

Développement Java

MÉTHODOLOGIES DE GESTION DE PROJET



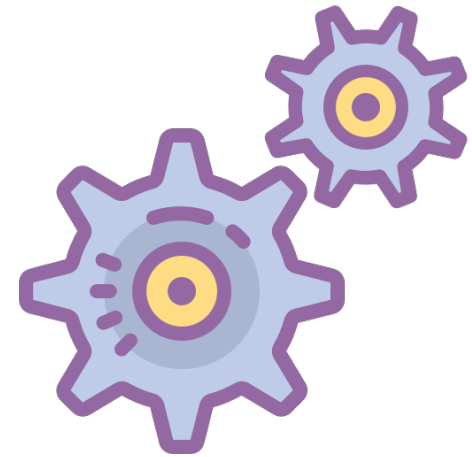
Objectifs pédagogiques

- Connaitre et comprendre les principales méthodes de gestion de projet.



C'est quoi la gestion de projet ?

- Démarche structurante assurant le bon déroulement d'un projet avec :
 - une planification
 - une gestion des ressources humaines
 - un suivi des enjeux financiers



Les différentes méthodologies

- **Méthodes traditionnelles**

Méthodes les plus utilisées en gestion de projet, chaque étape doit être terminée pour passer à la suivante

- **Méthodes Agile**

Moins rigides que les méthodes classiques, les méthodes Agile placent le besoin client au centre du projet

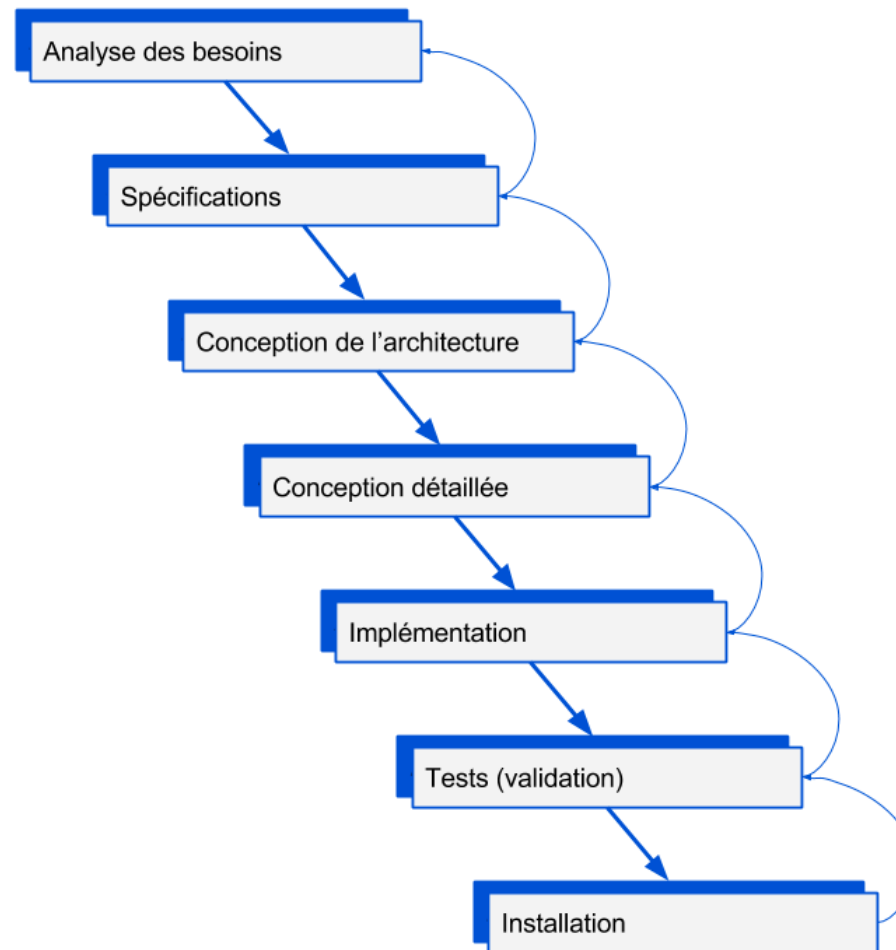
Les méthodes traditionnelles

- D'abord basé sur un cycle en cascade puis désormais sur un cycle en V
- L'équipe projet :
 - suit le cahier des charges en détails
 - Travaille sur la totalité du projet jusqu'à sa livraison
- Pas d'interaction avec le client qui recevra son projet une fois que celui-ci est terminé

Le cycle en cascade

- Présenté par Winston W. Royce en 1970
- Le modèle en cascade originel est hérité du BTP
- Basé sur deux idées fondamentales :
 - Une étape ne peut pas être débutée avant que la précédente ne soit achevée
 - La modification d'une étape du projet a un impact important sur les étapes suivantes

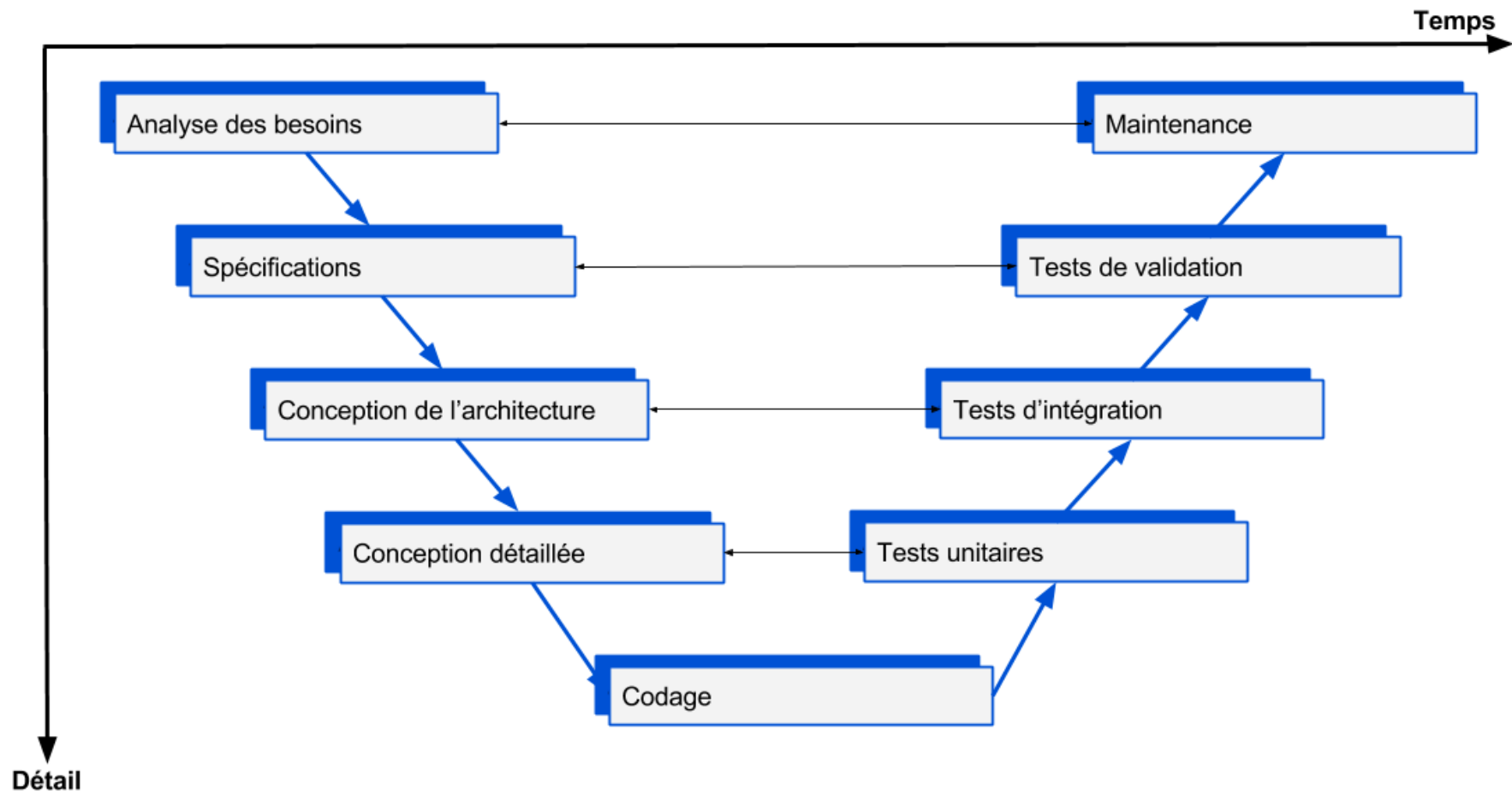
Aperçu du cycle en cascade



Le cycle en V

- En 1980, l'approche en cascade manque de réactivité
- Adoption du cycle en V
- Modèle découpé en 9 phases avec deux étapes :
 - Une étape de conception
 - Une étape de test

Le cycle en V



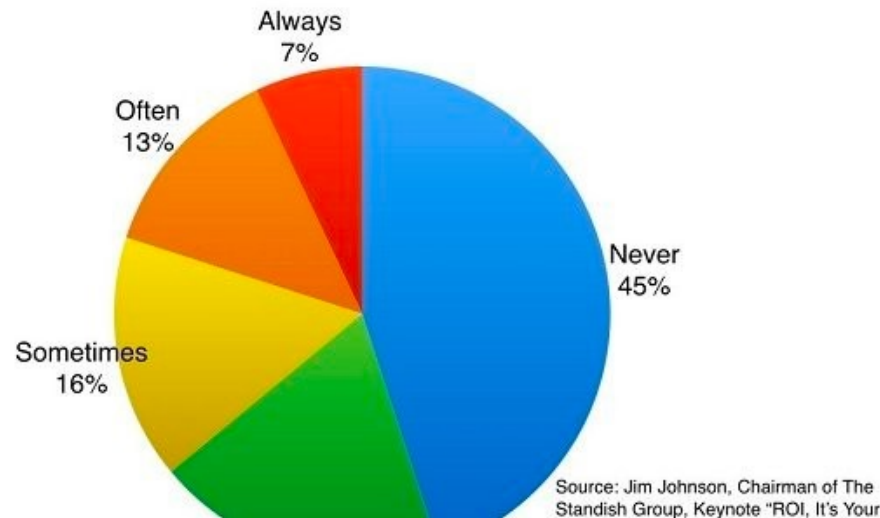
Les méthodes traditionnelles

- L'équipe s'engage sur un planning défini
- Méthode rigide : Peu changement possible
- Documentation pléthorique/surchargée
- Tests de validation tardifs pour l'équipe
- Recette tardive pour le client

Quelques chiffres

En 2009, d'après le « Standish Group Chaos Report » :

- Le taux de succès des projets informatiques en 2009 de 32%
- 45% des fonctionnalités ne sont jamais utilisées



...Vers l'agilité



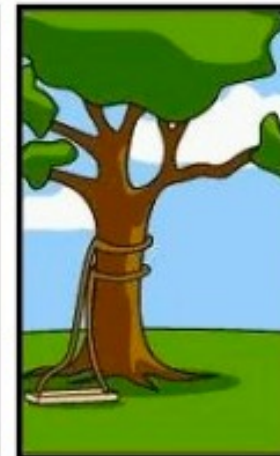
Comment le client a exprimé son besoin



Comment le chef de projet l'a compris



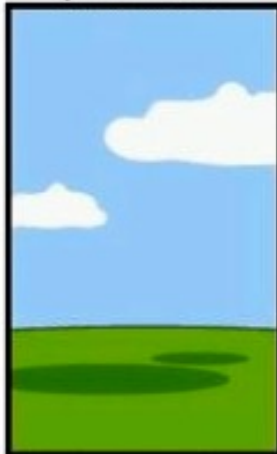
Comment l'ingénieur l'a conçu



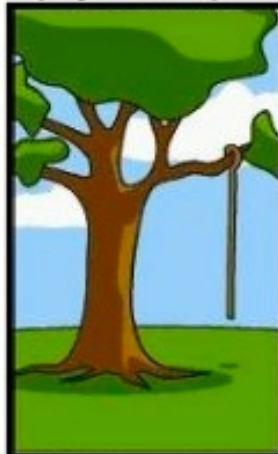
Comment le programmeur l'a écrit



Comment le responsable des ventes l'a décrit



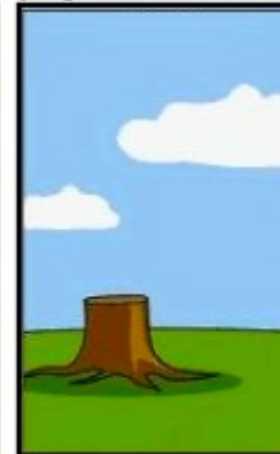
Comment le projet a été documenté



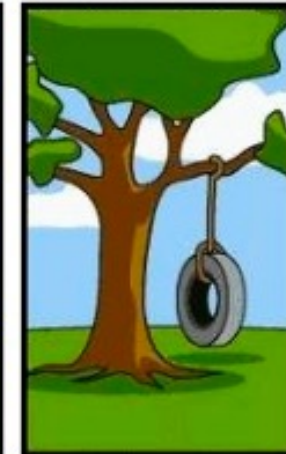
Ce qui a finalement été installé



Comment le client a été facturé



Comment la hotline répond aux demandes



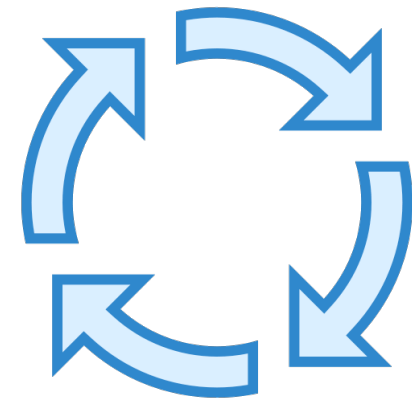
Ce dont le client avait réellement besoin

...vers l'agilité

- **Périmètre fonctionnel** du projet **changeant** au cours du projet
- **Besoin client changeant**, validation régulière de ce qui est réalisé par l'équipe par le client
- **Importance** de traiter ces risques techniques par une **validation technique régulière** du produit
- **Nécessité de livrer** rapidement une **première version**

Comprendre

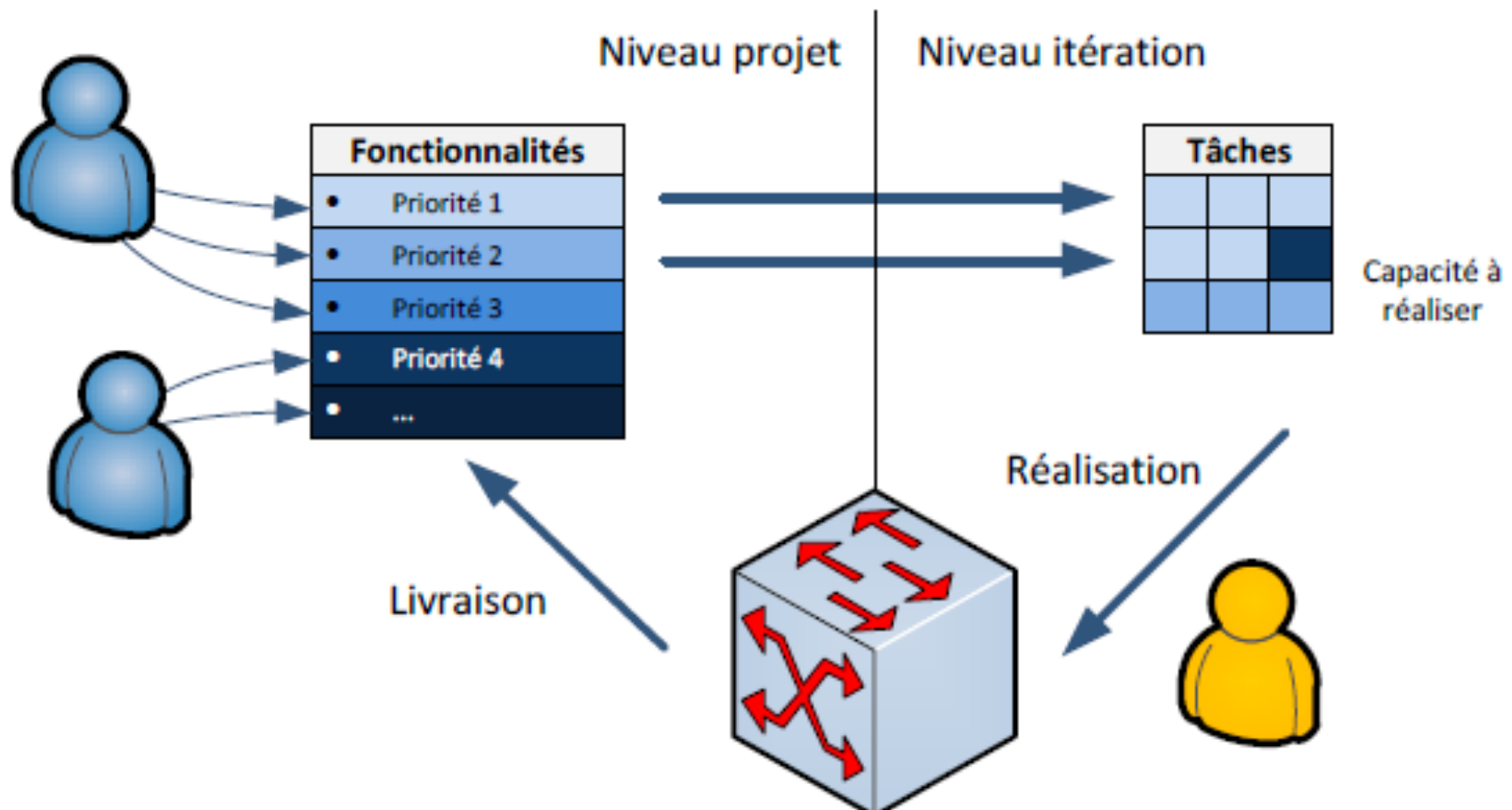
LES MÉTHODES AGILES



Les méthodes agiles

- Les méthodes agiles sont itératives et participatives
- Développement de petits morceaux validé par le client avant de passer aux suivants
- Le client pilote le développement à travers le choix des fonctionnalités qu'il souhaite voir développer
- Une itération prend 2 à 4 semaines

Les méthodes agiles



Le manifeste Agile – les Valeurs

Le ***Manifeste*** est un texte rédigé par 17 experts du développement d'applications informatiques - Février 2001

- <https://agilemanifesto.org/>
- **Personnes et interactions** plutôt que processus et outils
- **Logiciel fonctionnel** plutôt que documentation complète
- **Collaboration avec le client** plutôt que négociation de contrat
- **Réagir au changement** plutôt que suivre un plan

Le manifeste Agile – les Principes 1/4

- **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.

Le manifeste Agile – les Principes 2/4

- 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**

Le manifeste Agile – les Principes 3/4

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

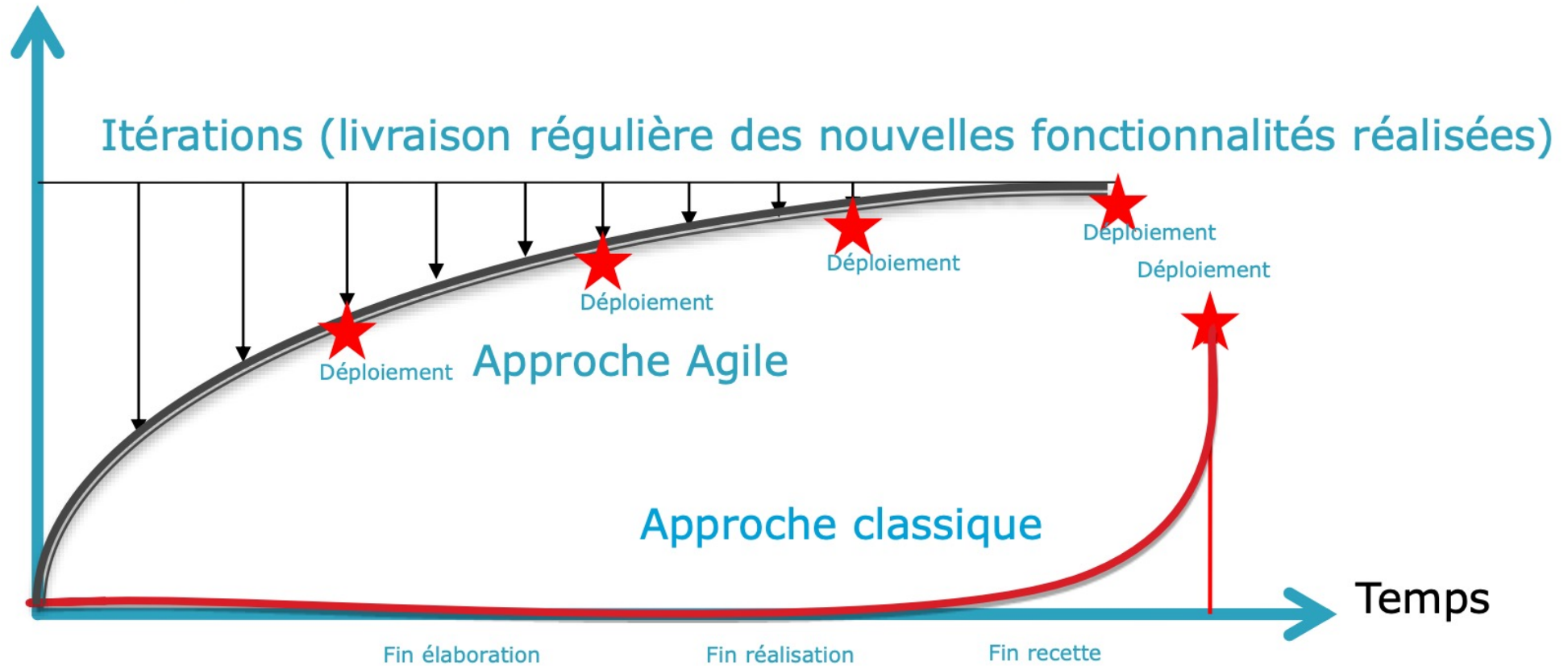
Le manifeste Agile – les Principes 4/4

- **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.

Pilotage par la valeur métier

Valeur Métier

(valeur pour l'utilisateur)



Atelier : W & STOP !



Les règles : 3 Tours

1. Suivre les instructions de l'animateur au tour 1
2. Suivre les instructions de l'animateur au tour 2
3. Suivre les instructions de l'animateur au tour 3 ;)

Règles : W & STOP !

Tour 1 : Lorsque je dirais Walk, marchez ! et Stop, arrêtez vous !

1. WALK = marcher et STOP = s'arrêter
2. WALK = s'arrêter et STOP = marcher

Tour 2 : Lorsque je dirais Name, dites votre nom ! Et clap, tapez des mains !

1. NAME = dire votre nom et CLAP = tapez des mains (toujours avec W & S)
2. NAME = tapez des mains et CLAP = dire votre nom

Tour 3 : Lorsque je dirais Dance, dansez ! Et Jump, Sautiez !

1. DANCE = danser et JUMP = sauter
2. DANCE = sauter et JUMP = danser

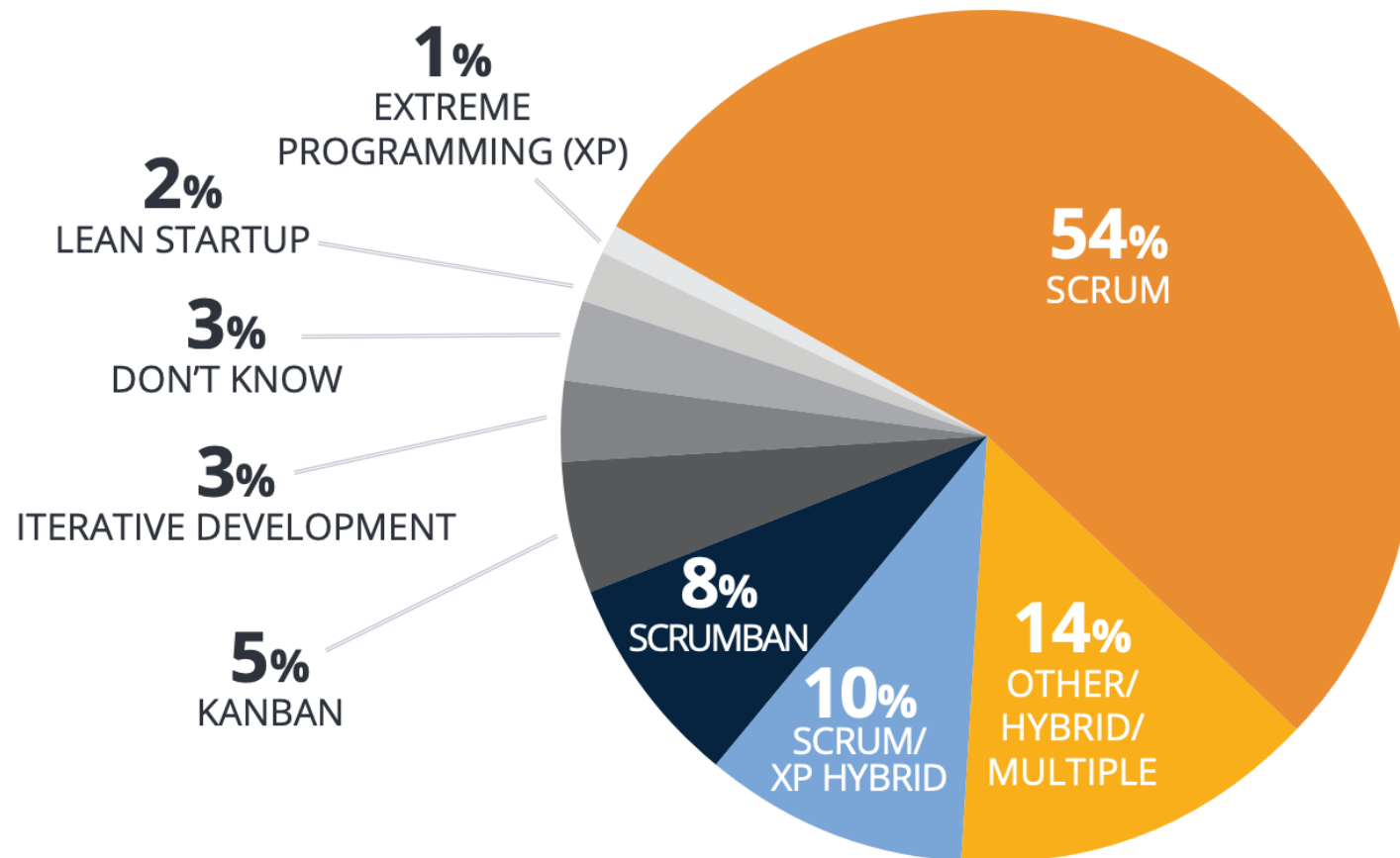
DÉBRIEF & FEEDBACK

Le changement nécessite un temps d'adaptation

Lorsque l'on s'habitue à changer, il devient plus facile, plus rapide et moins douloureux.



Les méthodes agiles



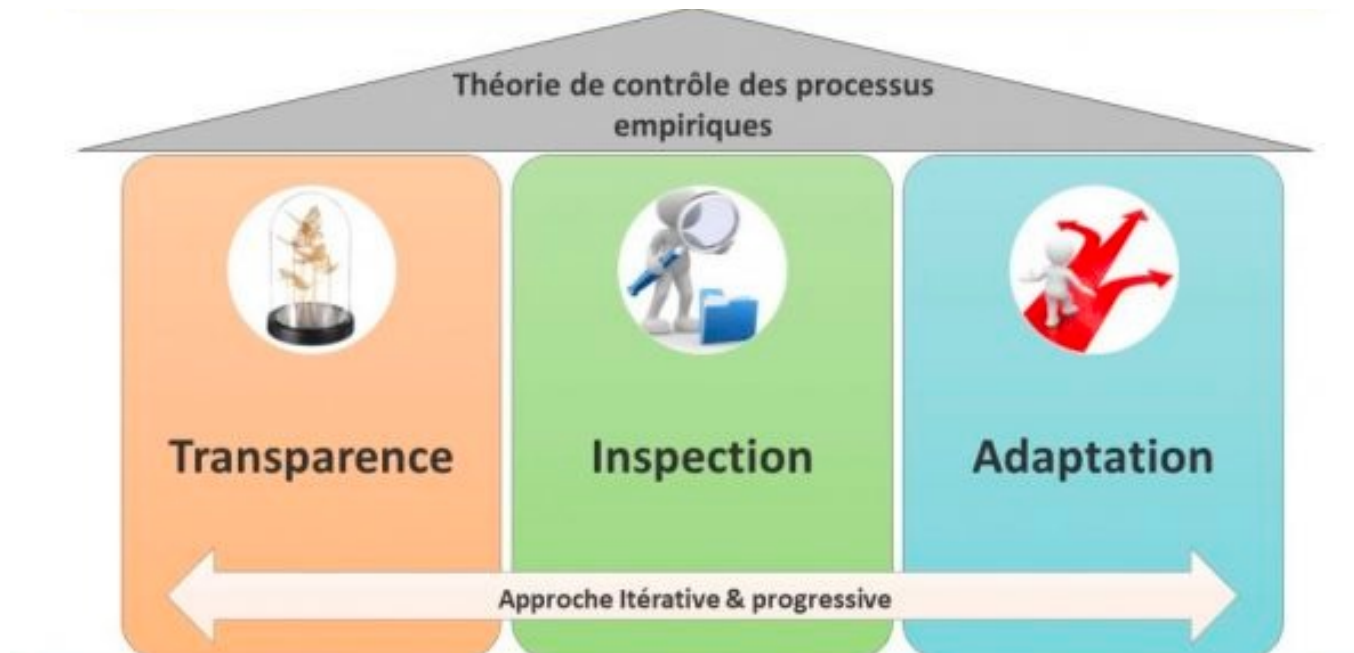
13th Annual State Of Agile Report (2018-2019)

Focus sur Scrum

- Ken Schwaber et Jeff Sutherland sont les concepteurs de Scrum
- Cadre de travail utilisé dès 1990 pour le développement de produit complexe
- Scrum n'est pas une procédure ou une technique pour construire des produits
- Le cadre Scrum définit plusieurs éléments essentiels à la réussite
 - L'équipe et Le rôle des membres
 - Les événements
 - Les artefacts
 - Les règles

Les trois piliers de Scrum

- Scrum repose sur trois piliers fondamentaux

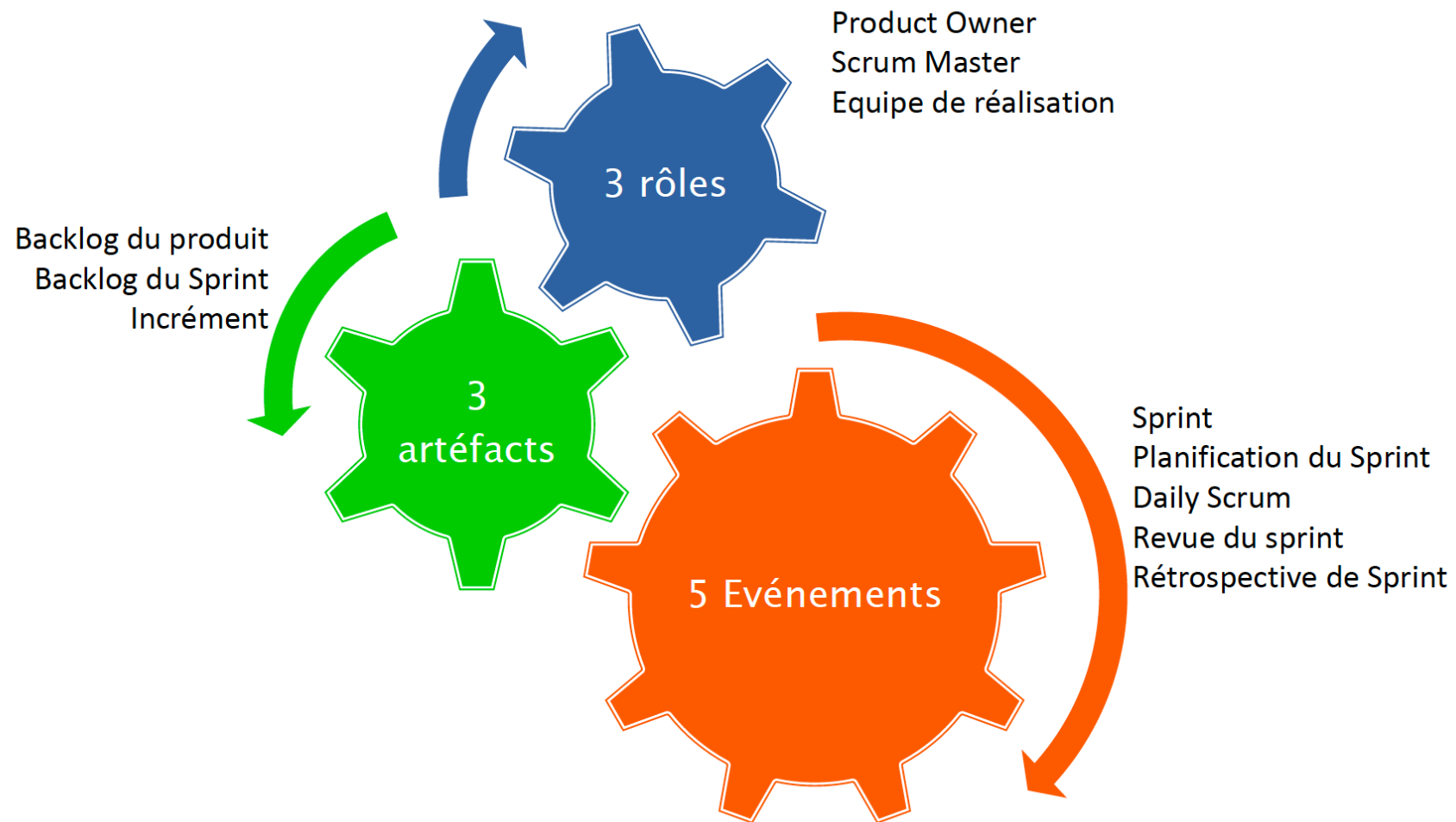


- Pour les appliquer, on utilise une **méthode empirique**.

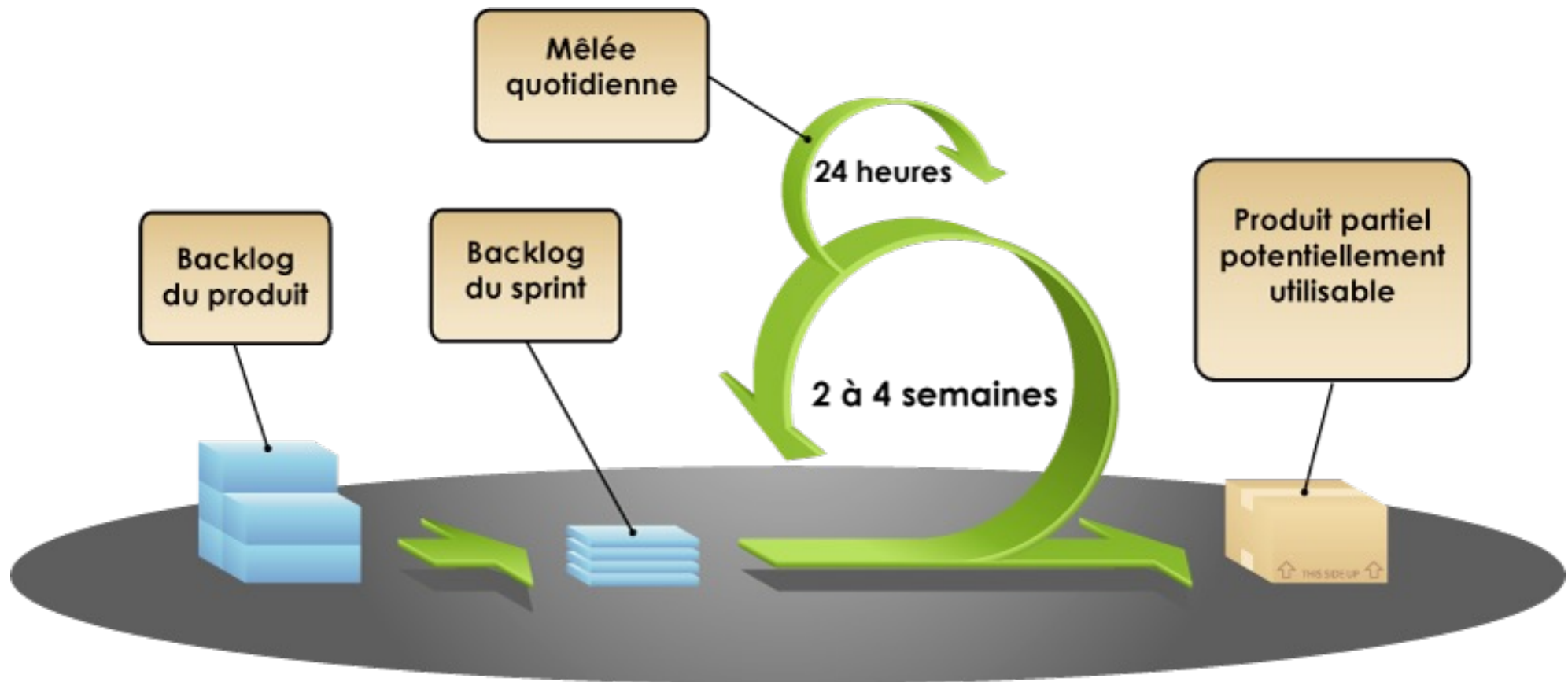
Empirisme ?

- L'empirisme est une doctrine épistémologique
 - Basé sur l'expérience
 - nécessite une bonne visibilité
 - Inspections fréquentes
 - Adaptations aux plans
- Lors de mêlées quotidiennes, on s'appuie sur l'expérience du jour passé pour adapter le planning des jours restant dans le sprint
- lors des revues de sprint, on tient compte de l'expérience du sprint passé pour adapter le planning de la release, portant sur les sprints à venir

Vue d'ensemble de Scrum



Vue d'ensemble de Scrum

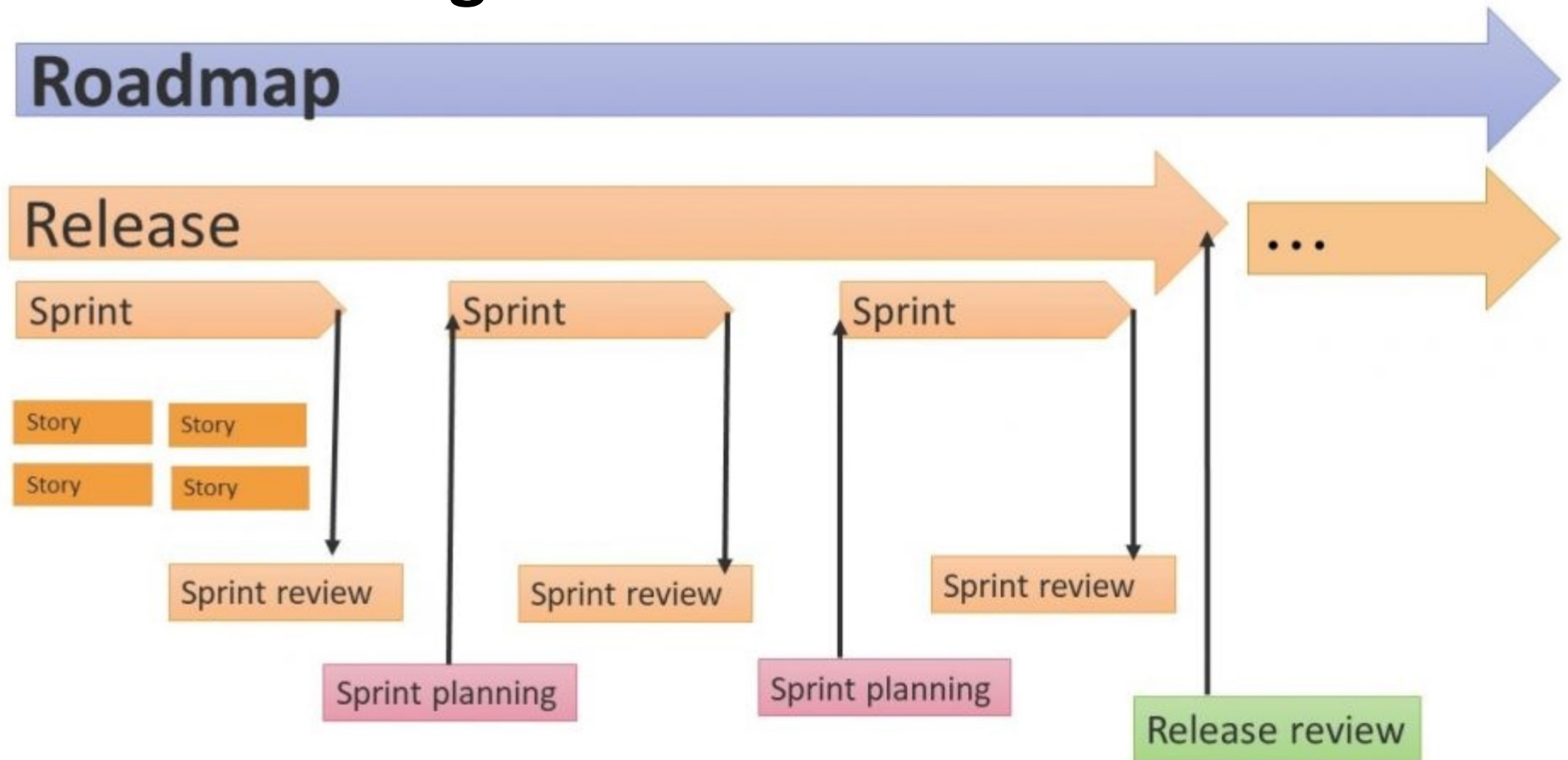


COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Les Sprints

- Les projets Scrum progressent par une série de sprints
- La durée d'un sprint est de 2 à 4 semaines
- Une durée constante apporte un meilleur rythme
- Le produit partiel est conçu, codé et testé pendant le sprint

Planning



Déroulement du Sprint

Chaque développeur s'approprie des tâches des User Stories de l'itération

- Premières tâches attribuées à la réunion de sprint
- Ensuite, au cours des réunions quotidiennes

Tous les matins, réunion café/standup, meeting/scrum meeting pour débloquer les problèmes et mesurer l'avancement

Au bout de l'itération, seules les User Stories complètes sont livrées

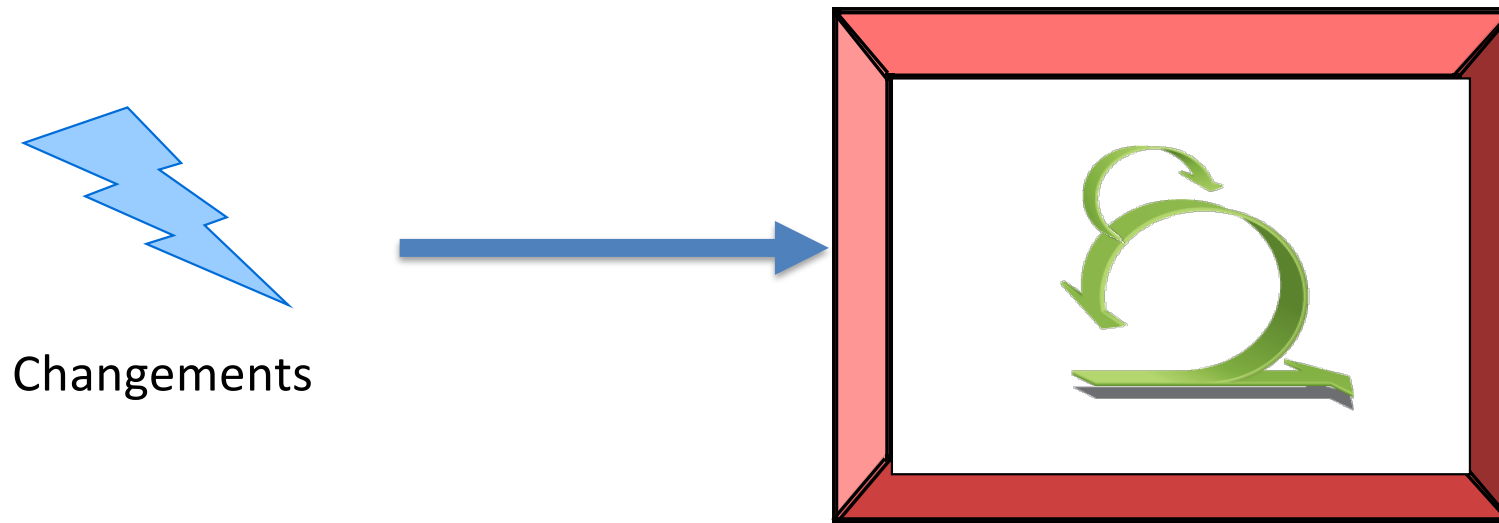
La séquence est différente



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

...Les équipes Scrum font un
peu de tout tout le temps

Sprint et stabilité



- Le Sprint a une durée et des fonctionnalités planifiées
- Permet de différer un changement au prochain sprint

Question 1/2

Sur quel type de contrôle de processus est basé Scrum ?

- Empirique
- Hybride
- Définie
- Complexe

Question 2/2

Quels sont les trois piliers de la théorie et du contrôle empirique de processus ?

- Transparence, Elimination des déchets, Kaizen
- Planification, Démonstration, Rétrospective
- Respect des personnes, Kaizen, élimination du gaspillage
- Inspection, Transparence, Adaptation

LE CADRE SCRUM



Le cadre Scrum

Rôles

- Product Owner
- ScrumMaster
- Equipe

Cérémonial

- Planification du sprint
- Revue du sprint
- Retrospective
- Scrum quotidien

Artefacts

- Backlog de produit
- Liste des tâches
- Burndowns
- Vélocité

Le cadre Scrum :

Rôles

- Product Owner
- ScrumMaster
- Equipe

Processus

- Planification du sprint
- Revue du sprint
- Retrospective
- Scrum quotidien

Artefacts

- Backlog de produit
- Liste des tâches
- Burndowns
- Vélocité

Product Owner

- Définit les fonctionnalités du produit
- Choisit la date et le contenu de la release
- Responsable du **ROI**
- Définit les priorités dans le **backlog** en fonction de la valeur « métier »
- Ajuste les fonctionnalités/priorités à chaque sprint si nécessaire
- Accepte ou rejette les résultats

ScrumMaster

- Représente le management du projet
- Responsable de faire appliquer par l'équipe les valeurs et les pratiques de Scrum
- S'assure que l'équipe est complètement fonctionnelle et productive, élimine les obstacles
- Facilite une coopération poussée entre tous les rôles et fonctions
- Protège l'équipe des interférences extérieures

Équipe Scrum

- De 5 à 10 personnes, regroupant tous les rôles
- Architecte, concepteur, développeur, spécialiste IHM, testeur
- A plein temps sur le projet (de préférence)
- L'équipe s'organise par elle-même
- L'équipe ne doit pas changer pendant un Sprint

Question

Qui est responsable de l'enregistrement et des estimations de travail pendant un Sprint ?

- Le Product Owner
- Le Scrum Master
- L'équipe de développement
- Le plus jeune membre de l'équipe

Le cadre Scrum :

Rôles

- Product Owner
- ScrumMaster
- Equipe

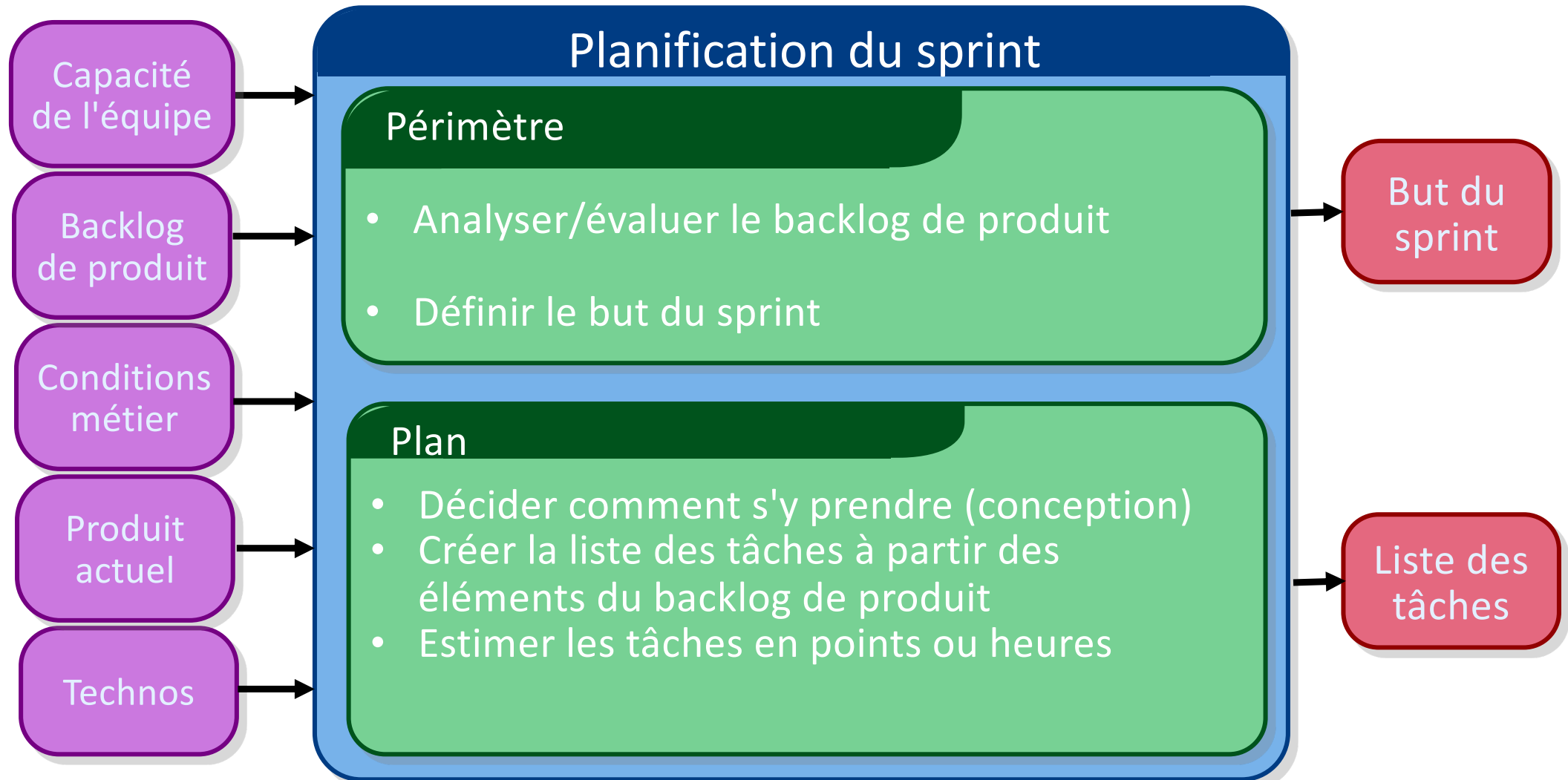
Cérémonial

- Planification du sprint
- Revue du sprint
- Retrospective
- Scrum quotidien

Arts

- Log de produit
- Liste des tâches
- Burndowns
- Vélodité

La planification du Sprint



Définir le but d'un Sprint

Important de définir ce sur quoi va porter le travail du sprint

Application BD

Faire tourner l'application sur une base MySQL en plus d'Oracle.

Sciences de la vie

Offrir les fonctions pour les études génétiques.

Services financiers

Offrir plus d'indicateurs que le produit ABC sur les données de streaming .

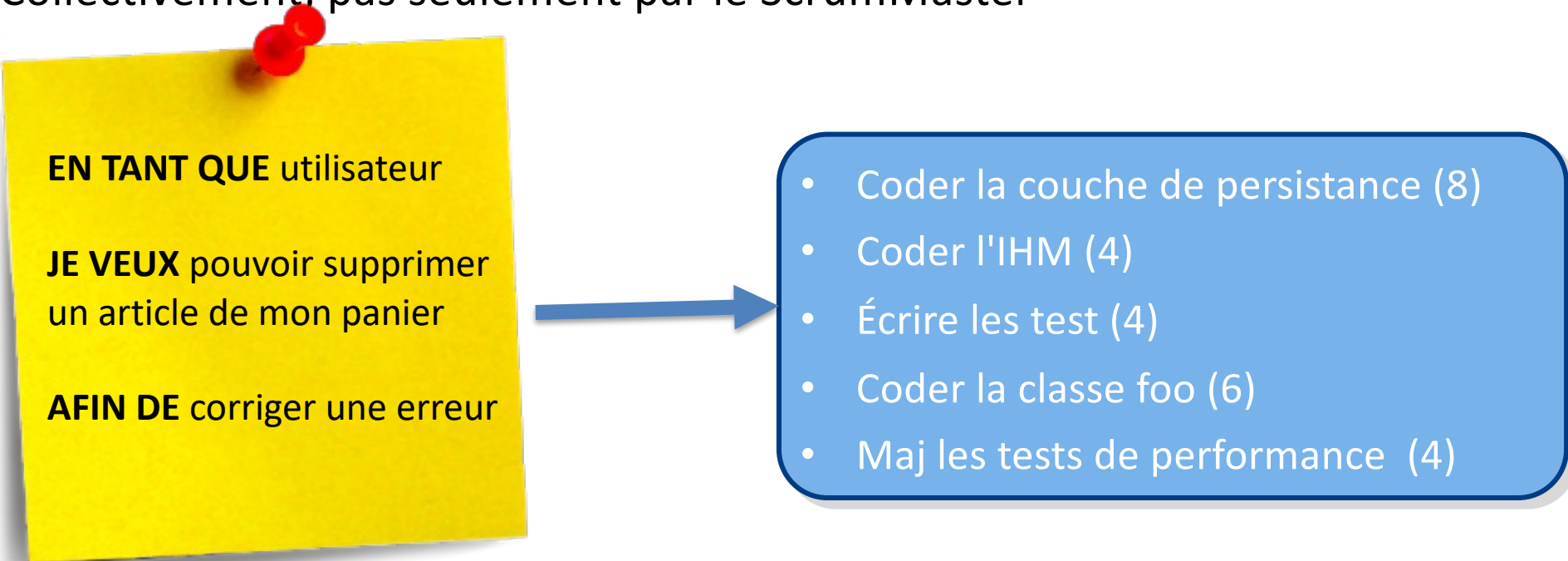
Planification du Sprint (Sprint planning)

- L'équipe choisit, à partir du backlog de produit, les éléments qu'elle s'engage à finir
- Les tâches sont identifiées et estimées
- Collectivement, pas seulement par le ScrumMaster

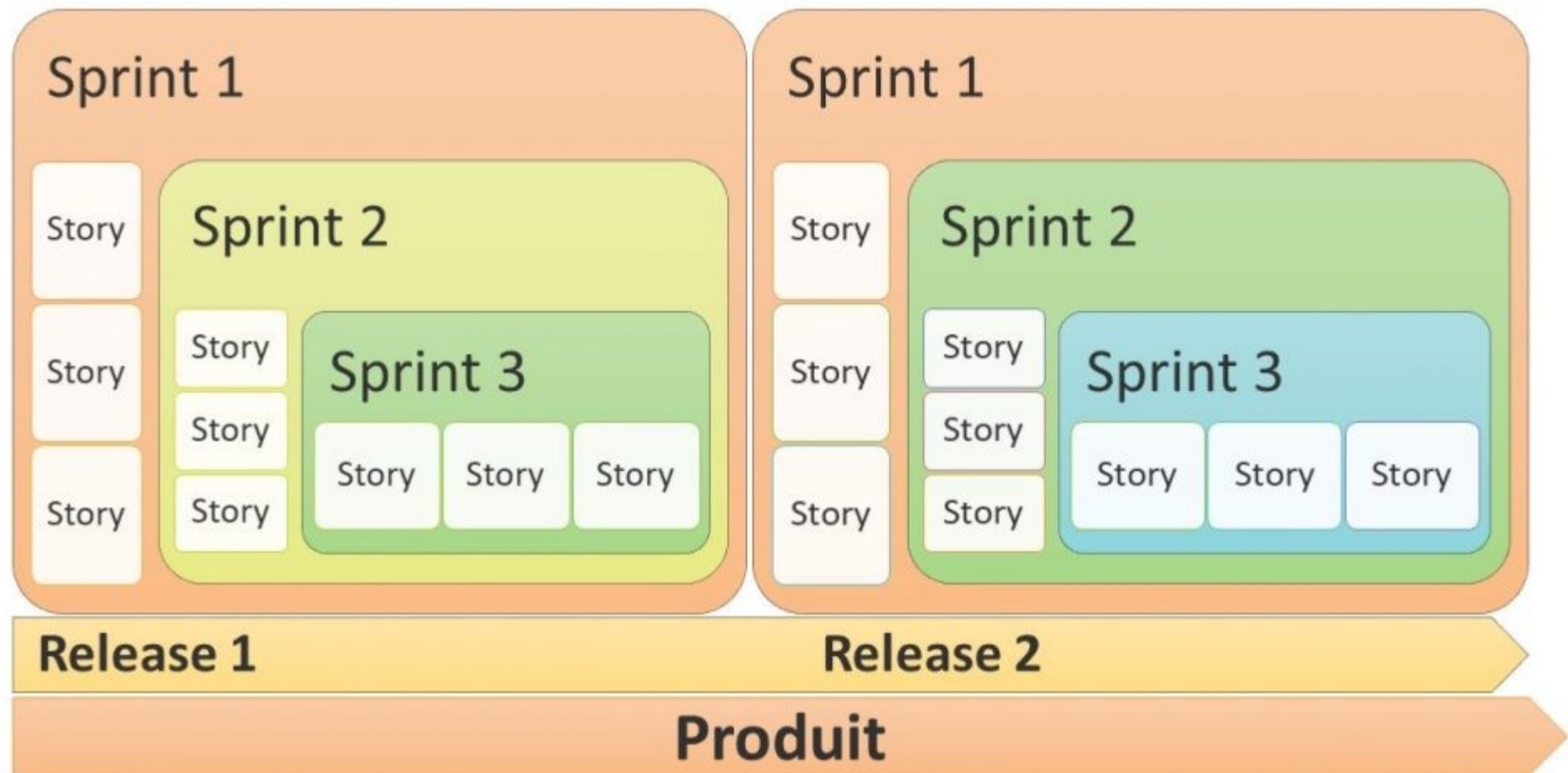
EN TANT QUE utilisateur

JE VEUX pouvoir supprimer
un article de mon panier

AFIN DE corriger une erreur

- 
- Coder la couche de persistance (8)
 - Coder l'IHM (4)
 - Écrire les test (4)
 - Coder la classe foo (6)
 - Maj les tests de performance (4)

Démarche itérative du projet



Revue de Sprint (Sprint Review)

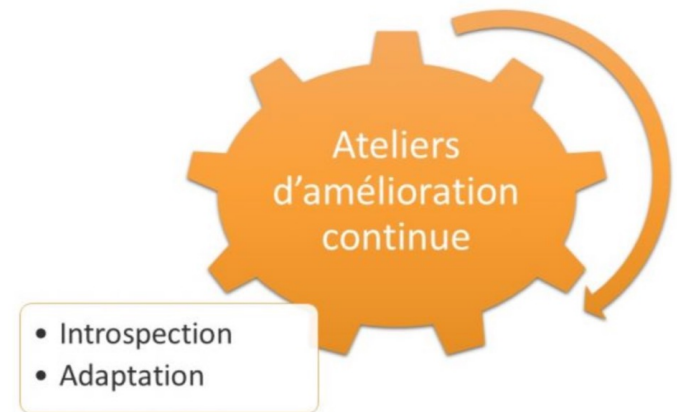
- L'équipe présente ce qu'elle a fait pendant le sprint
- Démo des nouvelles fonctionnalités ou de l'architecture
- Réunion Informel
- Préparation inférieur à 2h, pas de slides
- Toute l'équipe participe, on invite du monde
- Calcul de la vélocité

Rétrospective du Sprint (Sprint Retro)

- Faite à la fin de chaque sprint, 45 min par semaine de Sprint
- Réflexion sur ce qui marche et ce qui ne marche pas
- Toute l'équipe participe (ScrumMaster, Product Owner, Equipe)
- Eventuellement clients et autres intervenants

Retour sur pratiques :

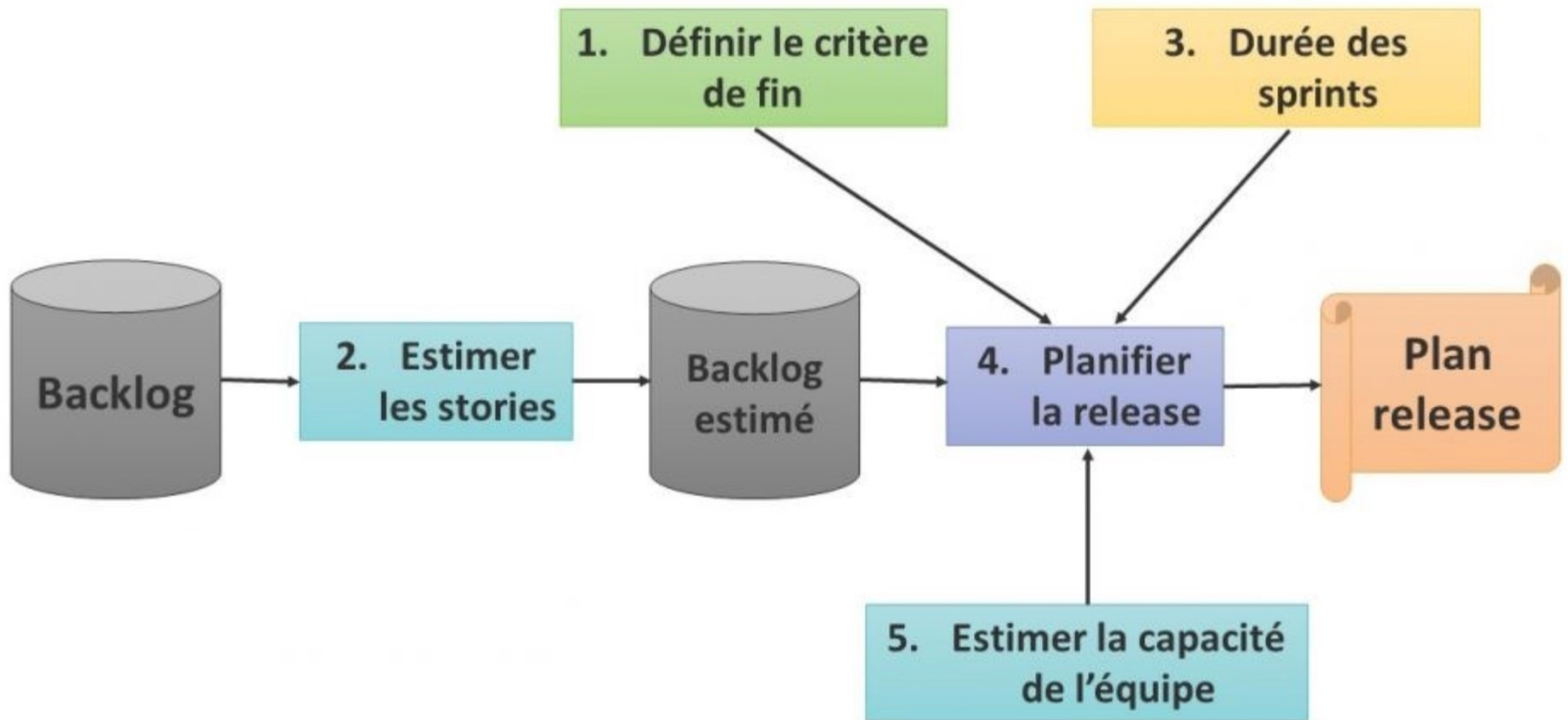
- Commencer à faire
- Arrêter de faire
- Continuer à faire



Mêlée quotidienne (Daily Scrum)

- Réunion journalière, dès le matin
- Maximum 15 minutes, Debout
- Pas fait pour résoudre les problèmes
- Tout le monde est invité mais seuls les membres de l'équipe peuvent parler
- Permet d'éviter l'organisation d'autres réunions
- **3 questions :**
 - Ce que j'ai fait hier ?
 - Ce que je compte faire aujourd'hui ?
 - Quels sont ou bien vont être les obstacles rencontrés ?

Les étapes pour garder le CAP !



Critères d'acceptation et critères de fin

- Rédigés par le Product Owner (en collaboration)
- Accompagne chaque User Story
- Représentent un ensemble de conditions pour que la story soit complète et terminée
- Utilisation de la structure GIVEN-WHEN-THEN
 - Étant donné, Quand, Alors

GIVEN – WHEN - THEN

- GIVEN : l'état du logiciel avant l'exécution de la user story
- WHEN : un évènement qui déclenche un processus
- THEN : l'état du logiciel après exécution

Exemple d'une User story :

ETANT DONNÉ que je suis sur la page de connexion ET que j'ai entré un login et un mot de passe dans le formulaire correspondant

QUAND je clique sur le bouton « Se connecter »

ALORS j'arrive sur la page d'accueil du site

Un critère d'acceptation doit proposer un point de vue Utilisateur

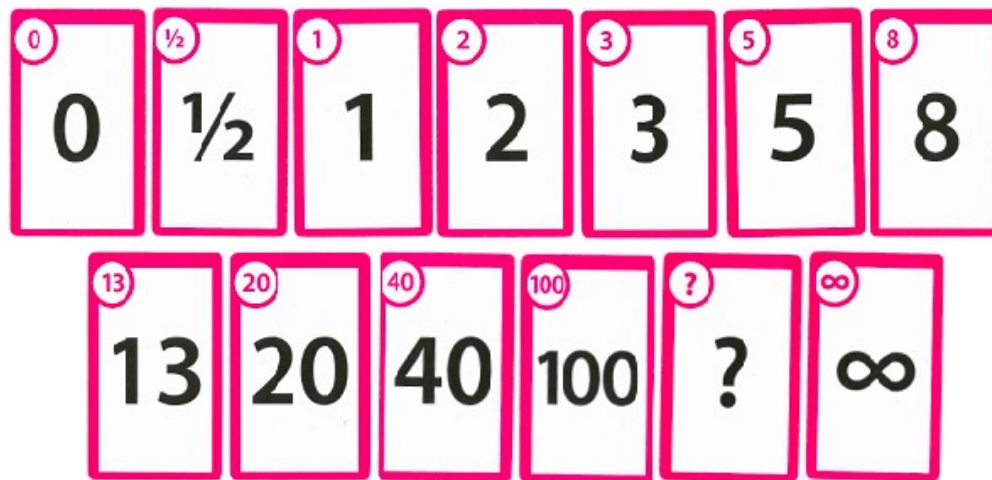
Pensez à S.M.A.R.T

S.M.A.R.T

- **Specific** : compréhensible, facile à reproduire
- **Measurable** : quantifiable et observable
- **Achievable** : possible à réaliser (sans complexité)
- **Relevant** : approprié à la User Story en question
- **Time bound** : avec un moment d'application circonscrit dans le temps

Planning Poker : Complexité

- Session de l'estimation de la complexité des éléments
- Choix démocratique via un jeu de carte



Poker Planning : Fonctionnement

- Chiffrage d'une tâche *étalon* (valeur à définir en commun)
- Le **Product Owner** présente une nouvelle story
- Les membres interrogent le **PO** pour comprendre la story
 - Chaque participant choisit sa propre estimation
 - Chaque participant dévoile ses cartes en même temps
 - Discussions des différences d'estimations
 - Nouvelle estimation jusqu'à être d'accord
- Le PO passe à la story suivante ...

Complexité et durée ?

- L'estimation de la complexité prends en compte :
 - l'effort et la difficulté à faire pour développer la demande
 - les risques, inconnues imaginé pour le développement
 - les dépendances potentielles avec des éléments extérieurs
- Une tâche ne doit pas être estimée si elle est incomplète
- Après le Sprint 0, la vélocité de l'équipe sera connue :
 - 12 tâches de 100 points réalisé en 10 jours de travail
- **L'équipe aura donc une vélocité de 100 points par Sprint**

Question

Quelle est la principale raison pour le Scrum Master d'être présent au DailyScrum ?

- Il ne doit pas être présent mais doit s'assurer que l'équipe de développement a une mêlée quotidienne
- Pour s'assurer que chaque membre de l'équipe répond aux trois questions dans le bon ordre de l'équipe
- Pour écrire toutes les modifications apportées au Sprint précédent, l'ajout d'éléments, le suivi des progrès
- Pour recueillir l'état et la progression à signaler à la direction

Le cadre Scrum

Rôles

- Product Owner
- ScrumMaster
- Equipe

Processus

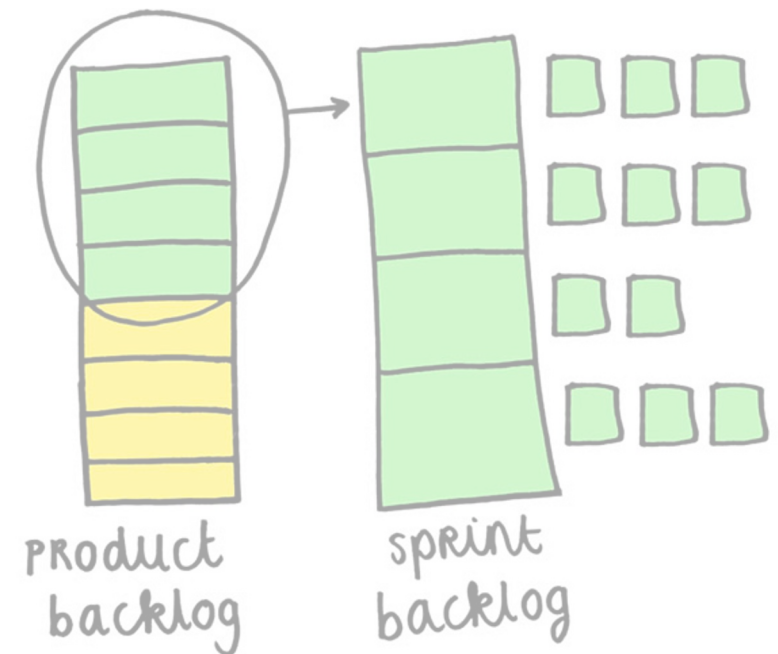
- Sélection du sprint
- Revue du sprint
- Retrospective
- Scrum quotidien

Artefacts

- Backlog de produit
- Liste des tâches
- Burndowns
- Vélocité

Backlog de produit (Product Backlog)

- Exprimé pour que chaque élément apporte de la valeur aux utilisateurs ou clients du produit : **User Story**
- Liste de tout ce qui va entraîner du travail pour l'équipe
- Les priorités sont définies par le Product Owner
- Les priorités sont revues à chaque sprint
- Exigences du client



User story

EN TANT QUE rôle **JE VEUX** besoin **AFIN DE** valeur métier



Exemple de Product Backlog

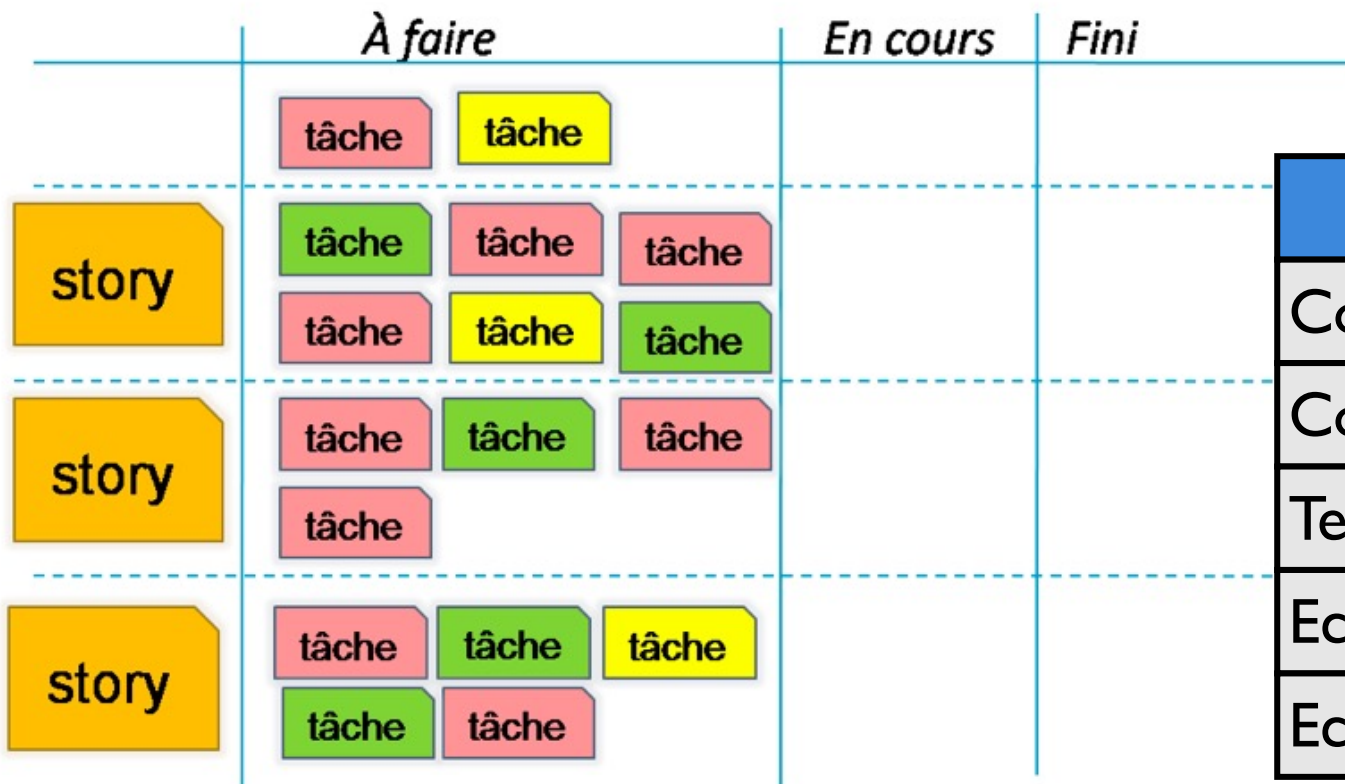


Product backlog (User stories)	Estimation
En tant qu'invite, je peux faire une réservation	4
En tant qu'invité, j'annule une réservation	5
En tant qu'invité, je change les dates d'une réservation.	3
En tant qu'employé de l'hôtel, je produis les rapports de revenu par chambre	8
...	8
...	30
...	...

Exemple du Scrum Board

Sprint 3 : début le 15/3, fin le 29/3

But : *lorem ipsum*



Tâches

Coder l'IHM

Coder couche métier

Tester l'intégration

Ecrire l'aide en ligne

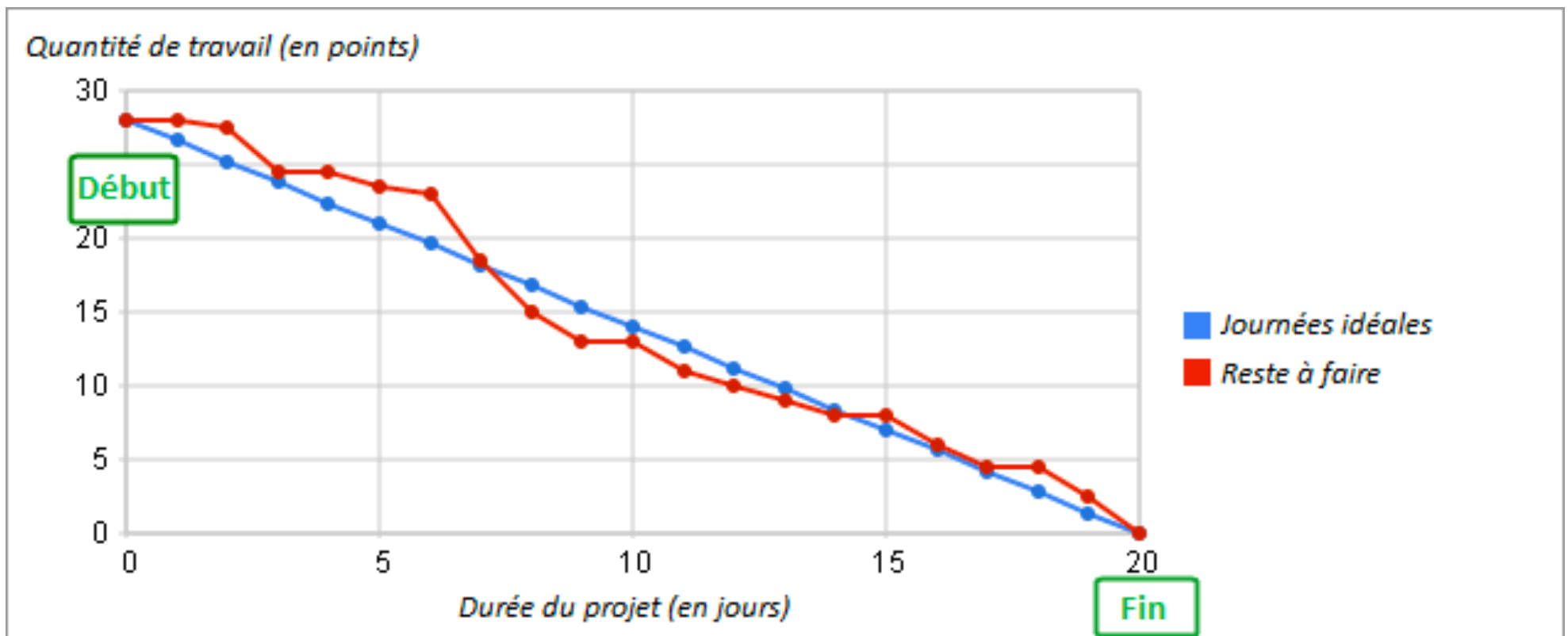
Ecrire la classe foo

Vie de la liste des tâches

- Chacun s'engage sur le travail qu'il choisit
- Le travail n'est jamais attribué par un autre
- L'estimation du reste à faire est ajustée tous les jours
- Si un travail n'est pas clair, définir une tâche avec plus de temps et la décomposer après est possible
- Mise à jour du travail restant quand il est connu

Le burndown chart

- Montre le RAF dans le backlog Sprint (par jour ou par Sprint)



Calcul de la vélocité 1/2

- Indicateur qui détermine l'effort qu'est capable de fournir une équipe de développement pour la réalisation des tâches programmées dans un sprint
- Exprimée en nombre de points
- Calculée à l'issue d'un sprint

Calcul de la vélocité 2/2

- Le Product backlog comporte un certain nombre de fonctionnalités à réaliser
- Il suffit d'additionner les points des US réalisés et livrés durant le sprint

Exemple : À la fin du sprint, l'équipe a réalisé 60 points

Généralement, la vélocité augmente au début pour se stabiliser

- Les membres ne se connaissent pas nécessairement

Elle reste une estimation de l'effort que l'équipe est capable de produire durant un Sprint

Scrum à grande échelle

- Une équipe typique c'est 7 ± 2 personnes
- Le changement d'échelle se fait par la collaboration de plusieurs équipes
- Facteurs dans la scalabilité
 - Type d'application
 - Taille de l'équipe
 - Répartition géographique des équipes
 - Durée du projet
- Scrum a été utilisé pour des projets de plus de 500 personnes

Scrum et mauvaises pratiques 1/2

Le stand-up : Une réunion de reporting au Scrum Master. L'heure de la réunion a été décidée par Le Grand Chef pour s'assurer que les salariés n'arrivent pas trop tard au bureau, vu que cette réunion est quotidienne et obligatoire.

Le Scrum Master : Il est là pour faire du flicage. Il rédige les mails qui résument sur quoi travaille chaque développeur. Il remplit par ailleurs la feuille de présence en fonction de qui est à l'heure ou en retard au stand-up. Il sert aussi de tampon aux Chefs, c'est à dire qu'ils peuvent l'accuser de tous les problèmes.

Le Product Owner : Il assigne les tâches aux développeurs. Il est également en contact permanent avec les clients, ou plutôt, il s'assure que les développeurs ne soient jamais en contact direct avec les clients. Toute communication doit passer par lui.

La Planification d'Itération : Tout le monde se retrouve en même temps dans la même salle pour s'engager individuellement sur ses propres tâches. On peut dire que les backlogs sont individuels.

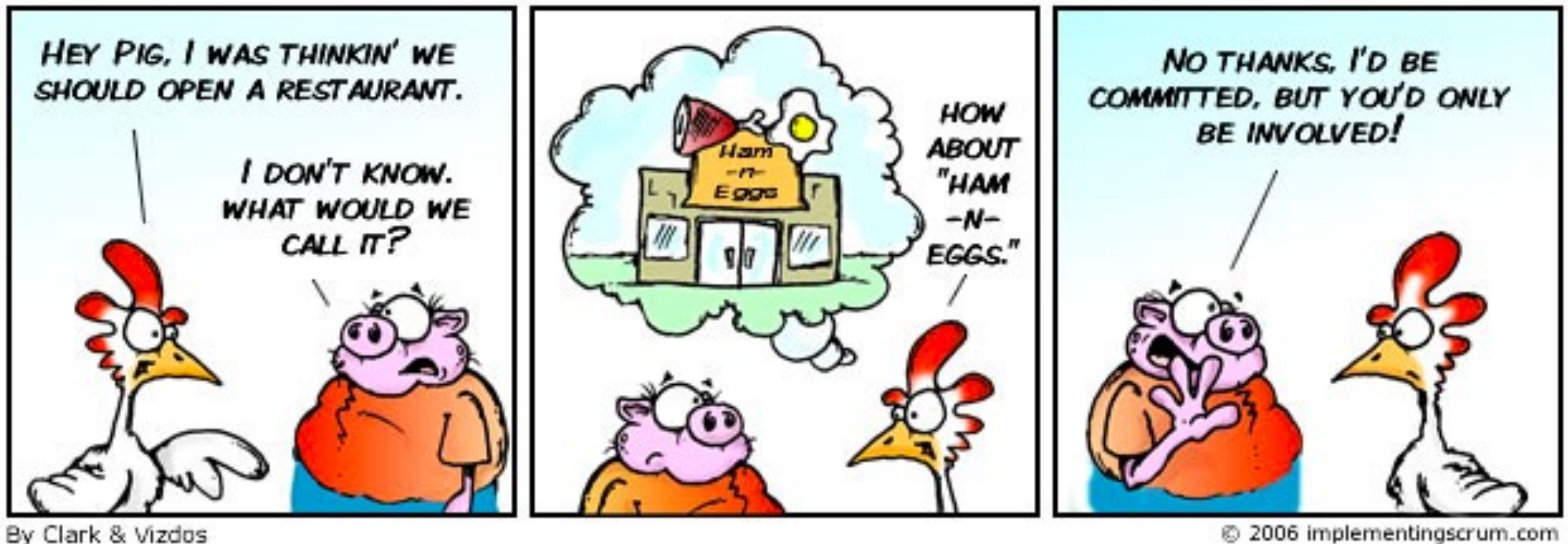
Scrum et mauvaises pratiques 2/2

La Revue d'Itération : Il y a bien un moment où chacun est convoqué pour s'assurer que les feuilles de temps sont bien remplies, que l'outil de gestion de "backlog" est à jour — à l'heure près, bien entendu. Une démo de ce qui a été construit ? Absolument pas. Pour quoi faire ?

Le processus de release : Très chaotique, c'est la punition obligatoire à tour de rôle. On a beau s'y prendre à chaque fois plus tôt, il y a toujours des problèmes qui empêchent de terminer à une heure normale. Bien entendu, c'est le vendredi soir. Et surtout, n'espérez pas trop avoir de soutien : quand vous cherchez votre N+1 pour un arbitrage de dernière minute, vous vous rendez alors compte qu'il n'est plus dans les locaux. Vous comprenez qu'il a bien fait attention à ne pas partir par le chemin qui passe devant votre bureau, on ne sait jamais, vous auriez pu lui demander d'assumer son rôle.

La Rétrospective : La réunion a été arrêtée il y a plus d'un an, ça ne servait à rien, tout le monde se plaignait mais rien ne s'améliorait. À noter, seul le Scrum Master a le droit d'organiser une Rétrospective. Il ne faudrait pas que les développeurs discutent entre eux, sans aucune oreille indiscrete dans la salle.

Scrum et Imposture ?



- The Chicken and Pig Story.....an Agile Anti Manifesto

Kanban agile : les principes

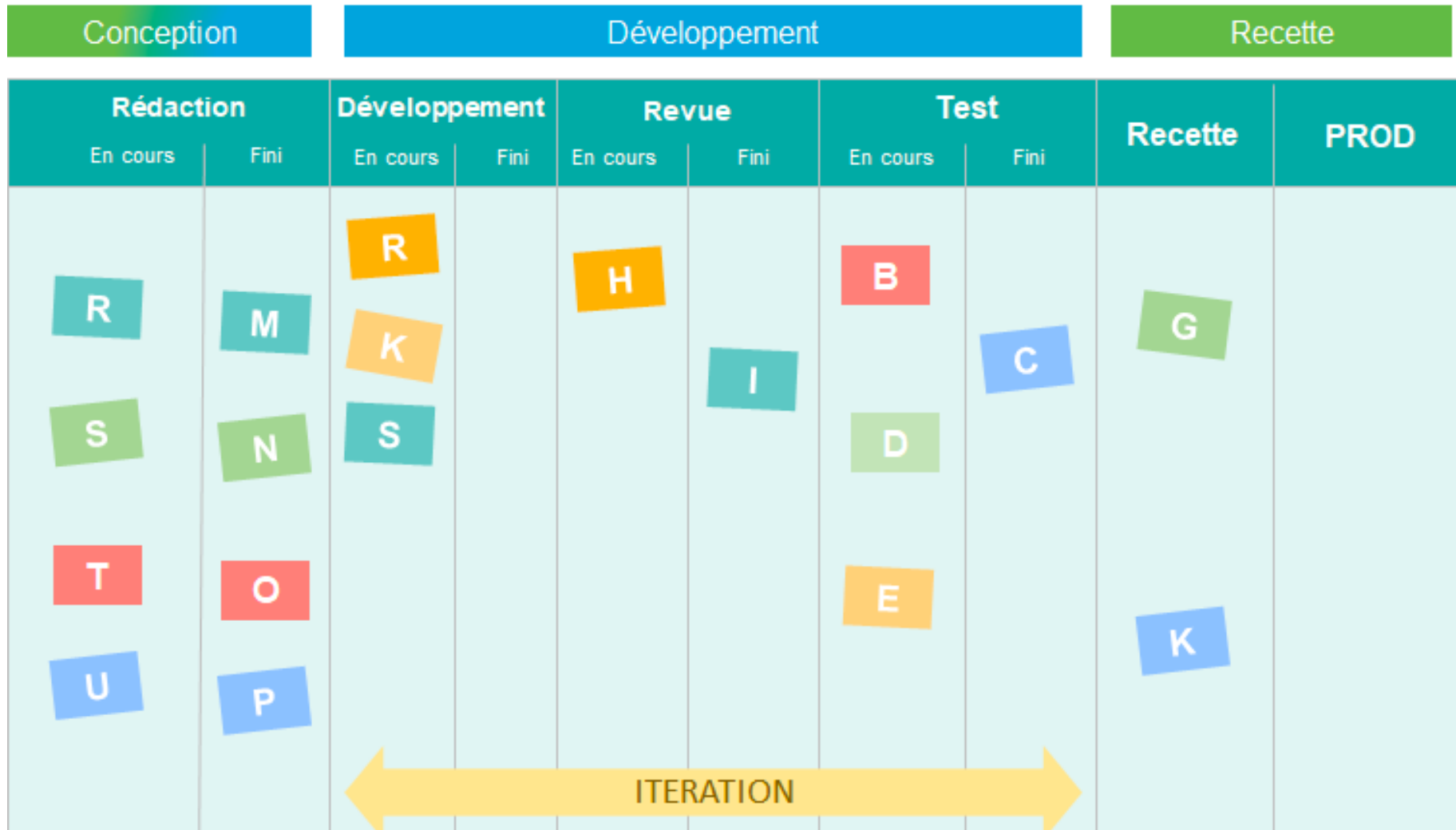
Visualisez le workflow :

- Divisez votre travail, décrivez chaque élément sur une fiche et mettez la au mur.
- Tracez des colonnes, donnez-leur le nom des étapes du workflow et placez y les éléments de travail.

Limitez le TAF (travail à faire) : fixez des limites précises indiquant combien d'éléments peuvent être placés dans chaque étape du workflow.

Mesurez et optimisez le temps de cycle (temps moyen pour traiter complètement un élément, appelé "lead time" en anglais), optimisez le processus pour que le temps de cycle soit aussi court et prévisible que possible.

Exemple de Kanban



Les apports de Kanban

Obliger à résoudre les problèmes (sinon la chaîne se bloque)

Mettre l'accent sur les goulots d'étranglement et encourager la collaboration pour les lever

Permet de fluidifier le travail

Mettre en exergue la notion de fini

Est compatible avec Scrum

Atelier : Offing the offsite

- Formation de groupes de 2 personnes
- Vous représentez des Start-up qui vont émerger sur le marché
- Start-ups composées d'un client et d'un développeur

Les règles : Deux Scénarios

- Scénario 1
 - Simulation de la création d'un logiciel, avec le client hors site (Traditionnel)
- Scénario 2
 - Simulation de la création d'un autre logiciel, , avec le client sur site (Agile)

<https://www.jamesshore.com/v2/projects/offing-the-offsite-customer>

Règles : Offing the Off-Site Customer !

Scénario 1 : *Off-site customer*

Le client en action :

- Le client a 10 min pour écrire la description complète de la vision produit
- Ce document ne doit pas contenir d'images, uniquement du texte
- Le développeur n'a pas le droit de participer / de regarder

Le client remet le document (cahier des charges) au développeur

Le développeur en action :

- Le développeur a 10 min pour réaliser le produit
- Le client n'a pas le droit de participer/ collaborer

Règles : Offing the Off-Site Customer !

Scénario 2 : *On-site customer*

Le client et développeur en action : (assis ensemble)

- L'équipe a 10 min pour réaliser le produit (duplicata de la vision produit)
- Seul le client peut examiner la vision du produit
- Seul le développeur peut utiliser un crayon (et la feuille A4)
- Communication verbale illimitée
- L'utilisation de trombone ou autre pour mesurer est possible

Le présentateur joue le rôle de *Market Place* et juge du succès ou échec de chaque entreprise

Il compare chaque produit à la vision produit et annonce l'avancée de l'entreprise selon les règles ci-dessous

Règles : Offing the Off-Site Customer !

Echec :

- L'équipe n'a pas recopié les trois éléments en ligne droite

Survie :

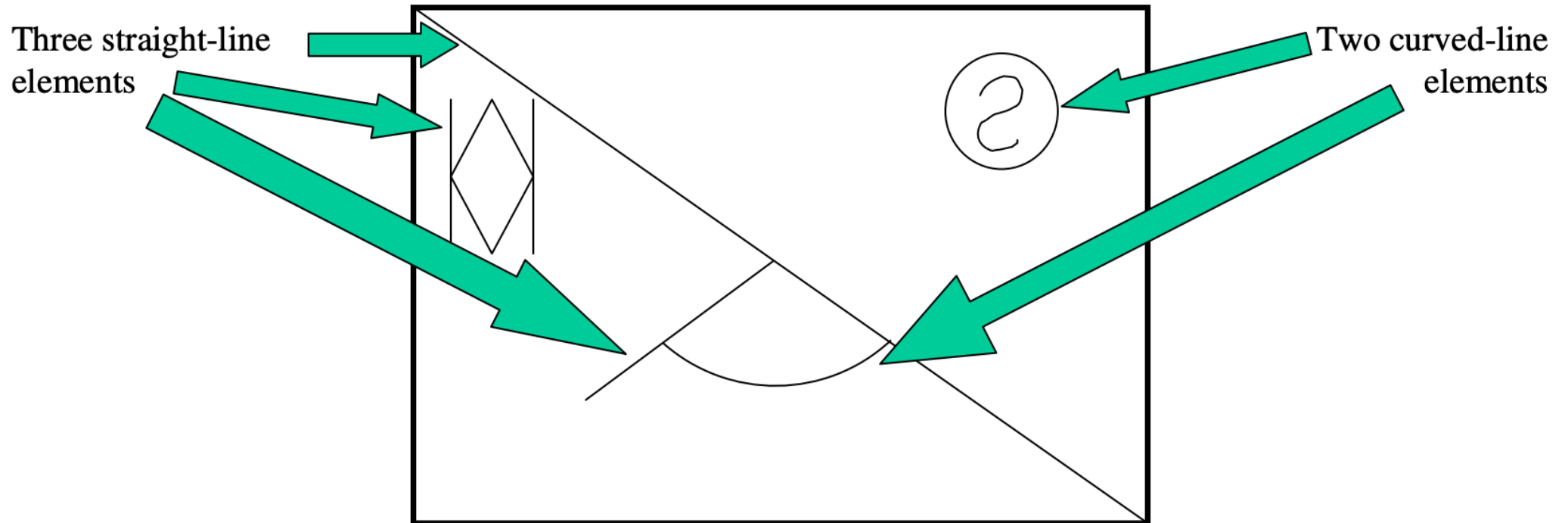
- L'équipe a recopié exactement les trois éléments en ligne droite
- Ils atteignent le marché attendu mais pas d'innovation

Succès :

- L'équipe a recopié exactement les trois éléments en ligne droite, les deux éléments en ligne courbés
- Vous êtes millionnaires, Félicitations !

Règles : Offing the Off-Site Customer !

Sample Product Vision



DÉBRIEF & FEEDBACK

- 90 % des startups échouent....
- Mise en évidence de la richesse de la conversation
 - **User Story** vs **Spécifications traditionnelles**
- Problématiques de terminologies
 - Que voulait-il dire en écrivant ces spécifications ?



- FIN -