

Comprendre la démarche Agile

Préparation à la certification scrum.org

Une formation de SQLI Institut

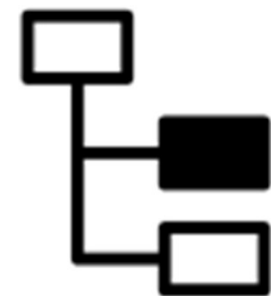


Créée et animée par Jean-Baptiste RICHARD



Sommaire

■ Introduction	3
■ Cycle projet	23
■ Rôles	33
■ Exigences	49
■ Évènements	69
■ Artefacts	81
■ Références	91
■ Mise en œuvre	109
■ Certification	123



Introduction

➔	Votre état des lieux	3
■	Présentations	9
■	Organisation	15



Démarrons par un jeu Agile

Pour établir VOTRE état des lieux

- Quelle est notre vision de l'Agilité à ce stade ?
- Notons ensemble :

⊘ Ce que vous pensez n'avoir pas totalement assimilé dans la démarche Agile (selon Scrum)

Ⓐ Ce que vous considérez comme particulièrement efficace dans l'Agilité (et Scrum)

✓ Ce que vous souhaitez avoir atteint à la fin de notre formation
-> Pour dire « ça m'a été utile ! »



Ce que vous pensez n'avoir pas totalement assimilé Dans la démarche Agile (selon Scrum)

- Cérémonies pas exhaustives
- Rôles
- Entendu parle d'agilité
Mais je ne sais pas trop ce que je sais
- Suivi par burn down chart vélocité



Ce que vous considérez comme Particulièrement efficace dans l'Agilité (et Scrum)

- Itérations = souplesse
- Implication du client pendant le cycle de validation
- Périodicité itération
- Rapidité adaptation et souplesse
- Sprint review demo tous les 15 jours = feed-back rapide



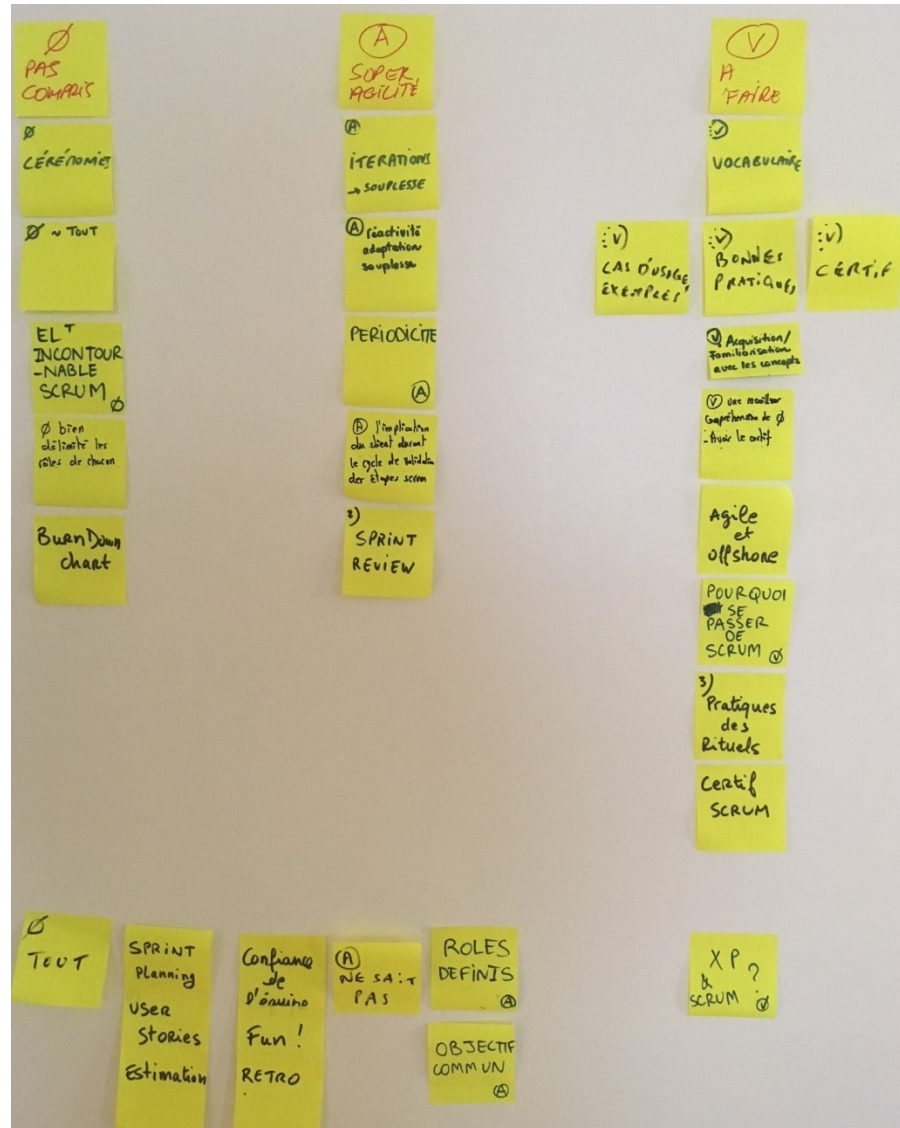
Ce que vous souhaitez avoir atteint à la fin de notre formation -> Pour dire « ça m'a été utile ! »

- Vocabulaire
- Meilleure compréhension et certification
- Cas d'usage et exemples plus coté utilisateur
- Bonnes pratiques (informaticiens pas formés)
- Certification
- Incontournable de Scrum
- Comment ou pourquoi s'en passer ?
- XP VS SCRUM
- Familiarisation avec les concepts
- Agilité offshore avec équipes distribuées (langue et décalage horaire)
- Parler de Scrum en pratique en plus de théorique
- Partir certifié mais surtout bonnes pratiques pour être efficace



Votre état des lieux

PostIt des participants – Backlog du projet *aAgile*



Introduction - Présentations

■	Votre état des lieux	3
➔	Présentations	9
■	Organisation	15



Présentons-nous

- Comment vous appelez-vous ?
- Où travaillez-vous ?
- Quel est votre activité ?
- D'où venez-vous
et quelle est votre expérience ?
- Qu'attendez-vous **CONCRETEMENT**
de cette formation ?
 - Sélectionnez 3 PostIt « prioritaires »
 - Énoncez ce qui doit différencier cette session
- Autres informations
vous concernant



Votre présentation

(Voir détail dans le document de présentation)

- **Thomas BARNABAUX**
 - Capsule technology : Program manager (chef de projet)
 - tbarnaba@qualcommmlife.com
- **Marwen DALY**
 - ILYEUM SAS : Consultant Sharepoint
 - daly.marwen@gmail.com
- **Bruno FAUVE**
 - Hachette livres : grands comptes et projets direction client
 - bfauve@hachette-livre.fr
- **Thomas FOUREST**
 - ITN (Éditeur de progiciel dans l'assurance) : Consultant
 - thomas.fourest@gmail.com
- **Aurélie SAGER**
 - Capsule technology : Chef de projet département R&D
 - aureliesager@gmail.com
- **Sing SITHAKOUL**
 - CELAD / Allianz : Squad animateur depuis janvier sur indemnisation
 - sithako@allianz.fr



Présentation de votre formateur

Jean-Baptiste RICHARD

- Consultant en Analyse et conception de Systèmes d'Informations
- Formation d'ingénieur informaticien
- développeur, chef de projets, puis conseil indépendant
- Objectif :
« Démilitariser les relations entre les utilisateurs et les informaticiens »
- Missions :
 - Diffuser les bonnes pratiques de l'Agilité
 - Modéliser pour recueillir les besoins
 - Concevoir des logiciels, bases de données, IHM, sites Web
 - Proposer des langages communs
- A conçu ces supports de formation et anime la session
- jbrichard@1024conseil.fr



Introduction - Organisation

■	Votre état des lieux	3
■	Présentations	9
➔	Organisation	15



Notre planning quotidien

- Début de la formation : 9h15
- Pause du matin : 10h30
- Repas de midi : 12h00
- Reprise après le repas : 13h00-14h00
- Pause de l'après-midi : 15h30
- Fin de la formation : 17h15
- + Discussions informelles ...



Nos différents rendez-vous

■ Jour 1

- Révision
- Objectif : Se Replonger dans l'Agilité et dans ses pratiques en nous appuyant sur la réalisation de notre formation (projet *aAgile*)

■ Jour 2

- Certification
- Objectif : Se préparer à la certification et la passer

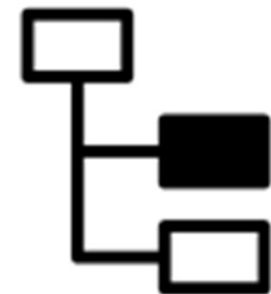


Contenu de notre formation

9 chapitres pour pratiquer en vraie grandeur

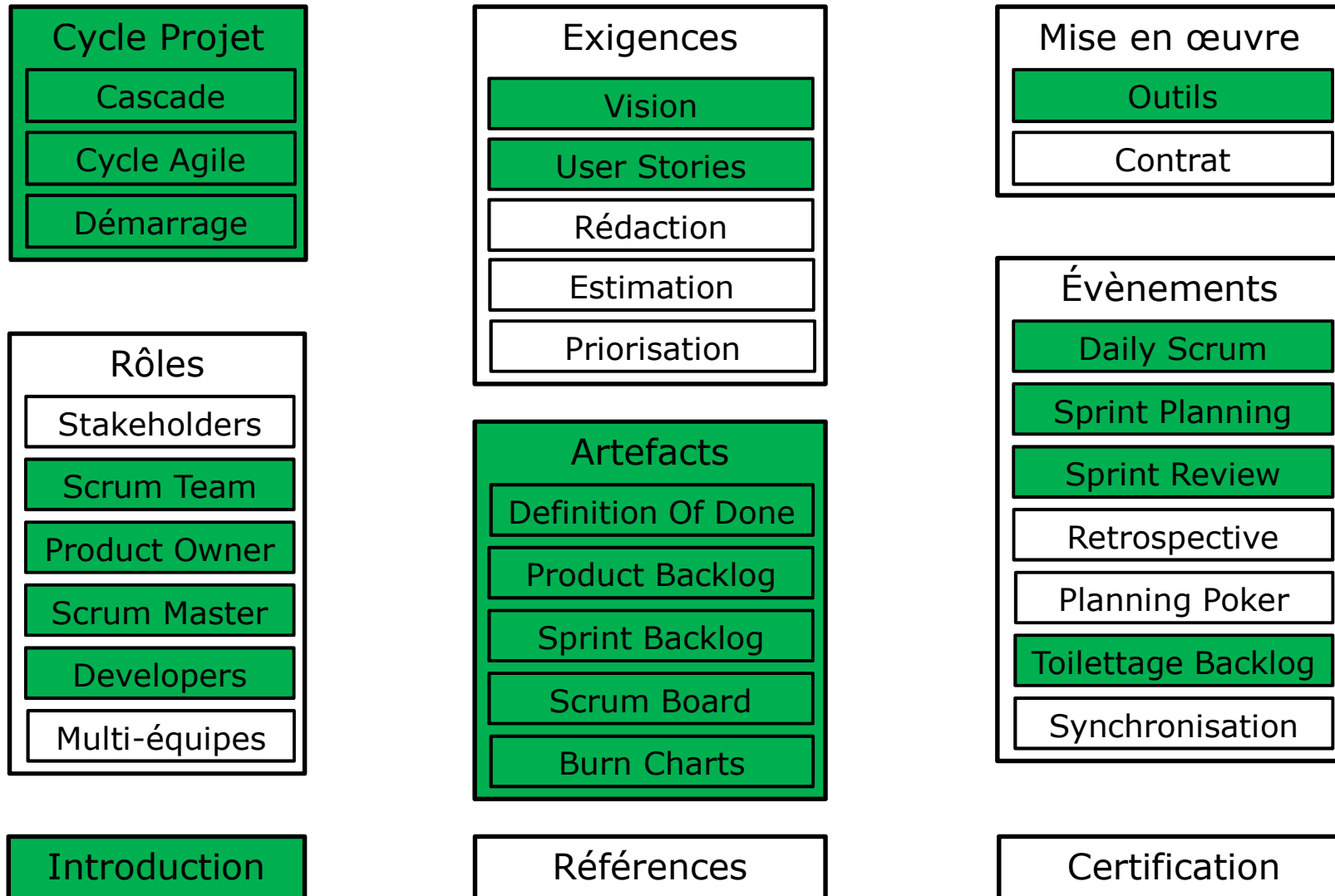
- **Introduction** 3
 - Comment va se passer cette formation
- **Cycle projet** 23
 - Approche traditionnelle VS Agile
- **Rôles** 33
 - Qui fait quoi dans l'équipe Agile ?
- **Exigences** 49
 - Comment gérer les exigences en mode Agile ?
- **Évènements** 69
 - Les « cérémonies » qui rythment un projet Agile
- **Artefacts** 81
 - Les éléments utiles pour organiser un projet Agile
- **Références** 91
 - Les origines de l'Agilité
- **Mise en œuvre** 109
 - Comment faire vivre les projets Agiles ?
- **Certification** 123
 - Préparation à l'examen de certification

Voir sommaire en page 2



Où en sommes-nous ?

Panorama des sujets



Organisons nos échanges

Quelques règles entre nous

- **Nous sommes tous là pour apprendre**
 - Soyons honnêtes et bienveillants
 - Évitions les discussions en aparté (tout le monde peut en bénéficier)
 - Concentrons-nous sur notre cours (Téléphones en mode silencieux)
- **Vous pouvez poser des questions à tout moment**
 - Nous sommes là pour progresser ensemble
 - Tout le monde bénéficiera de la réponse
- **Je ne suis moi-même pas omniscient :**
 - Je peux vous répondre « Je ne sais pas »
 - et vous proposer de creuser votre question avec le groupe



Votre support de cours

- Prenez vos notes sur ce support papier
- Le formateur complètera « en direct » la version projetée, à partir de nos échanges
- Et vous transmettra la version complète annotée Au format Acrobat (PDF)



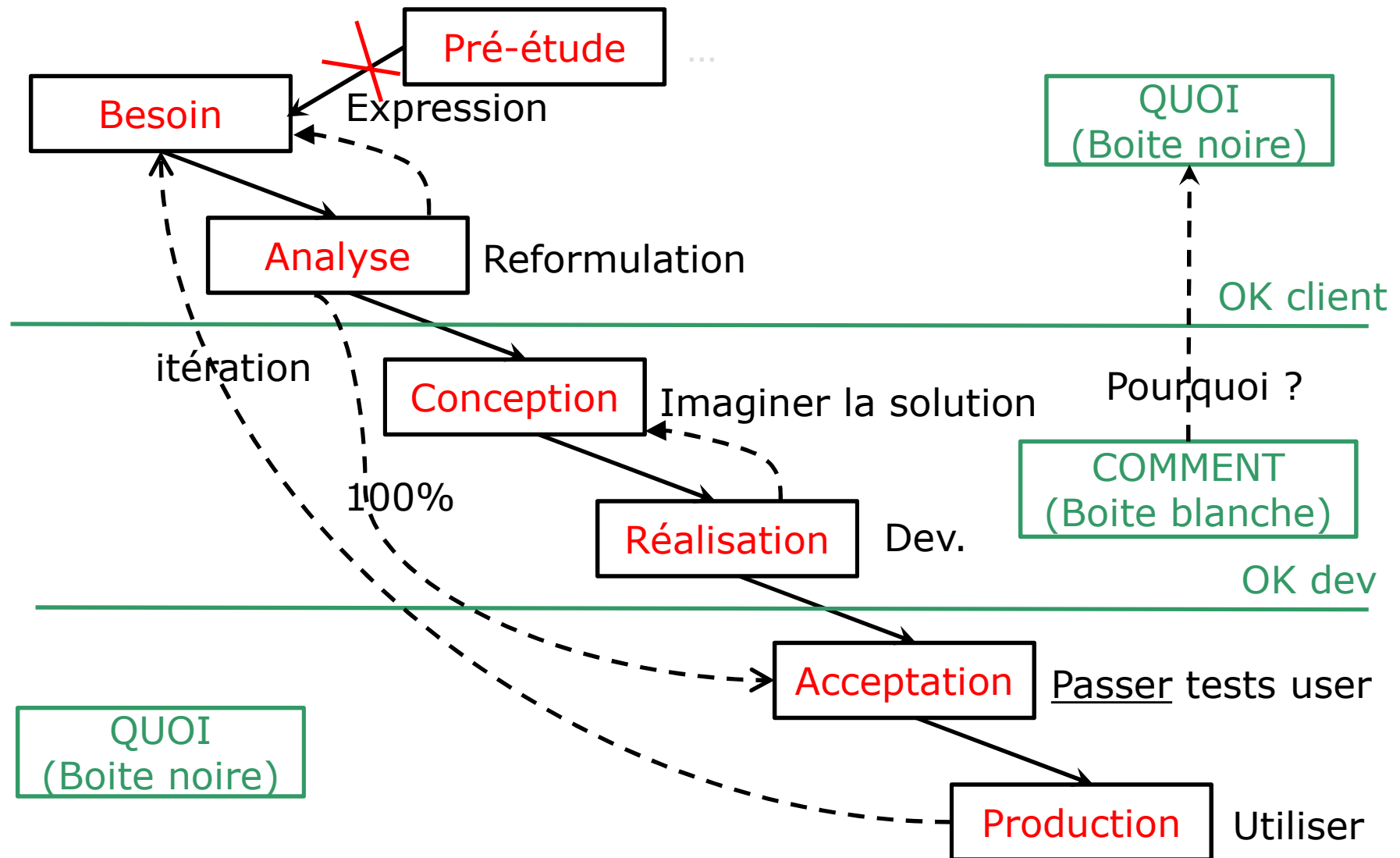
Cycle projet

■ Cycle en cascade	24
■ Cycle Agile	25
■ Paramètres fixes et variables	26
■ Prototypage	27
■ Initialisation	29
■ Comparatif des approches	29



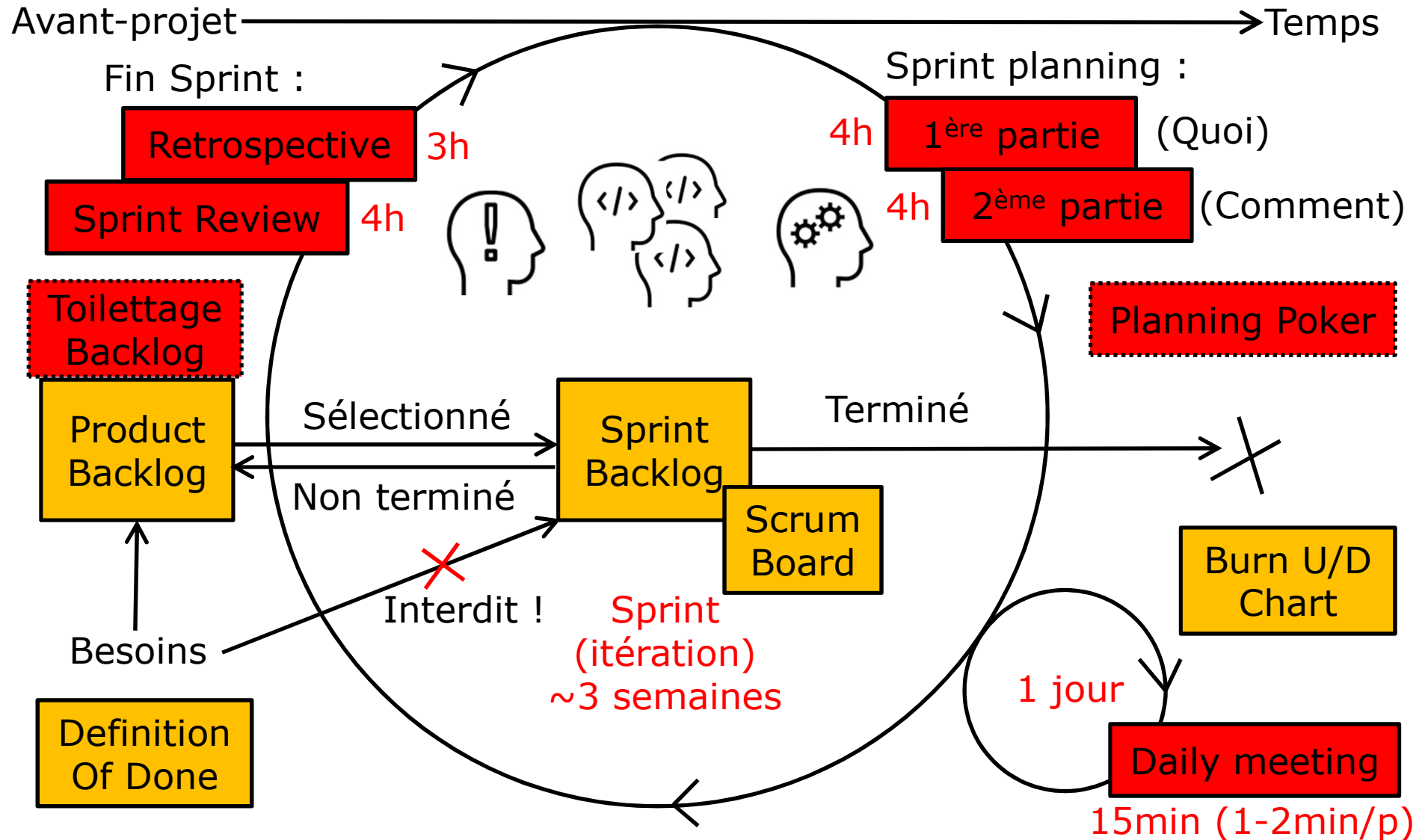
Le cycle de vie « traditionnel » (Cascade, V...)

Tel que nous le connaissons



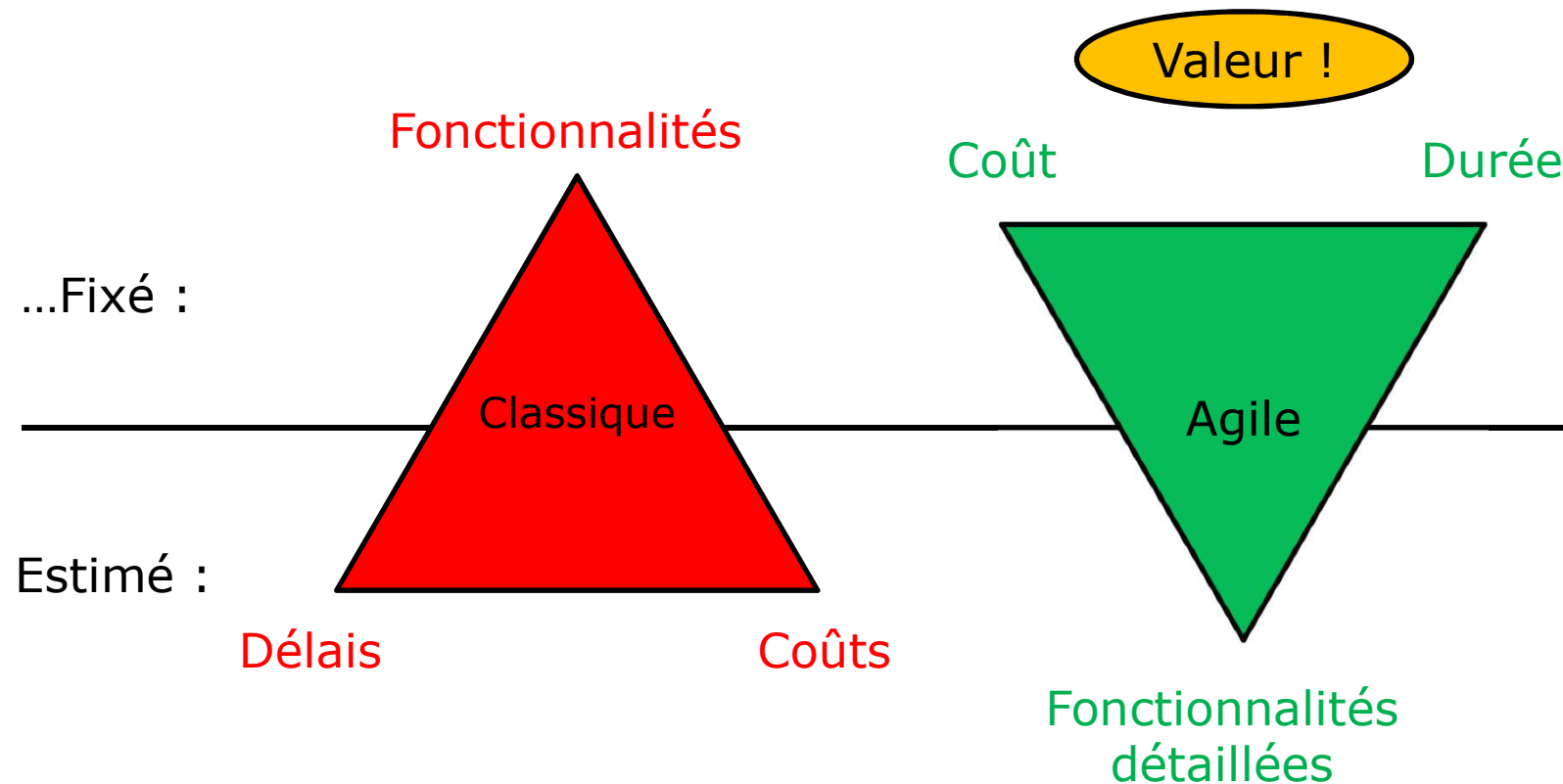
Le cycle de vie « Agile » type SCRUM

Principes et éléments

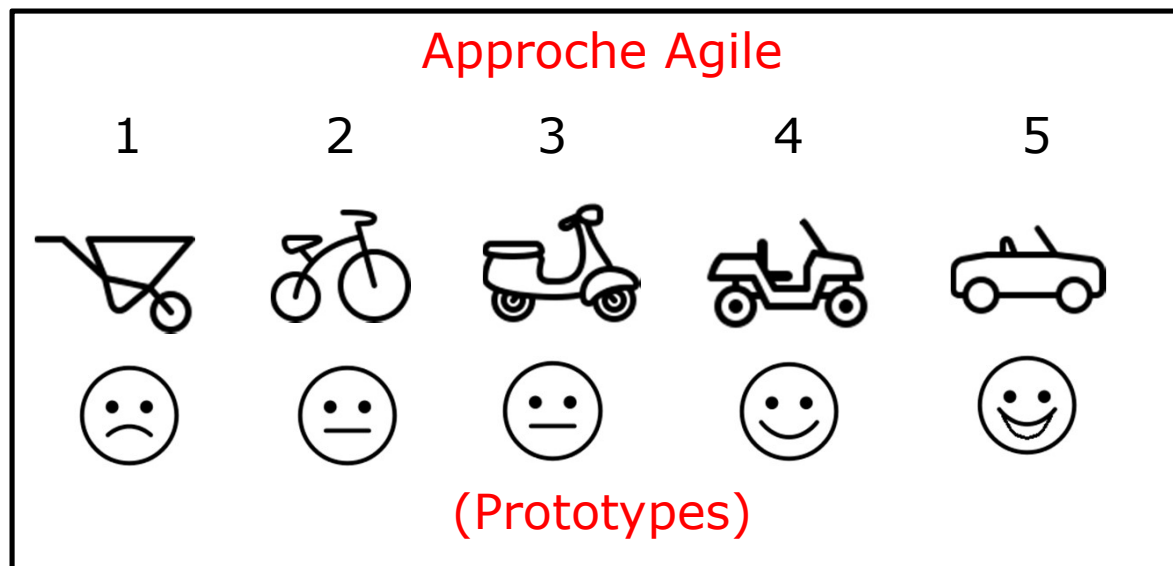
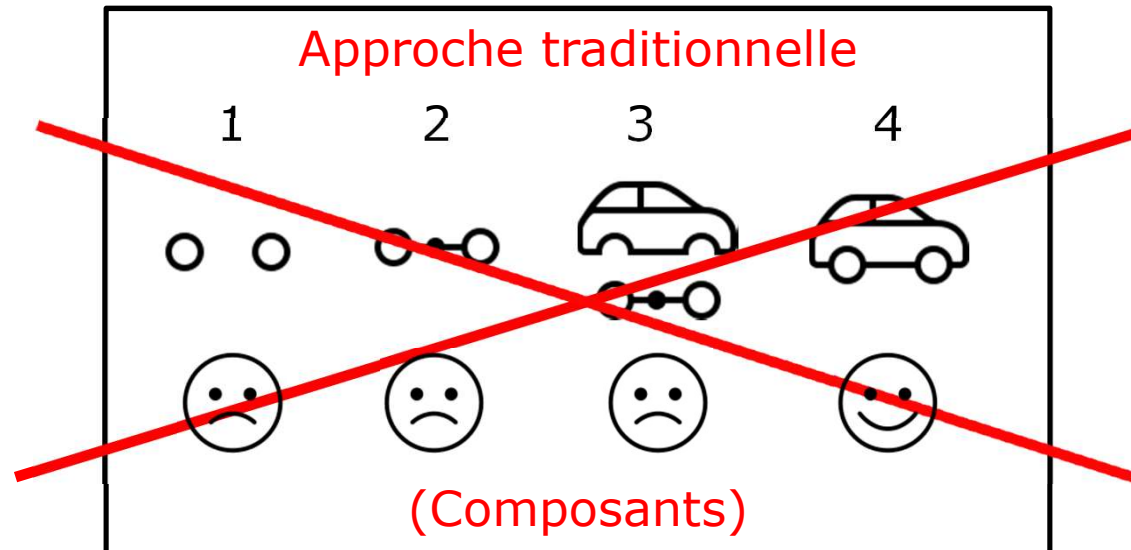


Les paramètres du projet

Ce qui est fixe, ce qui peut varier



Construction par prototypage successif et pas composant par composant

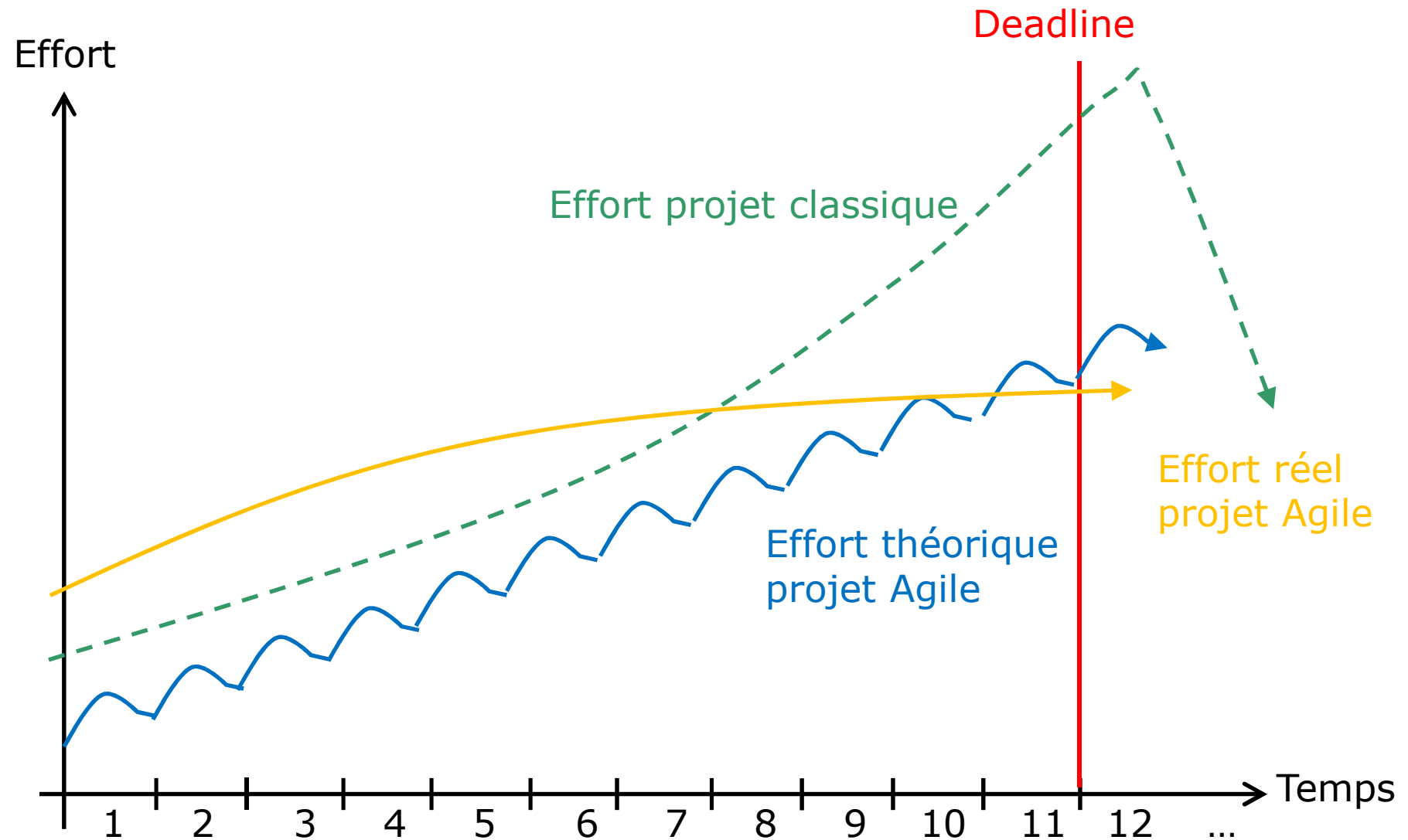


*D'après
Henrik KNIBERG*



Effort dans le temps

Comparatif entre Agilité et Traditionnel

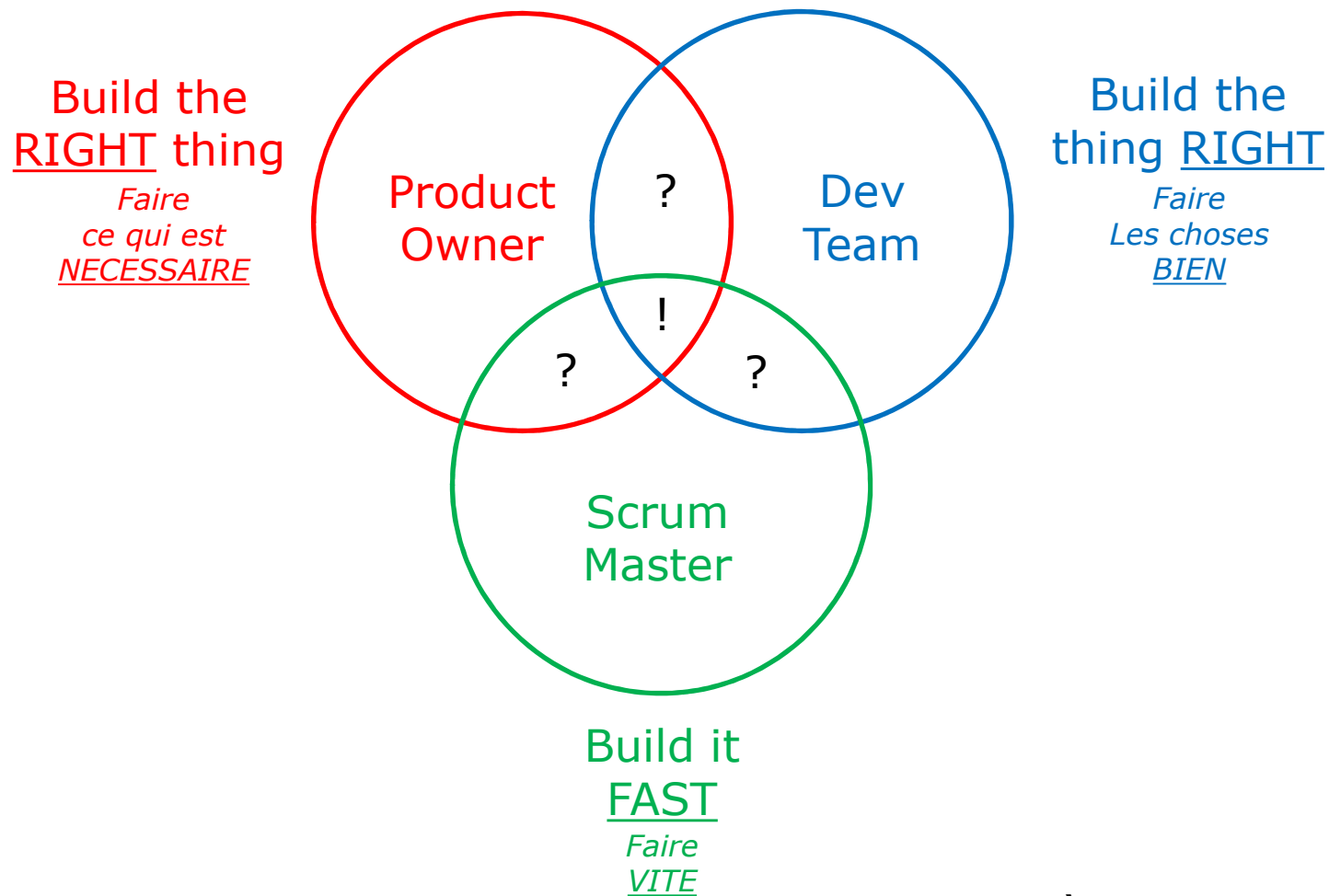


Rôles

■ Stakeholders	34
■ Objectifs	38
■ Scrum Team	39
■ Product Owner	40
■ Scrum Master	41
■ Développeurs	42
■ Capacités	44
■ Multi-équipes	46



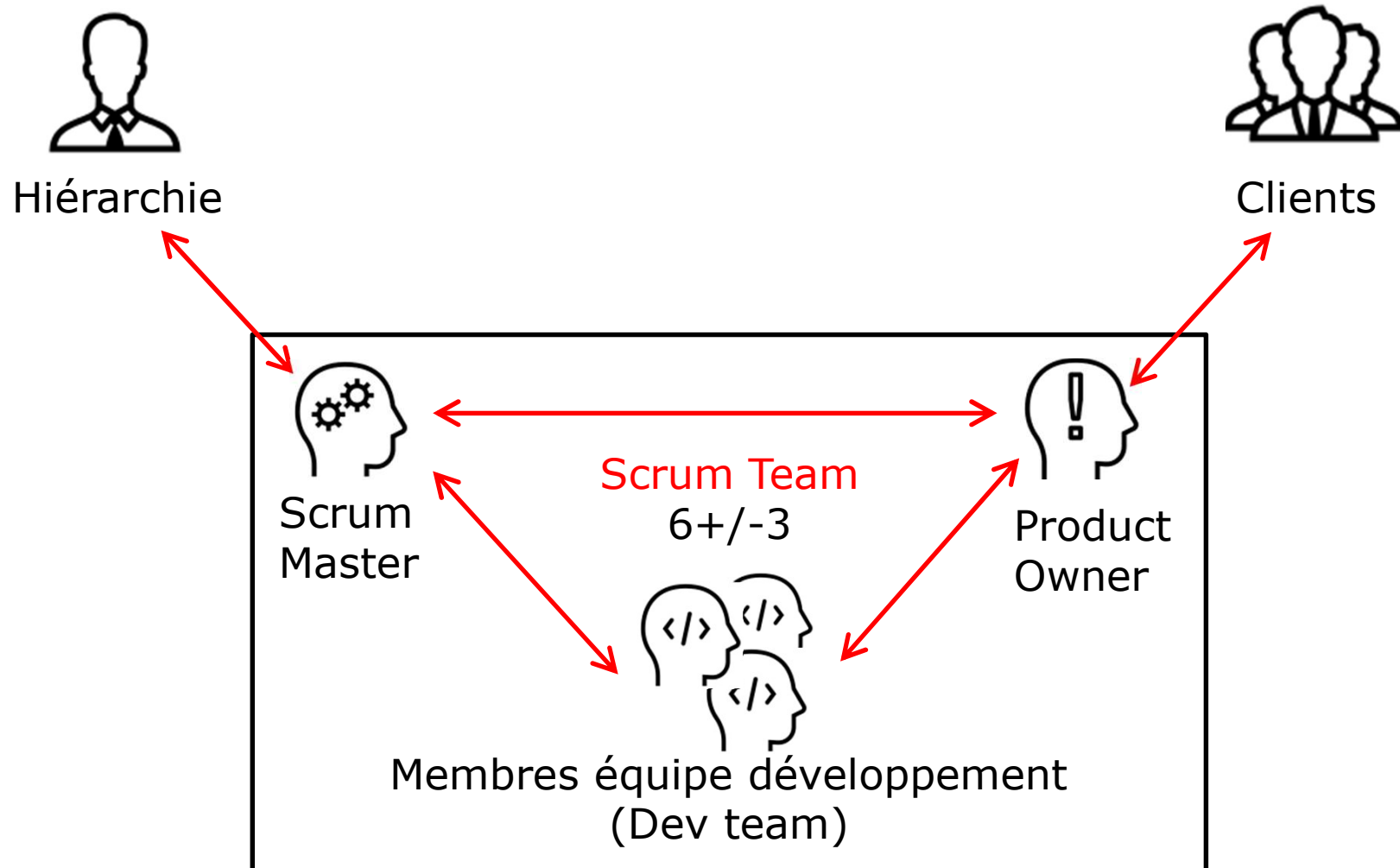
Objectifs des membres de l'équipe



(D'après Henrik KNIBERG)



Les rôles de l'équipe Scrum (Scrum Team)



Le Product Owner

- Est responsable de la valeur produit
- Transmet la « Vision » du produit à l'équipe
- Garantit les intérêts des utilisateurs
- Gère le coût total de possession du produit
- A le dernier mot sur les priorités
- Est passionné par le produit
- Est reconnu comme « autorité »
- Synchronisation (si multi-équipe)



Le Scrum Master

- (Coach Agile) -> Peut être joué par quelqu'un autre
- (Garant du succès de Scrum) -> Au final, ce sera l'équipe !
- Garantit le respect des règles de Scrum (surtout au début)
- Facilitateur
(anime les cérémonies si besoin)
- Protège l'équipe projet
des agressions de l'extérieur !!!
- Supprime les impédiments !!!
- Communique avec la hiérarchie
(Direction)
- Synchronisation
(si multi-équipe)



Les « Developers » (Dev Team)

Membres de l'équipe de développement

- Convertissent les éléments du Product Backlog en incréments du produit « Fini » potentiellement livrable
- On ne gère pas la vélocité en staffant
- Plurifonctionnels
- Pas de « titres » dans l'équipe
- Pas de sous-équipe
- Responsabilité GLOBALE sur le produit (Personne n'est propriétaire d'une US)



Quels rôles jouons-nous dans NOTRE équipe ici ?

Pour réaliser notre formation Agile

■ Équipe du projet *aAgile* (notre Formation)

☐ Product Owner

- Formateur : Jean-Baptiste RICHARD

☐ Scrum Master

- À choisir parmi les stagiaires

☐ Membres de l'équipe

- Stagiaires : VOUS tous !

■ Autres parties prenantes (stakeholder)

☐ Client (qui paye ?)

- DSI ?
- Responsable formation

☐ Experts

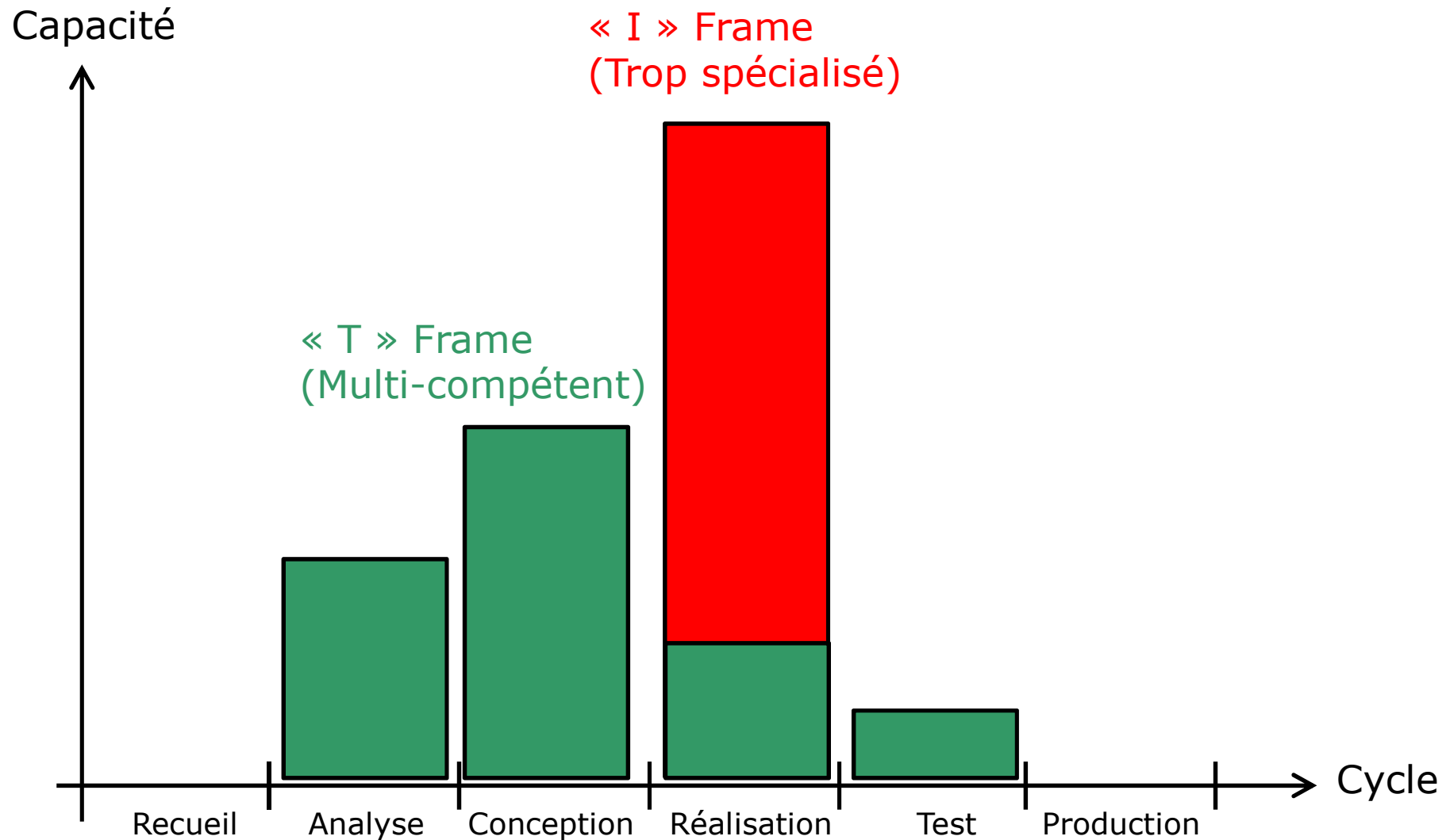
- Référents de l'Agilité : Ken SCHWABER, Henrik KNIBERG...

☐ Autres

- Ministre chargé(e) de la formation professionnelle

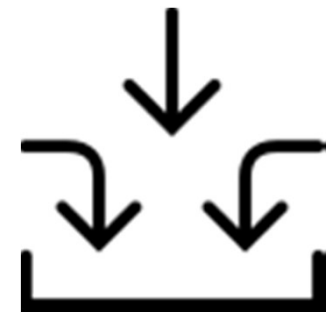


Capacités des membres de l'équipe Scrum



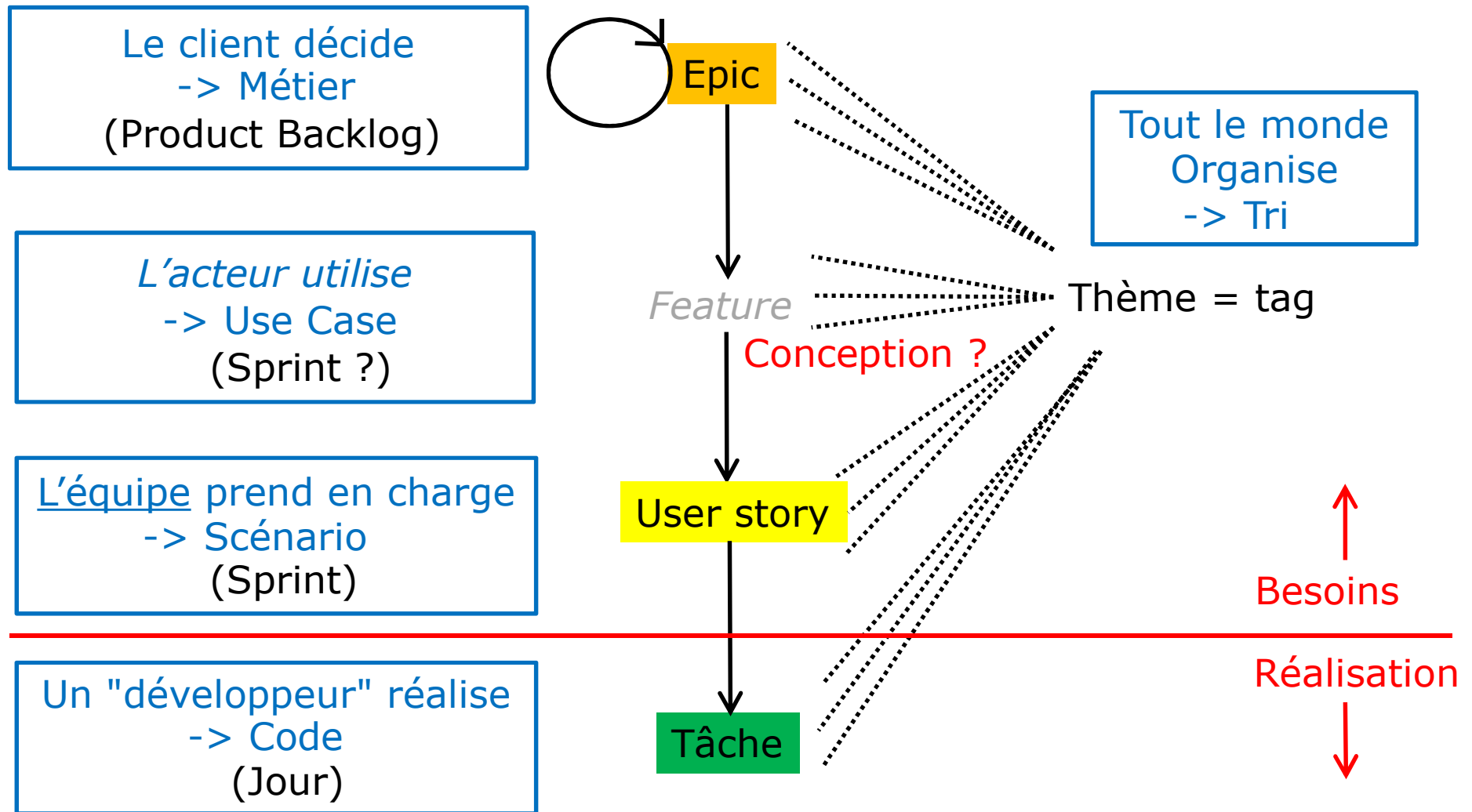
Exigences Agiles

■	Vision produit	50
■	Périmètre	52
■	Use Case	53
■	Typologie	54
■	Granularité	56
■	Rédaction	58
■	Estimation	61
■	Priorisation	62
■	Qualité	66



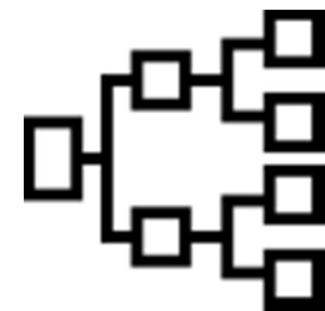
Organiser les User stories

Granularité des besoins Agiles



Exemples de granularités des User stories, Epics, Features, Tâches, Thèmes

- Epic
 - Client = Votre société
 - Démarrer l'Agilité dans ses projets
- Feature
 - Acteur = Participant
 - Pratiquer un cycle Agile complet
- User Story
 - Équipe = Nous
 - Maîtriser la « cérémonie » de Planning Poker
- Tâche
 - Membre = Formateur
 - Expliquer le principe, donner un exemple, pratiquer, débrief
- Thème
 - Gestionnaire = Pédagogue
 - Exemple, exercice, moyen mnémotechnique, question, difficulté



Estimation des User Stories

Pourquoi utiliser les Points (plutôt que les jours/h) ?

■ Si on estime en jours/hommes

- On est obligé de réévaluer les estimations de toutes les user stories à mesure que l'on constate que l'équipe devient plus (ou moins) efficace
- En appliquant des pourcentages, pour tenir compte des temps de réalisation constatés
- Jusqu'à obtenir des estimations en nombre de jours très éloignées de notre compréhension (23.5, 7.93...)

■ Si on estime en Points

- Il suffit de calculer notre vélocité en Points, à chaque fin de Sprint
- en tenant compte de ce qu'on a réellement été en mesure de réaliser
- pour pouvoir l'appliquer au prochain Sprint, sans nécessiter de toucher aux estimations de toutes les User Stories

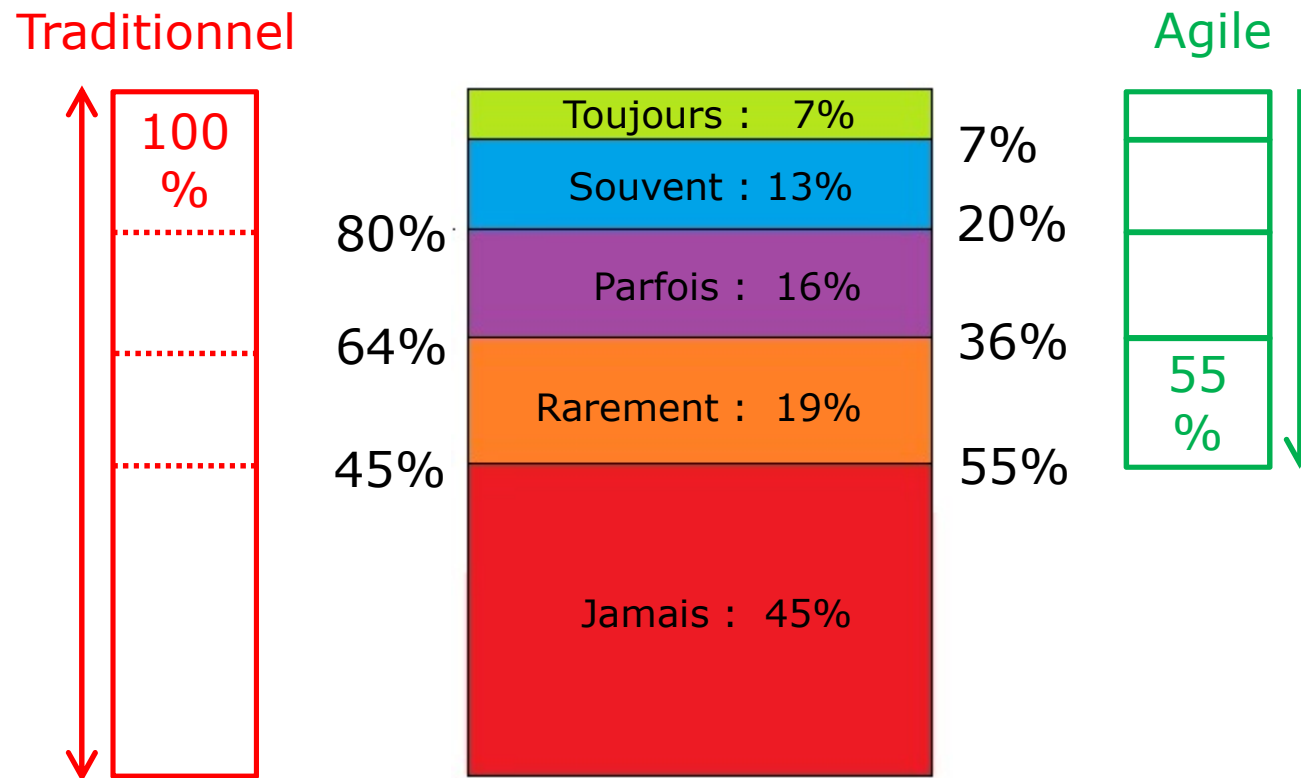
■ On peut aussi estimer en « tailles de Tee-Shirts »

- XS, S, M, L, XL...
- XS=1/4 - S=1/2 - M=1 - L=2 - XL=4



Taux d'utilisation des fonctionnalités des logiciels

Utilisation des fonctionnalités d'un système :



D'après : Standish Group (2000 projets de 1000 sociétés)



Évènements

■ Daily Scrum	70
■ Planning Poker	71
■ Sprint Planning	72
■ Sprint Review	74
■ Retrospective	75
■ Toilettage du Backlog	76
■ Scrum de Scrum	77
■ Obeya	78
■ Récapitulatif	79



Daily Scrum

Réunion debout, Daily/Standup meeting

- **Quand**
 - Au début (à la fin ?) de chaque jour travaillé, à heure fixe
 - 15 minutes maximum (1 à 2 minutes par personne)
- **Qui**
 - Tous les membres de l'équipe de développement sont présents
 - Product Owner optionnel (écoute)
 - Animée par le Scrum Master si besoin
- **Objectif**
 - Transparence = Tout le monde au même niveau d'info
- **3 questions**
 - Qu'ai-je fait depuis le dernier Daily Scrum ?
 - Que vais-je faire AUJOURD'HUI
 - Qu'est-ce qui entrave ma progression ou celle de l'équipe (impédiment) ?
- **Mise à jour du Scrum Board**
 - A Faire, En Cours, Terminé...
- **Règle TRES IMPORTANTE**
 - Réunion TRES courte (1 à 2 minutes par participant = 15 minutes)
 - Ne pas traiter une question en détail -> Prévoir une autre réunion



Sprint Planning – 1^{ère} partie

QUOI

- **Quand**
 - Au début de chaque sprint
 - ½ journée maximum (4 heures)
- **Qui**
 - Product Owner (Chef d'orchestre)
 - Équipe et Scrum Master
 - + experts...
 - Et C'EST TOUT !
- **Objectif**
 - Définir QUOI faire = Fixer le BUT du Sprint
 - Sélection dans le Product Backlog (priorisé)
 - En tenant compte de la dernière vélocité constatée



Sprint Planning – 2^{ème} partie

COMMENT (+ QUI + QUAND)

- **Quand**
 - Juste après la 1^{ère} partie du Sprint Planning
 - ½ journée maximum (4 heures)
- **Qui**
 - Équipe (Chef d'orchestre)
 - Scrum Master
 - Product Owner
 - + experts et intervenants souhaités
- **Objectif**
 - Définir COMMENT faire
 - Décomposer les (premières) Stories en tâches
 - Allouer les tâches aux membres de l'équipe



Sprint Review

Démonstration et retours client

- Quand
 - A la fin du Sprint
 - ½ journée maximum (4 heures)
- Qui
 - Product Owner
 - Équipe de Développement
 - Scrum Master
 - + Parties prenantes invitées par le PO
- Objectif pour l'équipe :
 - Présenter les fonctionnalités terminées ou non (justification)
 - Démonstration avec explication et réponses aux questions
 - Discussion sur ce qui s'est passé : Problèmes, résolutions
 - Étude et toilettage du Product Backlog (réalisable, évolutions)



Rétrospective

L'équipe étudie ce qui peut être amélioré

■ Quand

- ☐ A la toute fin du Sprint
- ☐ Un peu moins d'½ journée maximum (3 heures)

■ Qui

- ☐ Équipe
- ☐ Scrum Master (Animateur)
- ☐ Product Owner (facultatif, mais souhaitable)
- ☐ ET C'EST TOUT !

■ Objectif

- ☐ Apprendre (flux d'apprentissage)
(Parallèle au flux des fonctionnalités)

■ 2 questions principales

- ☐ Qu'est-ce qui s'est bien passé durant ce sprint ?
(Dont remerciements)
- ☐ Qu'est-ce qui peut être amélioré lors du prochain sprint ?

■ Jeux

- ☐ Ressenti, Roue (Start doing / Keep doing / Stop doing), bouquets
Team building, Brainstorming, démos, hackathon, Soap opéra tests, ...



Toilettage du Product Backlog (Grooming)

Revue, description et priorisation régulière

■ Quand

- ☐ A tout moment, quand le besoin s'en fait sentir
- ☐ Souvent lors du Sprint planning
- ☐ Peut aussi être programmé régulièrement

■ Qui

- ☐ Product Owner (seul ou en chef d'orchestre)
- ☐ Équipe ou membre de l'équipe, client, toute personne impliquée

■ Objectif

- ☐ Maintenir la liste du reste à faire au plus proche des besoins clients
- ☐ Rendre les user stories « ready » pour être sélectionnables dans un sprint

■ Actions concrètes sur le backlog

- ☐ Ajout d'informations
- ☐ Priorisation
- ☐ Suppressions...



Scrum de Scrum

Pour coordonner le travail des Scrum Team

■ Quand

- ☐ Typiquement 2 ou 3 fois par semaine (voire quotidiennement)
- ☐ Généralement tenue juste après la fin des Daily Scrum des équipes
- ☐ Timeboxé à 15 minutes (comme un Daily Scrum)

■ Qui

- ☐ un membre de chaque Équipe de Développement
- ☐ Pas nécessairement le Scrum Master (éventuellement le Product Owner)
- ☐ Peut regrouper les Équipes Fonctionnalités et les Équipes Composant

■ Objectif

- ☐ Faire ressortir les questions de dépendance et coordination
- ☐ Entre fonctionnalités et entre équipes

■ Les 3 questions deviennent

- ☐ Qu'a fait votre équipe depuis le dernier Scrum de Scrum ?
- ☐ Que va faire votre équipe d'ici le prochain Scrum de Scrum ?
- ☐ Qu'est-ce qui entrave la progression de votre équipe ?



Obeya - Réunions « grande salle »

Pour synchroniser le backlog entre équipes

■ Quand

- Régulièrement, pour préparer les Sprint plannings de chaque équipe
- Possible pour coordonner des revues de Sprint
- Également possible pour réaliser des rétrospectives « globales »

■ Qui

- Toutes les Scrum Team se réunissent dans une même grande salle
- Chacune avec son Product Owner

■ Objectif






































- Diminuer la charge de gestion de la communication entre les équipes

■ Fonctionnement

- La réunion commence par un introduction d'état des lieux faite par le Product Owner principal
- Chaque équipe se réunit dans un endroit différent de la salle
- Des intermédiaires font la navette entre les équipes pour apporter des demandes de travail, de l'aide, etc.



Récapitulatif de présence des rôles Lors des différentes cérémonies

 = Chef orchestre  = Participe  = Auditeur	 Product Owner	 Scrum Master	 Équipe Dev	 Expert	 Autre
Daily Scrum	 / 	 / 	 / 		
Planning Poker	 / 			()	
Sprint planning #1					()
Sprint planning #2				()	
Sprint revue					
Retrospective					
Backlog grooming		()	()		



Artefacts

■	Definition Of Done	82
■	Product Backlog	84
■	Sprint Backlog	85
■	Scrum Board	86
■	Burndown Chart	88
■	BurnUp Chart	89



Definition Of Done (DoD)

- **Règles communes à l'ensemble des User Story pour pouvoir les déclarer « TERMINEE »**
 - Checklist de tâches (éventuellement ordonnées selon un processus)
 - Affichée très clairement dans la « War room » (s'impose à tous)
 - Une User Story ne peut être démontrée en revue que si elle satisfait à la Définition Of Done
- **Inclut par exemple (pour un incrément à livrer en fin de Sprint) :**
 - Développement logiciel livré en environnement de test
 - Tests unitaires OK 100%
 - Tests d'intégration validés
 - Complexité cyclomatique du code mesurée par l'outil maison inférieure à 35%
 - RSSI OK sur les exigences de sécurité
 - Documentation admin minimale rédigée
- **Plusieurs DoD éventuelles**
 - Pour un incrément, une release, un déploiement
 - Pour un site Internet, une appli smartphone, un Web service...
- **Définition of « UnDone »**
 - Pas nécessaire pour dire « C'est terminé »
 - Au choix (à préciser) :
 - Jamais fait
 - A faire dans un Sprint suivant
 - Délégué à une autre équipe / projet



Notre Definition Of Done

Pour notre formation – Projet *aAgile*

■ Done :

- ☐ On a parlé du thème
- ☐ 1 slide présenté et annoté avec nos discussions
- ☐ 100% des participants disent « Principe compris »
- ☐ Terminologie fournie (FR et EN)
- ☐ Un exemple, voire une anecdote
- ☐ Le formateur s'engage à avoir délivré la valeur
- ☐ Le formateur a répondu à 100% des questions
- ☐ Références, dont ouvrages, sites web, autres slide si existant
- ☐ Les stagiaires déclarent être capables d'essayer cette pratique
- ☐ Le support annoté a été envoyé à tous les stagiaires
- ☐ Savoir si le sujet fera l'objet d'une question dans la certification
- ☐ Savoir s'il y a des questions pièges sur le sujet (théorie VS pratique)

■ UnDone :

- ☐ Bibliographie exhaustive
- ☐ Le support envoyé a été lu par 100% des stagiaires
- ☐ Développement d'un vrai projet sur le sujet



Product Backlog

Reste à faire global du produit

- Tout ce qui pourrait être UTILE pour le produit
= Besoins
 - Fonctionnalités
 - Technologies
 - Améliorations
 - Corrections de défauts
- Sous la responsabilité du Product Owner
 - Origine UNIQUE des exigences
- Liste ordonnée du plus au moins vital
 - Plus un besoin est prêt à être réalisé, plus il est détaillé
 - Description
 - Valeur
 - Estimation de l'effort
 - Jamais terminé / toiletté régulièrement



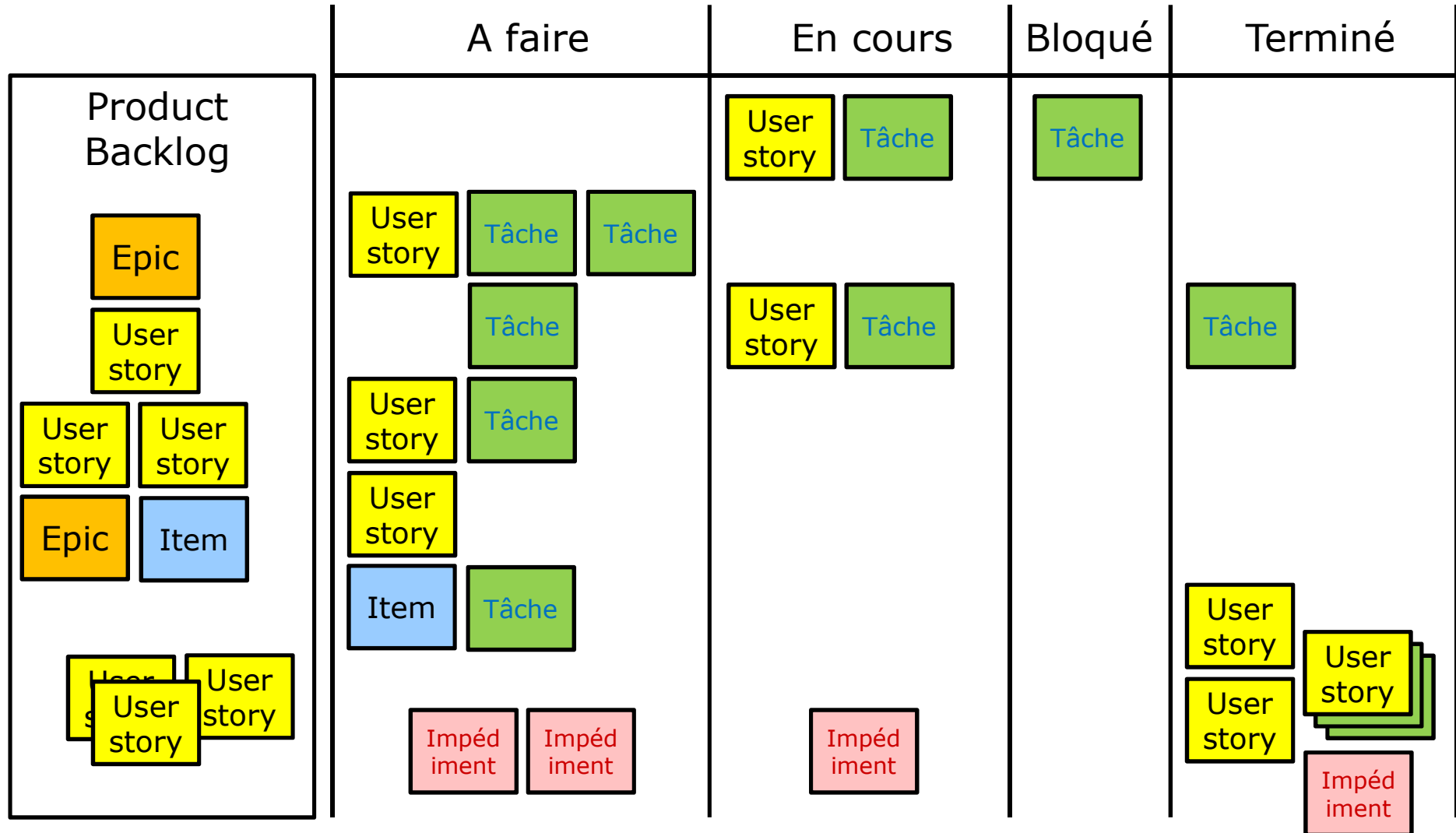
Sprint Backlog

Reste à faire de l'itération en cours

- Ensemble des éléments du Product Backlog sélectionnés pour CE sprint
 - Permettant d'atteindre l'objectif du sprint
 - Planifié pour fournir l'incrément du produit
- Liste des tâches visible
 - Propriété de l'équipe de développement
 - Mise à jour par elle pendant le Sprint



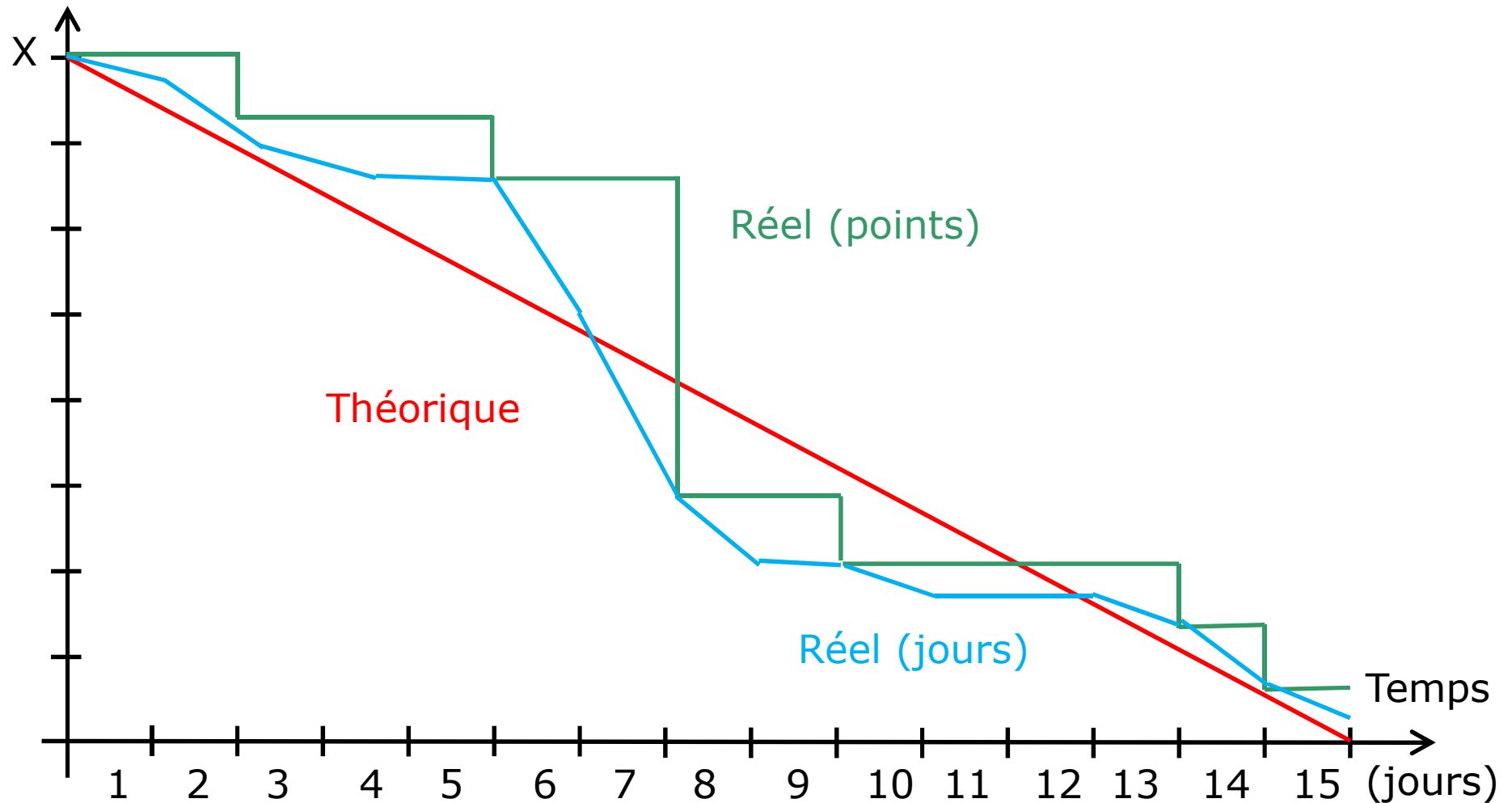
Scrum Board



Burndown chart

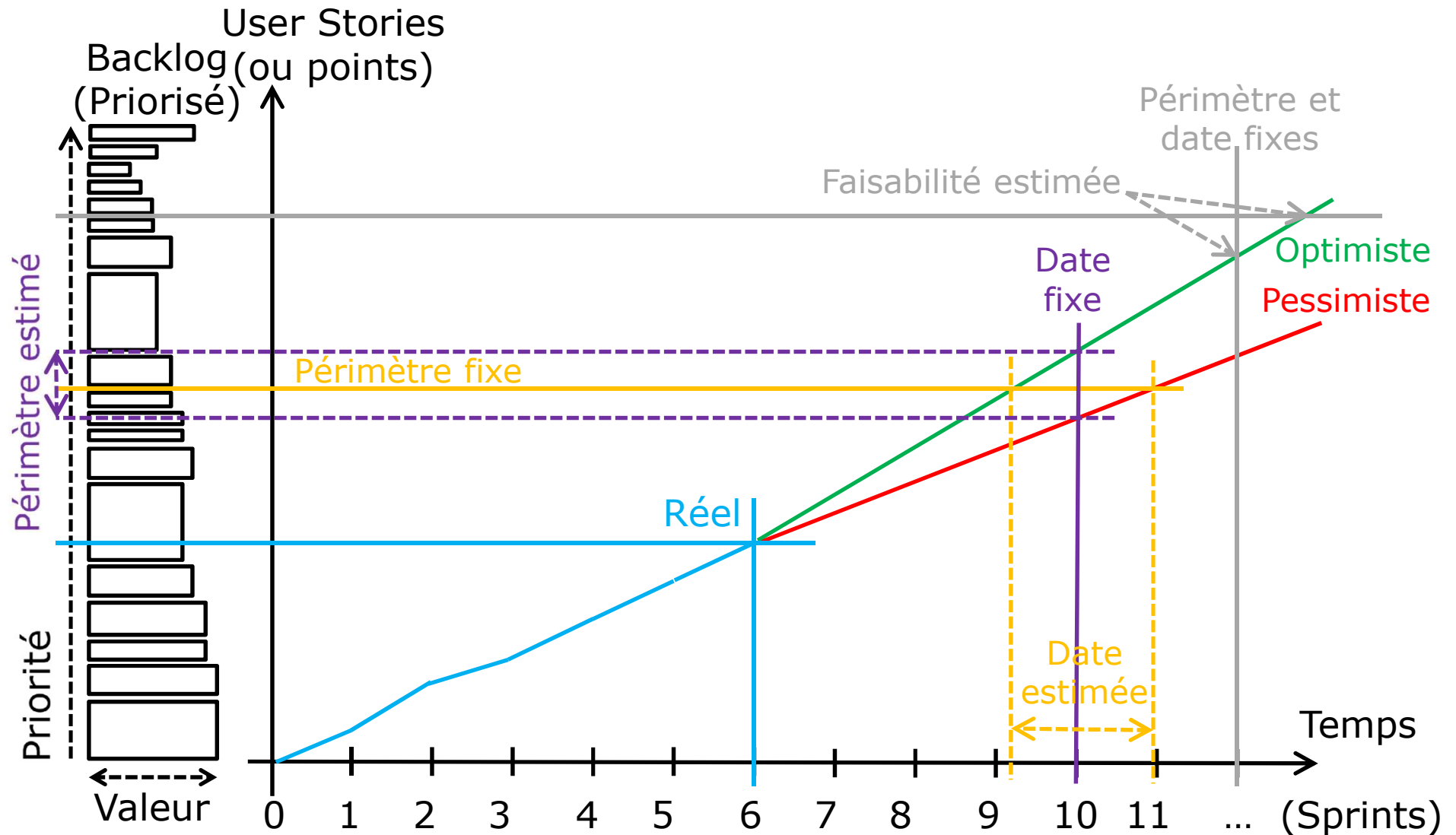
Pour un sprint

Reste à faire
(Vélocité)



BurnUp chart

Pour la totalité du projet



Références

■ Historique	92
■ Piliers	93
■ Manifeste	94
■ 12 principes	95
■ Bibliographie	96
■ Sites Web	97
■ Une mode ?	98
■ Autres démarches	99
■ Glossaire	107



Les piliers de l'Agilité

■ Transparence

- Les aspects significatifs du processus doivent être visibles pour les responsables de ses produits

■ Inspection

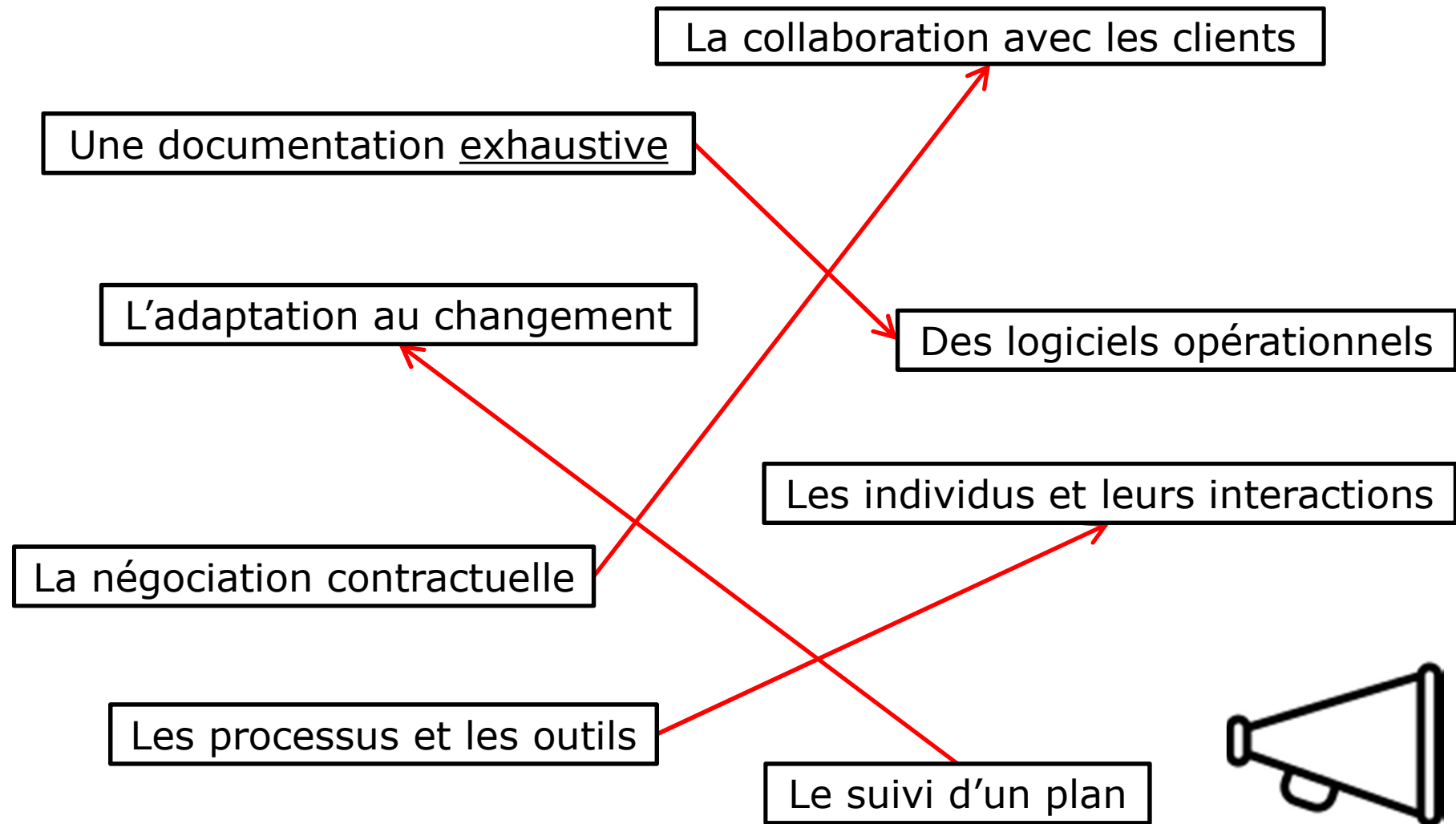
- Chaque aspect du processus doit être inspecté fréquemment pour détecter les variations inacceptables

■ Adaptation

- Les ajustements doivent être faits aussi rapidement que possible pour minimiser les déviations ultérieures



Le manifeste Agile



Les 12 principes Agiles

- 1) La plus haute priorité est de **satisfaire le client**, en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée
- 2) Accueillir positivement les **changements** de besoins
Exploiter le changement pour donner un avantage compétitif au client
- 3) **Livrer** fréquemment un logiciel opérationnel
- 4) Les utilisateurs ou leurs représentants et les développeurs doivent travailler **ensemble** quotidiennement tout au long du projet
- 5) Réaliser les projets avec des personnes **motivées**,
et leur faire confiance pour atteindre les objectifs fixés
- 6) La méthode la plus simple et la plus efficace
pour transmettre de l'information est le dialogue en **face à face**
- 7) Un logiciel **opérationnel** est la principale mesure d'avancement
- 8) Les processus agiles encouragent un rythme de développement **soutenable**
Un rythme constant devrait pouvoir être maintenu indéfiniment
- 9) Une attention continue à l' **excellence** technique et à une bonne conception
- 10) La **simplicité** est essentielle
- 11) Le meilleur travail émerge d'équipes **auto-organisées**
- 12) À intervalles réguliers, l'équipe réfléchit aux moyens
de **régler** et modifier son comportement en conséquence



Bibliographie

■ Le guide SCRUM :

- Gratuit, simple et précis, par les auteurs de la démarche
- Version française : <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-French.pdf>
- La version originale en anglais : <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>

■ « AGILE Project Management with Scrum »

- Ken Schwaber - Microsoft
- Par l'un des co-auteurs de la démarche
- <http://amzn.to/1s1bnGv>

■ « Le guide pratique de la méthode agile la plus populaire »

- Claude Aubry - DUNOD
- <https://www.amazon.fr/Scrum-guide-pratique-m%C3%A9thode-populaire/dp/210059446X>

■ « SCRUM et XP from the trenches »

- Enrik KNIBERG – Info Q – Téléchargement gratuit
- Des tonnes de conseils pratiques expliqués simplement et intelligemment illustrés
- <http://www.infoq.com/minibooks/scrum-xp-from-the-trenches-2>

■ « Rédiger des cas d'utilisation efficaces »

- Alistair Cockburn – Eyrolles
- Pour tirer davantage partie de ce fantastique moyen au service du recueil des besoins, y compris dans le cadre d'une démarche Agile
- <http://amzn.to/ZekUzp>



Sites Web

- **Le site "Scrum" de Ken Schwaber**
 - Co-auteur de la démarche
 - <http://www.scrum.org/>
- **Le site "Scrum Guides" de Jeff Sutherland**
 - Co-auteur de la démarche
 - <http://www.scrumguides.org/>
- **Le site d'Alistair Cockburn**
 - Agiliste et utilisateur de Use Case de la 1ère heure
 - <http://alistair.cockburn.us/>
- **Le site de Thomas Jaskula**
 - sur l'Agilité, le refactoring...
 - <http://blogs.developpeur.org/tja/>
- **Quiz test de certification scrum.org PSM1 et PSPO**
 - <http://mlapshin.com/index.php/scrum-quizzes/>



Autres démarches Agiles

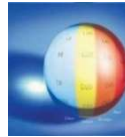
■ Dynamic Systems Development Method (DSDM)

- Autonomie, adaptabilité, itérations non fixes



■ Crystal Clear

- Modélisation collaborative



■ Rational Unified Process (RUP)

- Itérative, incrémentale, centrée sur architecture



■ eXtrem Programming (XP)

- Orientée programmation, bonnes pratiques
 - Pair Programming
 - Test Driven Development (TDD)



■ Kanban

- Scrum Board, Work In Progress

看板

■ LEAN

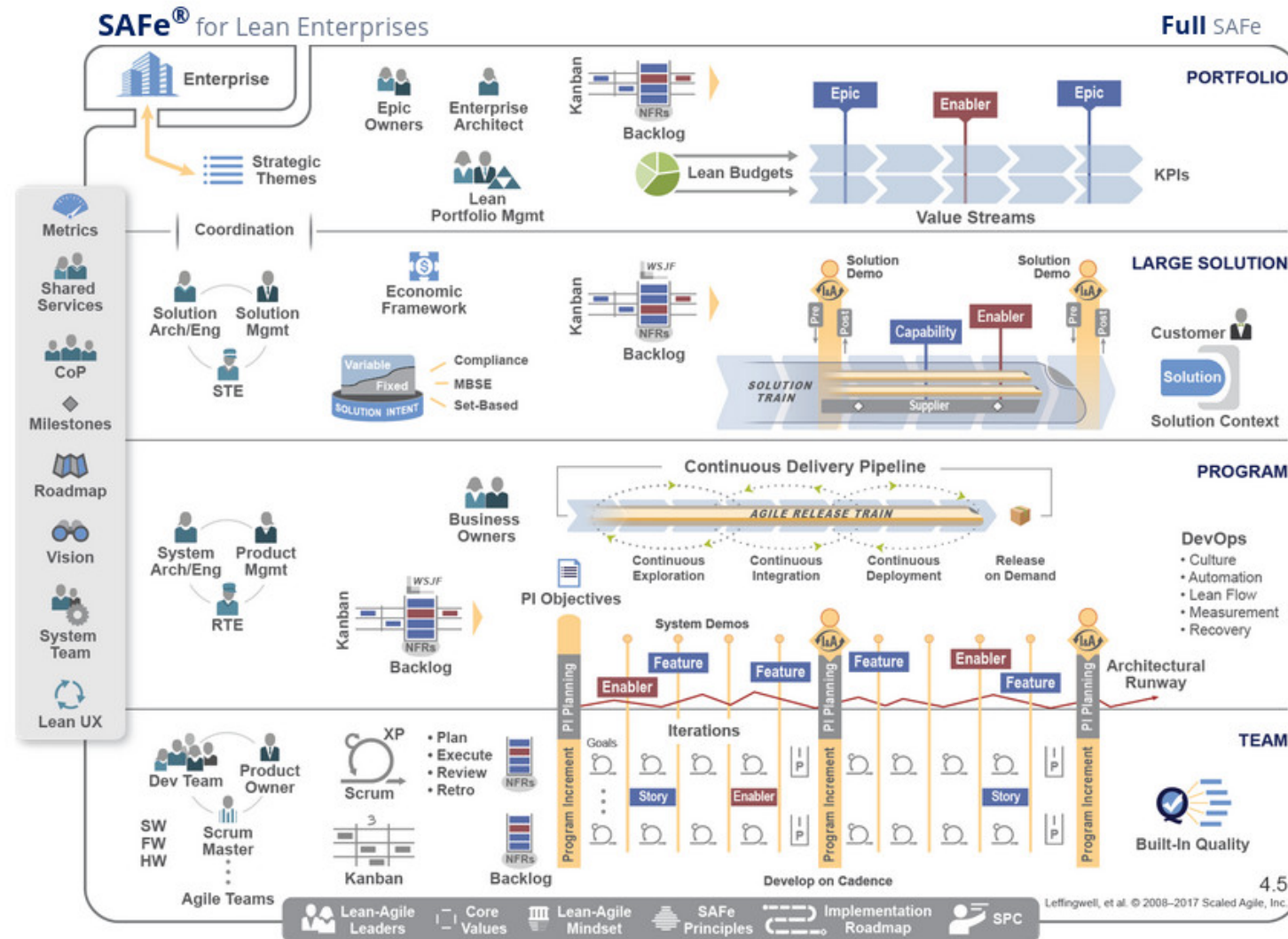
- Plus globale dans l'entreprise, pour éviter le gaspillage

■ ...



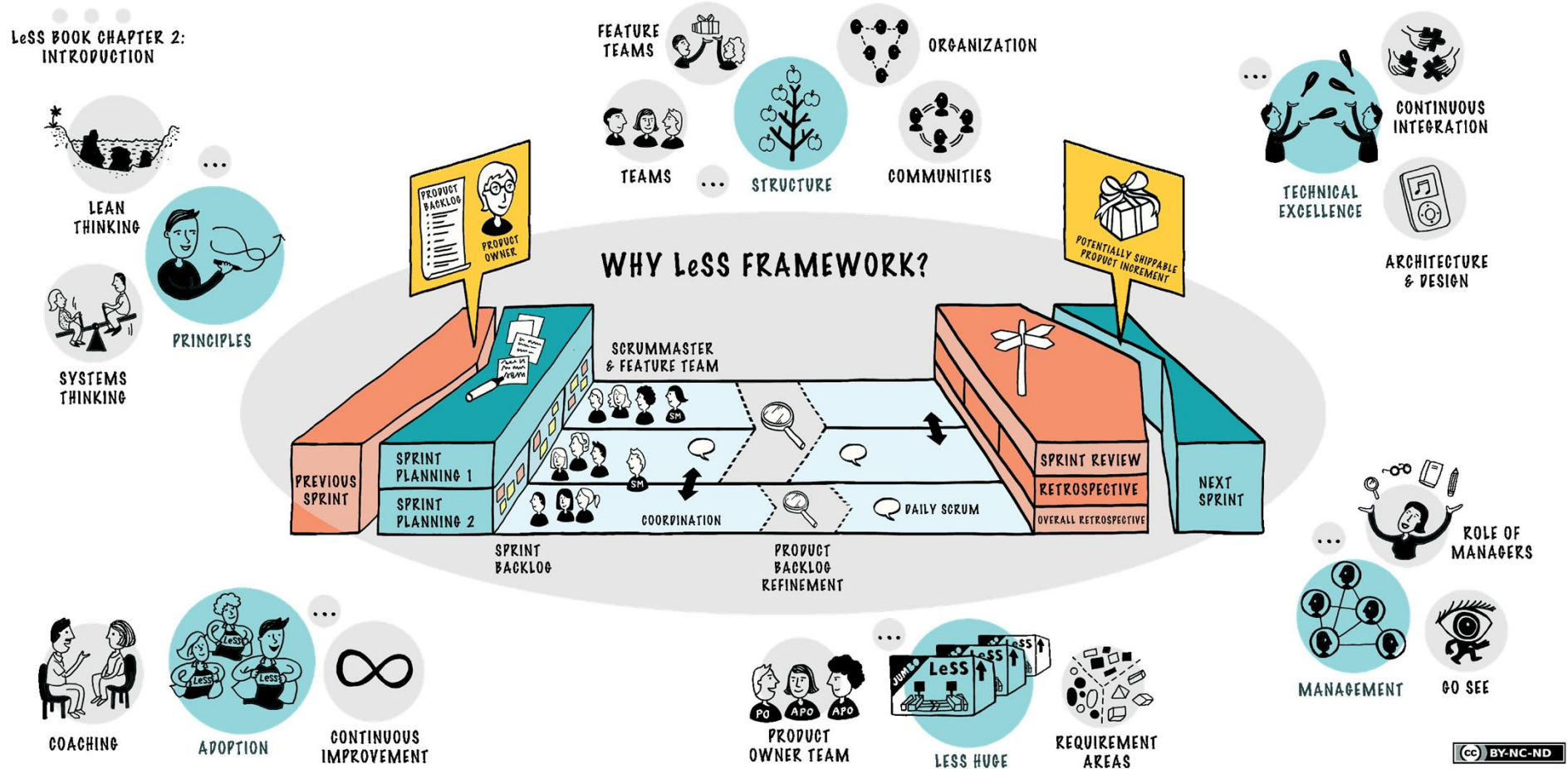
SAFe : Scaled Agile Framework (for enterprise)

Intégrer LEAN, Scrum, XP... dans l'entreprise



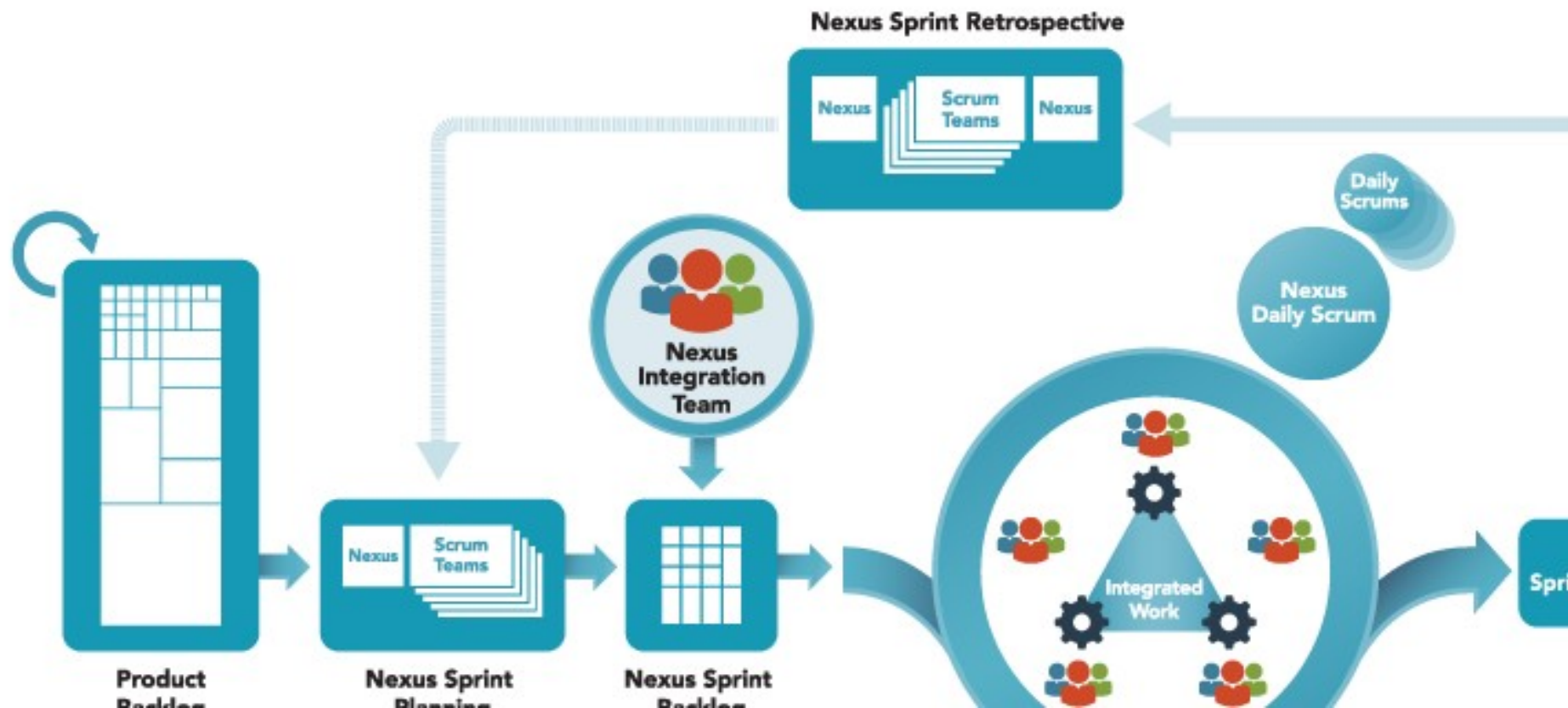
LeSS : Large Scale Scrum

Développements Scrum à grande échelle



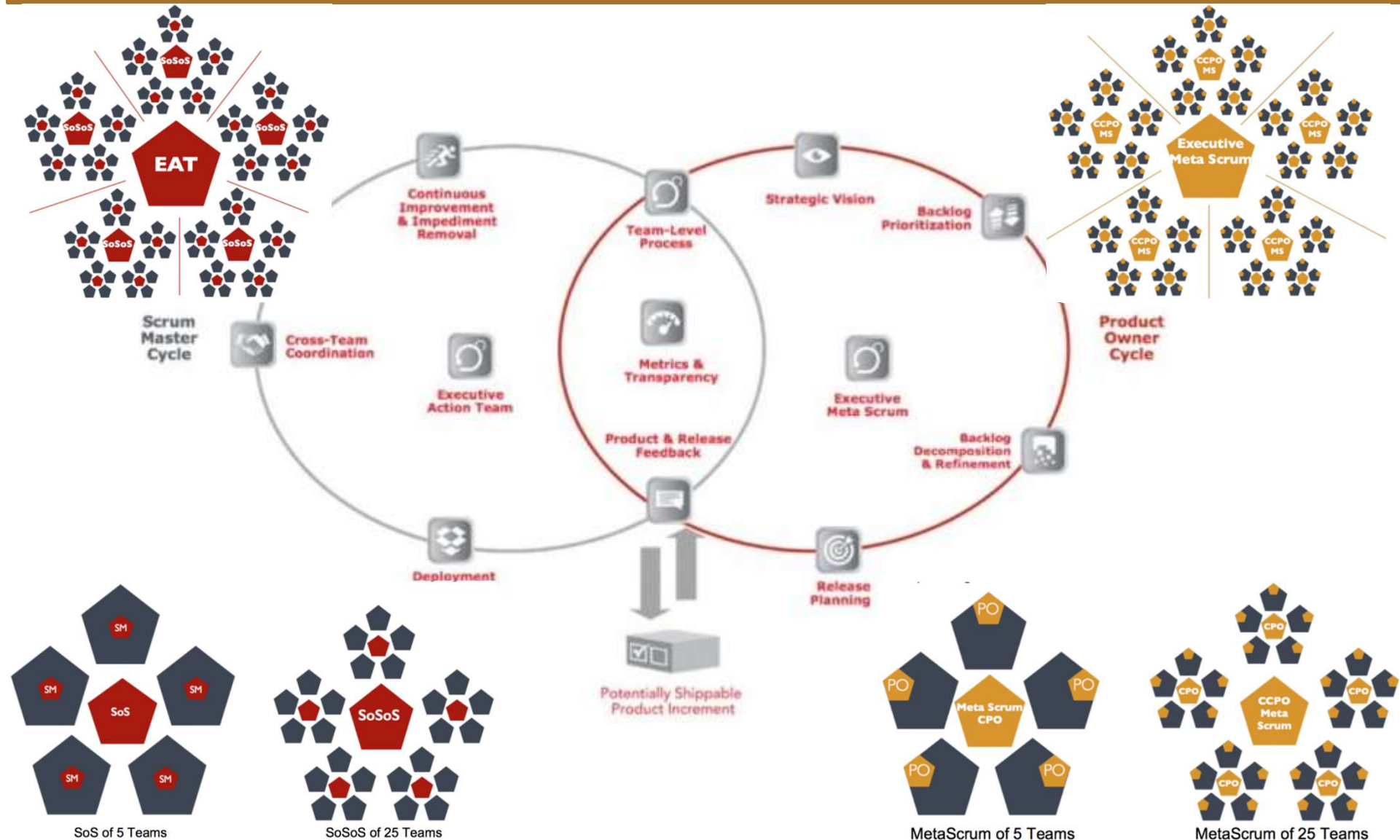
Nexus :

La vision de Ken Schwaber, co-créateur de Scrum



Scrum@Scale

La vision de Jeff Sutherland, co-créateur de Scrum



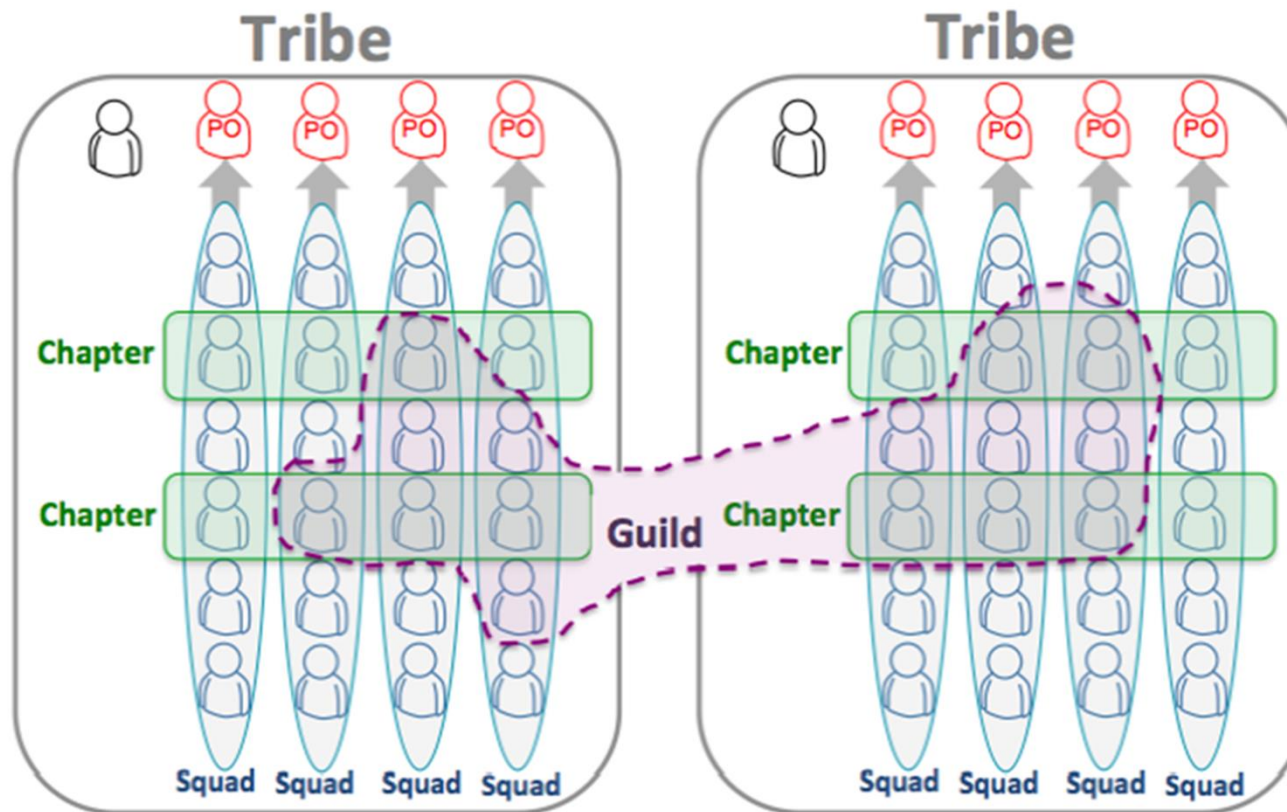
Scrum @ Spotify

Vision pragmatique de Henrik Kniberg

Scaling Agile @ Spotify

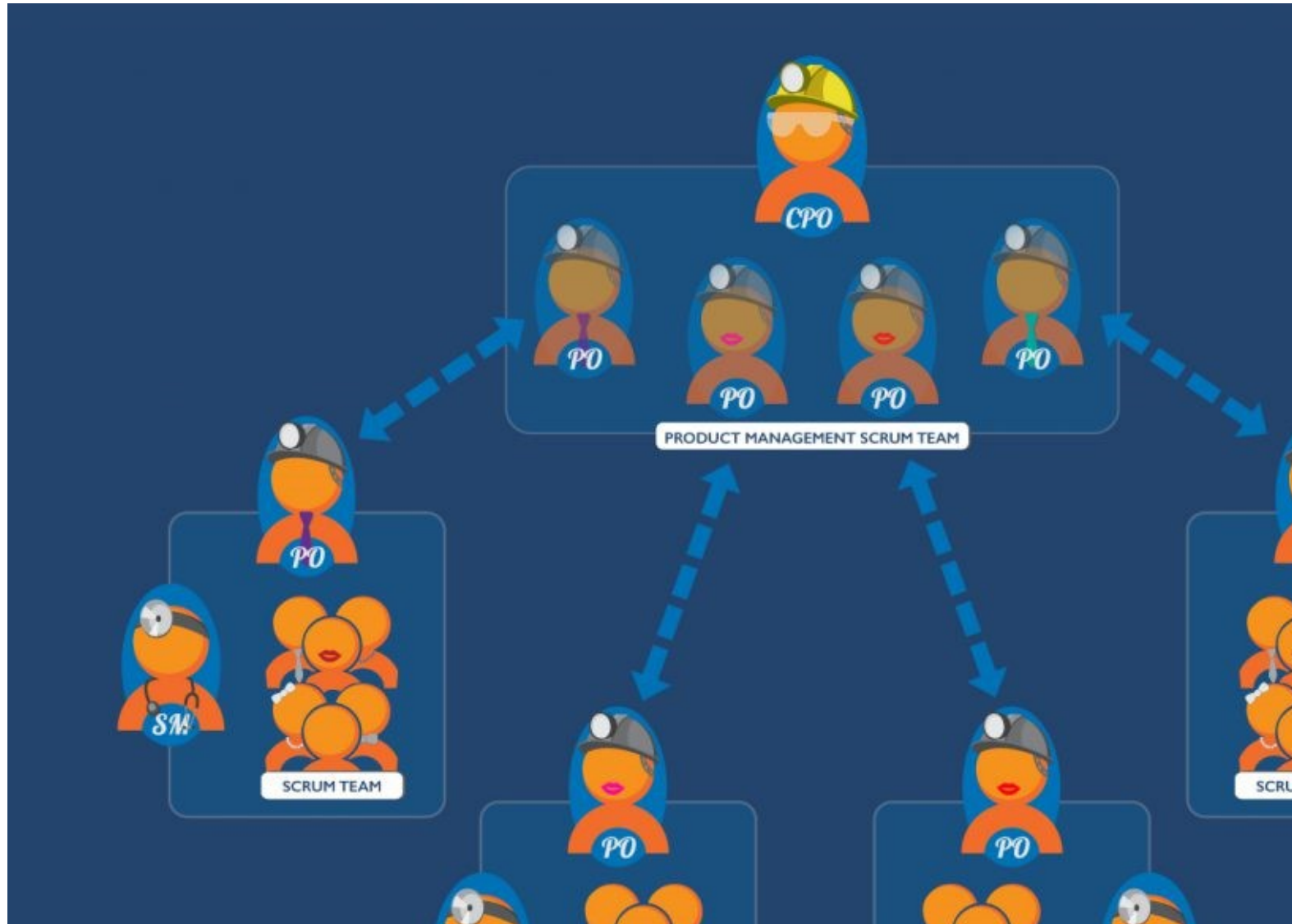
with Tribes, Squads, Chapters & Guilds

Henrik Kniberg & Anders Ivarsson
Oct 2012



SSWS : Scaled Scrum With Scrum

Scrum de A à Z sur toute l'échelle



Glossaire de notre formation

Définitions et traductions anglais/français

- Timebox : Contraint par le temps
- Minimum Valuable Product (Produit Minimum Viable)
- Scrum (Mêlée au Rugby) : Démarche Agile à l'état de l'art
- Sprint (itération) : Cycle de développement complet (+/- 3 semaines)
- Backlog (Reste à faire) : Du produit ou de l'itération (Product / Sprint)
- TDD (Test Driven Development) : Développement dirigé par les tests (Agiles)
- Client : Acteur (utilisant le produit) ou payeur ?
- User Story : Exigence Agile
- Post mortem = Rétrospective
- Incrément : Livré à la fin d'une itération (prototype, y compris simulation)
- Release (livraison) : Version complète en production
- LEAN (Maigre, en anglais) : Démarche Agile au niveau d'une entreprise
- DEVOPS : Intégration des développement et de la production
- AMOA : Assistance à Maîtrise d'OuvrAge = Business Analyst
- MOE (Maîtrise d'Œuvre) : Ceux qui vont développer le produit
- MOA (Maîtrise d'OuvrAge) : Ceux qui vont utiliser le produit
- BA (Business Analyst) : Analyste des besoins utilisateurs



Mise en œuvre

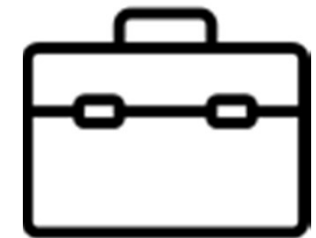
■ Matériel et outils	112
■ Exemples	114
■ Bonnes pratiques	116
■ Contractualisation	117
■ TDD	118
■ Refactoring	119
■ Notre rétrospective	120
■ VOTRE programme	122
■ Coordonnées	123



Matériel

Kit pour démarrer dans de bonnes conditions

- « War room »
 - local dédié
- PostIt notes
 - Différentes couleurs (US, Epic, Item, Tâche, impédiment)
- Feutres
 - Visibilité des PostIt
- Cartons « plumes » (2 bostols sur mousse)
 - Pour déplacer aisément backlogs, Scrum Board, Burn Charts...
- Paperboard ou mieux tableaux effaçables
 - Pour réaliser des schémas avec PostIt et « flèches »
- Jeux de Planning Poker
- + Webcam, micro, dispositifs de visio-conférence ?
 - Si on doit travailler à distance
- + Pourquoi pas des « Smartboards » (chers) ?
 - Vidéoprojecteurs dynamiques et partageables



Outils ?

■ Réfléchir avant de choisir/d'utiliser les outils

- Quels usages ?
- Quelles configurations ?
- Le meilleur outil, c'est celui que tout le monde sait (bien) utiliser

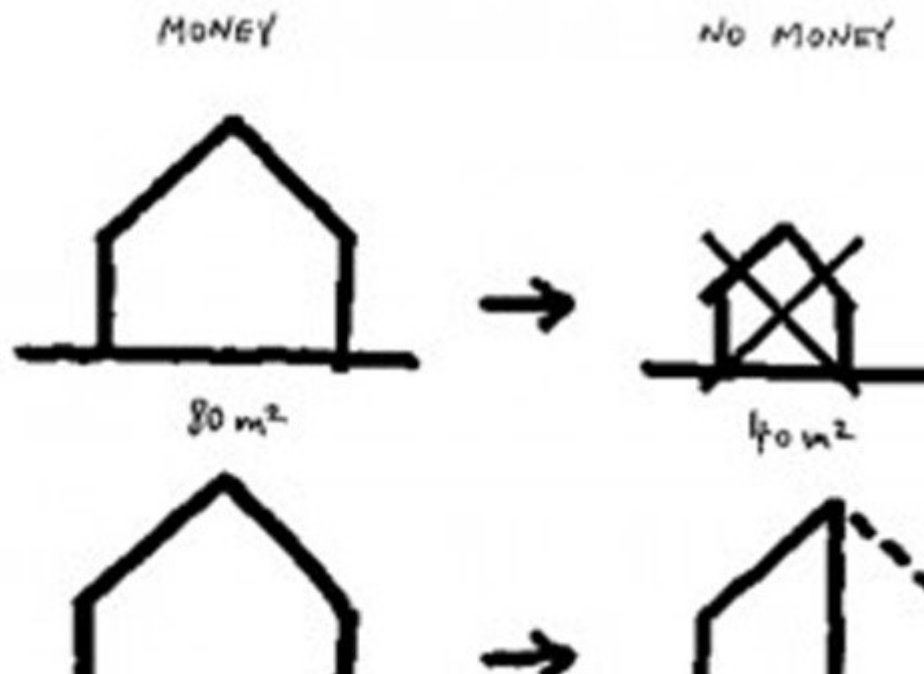
■ Exemples d'outils

- JIRA (de Atlassian)
 - Très fréquemment utilisé pour gérer des projets Agiles
 - <https://fr.atlassian.com/software/jira>
 - Souvent associé à Confluence (gestion des exigences)
- Trello (Racheté par Atlassian)
 - Outil collaboratif plus succinct
 - <http://trello.com/>
- Kanboard
 - Pour gérer des Kanban (Scrum Board)
 - <http://kanboard.net/>
- IceScrum, iObeya...



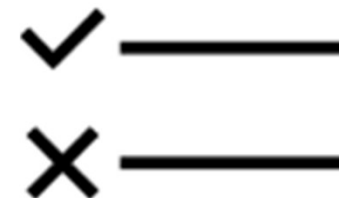
Agilité dans le domaine de l'architecture

Alejandro Aravena, lauréat 2016 du Pritzker Prize



Autres bonnes pratiques

- Rédiger un glossaire dès que des termes sont ambigus
- Pas de « Pipotron » <http://www.lepipotron.com/>
- N'utiliser que ce qui fonctionne « sous les bombes »
- Rester le plus longtemps possible en Analyse
- Remonter de la conception vers l'analyse par la recherche de causalité (5 Pourquoi de Ishikawa)
- Penser « Produit Minimum Viable »
- Co-construire avec les utilisateurs
- Utiliser les PostIt « à l'envers »
- « Sauvegarder » les paperboard en les photographiant
- Pour prioriser et sélectionner des US, se poser la question : « Si on stoppe à la fin de cette itération »



Contractualisation

■ Objectif

- ☐ Si on va plus vite, tout le monde y gagne
- ☐ Si on va plus lentement, ça coûte à tout le monde
- ☐ -> Tout le monde a intérêt à la réussite

■ Risques pour un contrat au forfait

- ☐ Il y a toujours un risque à l'estimation initiale
- ☐ On prend des marges dans les 2 cas
- ☐ Réévaluation à chaque sprint = réduction du risque en Agile

■ Diviser le contrat en 2 parties

- ☐ 1^{ère} partie : Vision du produit, périmètre, feuille de route
- ☐ 2^{ème} partie : détails d'implémentation (après 1^{ère} partie)

■ Approches efficaces

- ☐ fixation des prix de type « pièce et main d'œuvre »
- ☐ Inclure des clauses de remplacement
- ☐ Changements dans l'ordre d'implémentation à périmètre figé



Refactoring du code

(Pratique Agile issue de XP)

- Il est illusoire d'écrire d'emblée du code « réutilisable » efficacement dans TOUTES les conditions futures
 - Car on privilégie souvent une optique de réutilisation a priori
 - Et ce sera peut-être un autre besoin de réutilisation qui apparaîtra
- Principe du Refactoring (dans le cadre du Test Driven Development)
 - On écrit d'abord un code simple
(de qualité, mais pas spécialement prévu pour être réutilisable)
Juste nécessaire pour que les tests unitaires soient validés
 - On sait que ce code fonctionne, mais qu'il est peu réutilisable
ce qui crée de la « dette technique »,
qui fera baisser notre vélocité à terme si on ne modifie rien au code
 - Une fois qu'on a plusieurs User Stories nécessitant un code similaire,
On retravaille le code pour le rendre réutilisable
(en connaissance de cause, donc dans une direction UTILE)
 - À iso-fonctionnalité (sans changer du tout le besoin réalisé)
 - Ainsi, si le nouveau code échoue à passer les tests unitaires,
on peut toujours revenir à l'ancien code



VOTRE programme pour la suite : Scrum Board personnel

- Que prévoyez-vous de faire de ce que vous avez appris, une fois de retour dans votre entreprise ? (À vous de compléter ici) :
☐

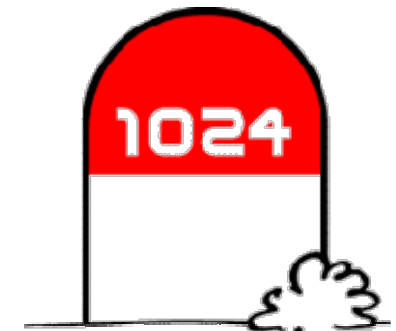
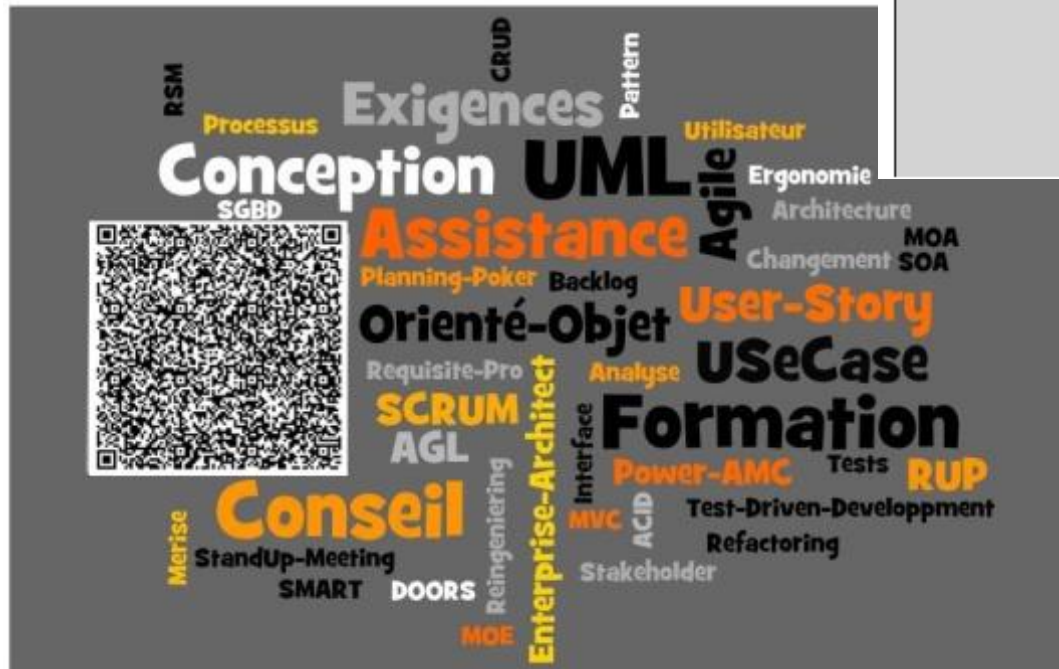
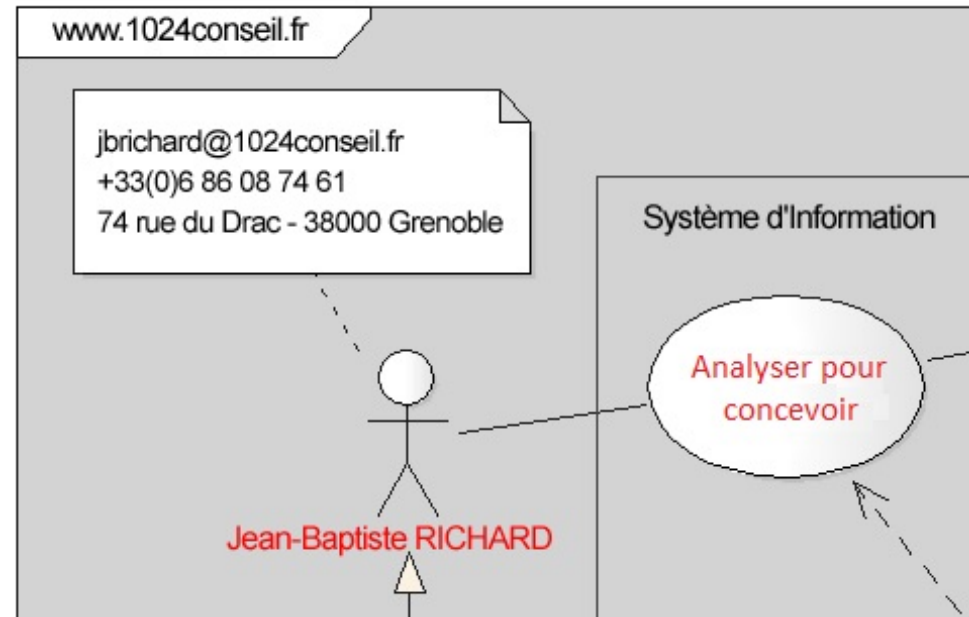


N'hésitez pas à me contacter

1024 Conseil

Jean-Baptiste RICHARD

jbrichard@1024conseil.fr



Certification

■ Test blanc	124
■ Site scrum.org	125
■ Examen	139
■ Conseils	141
■ Révision	142
■ Notes	154



Site de Mikhail Lapshin

Pour passer des tests BLANC

- Aller sur le site Web de Mikhail Lapshin
 - <http://mlapshin.com/index.php/scrum-quizzes/>
- Testez les deux modes d'examen blanc
 - Learning mode : Pas de temps limites + explications
 - Real mode : Temps limite en conditions réelles



Inscription sur le site scrum.org

Pour le passage de la certification

- Aller sur le site de Scrum.org
 - www.scrum.org
- Se créer un compte (si pas déjà créé précédemment)
 - Register
- Valider l'email de création de compte
 - Cliquer sur le lien
- Lancer la certification
 - ATTENTION : Quand tout le monde est prêt !!!
 - Get certified / Professionnal XXX / XXX Assessment / Start Assessment
 - Saisissez l'assessment password
 - Cliquez sur START (ATTENTION : Le test démarre et le compteur tourne !!!)

