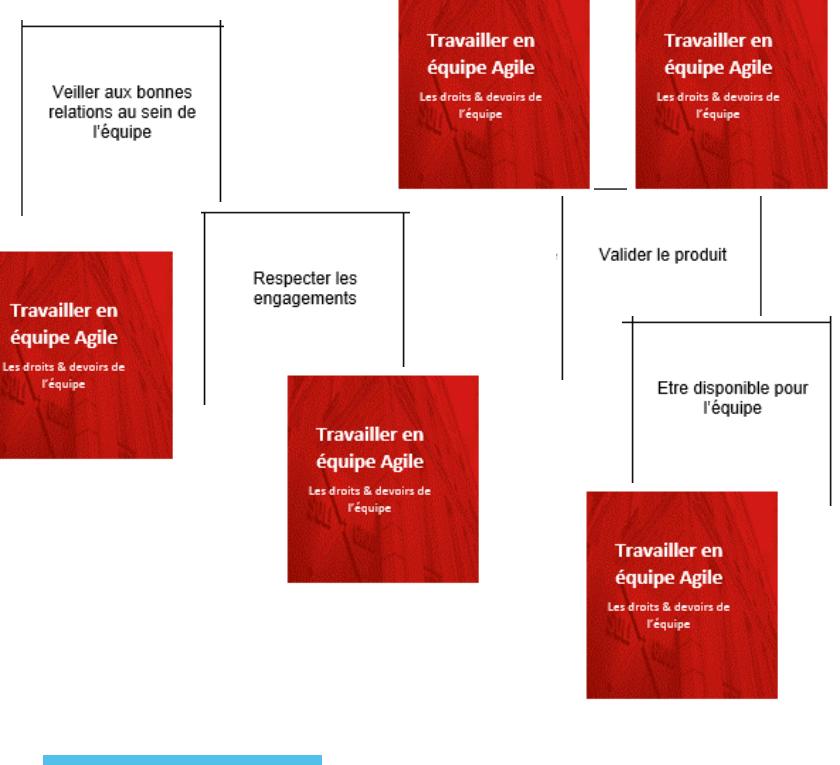
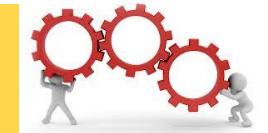


➤ [HTTPS://COACH-AGILE.COM/SERIOUS-GAME-AGILE-BURGER-QUIZ/](https://coach-agile.com/serious-game-agile-burger-quizz/)



LES DROITS ET LES DEVOIRS

S Q L I 2 0 1 9





LES RÔLES DE SCRUM



Protecteur et facilitateur, il favorise le travail en équipe et la collaboration

➤ UN LEADER AU SERVICE DE L'ÉQUIPE PROJET

Membre de l'équipe projet

Facilitateur pour les problèmes non techniques de l'équipe

Fait en sorte que l'équipe soit productive et reste concentrée sur le véritable objectif du projet

Aide l'équipe à travailler de façon autonome et à s'améliorer constamment.

Communique avec le management.



Prend les décisions importantes sur l'orientation du produit mais n'a pas de rôle hiérarchique au sens traditionnel

➤ LE RESPONSABLE DES FONCTIONNALITÉS À DÉVELOPPER

Est le représentant du « métier »

Regroupe les besoins utilisateurs escomptés et les met à disposition régulièrement et progressivement

Définit l'ordre dans lequel ces éléments seront développés

Valide les éléments qui lui sont présentés.



Membre de l'équipe

S'autogérant, ils décident seuls du « comment » arriver au produit final.

➤ IMPLIQUÉS ET RESPONSABILISÉS

Membre de l'équipe projet

Disposent de toutes les compétences nécessaires au développement du produit

S'organisent eux-mêmes, pas de notion de hiérarchie interne

Sont investis avec le pouvoir et l'autorité pour faire ce qu'ils ont à faire

LA CRÉATION ET LA GESTION DU BACKLOG

DÉFINITION

➤ **LISTE PRIORISÉE D'ÉLÉMENTS À PRODUIRE DURANT LE PROJET ET DEVANT CRÉER DE LA VALEUR**

➤ **LES ÉLÉMENTS DU PRODUCT BACKLOG DOIVENT ÊTRE :**

 Décrits de façon homogène (granularité, méthode de description)

 Estimés

 Priorisés

➤ **VIVANT**

 Réordonnancé selon les changements de vision

 Affiné lorsque nécessaire (« Grooming »)

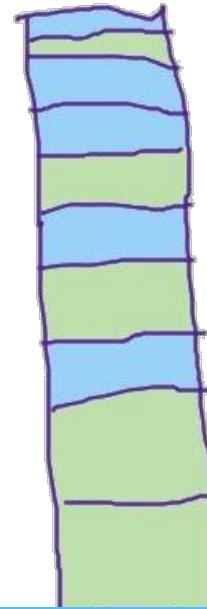
 Modifié = ajout ou suppression d'éléments

➤ **PARTAGÉ ET UTILISABLE**

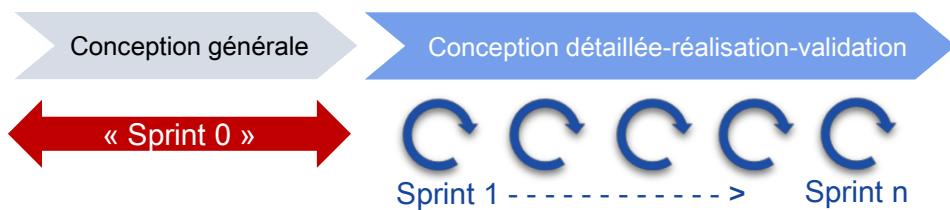
 Les éléments les plus importants sont suffisamment détaillés pour les prochaines itérations (« Prêts »)

COMMENT L'INITIALISER

- LA PÉRIODE D'UN PROJET DÉDIÉE À L'ÉLABORATION DU PRODUCT BACKLOG EST PARFOIS APPELÉE SPRINT 0
- ELABORER LE BACKLOG NE SIGNIFIE PAS QU'IL DOIT ÊTRE EXHAUSTIF
- LE BACKLOG RECENSE L'ENSEMBLE DES ÉLÉMENTS PRIORISÉS MAIS TOUS LES ÉLÉMENTS NE DOIVENT PAS NÉCESSAIREMENT ÊTRE PRIORISÉS AVEC LE MÊME NIVEAU DE DÉTAIL
- LES ÉLÉMENTS CONSIDÉRÉS COMME ÉTANT LES PLUS PRIORITAIRES DOIVENT ÊTRE PLUS DÉTAILLÉS CAR ILS SERONT ADRESSÉS EN PREMIER



Le Product Backlog est de la responsabilité du Product Owner



L'ITÉRATION / SPRINT 0

➤ QUI

Le Product Owner

Le Scrum Master

Les parties prenantes : clients,
utilisateurs (inc. production),
architectes...

Les membres identifiés de
l'équipe de réalisation

➤ COMBIEN DE TEMPS

Le temps nécessaire pour
permettre un démarrage sain
Entre 2 jours et plusieurs
semaines.

➤ OBJECTIFS

Établir la vision du produit

Identifier la liste grossière des
fonctionnalités (Product Backlog)

Les prioriser - Périmètre.

Identifier le coût et la durée

Définir les aspects techniques

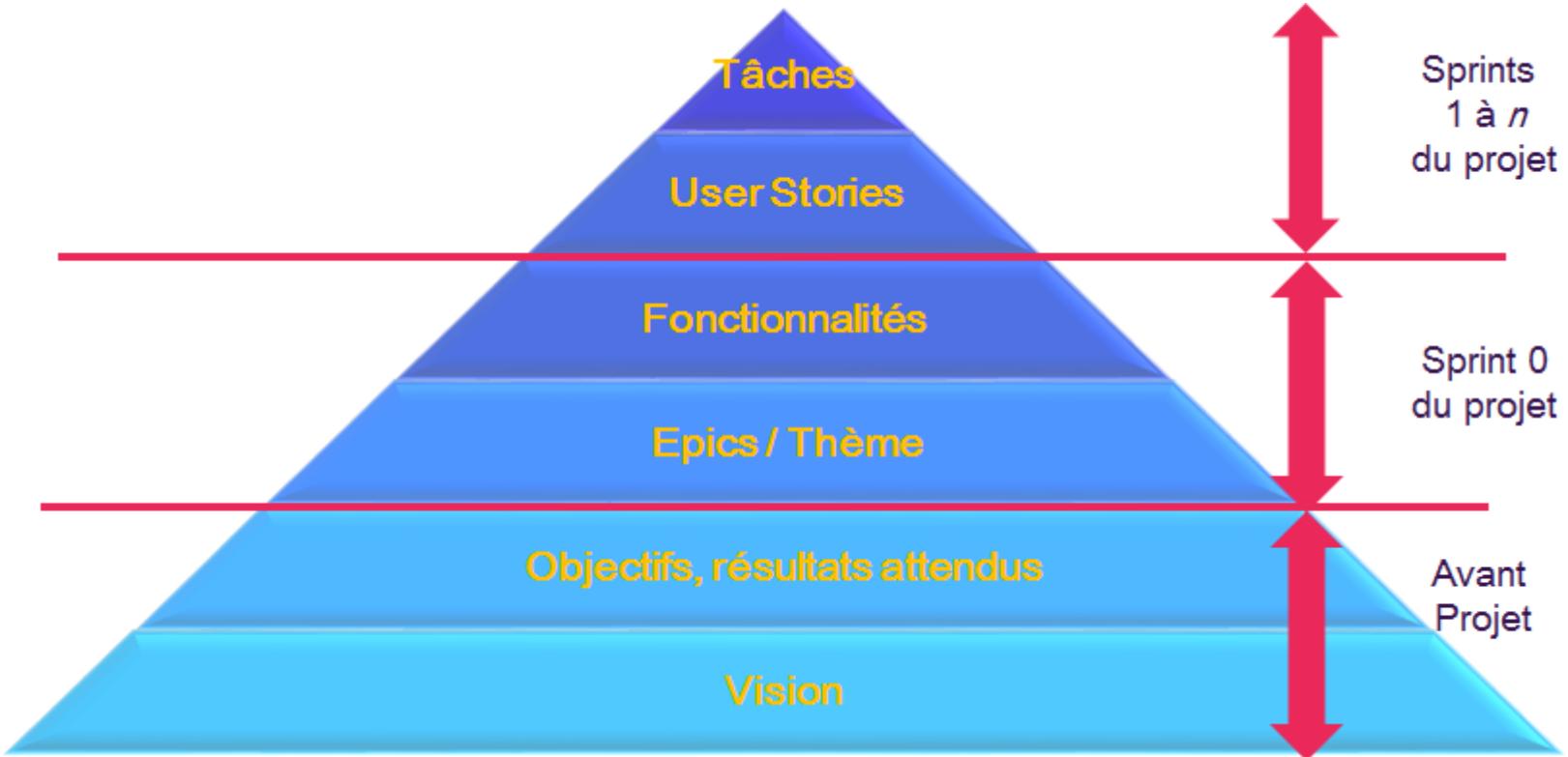
- architecture,
- exploitation,
- environnement et outillage

Définir l'organisation

- durée des itérations.
- plan de release.



DIFFÉRENTS NIVEAUX DE DÉFINITION DU PRODUIT



QUELQUES REPÈRES DE TAILLE

- ON PEUT RÉPARTIR LES ÉLÉMENTS DU PRODUCT BACKLOG SELON DES FONCTIONS, CES DERNIÈRES FAISANT PARTIE D'ÉPOPÉES OU THÈMES.

	Unité de Temps	Échelle pour l'entité
Epopée / Thème	Mois	Produit
Fonction	Semaines	Version / Release
User Story	Jours	Sprint
Tâche	Heures	Sprint

LE BACKLOG PRODUIT CONCRÈTEMENT

➤ ON PEUT ENRICHIR LE BACKLOG DE PRODUIT AVEC DES INFORMATIONS COMPLÉMENTAIRES

Démo : Comment va-t-on pouvoir prouver que l'élément demandé a bien été réalisé ?

- Ce sont les cas de tests relatifs à l'élément

Notes : Informations complémentaires nécessaires à la compréhension de l'élément

- Peut contenir des informations sur comment l'élément va être réalisé

ID : Numérotation unique de l'élément

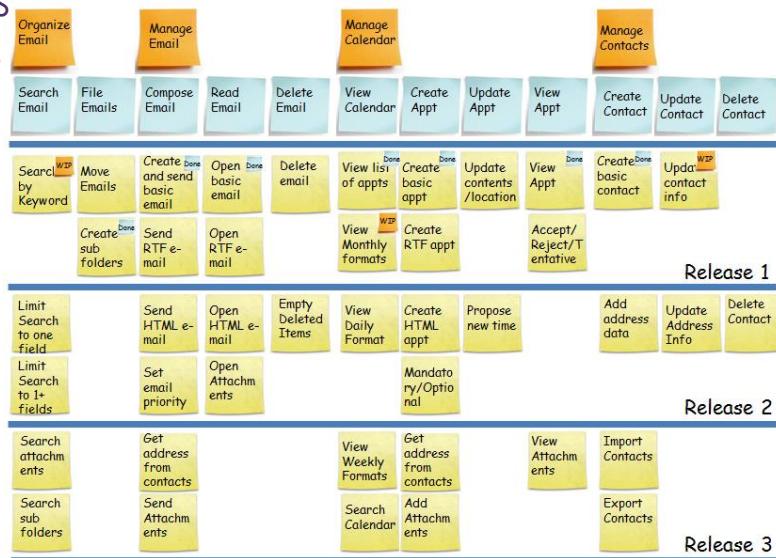
➤ EXEMPLE DE BACKLOG :

ID	Nom	Importance	Démonstration	Notes	Estimation
1	Mettre à jour le consommé en temps réel	30	Authentification, ouvrir compte utilisateur, puis consommer 3 unités et vérifier la mise à jour du solde utilisateur	Optimiser requêtes par PP. La performance système sera testée à S+3	20
2	Voir historique des transactions via interface client	10	Authentification, ouvrir page histo, vérifier nouvelle consommation et infos durée heure	Utiliser une indexation pour limiter volume des requêtes	15

LE STORY MAPPING

➤ UNE VUE COMPLÈTE DU PRODUIT À DÉVELOPPER

1. Identifier les utilisateurs : les acteurs -> Personas ou les Automates (= autres systèmes)
2. Identifier les grands thèmes et fonctions dont ils ont besoin
3. Identifier les User Stories pour chaque fonctions
4. Trier les User Stories pour identifier des Releases
 - Produit Minimum Viable (MVP)
5. Les faire estimer par l'équipe



PARTICIPANTS

- Responsable(s) IT / produit
- Product Owner / Scrum Master
- Scrum Team / Architecte / Sécurité...
- Utilisateurs clés

Exemple : Logiciel de messagerie
 Épopées > Fonctions > Stories
 source : <http://www.agilebuddha.com>



LE STORY MAPPING (1/6)

S Q L I 2 0 1 9



➤ CONTEXTE :

Développer une application mobile sur le trekking permettre aux utilisateurs de trouver des itinéraires appropriés à leurs envies

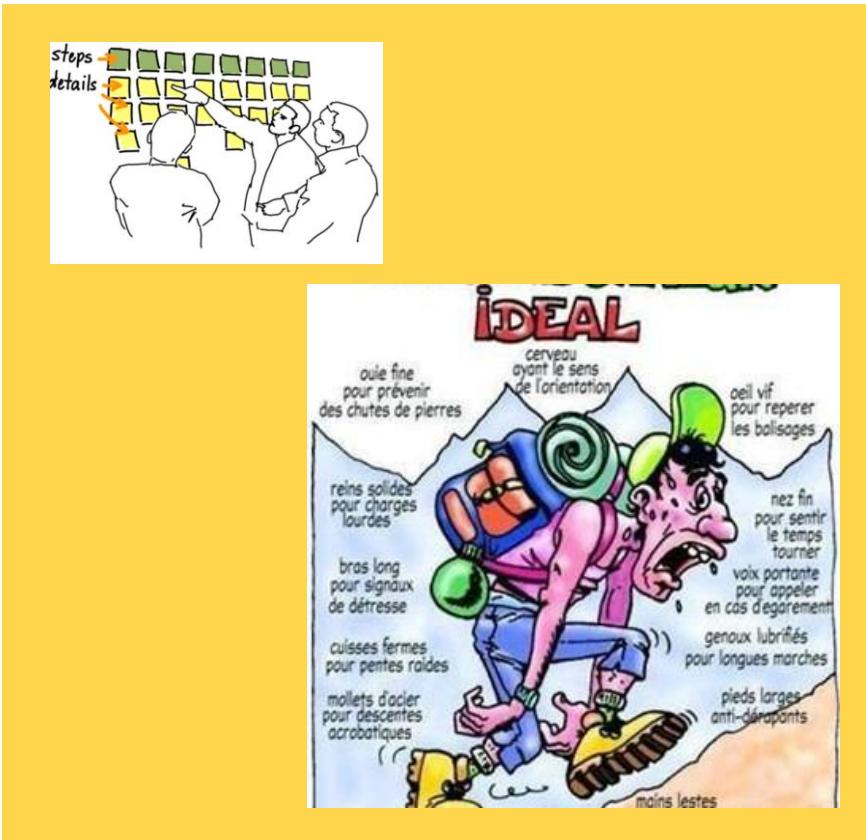
➤ INITIER LA STORY MAPPING DE CETTE APPLICATION

le fil rouge



LE STORY MAPPING (2/6)

SQLI 2019



> LE CLIENT :

Le DSI du conseil départemental de Savoie

> LE PROJET :

Une application mobile sur le trekking

> LE BUT :

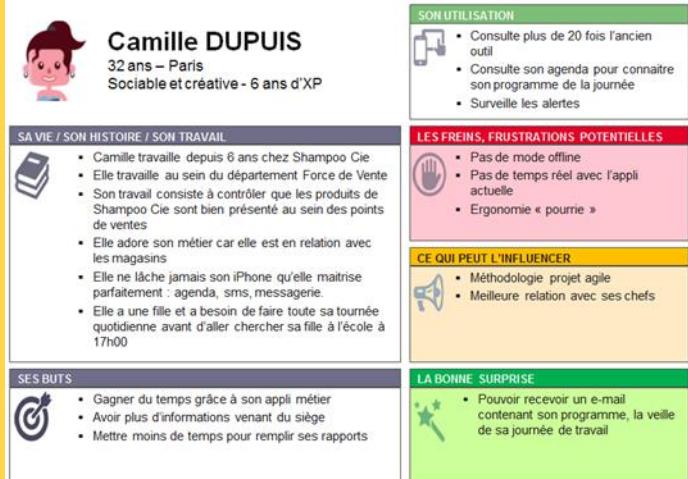
Permettre aux trekkeurs de trouver des itinéraires appropriés à leurs envies

> ÉLÉMENTS COMPLÉMENTAIRES :

L'application devra également proposer d'autres fonctionnalités telles que:

- La recherche d'itinéraires
- La visualisation des itinéraires
- La recherche de « bons plans » : restauration, produits du terroir, hébergement...
- La mise en relation avec des fournisseurs de services : guides, équipement spécialisé....

LE STORY MAPPING (3/6)



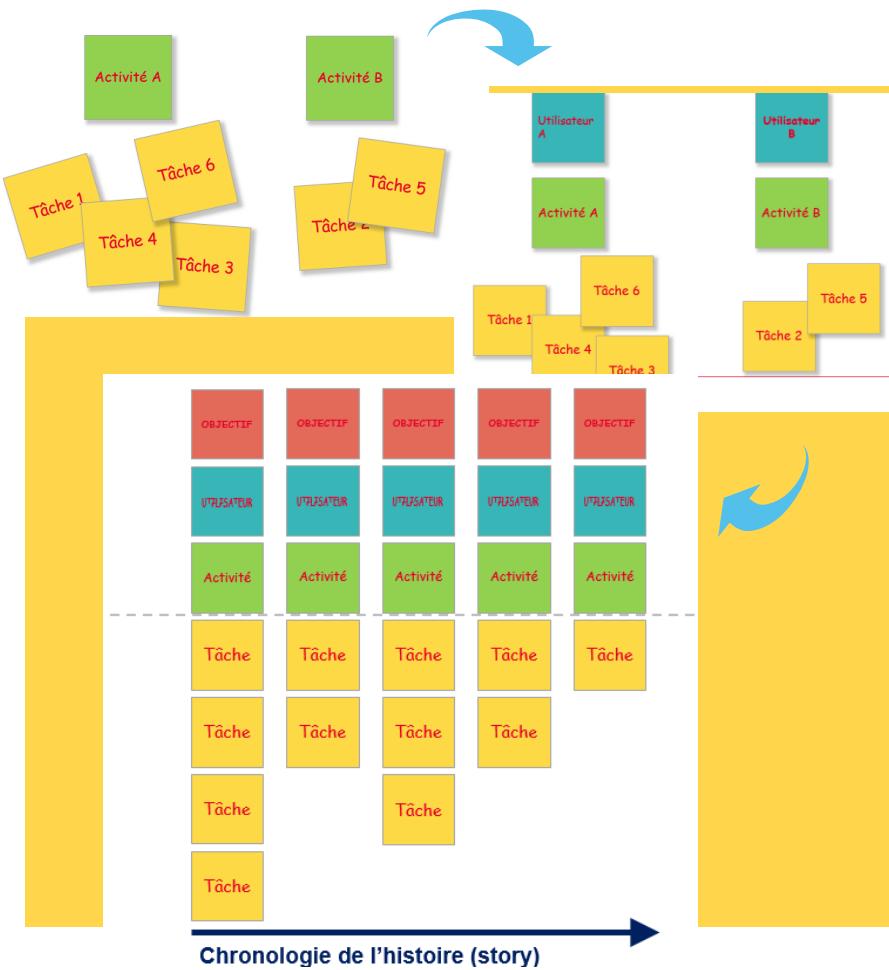
➤ ETAPE 1 : POUR QUI ?

Personnaliser les utilisateurs du Produit/Service pour mieux les comprendre

- Aider l'équipe de réalisation à comprendre le point de vue utilisateur
- Faciliter les discussions sur les usages et les fonctionnalités
- Être en empathie / communiquer
- Rationaliser les décisions et prioriser



LE STORY MAPPING (4/6)



➤ ETAPE 2 : QUOI ?

- Identifier les macro-activités du produit (~les thématiques du produit)
- Puis pour chacune des macro-activités, identifier les grandes étapes à effectuer dans l'ordre chronologique
- En précisant qui les effectuent



LE STORY MAPPING (5/6)

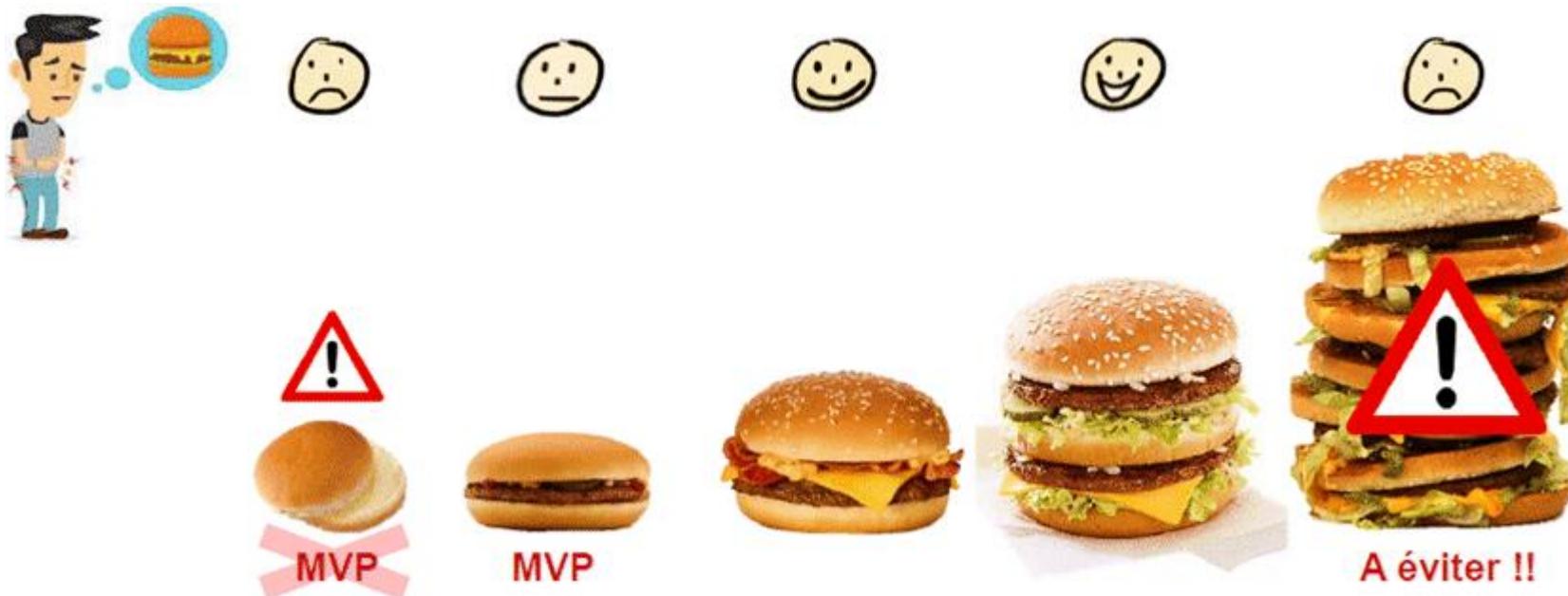


➤ ETAPE 3 : PRIORISATION

- Prioriser les activités par ordre d'importance (valeur) métier
- Identifier ce qui est absolument nécessaire: MVP
- Définir les différentes releases

MINIMUM VIABLE PRODUCT (MVP)

S Q L I 2 0 1 9



MINIMUM VIABLE PRODUCT (MVP)





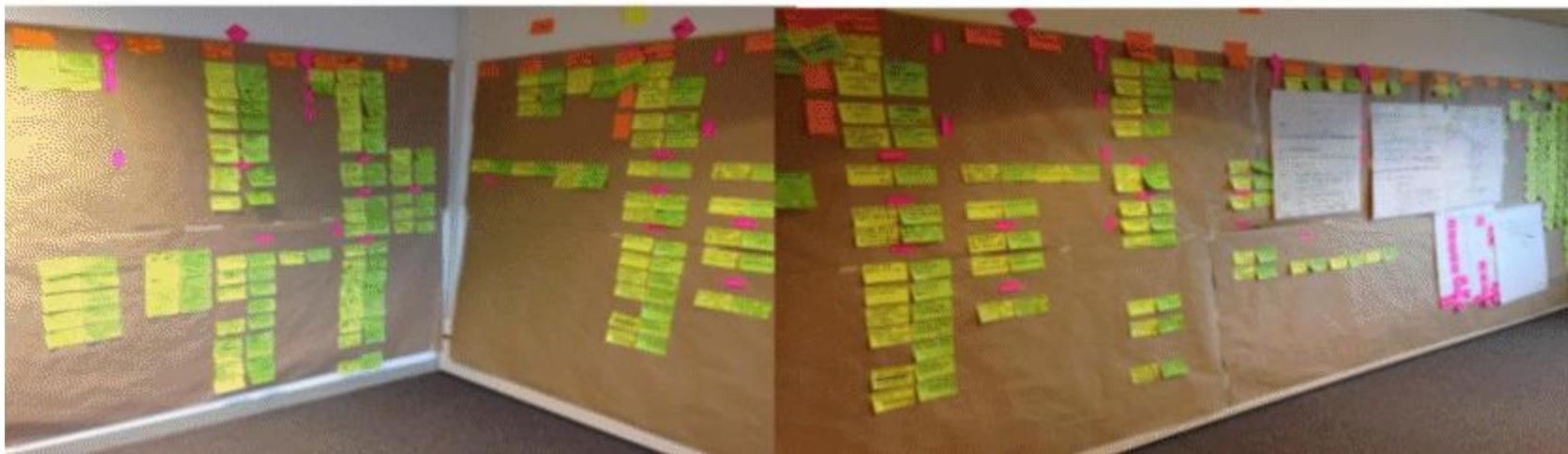
LE STORY MAPPING (6/6)

S Q L I 2 0 1 9



Un Story Mapping prend de la place

- Pour un projet raisonnablement complexe, il faut souvent plusieurs murs



Pause



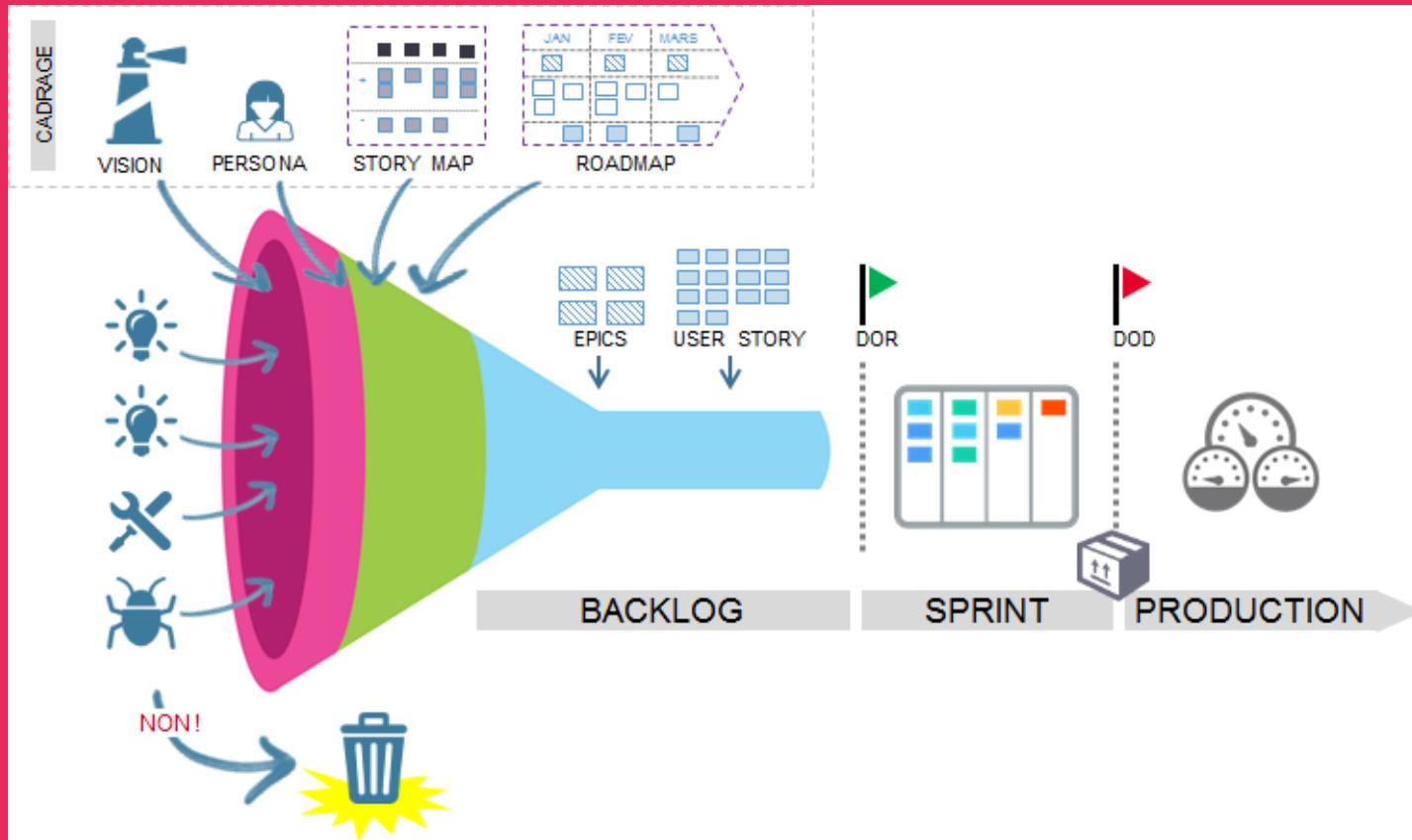
ENERGIZER



LE NŒUD HUMAIN



— AVANCER PAS À PAS ET PRÉCISER AU FUR À MESURE

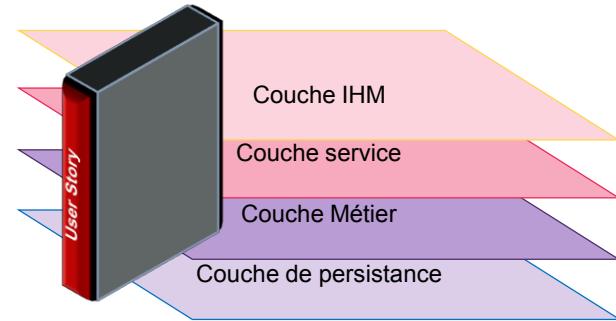
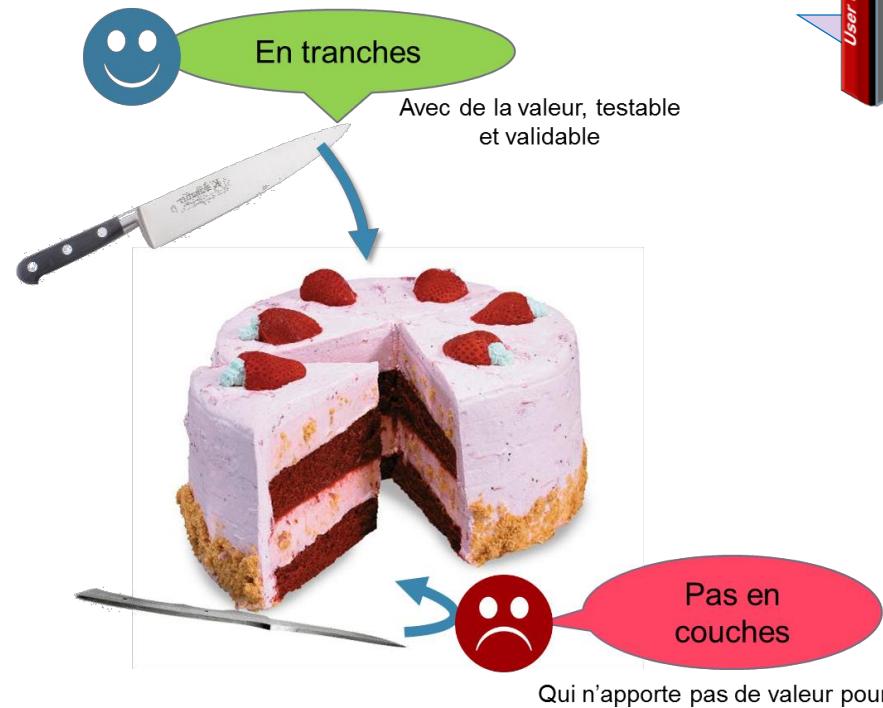


L'ITEM DE BACKLOG

DÉFINITION D'UN ITEM DE BACKLOG

SQLI 2019

- IL DÉFINIT UNE FONCTIONNALITÉ COHÉRENTE traversant toutes les couches fonctionnelles.



DÉFINITION D'UN ITEM DE BACKLOG

➤ DESCRIPTION « STANDARDISÉE » D'UNE USER STORY

En tant que <rôle>, je peux/veux <action> afin de <résultat>

Idéalement, formulée en une ou deux phrases



➤ IMAGINONS LA DESCRIPTION SUIVANTE :

Le produit devra fonctionner avec un moteur à essence, équipé de quatre roues, de pneus gomme, disposer d'un volant, structure en acier.

➤ TOUT LAISSE SUPPOSER QUE LE PRODUIT DÉCRIT EST UNE VOITURE...

➤ ET AVEC CETTE NOUVELLE DESCRIPTION ?

En tant qu'utilisateur, je veux tondre ma pelouse plus rapidement et plus facilement.
En tant qu'utilisateur, je veux pouvoir être confortablement assis lorsque je tonds ma pelouse.

En tant qu'utilisateur, je veux pouvoir parcourir le terrain et manœuvrer.

EXAMPLE DE USER STORY

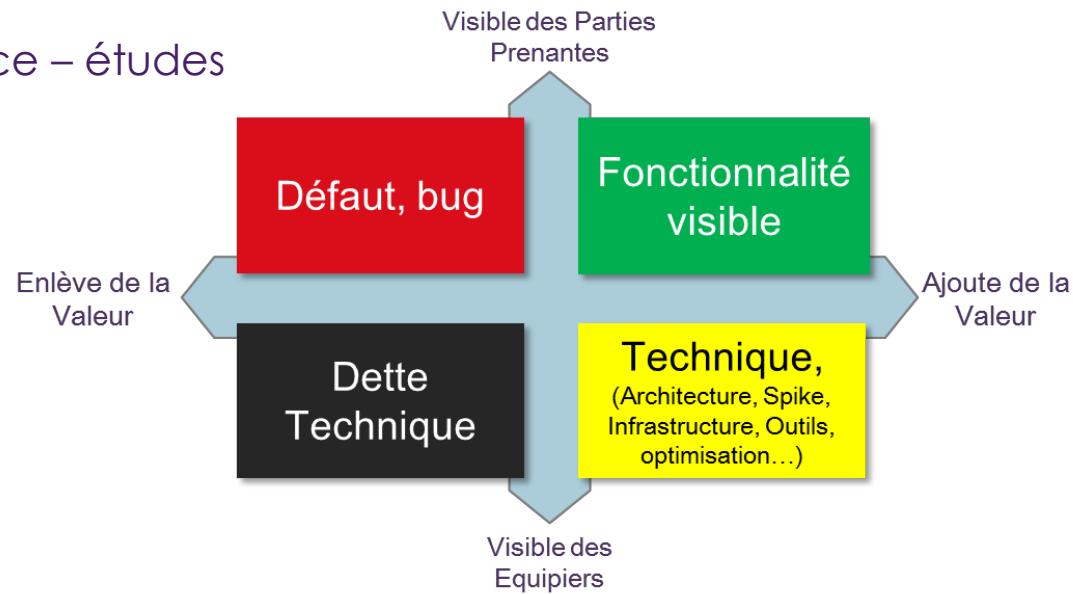
**En tant que <voyageur fréquent>,
je peux <faire une nouvelle
réservation d'un voyage déjà
effectué> afin de <gagner du
temps>**

DÉFINITION D'UN ITEM DE BACKLOG

SQLI 2019

- IL DÉFINIT UNE FONCTIONNALITÉ COHÉRENTE
- OU BIEN REPRÉSENTE

- Les défauts/bugs
- Les améliorations techniques (refactoring, automatisation, outils)
- Les besoins en connaissance – études méthodologiques (spike)



Sources:

- What colours is your backlog, at <http://philippe.kruchten.com/talks>
- L'art du maniement des exigences agiles – Alex Boutin (Scrum Day 2015)

ET LES EXIGENCES NON FONCTIONNELLES ?

➤ INTÉGRÉE À LA VISION DU PRODUIT

- Ex : Conformité à un standard

➤ FORMALISÉE DANS LA STORY « FONCTIONNELLE », COMME CRITÈRE D'ACCEPTATION

- Ex : L'écran doit apparaître en moins de 2 secondes

➤ LISTÉE DANS LA DÉFINITION DU FINI

- Ex : Les 3 langues Eng, Fra & Spa sont disponibles
- Ex : Compatible avec les navigateurs IE, Firefox & Chrome

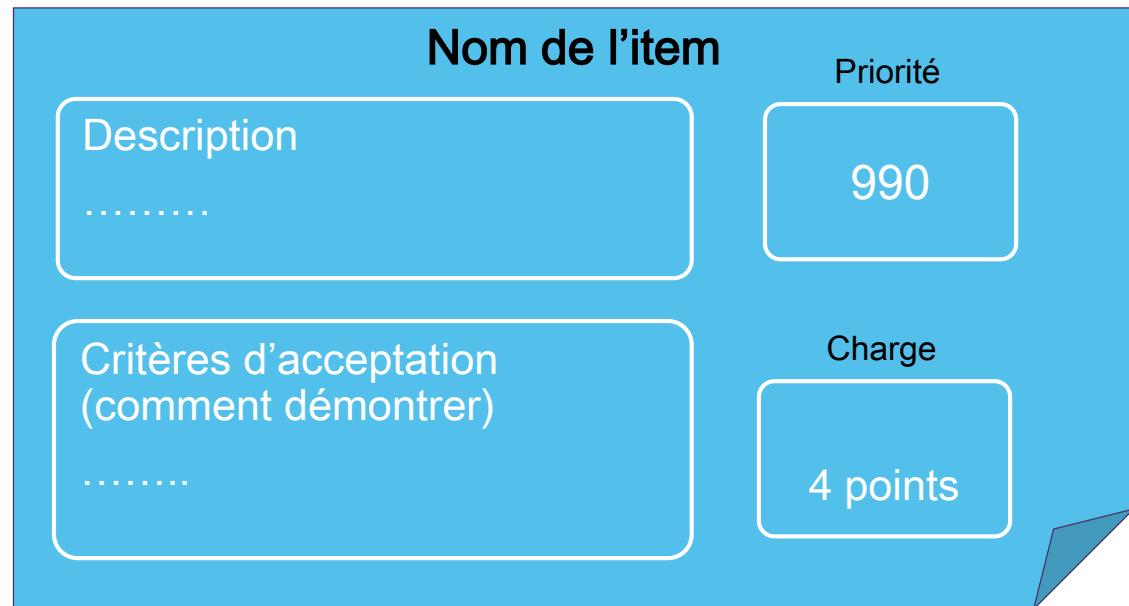
➤ LISTÉE DANS LE BACKLOG, UNIQUEMENT SI ELLE PEUT ÊTRE DÉCLARÉE COMME FINIE

- Ex : Définition de l'interface graphique

REPRÉSENTATION D'UN ITEM

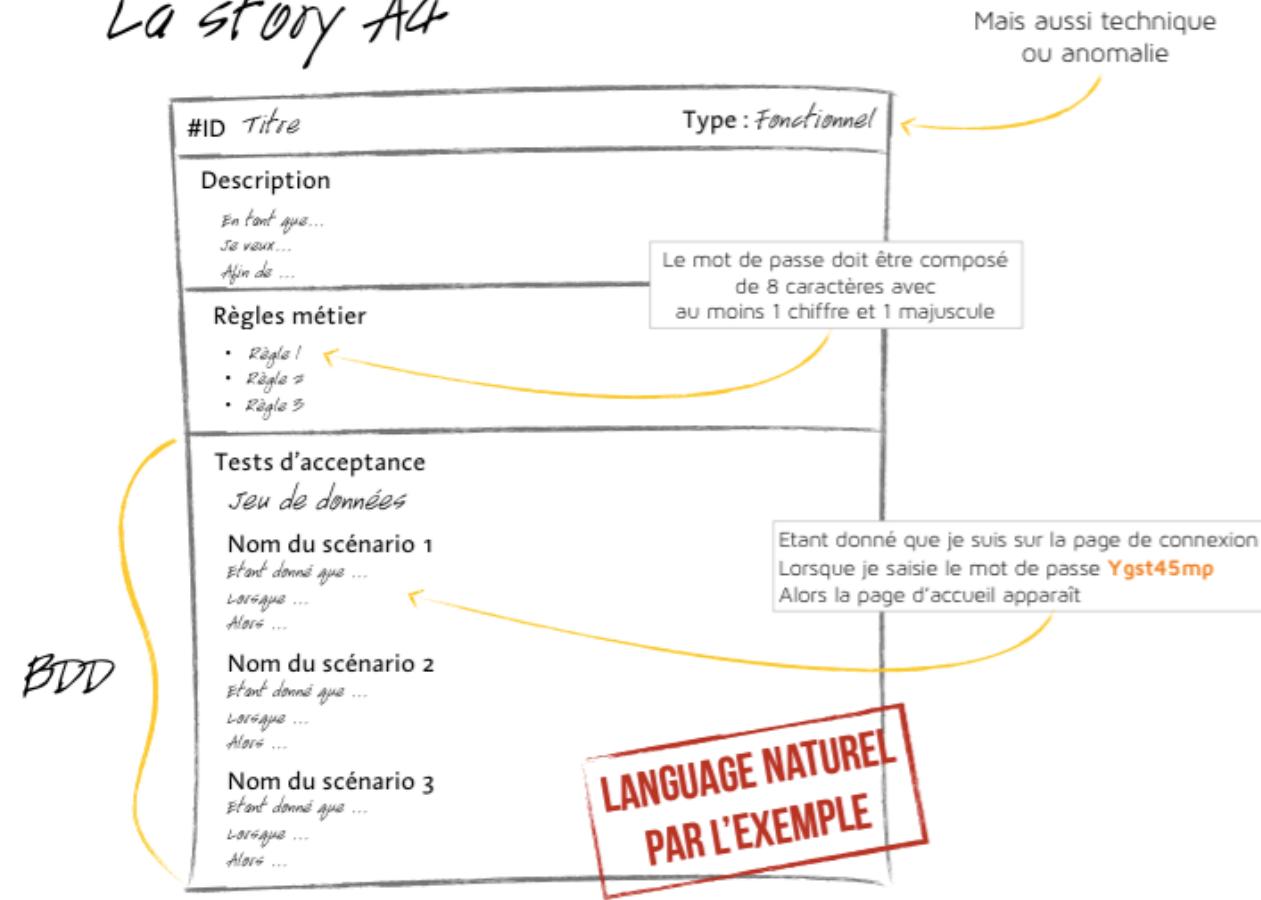
➤ UN ITEM DE BACKLOG EST DÉFINI PAR

- Une description
- Des critères d'acceptation
- Une estimation
- Un ordonnancement



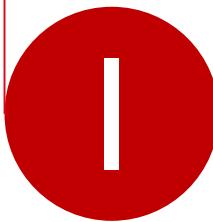
EXEMPLE FORMAT D'USER STORY

La story A4



CRITÈRES DE DÉFINITION D'UNE US

Indépendante pour faciliter son traitement.



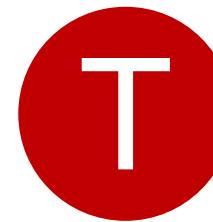
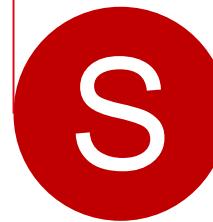
Elle est **négociée**, discutée dès les réunions d'estimation et de planification de l'itération mais aussi tout au long de ce dernier.

Elle est source de **valeur** pour le Client final ou l'utilisateur.



Elle est **estimée** par les équipes de développement; une estimation relative c'est à dire les unes par rapport aux autres, en story points.

Le plus souvent petite (**small**) car susceptible d'être traitée (livrée et testée) par l'équipe sur une seule itération de 2 à 3 semaines.



Une User Story "de qualité" est avant tout **testable**, déjà dans sa forme et surtout dans le sens où les critères d'acceptation sont envisagés d'entrée.

CRITÈRES DE DÉFINITION D'UNE US

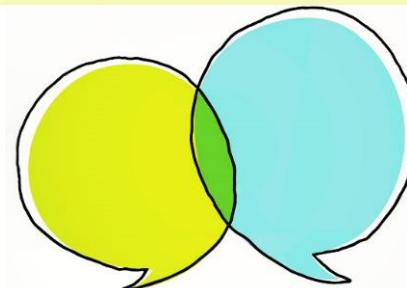
➤ LA RÈGLE DES 3C

CARTE



L'**histoire est courte**, une ou deux phrases et peut être écrite sur une carte 8x13 cm, c'est mieux :
« En tant qu'utilisateur, je veux faire cela »

CONVERSATION



Les détails de l'**histoire sont discutés** par les équipes avec le métier, les ergonomes, les analystes et concepteurs.

CONFIRMATION



L'**histoire est confirmée** par des tests d'acceptation rédigés au même moment que celle-ci, au dos de la carte.

STRUCTURE DU BACKLOG

➤ EXEMPLE 1 : DANS UN TABLEUR

Permet une gestion complète du Backlog (avancement, historique...) mais ne facilite pas le partage d'information

N° Epic	N° User Story	En tant que...	je peux	Afin de...	Comment démontrer	Ordre / Valeur	Taille	Statut
1	0	Administrateur commercial	Alimenter et modifier la grille commerciaux et mettre à disposition des informations de mes services proposés à mes clients	formaliser les accords entre la grille aux différents intervenants			164	
1	1	Administrateur commercial	Créer un nouveau contrat	pouvoir y rattacher les services contractualisés avec un client	je peux enregistrer un libellé de contrat, un client, et une date de fin	1480	1	
1	2	Administrateur commercial	Créer ou modifier un producteur client attaché à un contrat	pouvoir identifier les producteurs attachés à un contrat pouvoir y rattacher les services contractualisés avec un client	je peux enregistrer les coordonnées du producteur ainsi que le code SYSTEME_2 je le lie à un contrat	1475	2	
1	3	Administrateur commercial	Créer ou modifier un atelier attaché à un producteur d'un contrat	pouvoir identifier les ateliers attachés à un producteur pouvoir y rattacher les services contractualisés avec un client	je peux enregistrer les coordonnées de l'atelier lié à un producteur ainsi que son code SYSTEME_2	1470	2	
1	4	Administrateur commercial	Consulter la liste de tous les contrats avec leurs producteurs associés et leurs ateliers	avoir accès aux données existantes, et ainsi éviter de créer des doublons de tiers	Je visualise les contrats et accède à leurs producteurs et ateliers avec les éléments clé (RS / département /ville) si je le veux, je peux voir les informations détaillées du producteur et de l'atelier	820	3	

STRUCTURE DU BACKLOG

➤ EXEMPLE 2 : DANS OUTIL JIRA

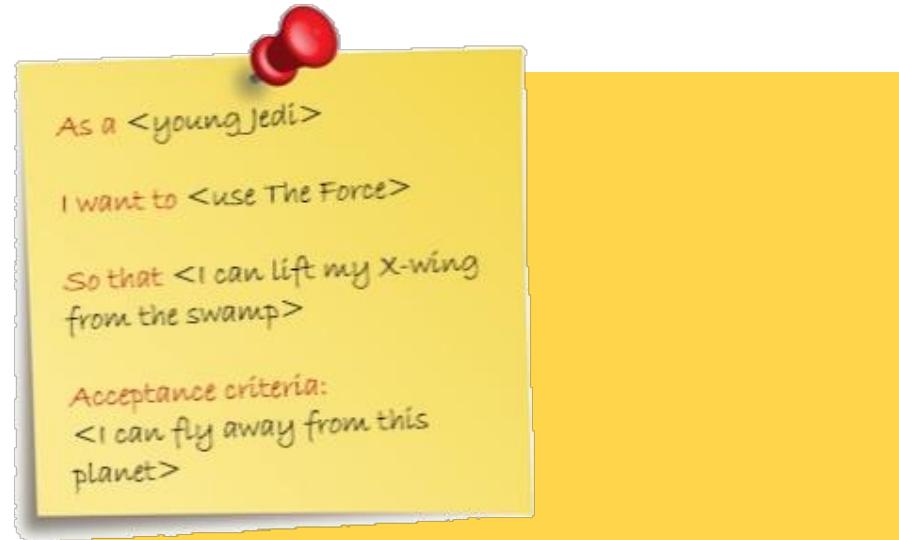
The screenshot shows the Jira RapidBoard interface for the project "PREDIRE". The board is titled "Backlog" and displays a list of user stories. The left sidebar features navigation links for "Tableaux de bord", "Projets", "Demandes", "Capture", "Tableaux", "Git", "Time in Status", "éazyBI", and "Créer". Below these are filters for "FILTRES RAPIDES": "Tâches techniques / Dev", "Tests", "User Stories", "Anomalies", "Mes demandes", and "Récemment mis à jour". The backlog lists 63 user stories under the heading "PREDIRE - Release 2". Each story includes a small icon, its ID, a brief description, and a status indicator. To the right of each story are three buttons: "Visualiser de maniè...", "Paramétrier l'applica...", and "Traitements automa...". The sidebar also includes sections for "VERSIONS", "ÉPICS", and "Tâches".

ID	Description	Status	Actions
PREDIRE-16	Visualiser la contraintes maximale	v1.0.0	Visualiser de maniè..., Paramétrier l'applica..., Traitements automa...
PREDIRE-28	Paramétrer les zones géographiques (par Poste Operateur)		Visualiser de maniè..., Paramétrier l'applica..., Traitements automa...
PREDIRE-35	Exclusion de contrainte		Visualiser de maniè..., Paramétrier l'applica..., Traitements automa...
PREDIRE-32	Contrainte directionnelle		Visualiser de maniè..., Paramétrier l'applica..., Traitements automa...
PREDIRE-33	Priorisation des contraintes : rémanence		Visualiser de maniè..., Paramétrier l'applica..., Traitements automa...
PREDIRE-34	Fenêtre temporelle Temps réel		Visualiser de maniè..., Paramétrier l'applica..., Traitements automa...
PREDIRE-30	Filtres et tris		Visualiser de maniè..., Paramétrier l'applica..., Traitements automa...
PREDIRE-39	Identifier les besoins de coordination		Visualiser de maniè..., Paramétrier l'applica..., Traitements automa...
PREDIRE-49	Accéder au journal de bord en lecture seule		Journal de bord, v1.0.0
PREDIRE-63	Reporting indicateurs		Journal de bord, v1.0.0
PREDIRE-60	Calcul de gravité Transit		Calcul, v1.0.0
PREDIRE-64	Calcul de gravité Tension		Calcul, v1.0.0
PREDIRE-133	Import de l'ensemble des seuils issues des situations réseau		Traitements automa...
PREDIRE-136	Couleur des défauts et contraintes		Visualiser de maniè..., Traitements automa...
PREDIRE-137	Afficher les regroupements de points horaires		Visualiser de maniè..., Traitements automa...
PREDIRE-138	Rafraîchissement automatique		Traitements automa..., Visualiser de maniè...
PREDIRE-139	Visualiser les contraintes par zone géographique (Poste Opérateur)		Visualiser de maniè..., Traitements automa...
PREDIRE-140	Filtre des Contraintes N répétées en N-1		Visualiser de maniè..., Traitements automa...



LES USER STORIES

SQLI 2019



CARTE



L'**histoire est courte**, une ou deux phrases et peut être écrite sur une carte 8x13 cm, c'est mieux : « En tant qu'utilisateur, je veux faire cela »

CONVERSATION



Les détails de l'**histoire sont discutés** par les équipes avec le métier, les ergonomes, les analystes et concepteurs.

CONFIRMATION



L'**histoire est confirmée** par des tests d'acceptation rédigés au même moment que celle-ci, au dos de la carte.

➤ CONTEXTE :

Développer une application mobile sur le trekking permettre aux utilisateurs de trouver des itinéraires appropriés à leurs envies

➤ IDENTIFIER, DANS LA STORY MAP, 3 À 4 ITEMS LES PLUS IMPORTANTS ET LES AFFINER SOUS FORME DE USER STORIES RESPECTANT LA RÈGLE DES 3C

le fil rouge

AFFINAGE / TOILETTAGE BACKLOG

BACKLOG GROOMING / REFINEMENT

➤ OBJECTIFS

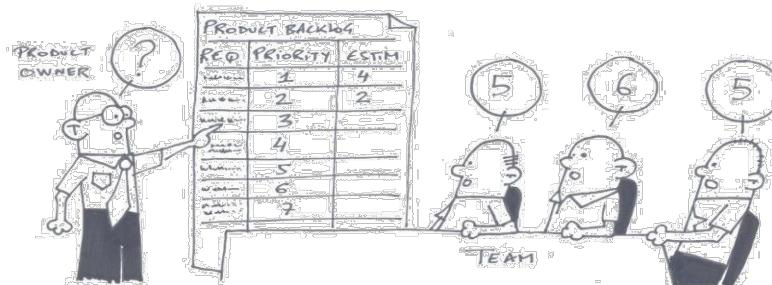
Redécouper et/ou ré-estimer certains éléments
Supprimer les éléments obsolètes

➤ QUAND

Régulièrement (ex 1 heure par semaine) si pas inclus dans la Planification d'Itération
A défaut, avant chaque release

➤ COMMENT

Redécouper et ré-estimer les éléments les plus prioritaires (urgents) et les plus gros



AFFINAGE / TOILETTAGE BACKLOG

S Q L I 2 0 1 9

➤ 10 STRATÉGIES DE DÉCOMPOSITION DES USERS STORIES

Par étapes d'un Workflow

Par scénario

Par séquence dans un scénario

Par opérations (CRUD)
(create, Retrieve,

Par format ou type de données

Par type d'entrée, sortie ou configuration

Par Persona ou rôle

Par niveau de connaissance

Par niveau de complexité

Par niveau de qualité attendu

LES RESPONSABILITÉS

- **LE PRODUCT OWNER A LA RESPONSABILITÉ DE LA PRIORISATION DES ÉLÉMENTS.**
 - Il peut changer l'ordre des éléments
 - Il peut modifier le contenu du Product Backlog en cours de projet
- **C'EST L'ÉQUIPE DE DÉVELOPPEMENT QUI ESTIME LA COMPLEXITÉ RELATIVE DE CHAQUE ÉLÉMENT**

Les estimations pourront être ajustées en cours de projet suivant l'évolution des développements et du besoin.

En tant qu'équipe de dev, il vous faut comprendre la finalité de ce que vous allez développer. Challengez le Product Owner !

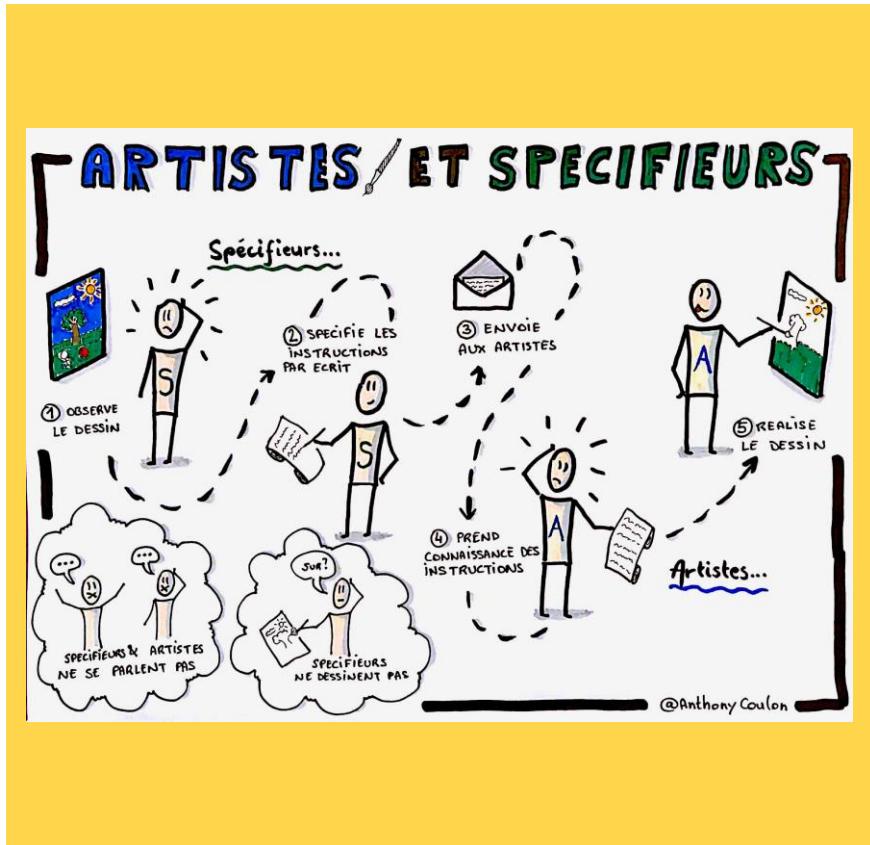


LA COMMUNICATION AU SEIN DE L'ÉQUIPE AGILE



ARTISTES ET SPÉCIFIEURS (1/3)

S Q L I 2 0 1 9

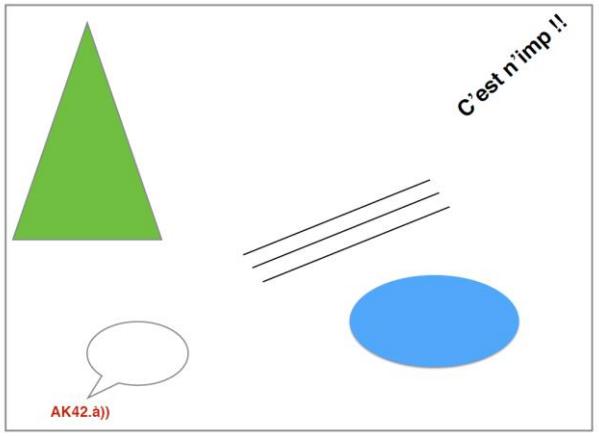


- 3 ÉQUIPES
- 1 DESSIN IMPOSÉ
- 2 RÔLES

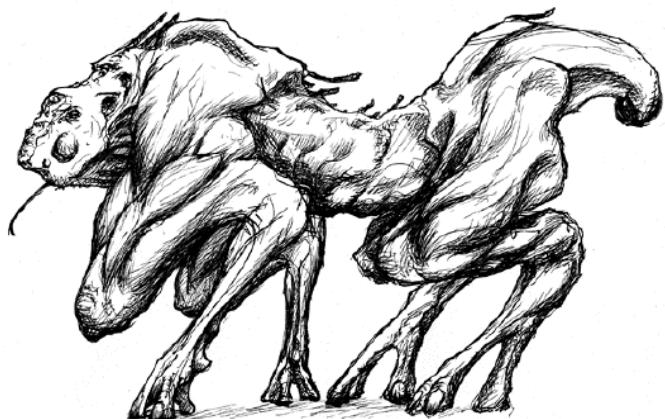
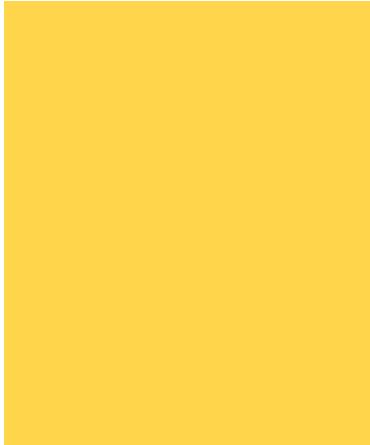


ARTISTES ET SPÉCIFIEURS (2/3)

S Q L I 2 0 1 9



DEBRIEF

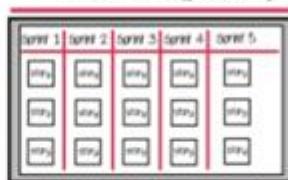
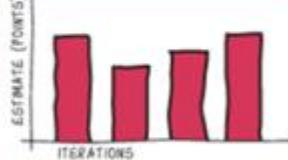
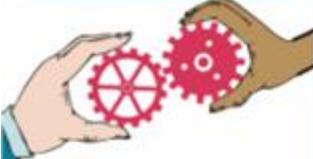
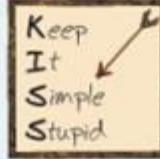




ARTISTES ET SPÉCIFIEURS (3/3)

Knowledge TRAIN

The 12 agile principles*

1 Satisfy the customer 	Welcome change 	Deliver frequently 	4 Work together 
5 Trust and support 	Face-to-face conversation 	Working software 	8 Sustainable development 
9 Continuous attention 	10 Maintain simplicity 	11 Self-organizing teams 	12 Reflect and adjust 

Copyright © 2015. Knowledge Train Limited. www.knowledgetrain.co.uk. Used with permission.

➤ RETOUR SUR LE
MANIFESTE AGILE

DEBRIEF

COMMUNIQUER AU SEIN DE L'ÉQUIPE

➤ LE QUESTIONNEMENT

Fermée, une seule réponse possible
(Oui/Non)

Ouverte, une réponse développée
Retour, renvoie la question à
l'interlocuteur

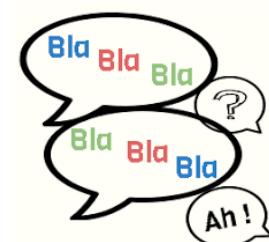
Reflet suscite l'explication d'un fait



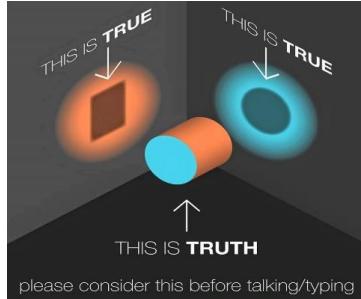
➤ LA REFORMULATION

Reformuler une phrase n'est pas répéter ce que l'autre a dit : Il s'agit de vérifier et de valider que l'on s'est bien compris

La reformulation améliore l'écoute, met en valeur, permet de nuancer, de relever un excès de langage



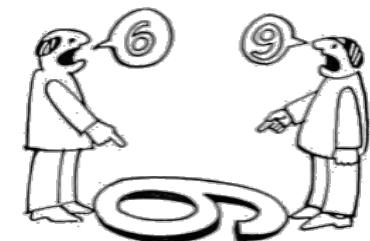
COMMUNIQUER AU SEIN DE L'ÉQUIPE



➤ L'EMPATHIE

Montrer votre empathie en restant centré sur votre interlocuteur, faites abstraction de votre point de vue. Essayer de comprendre son ressenti, son comportement, son analyse et de ses propres mots

Ne pas projeter sa propre vision, ne pas juger l'autre
Accepter la différence de vision



➤ L'ÉCOUTE ACTIVE

Nos attitudes démontrent que nous n'écoutons pas toujours, nous sommes déjà dans une posture de jugement, d'évaluation et interprétation.

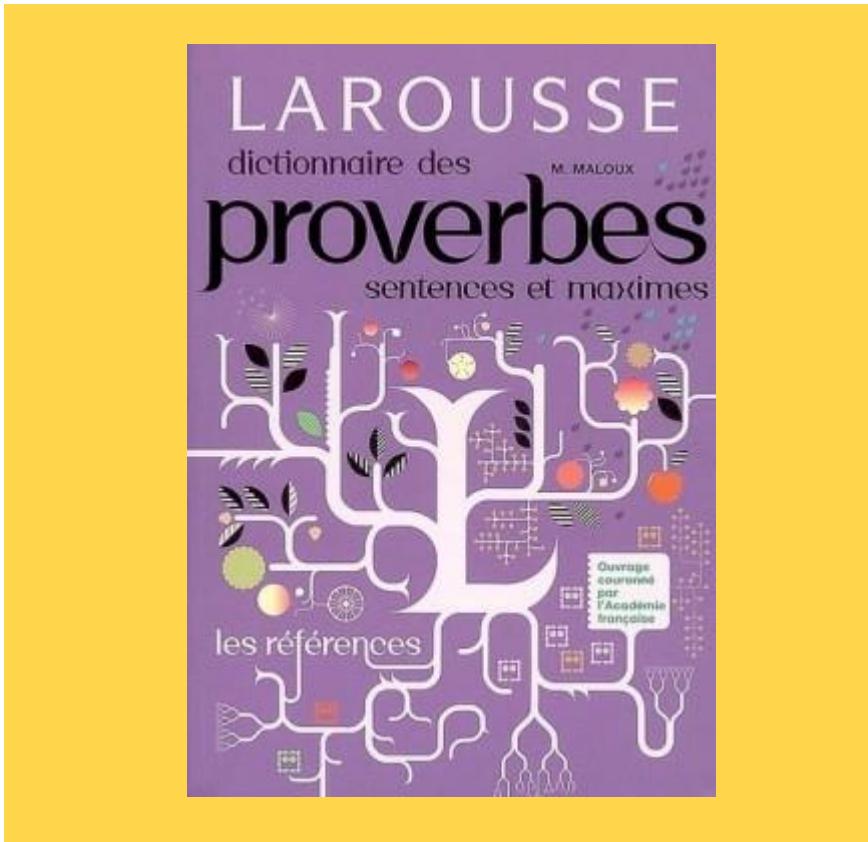
Exemple d'attitudes prouvant le manque d'écoute :

- Vous n'avez pas bien compris
- C'est une bonne idée, mais
- J'ai vécu une situation identique, je peux vous dire que...
- Excusez moi de vous couper la parole
- Si j'étais vous



L'USAGE DES QUESTIONS

S Q L I 2 0 1 9



LE JEU DU PROVERBE

FIN DE LA 1ÈRE JOURNÉE

➤ QUESTIONS



➤ FEEDBACK – LE ROTI



"**Excellente.** Voila une super réunion dont je vais bénéficier. Ça valait bien plus que le temps qu'on y a passé"



"**Bonne.** Voilà une réunion au dessus de la moyenne. J'ai gagné plus que le temps que j'y ai passé "



"**Juste Moyenne.** Je n'ai pas perdu mon temps, sans plus"



"**Utile** mais ça ne valait pas à 100% le temps que j'y ai passé. J'ai donc perdu du temps "



"**Inutile.** Je n'ai rien gagné, rien appris. J'ai vraiment perdu 1 heure ! "

Jean Claude Grosjean - www.qualitystreet.fr

C'EST
TOUT
POUR
AUJOURD'
HUI.