

# Printemps Agile Caen - 2017



Rémy  
Génin



# Printemps Agile – Merci aux sponsors !



RÉGION  
NORMANDIE



NATIXIS

INTERÉPARGNE



Calvados



LE DÉPARTEMENT



construire ensemble durablement



École Universitaire  
de Management



UNIVERSITÉ  
CAEN  
NORMANDIE



Caen Normandie  
DÉVELOPPEMENT



BASSE-NORMANDIE



Crédit Mutuel  
Normandie



Saagie



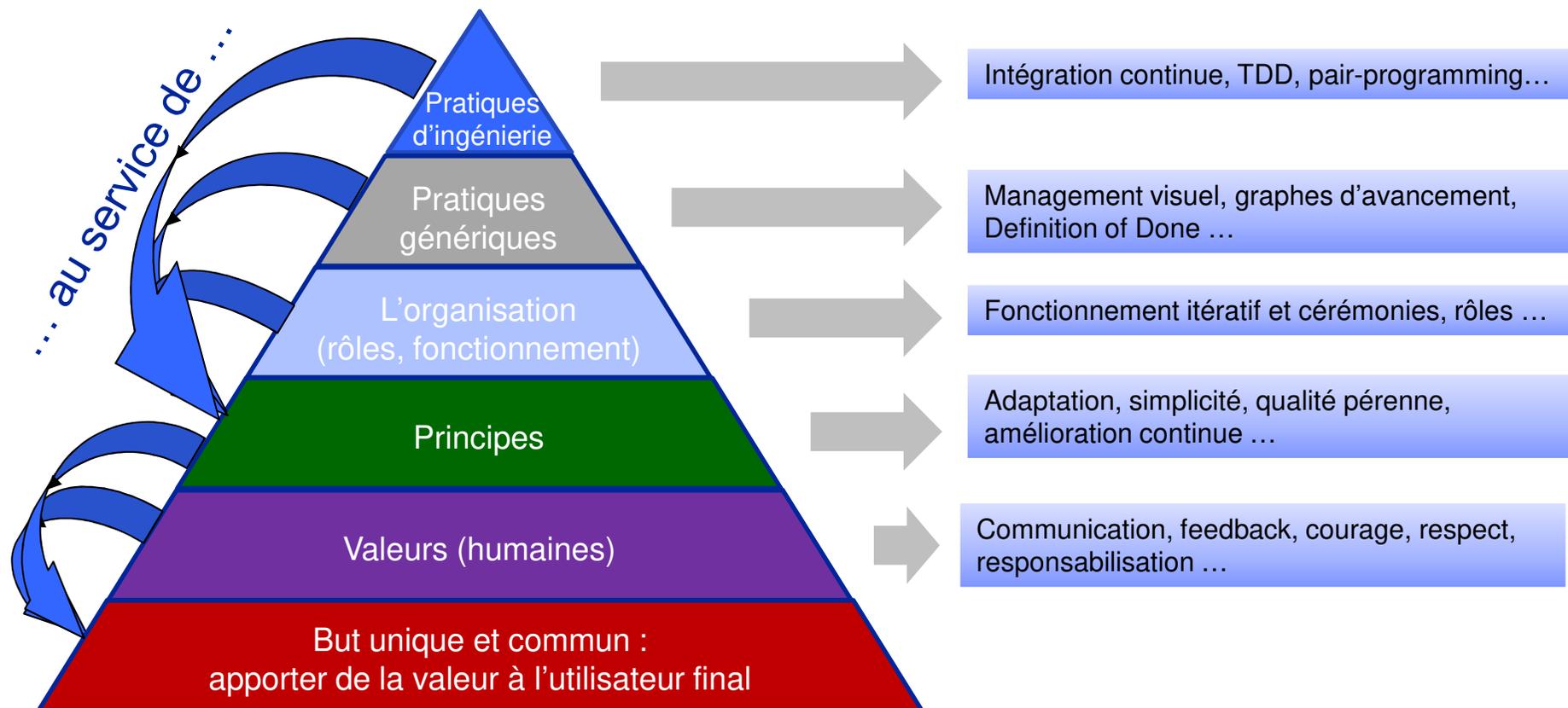
IMAGILE  
MH communication



**UNE VISION D'ENSEMBLE  
DE L'AGILITÉ**

# La Pyramide agile

L'Agilité est une méthode de gestion de projet centrée autour de l'apport de valeur à l'utilisateur final. Ses composants sont imbriqués et en cohérence.



# le Manifeste Agile

**En 2001, 17 pionniers de l'Agilité mettent en lumière  
4 + 12 principes :**

**« Cherchant par la pratique et l'entraide de meilleures façons de  
développer des logiciels, nous avons trouvé que »**

**il y a plus de valeur dans ... que dans ...**

les individus & leurs interactions  
un logiciel opérationnel  
la collaboration avec le client  
l'adaptation au changement

**HUMAIN**

**si nous accordons une valeur aux éléments à droite,  
ceux de gauche nous paraissent avoir  
une valeur encore plus grande**

<http://agilemanifesto.org/iso/fr/manifesto.html>

# le Manifeste Agile – 12 principes de plus

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

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

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

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

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**.

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.

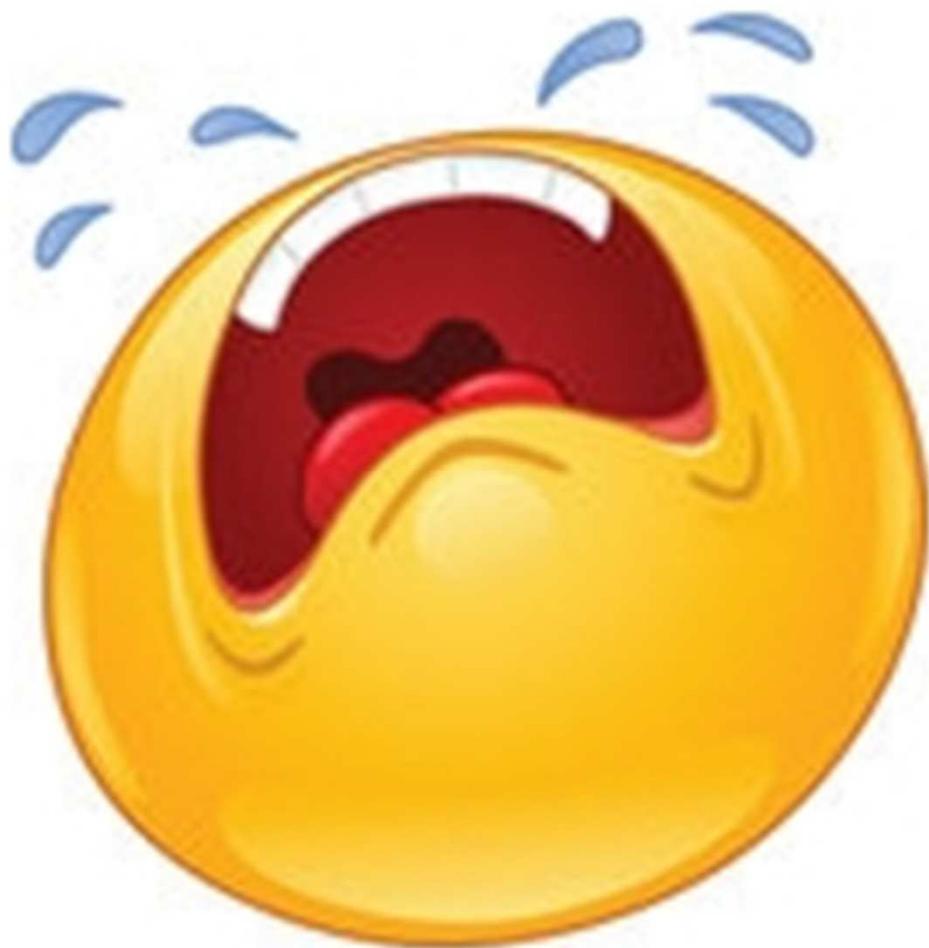
Les meilleures architectures, spécifications et conceptions émergent d'**équipes auto-organisées**.

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



## **LES JOIES DU CYCLE EN « V »**

# Le paradoxe du cycle en V / Waterfall



Scope  
fonctionnel



Devis  
estimations



Budget  
+ Délai



Contrat



Premiers  
changements  
(et jusqu'à 60%)

Infos  
détaillées

# La question des changements

Comment éviter  
de figer ce qui va  
changer ?



# La question des personnes

- **A leur arrivée, les collaborateurs ont envie**
  - de bien faire
  - d'apporter leurs compétences
  - leurs qualités, ce qu'ils sont

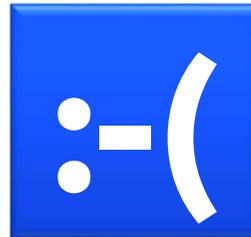
- **En contrepartie, ils souhaitent**
  - que leur travail, leur contribution soit reconnue comme apportant de la valeur
  - avoir une place légitime au sein de l'équipe

**MAIS ...**

**Personnel = "ressources"**

**Les "ressources" doivent être interchangeables**

**Les compétences doivent être dans un document**



**Obéir au règlement intérieur**

**Méfiance & sanctions possibles**

**Pression**

**Le contrat est formel : pas le droit à l'erreur**

**Organisation = Suivi strict des procédures**

**Engagement pris par le manager / commercial sans consulter ceux qui vont réaliser**

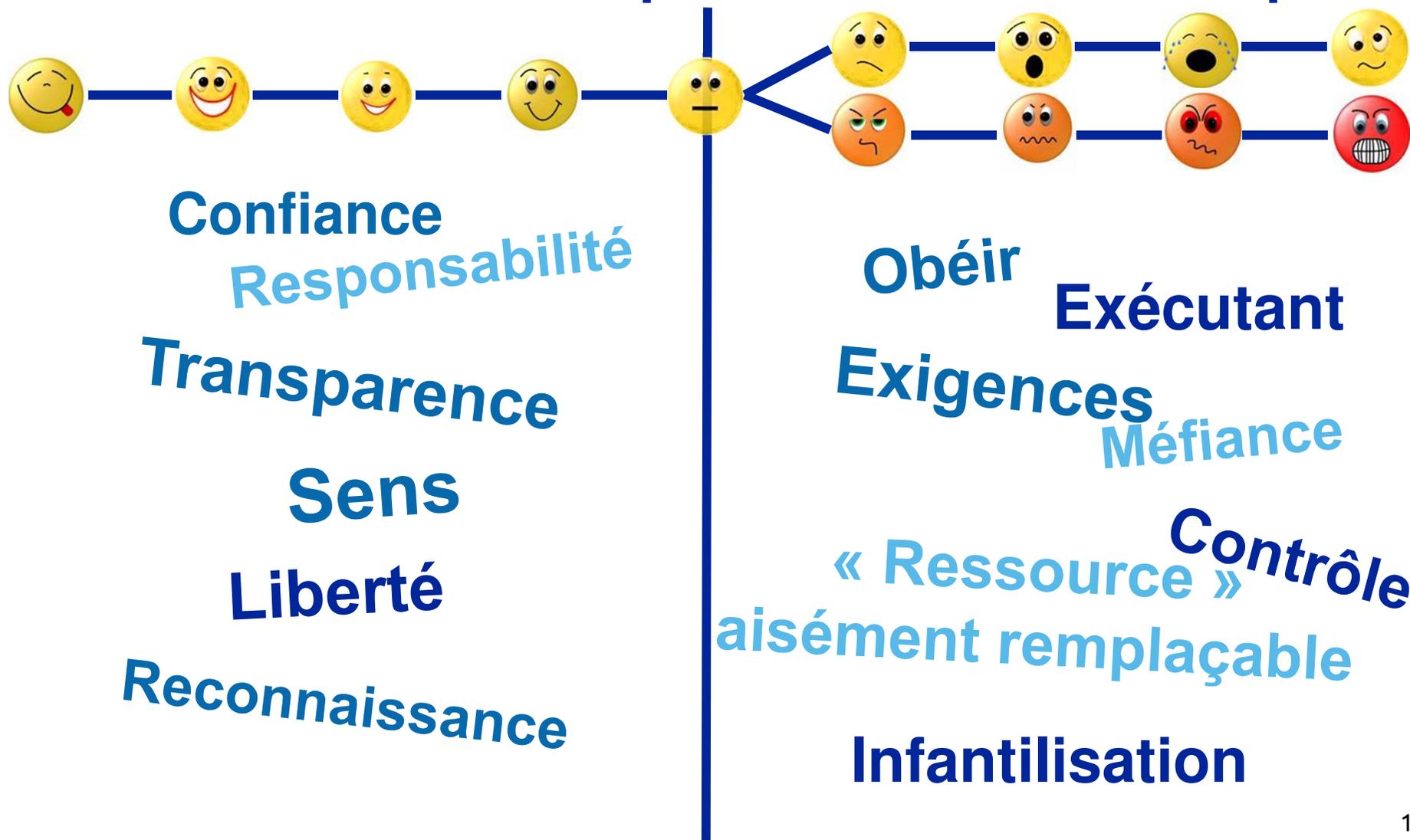
**Réorganisations trop fréquentes = déstructuration, perte de sens, peur, stress**



**PETIT DÉTOUR  
PAR LA SYSTÉMIQUE**

# C'est le système qui fait l'individu

L'environnement et le style de management conditionnent le comportement et l'état d'esprit



# Le système fait l'individu

L'environnement et le style de management conditionnent le comportement et l'état d'esprit



La MEME PERSONNE peut être



ou



C'est le **SYSTÈME** dans lequel elle va interagir qui détermine si elle produira à 20%, 50%, 100%, 150%, 200% de son rendement nominal



**ALORS ...  
QUE PROPOSE L'AGILITÉ ?**

# Objectif 1<sup>er</sup> : satisfaire en étant utile !

- **Notre plus haute priorité**  
(donc notre objectif principal, censé être atteignable)
  
- **est de satisfaire le client**  
(le client d'abord, pas le contrat)
  
- **avec des fonctionnalités à grande valeur ajoutée.**  
(apporter quelque chose d'utile, et même de très utile !)

# Qualité et pérennité : indispensables !

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

**Donc ...**



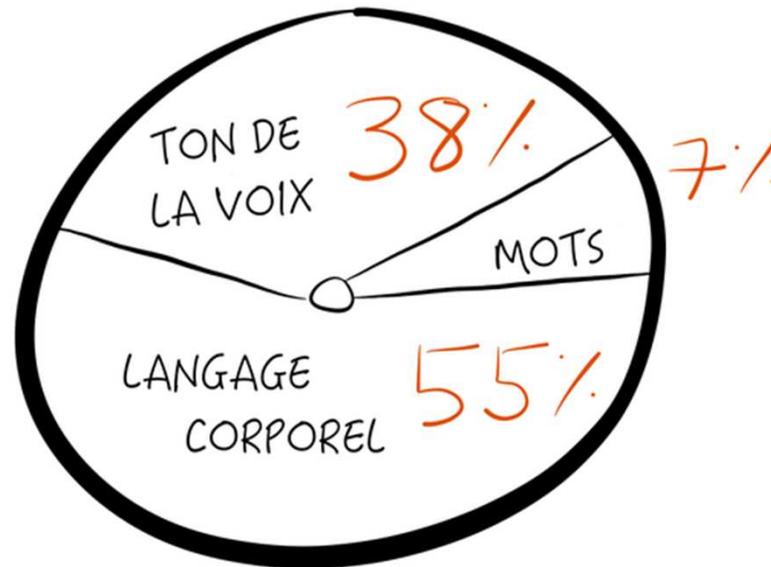
**grande qualité**



**industrialisation robuste  
pour une valeur pérenne**

# Dialogue en face à face.

- 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.



# Valeurs Agiles : officiellement humaines



**Bon sens**



**Communication & Feedback**



**Créativité & Innovation**



**Interactions  
Collaboration**



**Courage**



**Droit à l'erreur sans sanction pour apprendre et s'améliorer**

**F.A.I.L.**

First Attempt In Learning



**Confiance**



**Responsabilité**



**Liberté sur la façon d'honorer la responsabilité**



**Respect**



**Vérité transparence**

- 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.

### Maintient de la qualité, et de l'efficacité, donc

- Satisfaction pérennisée du client
  - Réduction des coûts à moyen et long terme
- > bon pour les individus  
-> bon pour l'entreprise



**CONCLUSION :  
L'AGILITÉ MISE GROS SUR L'HUMAIN !**



**PAS DE PHILANTHROPIE, PAS DE BISOUMOURS,  
SEULEMENT DU GAGNANT/GAGNANT !**



**DES MÉTHODES ÉPROUVÉES  
UTILISANT  
DES PRATIQUES PRAGMATIQUES**

# Une démarche itérative et incrémentale

- **Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.**
- **Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.**

# Une démarche itérative et incrémentale

## ■ **Itération : "boite de temps"**

- de 1 à 4 semaines
- de durée fixe
- lorsqu'une itération se termine, une nouvelle commence
- dans la méthode « Scrum », une itération s'appelle un « sprint »

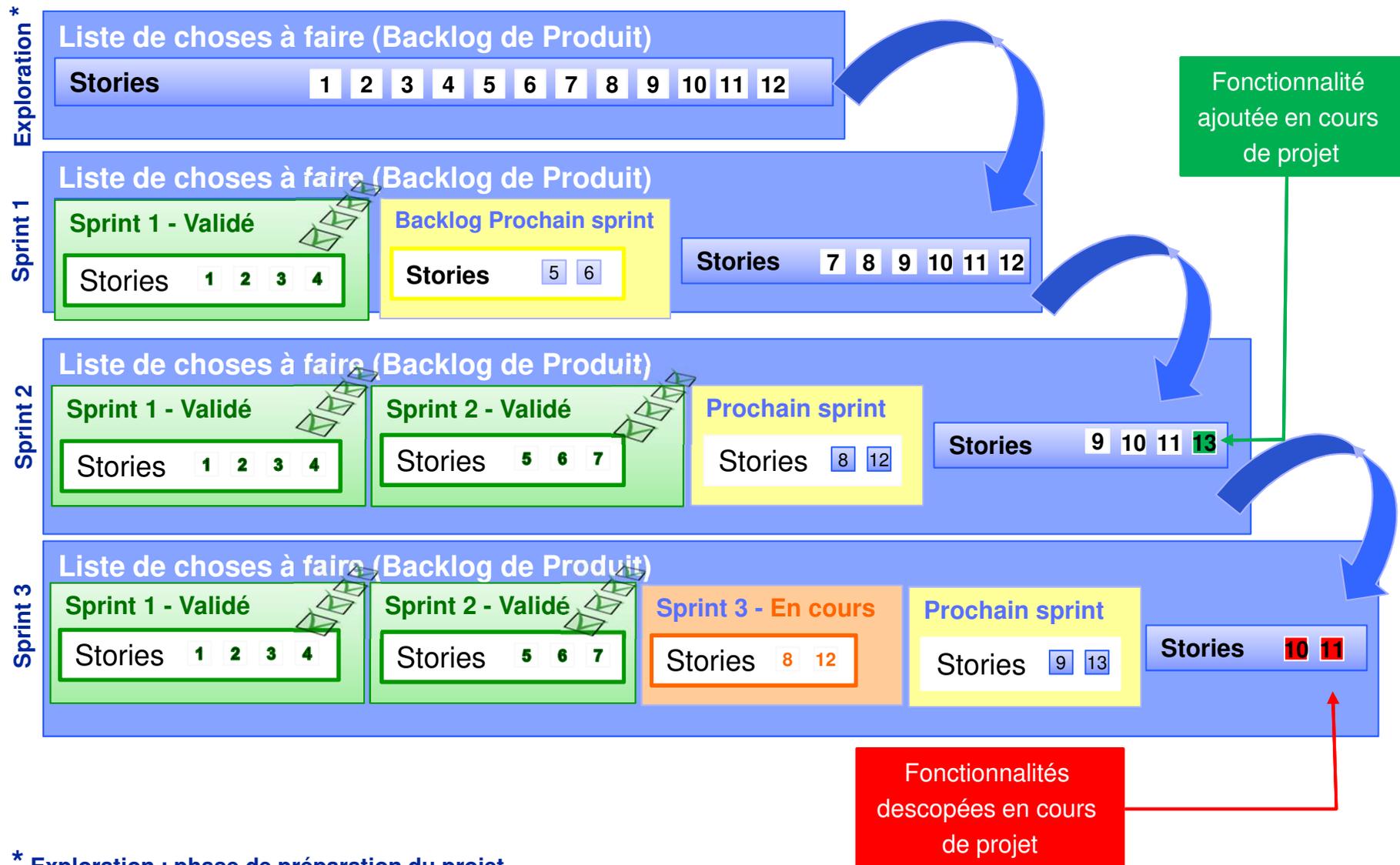
## ■ **Incrément**

- définition : quantité de quelque chose qu'on ajoute à une quantité existante
- Dans les méthodes agiles, l'incrément désigne un ajout de valeur métier supplémentaire

## ■ **Itératif et incrémental :**

à chaque itération,  
l'équipe ajoute un peu plus de valeur métier à l'application

# Une démarche itérative / incrémentale

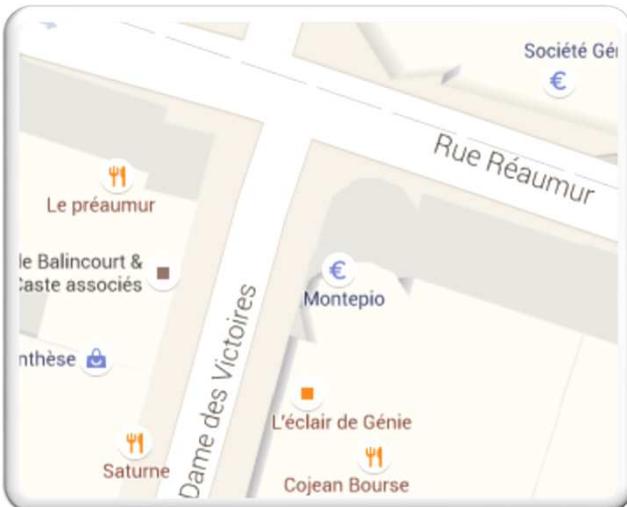


# User story : détail progressif



Dans un premier temps, le « backlog » ne contient pas de détail

Cela permet aux développeurs de comprendre de quoi il s'agit et de donner une estimation.



Au moment de réaliser une fonctionnalité, on ajoute « Suffisamment de détails »

Cela permet aux développeurs de la découper en petites tâches unitaires techniques pour la réaliser

**Pour ne détailler que l'utile ... au bon moment**

# Quelques gains du fonctionnement itératif

## Principaux gains attendus

- **feedback (client <--> développeurs)**
- **meilleure collaboration**
- **meilleure prédictibilité et suivi du projet facilité**
- **risques mieux maîtrisés**
- **changements fonctionnels / repriorisation faciles**
- **innovation / R&D plus facile à oser**
- **amélioration continue** (expliquée juste après)



**DÉMARCHE EMPIRIQUE**  
**+**  
**DROIT À L'ERREUR**  
**+**  
**ITÉRATIONS COURTES**  
**=**  
**AMÉLIORATION CONTINUE**

# Amélioration continue

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

## Comment faire concrètement :

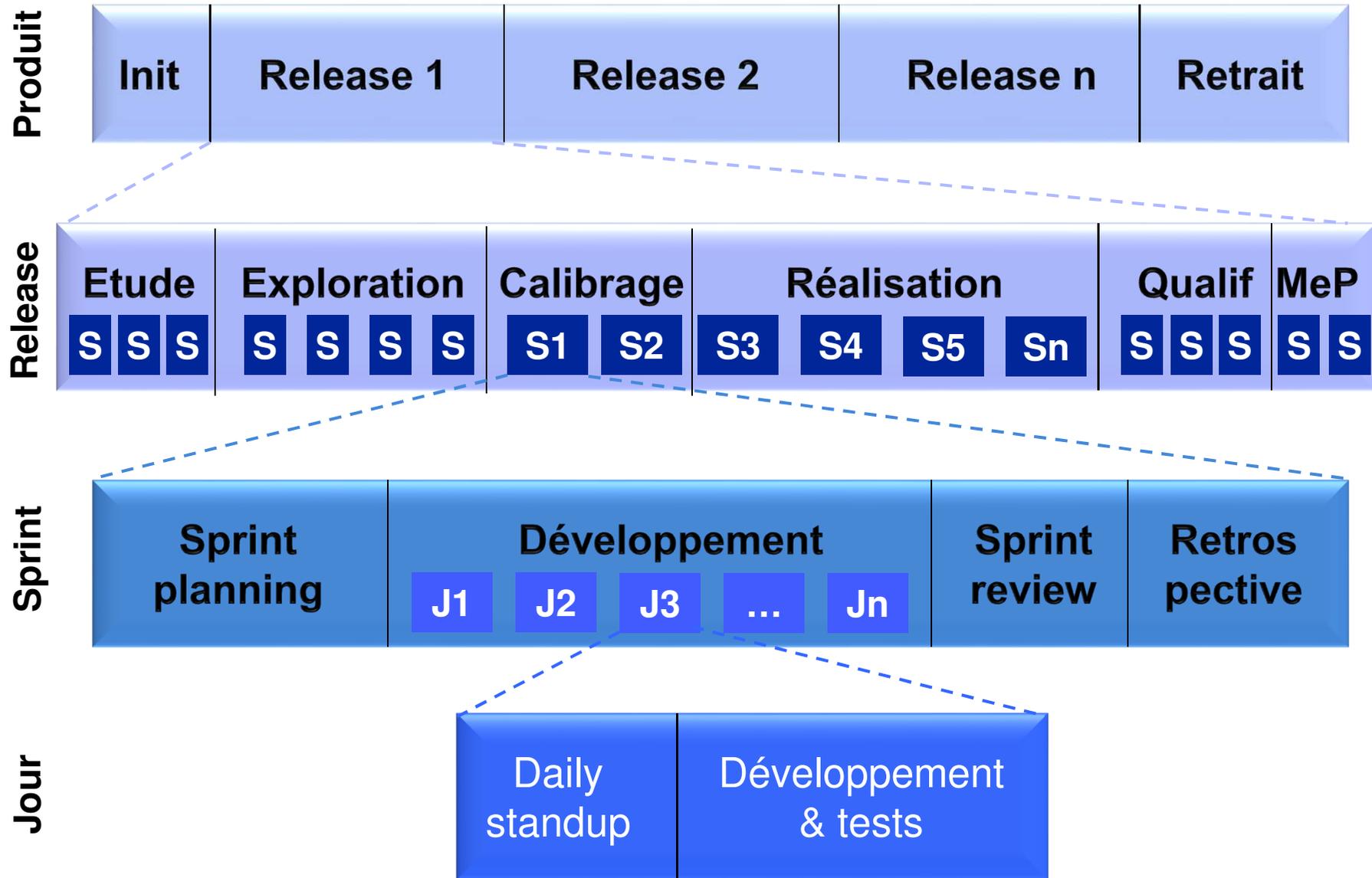
A chaque fin d'itération (ou à intervalle régulier), l'équipe investit quelques heures pour célébrer ses succès et améliorer son fonctionnement.

**Super efficace !**



**COMMENT CA POURRAIT MARCHER...**

# Vue globale



# Les 3 rôles de base (méthode Scrum)



« **Product Owner** » :  
*sait ce qu'il faut faire*  
(ce dont les utilisateurs finaux ont besoin)



**l'équipe de développeurs**  
*sait comment le faire*  
chaque développeur est concepteur, codeur, testeur technique, co-responsable du résultat, et engagé



« **Scrum Master** » :  
*sait comment organiser le travail*  
met en place et maintient l'Agilité sur le projet, en l'adaptant au contexte

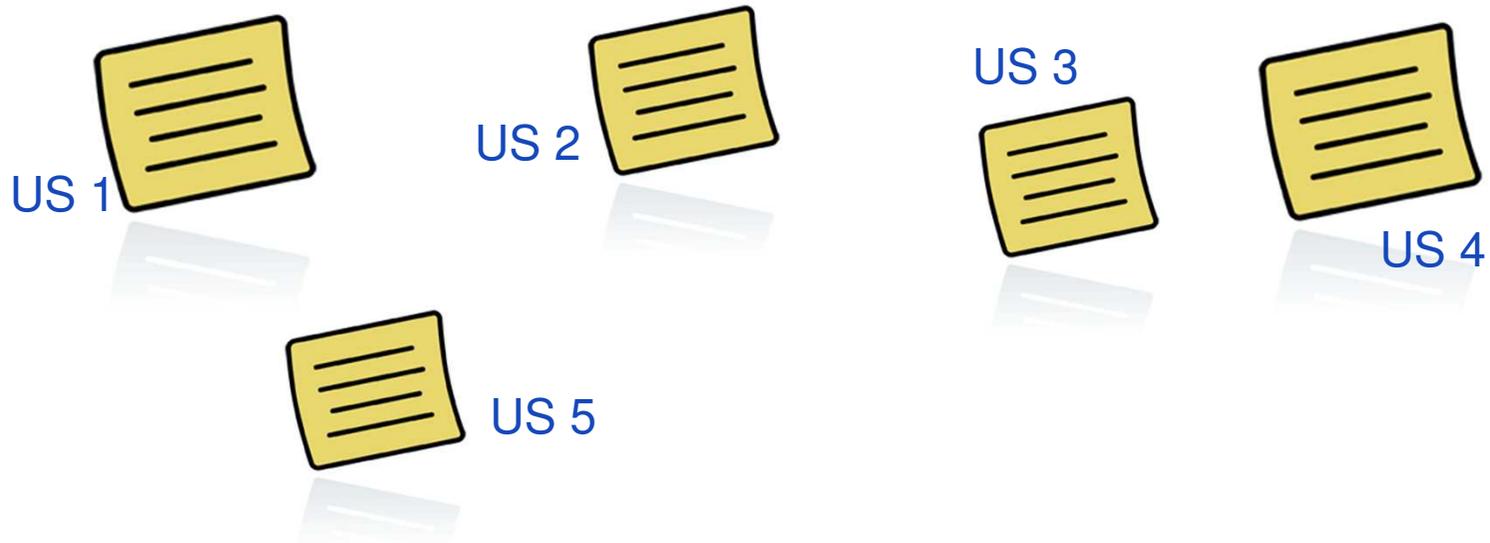


# Préparer le projet

# Préparation du projet – Backlog produit

## 1/ PO formalise des user stories :

« <un acteur du produit final>  
voudrait <fonctionnalité>  
car <raison> »



# Priorisation MoSCoW

## 2/ PO met les priorités métier

### MUST

L'indispensable, hors confort.  
Minimum vital coté fonctionnel

### SHOULD

Non indispensable,  
mais très utile

### COULD

Ajoute du confort d'utilisation

### WON'T HAVE

Pas le temps / budget !



# La Story Map

## Priorités MoSCoW

## Activités

	Clients	Facturation	Rendez-vous	Statistiques
<b>MUST</b>	Créer fiche client	Calcul facture Edition papier	Créer RDV Modif RDV Liste RDV	Clients impayés
<b>SHOULD</b>	Modif fiche client	Envoi mail	Suppr RDV	Clients fidèles
<b>COULD</b>	Supprime fiche client	Consultation internet	Tableau de bord	Tableau de bord
<b>WON'T HAVE</b>		Notification SMS		Mail automatisé

# La complexité technique (effort / story points)

Sert à estimer l'effort à fournir pour implémenter la fonctionnalité.

Sert ensuite à valoriser le backlog de produit, sans détail et sans homme-jour (!)



Les développeurs utilisent les chiffres de la suite de Fibonacci pour exprimer leur estimation de la difficulté pour faire une User story



1

2

3

5

8

13

21

34

55 ...

*Facile*

*Moins facile*

*Compliqué*

*Très compliqué*

# Complexité

# technique



**3/ les DEV estiment la difficulté à réaliser, avec la suite de Fibonacci**

Facile

**La « complexité technique » permet d'estimer l'effort sans avoir besoin de détail (inédit !)**

Difficile

1

2

34

55

c'est beaucoup plus facile que la référence

pareil

... plus compliqué

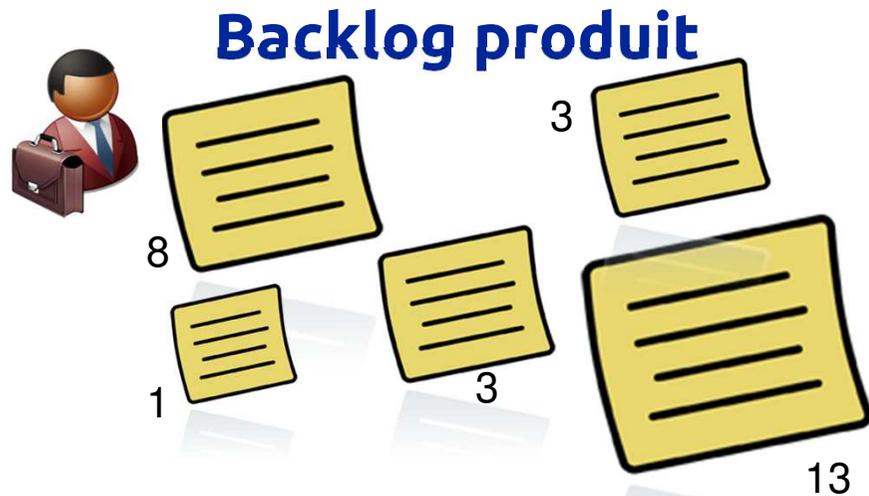
... encore plus compliqué

1 + 3 + 3 + 8 + 13 points  
Le backlog fait 28 points

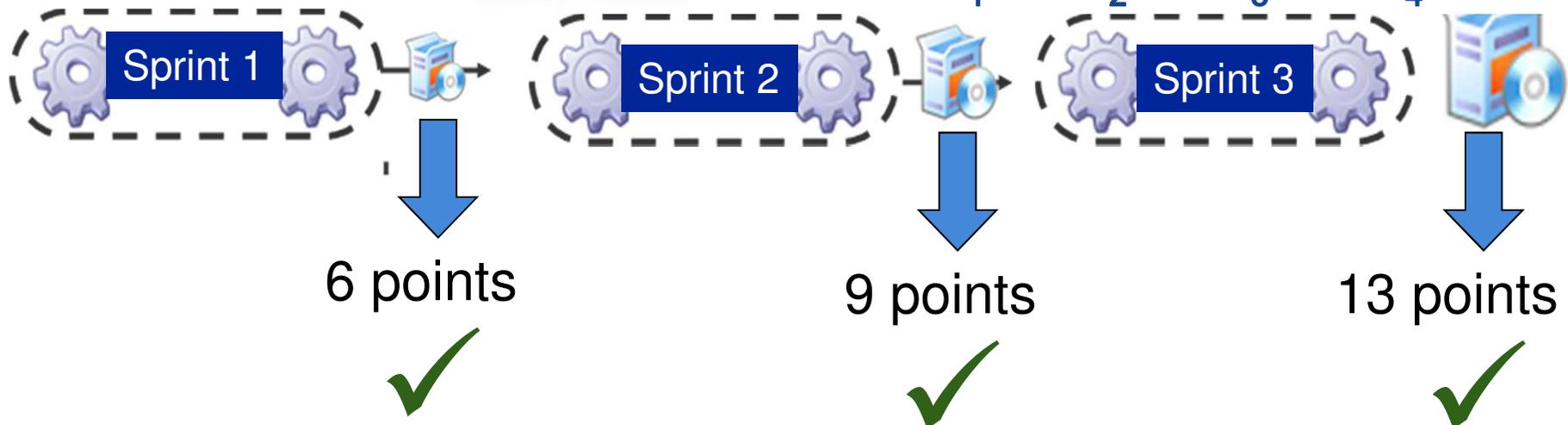
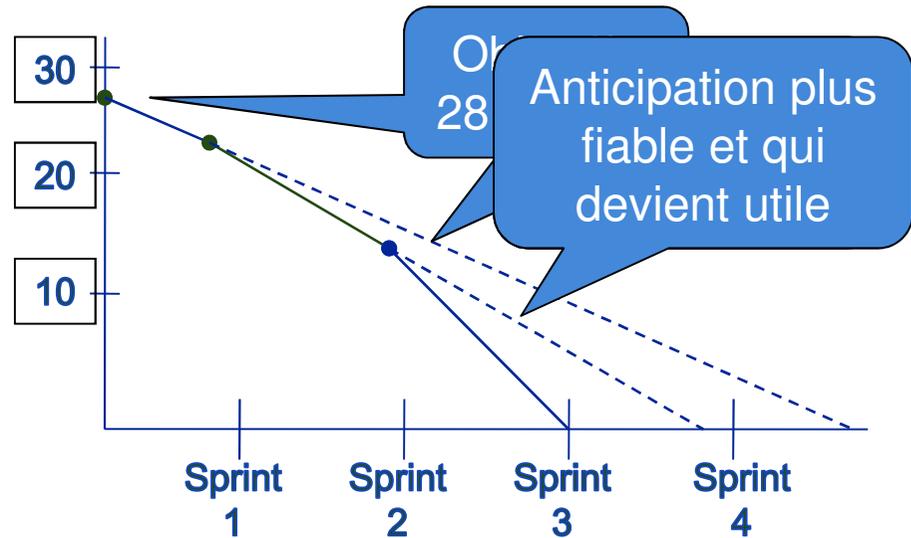


# Déroulement du projet

# Suivi global et reporting



### Burndown chart de produit

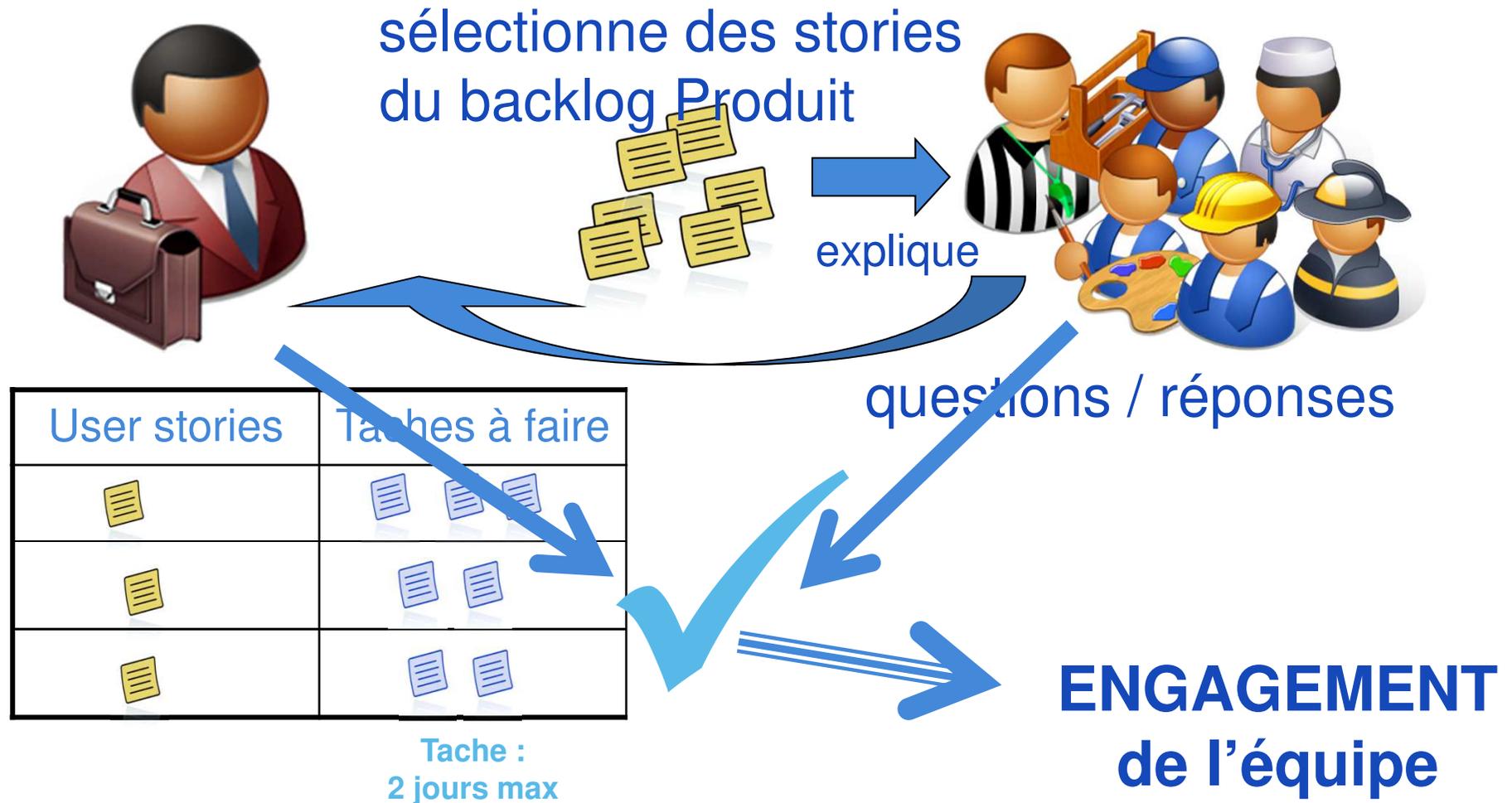




# **Fonctionnement d'un sprint**

# Déroulement d'un sprint

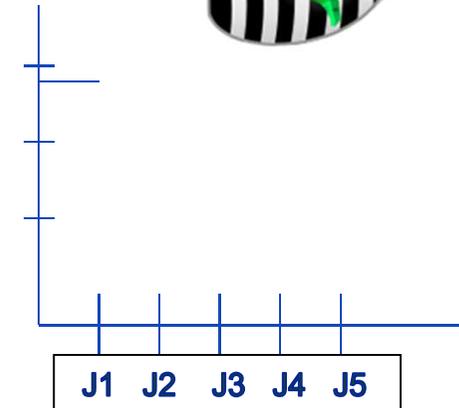
## 1/ Sprint planning => définir le contenu du sprint



# Déroulement d'un sprint

## 2/ Développement

User stories	Taches		
	A faire	En cours	Terminé
			
			
			



# Déroulement d'un sprint

## 2/ Développement. Jour 2

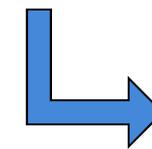
User stories	Taches		
	A faire	En cours	Terminé
	  		
	 		
	 		



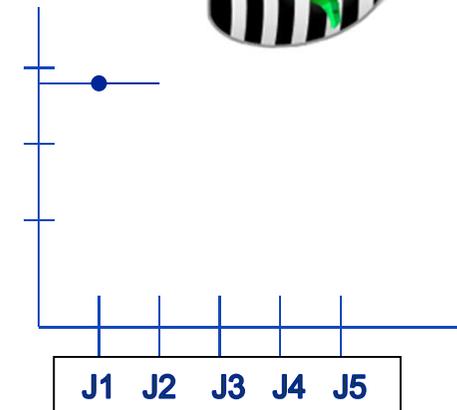
questions  
réponses



## Daily Standup (15mn)



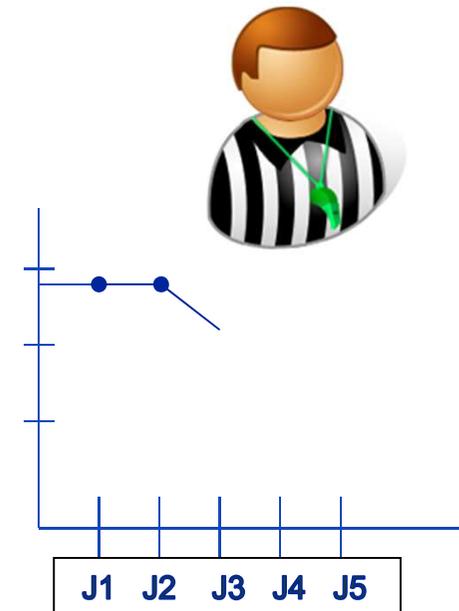
Qu'ai-je fait hier ?  
Que vais-je faire ?  
Suis-je bloqué ?



# Déroulement d'un sprint

## 2/ Développement. Jour 3

User stories	Taches		
	A faire	En cours	Terminé
	 		
			
			

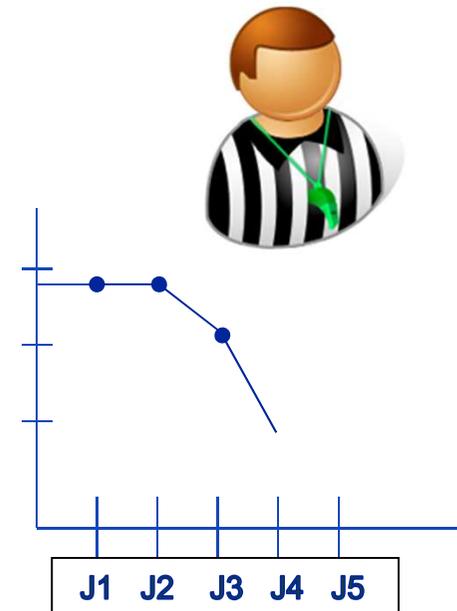


# Déroulement d'un sprint

## 2/ Développement.

### Jour 4

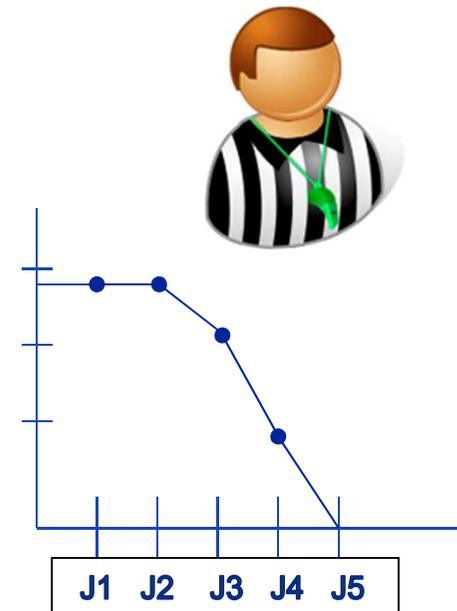
User stories	Taches		
	A faire	En cours	Terminé
			
			
			 



# Déroulement d'un sprint

## 2/ Développement. Jour 5

User stories	Taches		
	A faire	En cours	Terminé
			
			
			



# Déroulement d'un sprint

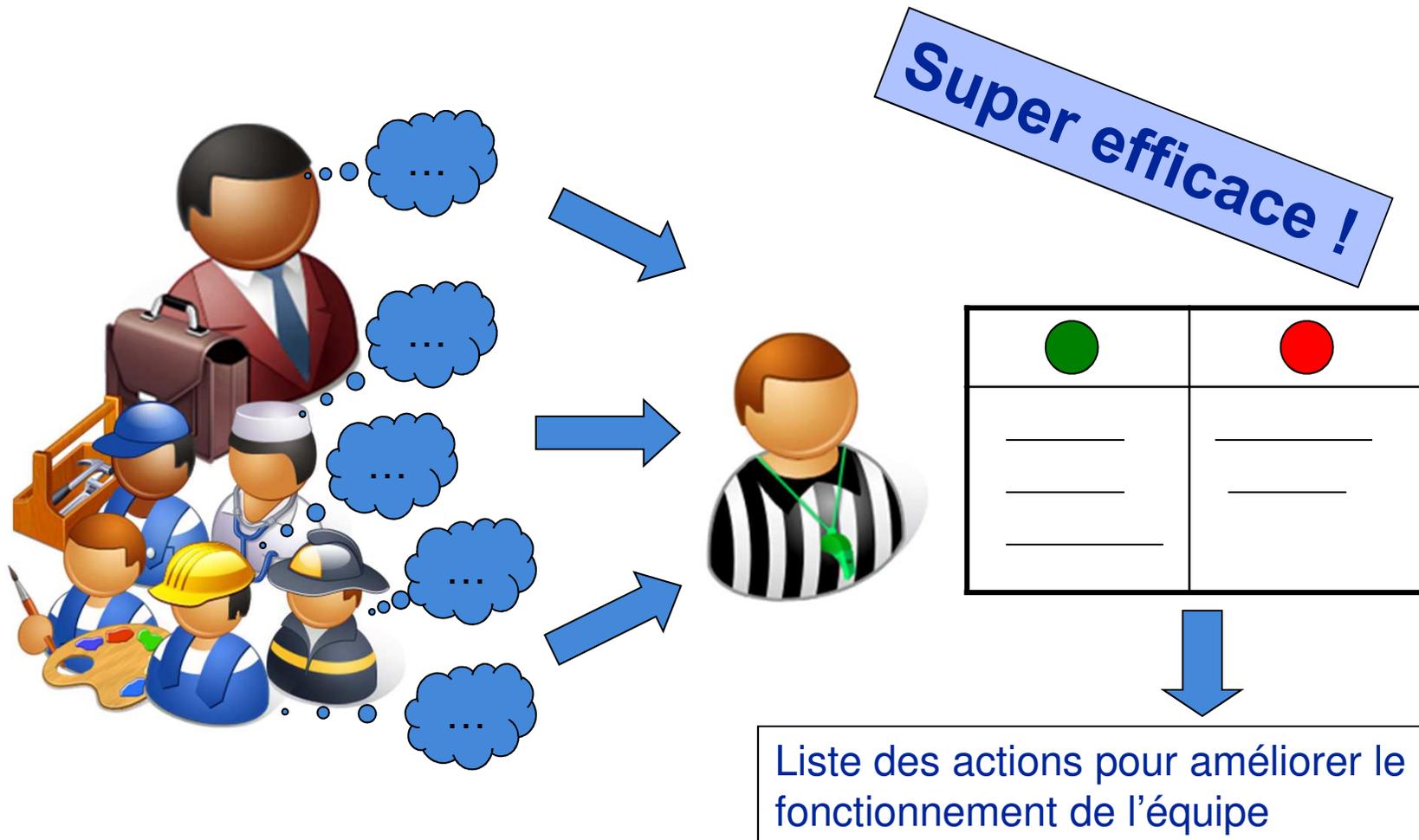
## 3/ Sprint review (Revue de sprint), pour :

- collecter du feedback au plus tôt
- valider le travail réalisé



# Déroulement d'un sprint

## 4/ Cérémonie : Rétrospective => amélioration continue





# **LES PRATIQUES AGILES UTILES POUR DÉMARRER**



# Quelques pratiques Agiles

## ■ **Management visuel**

- Permet de centraliser l'information, la partager et la rendre transparente

## ■ **Définir un vocabulaire commun**

- Créer un glossaire des termes fonctionnels clé.

## ■ **Travailler en binôme**

- Deux développeurs travaillent ensemble sur la même tâche. Après une phase d'apprentissage, un binôme peut aller plus que 2 fois plus vite qu'un développeur seul. Cela accroît la qualité, le respect des normes du projet et des DoD, réduit le nombre de bugs, favorise l'innovation et améliore la conception.

## ■ **Appartenance collective du code**

- Chaque développeur a pu travailler sur toutes les parties de l'application, et sait intervenir efficacement partout. Personne n'est devenu « indispensable » sur un bout de code. Tout le monde peut partir paisiblement en congés.

## ■ **Tester, et automatiser les tests**

- Permet de certifier en permanence que tout fonctionne, sans régression. Il n'y a plus de stress lorsqu'on modifie du code ancien, ou lorsqu'on met en production



# **L'AGILITÉ INTRODUIT DES CHANGEMENTS**

# Les bonnes idées

- **accepter la réalité lorsqu'elle s'est produite, et s'y adapter**
- **apprendre en faisant, avancer en marchant**  
**Oser apprendre de ses (petits) échecs**
- **avancer de manière itérative et incrémentale**
- **ne faire que ce qui est utile**
- **une proximité maximale.**  
**plateau commun et unique**
- **chercher des solutions ensemble,**  
**plutôt que des coupables.**

# L'Agilité – une dynamique d'équipe

- **rôles clairement définis**
- **rôles très respectés**
- **équipe petite (7, +/- 2),**  
(au besoin, on fait plusieurs équipes)
- **équipe autonome**
- **pas de hiérarchie interne**  
(à moins de 10, grâce à la responsabilité collective, pas besoin !)



**L'AGILITÉ PROPOSE  
DE NOUVEAUX PARADIGMES**

## Changements contractuels

- **Les spécifications ne sont plus figées.**
- **On se prépare à accueillir le changement grâce à l'itératif**
- **Le client donne son avis tout au long du projet**
- **L'équipe s'engage sur un nombre de points à réaliser après avoir constaté sa capacité de production par sprint**

# Les changements liés à l'agilité

## Changements organisationnels

- le nombre des rôles est réduit
- les rôles sont très respectés, valorisés, à spectre large
- le travail est itératif et incrémental
- les décisions sont prises collégialement
- il n'y a pas de hiérarchie interne dans l'équipe

# Des changements sont indispensables



“What if we don’t change at all ...  
and something magical just happens?”

# Accepter la réalité

et vous ...



... quelle pilule prenez-vous ?

**que l'agilité**



**soit avec vous !**