

Les fondamentaux d'Agile



Formation gratuite sur Agile et les frameworks Scrum et SAFe

Sommaire

Thématique

Approche projet

- ✓ Prédictif et adaptatif
- ✓ Cycle de vie prédictif
- ✓ L'incrémental
- ✓ L'itération
- ✓ Cycle de vie adaptatif
- ✓ Pour résumer

Agile et Scrum

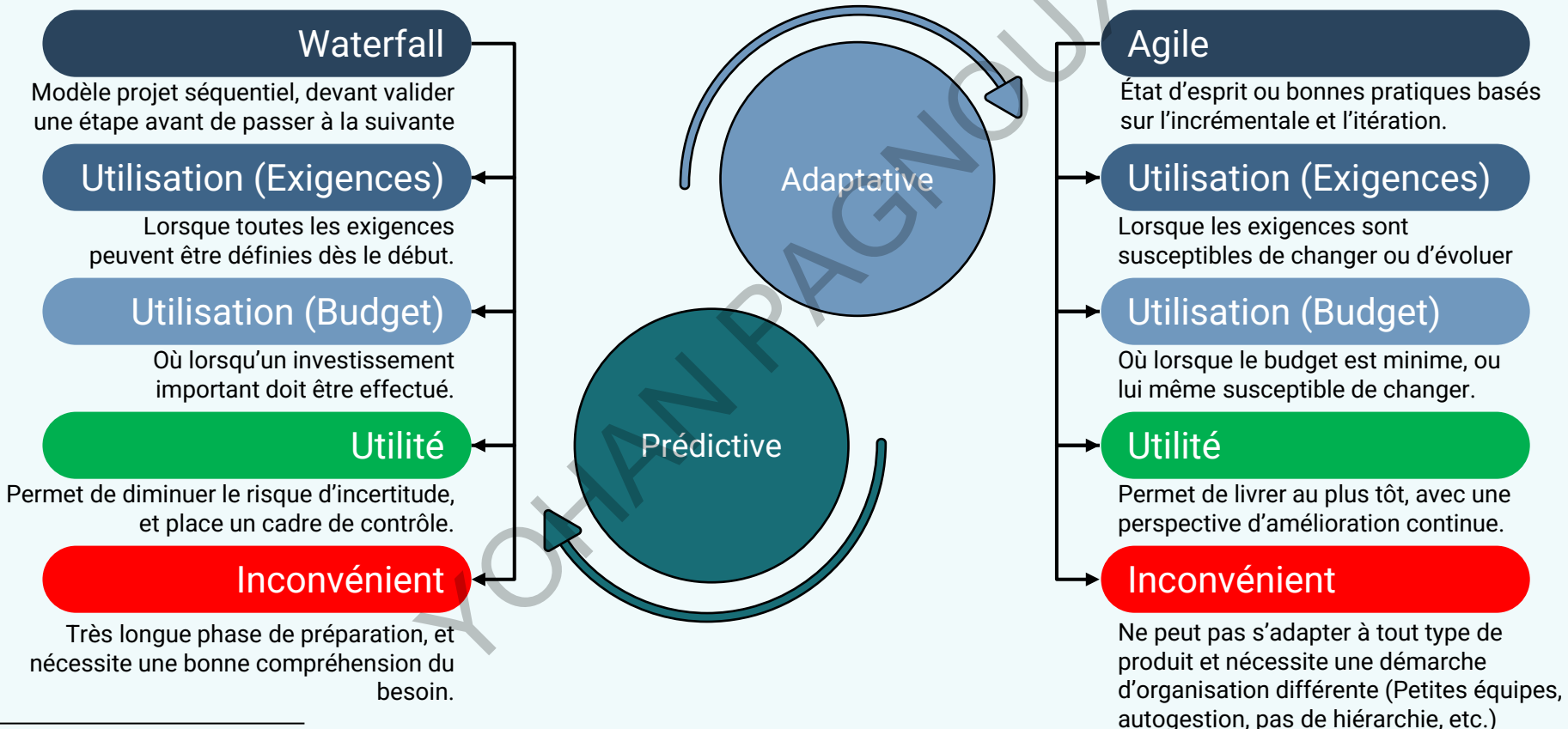
- ✓ Le manifeste Agile
- ✓ Frameworks issus d'Agile
- ✓ Différence entre méthode, framework et bonnes pratiques
- ✓ Scrum Guide

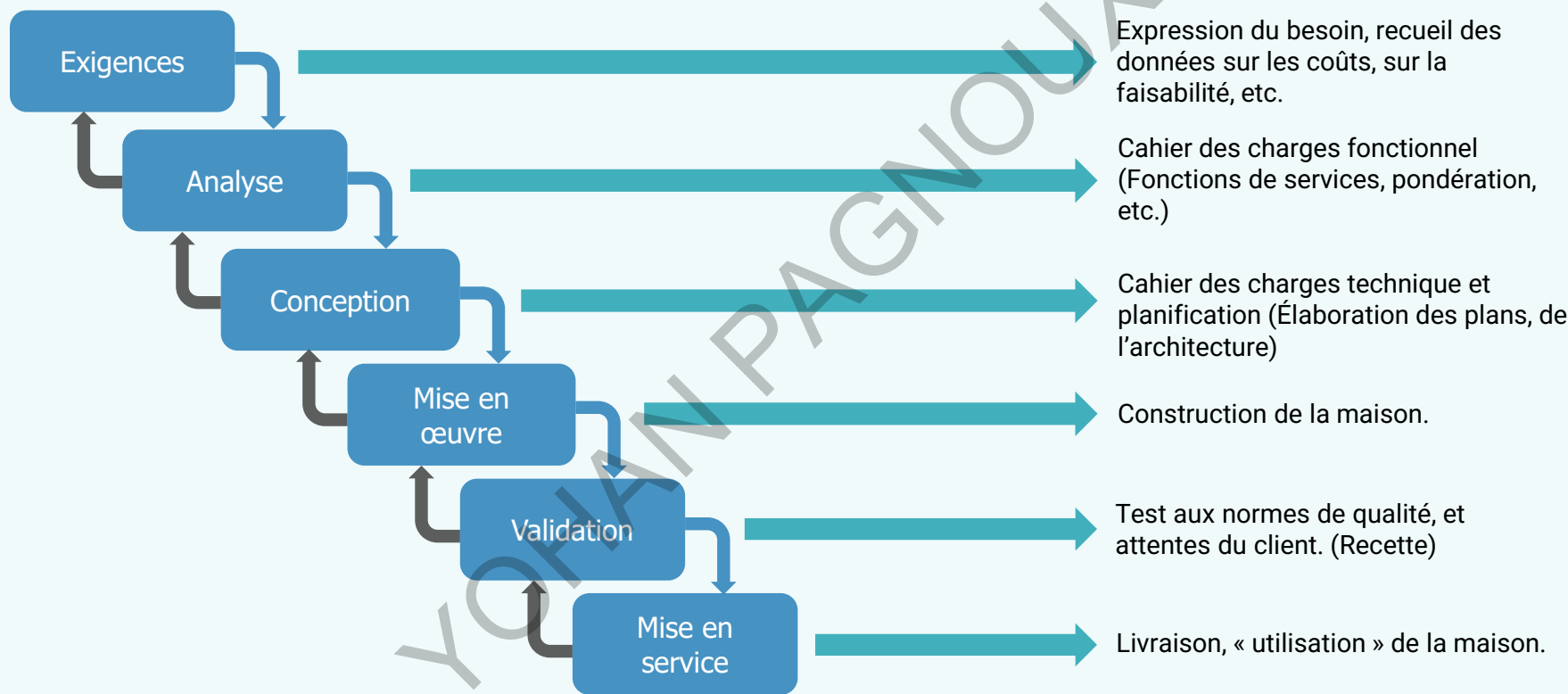
Fonctionnement

- ✓ La Scrum Team et son fonctionnement
- ✓ Les parties prenantes à l'équipe Scrum
- ✓ Fonctionnement global
- ✓ La gestion des Sprints
- ✓ Le Product Backlog (User Story)
- ✓ La release
- ✓ Daily Meeting
- ✓ Review de Sprint
- ✓ Rétrospective de Sprint
- ✓ Les limites d'Agile et de ses frameworks

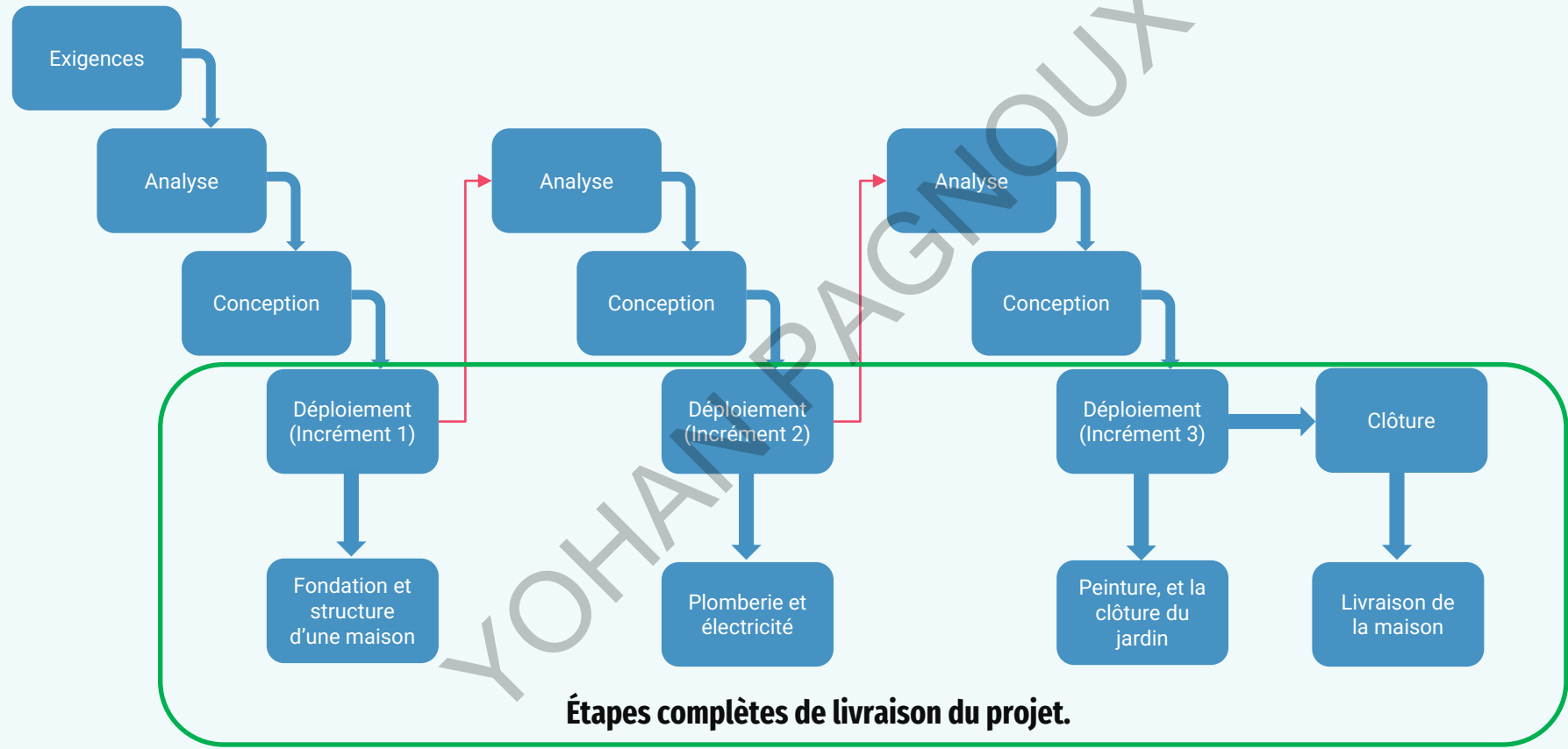
L'Agile à l'échelle

- ✓ Scaled Agile Framework
- ✓ La couche Essential
- ✓ La couche Large Solution
- ✓ La couche Portfolio
- ✓ La couche Full SAFe
- ✓ Le cycle de vie de SAFe
- ✓ Résumé global et conclusion





Approche prédictive : Waterfall (Exemple d'une construction de maison)





Recueil et analyse des besoins



Définition du logiciel



Développement du logiciel



Test technique du logiciel



Tests avec l'utilisateur



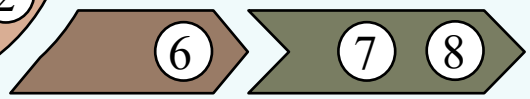
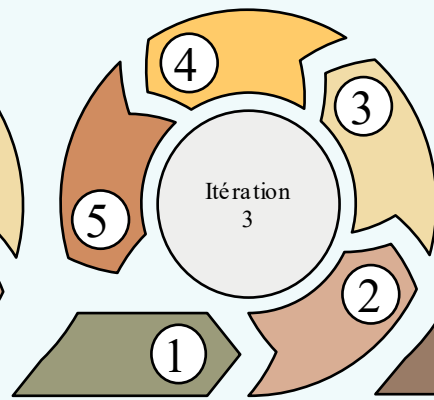
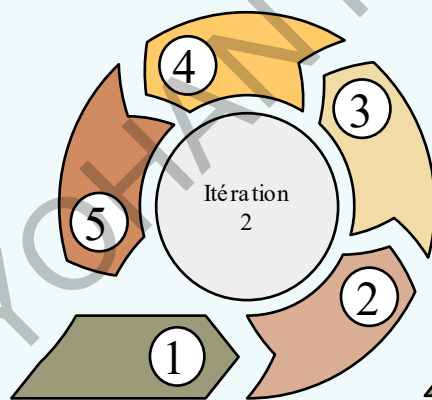
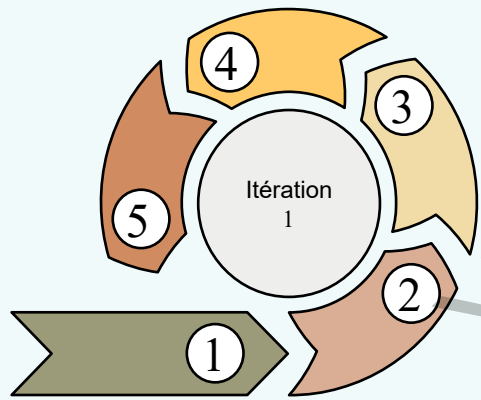
Test de la version finale

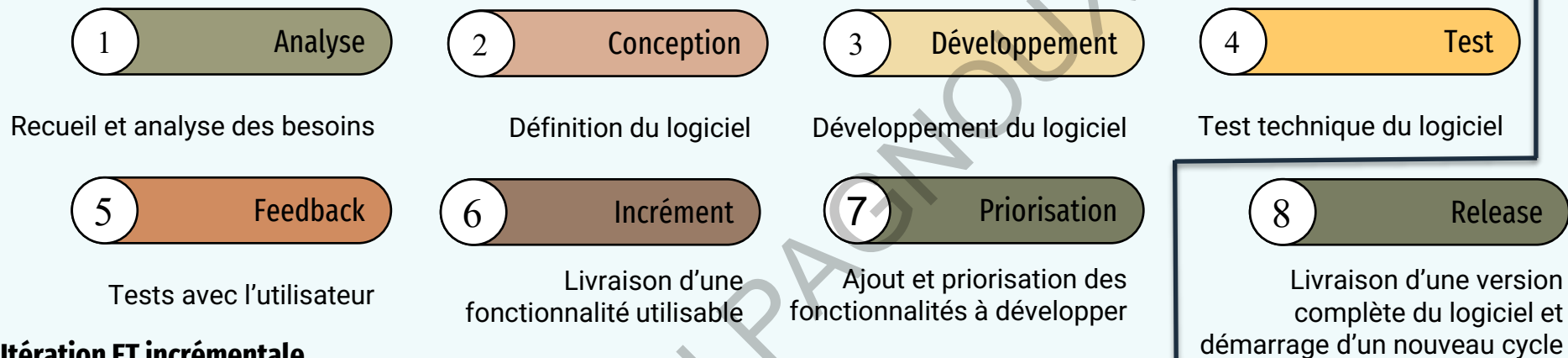
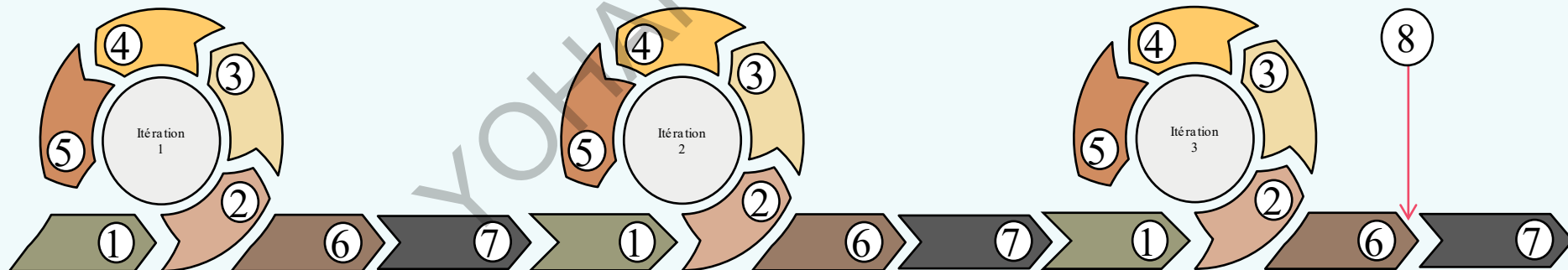


Mise à disposition du logiciel aux utilisateurs



Clôture du projet



**Itération ET incrémentale**

Approche prédictive

Approche qui permet de suivre un plan défini dès le départ, dont il faut suivre, et dont il faut se conformer pour pouvoir réussir le projet. Exemple : la mise en place d'un système d'information en totalité.

Approche adaptative

L'approche adaptative se base sur une approche itérative et incrémentale. Elle permet de livrer une fonctionnalité du produit (incrément) rapidement, à chaque itération, et de l'améliorer grâce aux feedbacks. Exemple : logiciel compta.

Étape par étape,
avec validation
entre chaque étape

Approche prédictive

Approche hybride

Approche adaptative

Itération, et
incrément à
chaque livraison

Approche hybride

L'approche hybride combine le prédictif et l'adaptatif. Elle peut être utile notamment pour gérer plusieurs livrables de manière différente. Ex : livraison d'un SI en prédictif, et création d'un logiciel compta installée dans le SI, en adaptatif.

La bonne approche ?

Il n'y a pas de meilleure approche : il faudra choisir la bonne en fonction du produit à développer, et du contexte de votre projet (contraintes, périmètre, budget, etc.)

Les limites des approches historiques

Le relancement économique après la Seconde Guerre a replacé la conception du produit au centre de tout, en se basant sur le besoin du client, et la valeur à obtenir grâce à des matériaux de substitution. (cf. Analyse de la valeur)

À mesure que les projets devenaient de plus en plus complexes, des évolutions se sont faites durant les années 60, 70, 80, 90, puis dans les années 2000. On a vu voir apparaître de nouvelles approches, avec de nouveaux outils, afin de toujours obtenir plus de valeur. Mais aussi d'optimiser et d'améliorer les organisations internes des entreprises.

Le Lean Manufacturing et le Lean Management

Le Lean Manufacturing a été développé dans les années 1950 au Japon par Toyota et consiste en une méthode d'optimisation de la production qui vise à minimiser les gaspillages et à améliorer la qualité, la rapidité et la flexibilité du processus de production.

Le Lean management a émergé à partir du Lean Manufacturing et consiste en une approche globale pour améliorer la performance d'une entreprise en identifiant et éliminant les gaspillages dans tous les aspects de l'organisation, incluant les processus administratifs et les relations avec les clients et les fournisseurs.

Le Lean et Agile

Le Lean a donné naissance, au travers de différents outils, à plusieurs bonnes pratiques, notamment l'automatisation, et l'amélioration continue. Et on a également vu apparaître le Six Sigma pour optimiser les process (au contraire du Lean qui élimine) et qui a pu être combiné avec le Lean pour donner le Lean Six Sigma. L'Agile s'est inspiré d'un grand nombre de ces pratiques pour baser son approche avec une incrémentation à chaque itération. L'objectif : la conception et l'amélioration continue d'un logiciel, et la mise en place d'automatisation et d'autoorganisation.

Formalisation de l'Agile et création du manifeste Agile

De nouvelles organisations avec des principes tels que la décentralisation des prises de décisions, la réduction des niveaux hiérarchiques, la gestion par le leadership, l'autoorganisation des équipes, l'utilisation stratégique de l'empirisme, l'amélioration continue, se mettent en place, et donnent naissance au terme « d'organisation Agile. »

17 spécialistes du développement de logiciels en 2001 se sont réunis pour créer ensemble le « manifeste Agile », spécialement conçu pour mettre en place des équipes autoorganisées, afin de concevoir un logiciel plus facilement.

Les créateurs du manifeste, et ses dérivés

Certains de ces spécialistes ont ensuite créé leurs propres pratiques « agiles » :

- Ward Cunningham l'inventeur du Wiki et qui a permis notamment l'élaboration de Wikipédia.
- Kent Beck créateur de l'extreme programming/XP (Framework Agile) et co-créateur de JUnit (Outil test unitaire)
- Ken Schwaber et Jeff Sutherland créateurs et promoteurs de Scrum (Framework Agile)
- Jim Highsmith prônant l'ASD (méthode de développement rapide d'applications)
- Alistair Cockburn inventeur du Crystal Clear (Framework Agile)
- Martin Fowler, Dave Thomas et Arie van Bennekum pour DSDM. Dynamic systems Development method framework Agile très peu connu en France

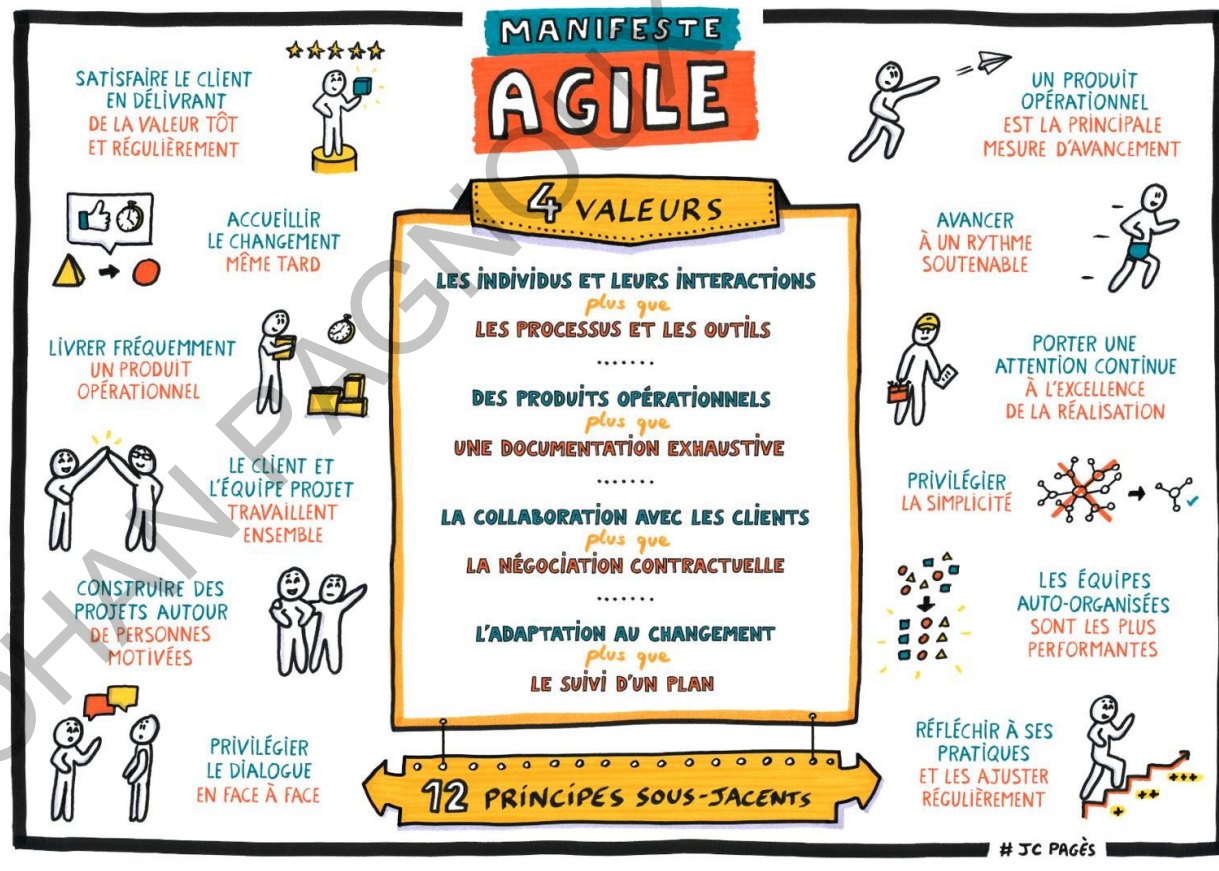
Démocratisation des frameworks Agile

Le développement de logiciel prend de plus en plus de place dans la numérisation de nos entreprises, avec le besoin d'être toujours plus rapide, en mettant en place des organisations efficaces. Scrum est un framework populaire dans l'industrie du développement logiciel qui est basé sur les principes d'Agile. Il se concentre sur la collaboration de l'équipe, l'adaptation aux changements et la livraison rapide et continue de valeur pour les clients. Tout comme Agile, ils sont avant tout focus sur la création d'un produit, plus que sur la gestion d'un projet.

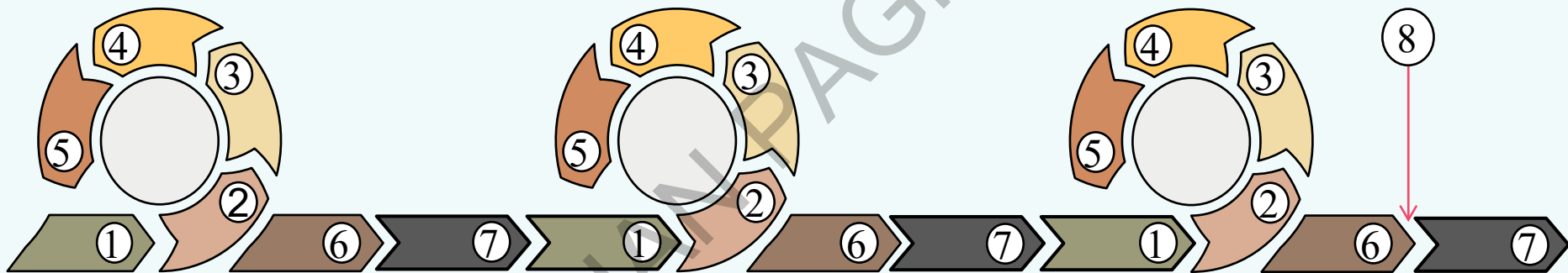
Le Manifeste Agile est un document fondateur publié en 2001 qui définit les principes fondamentaux pour une approche plus flexible et adaptative de la gestion de développement de logiciels.

Il comprend 4 valeurs et 12 principes qui mettent l'accent sur la collaboration, l'adaptation aux changements, la livraison fréquente et la satisfaction du client.

<https://manifesteagile.fr/>



Le manifeste Agile se matérialise donc sur une approche adaptative, afin d'installer une amélioration continue, une réponse au besoin de plus en plus précise au fur et à mesure du développement, permettre aux équipes de s'auto-organiser, et d'accepter le changement plus facilement. De plus, c'est une approche orientée « produit », et non pas « gestion de projet ».



Cette approche a donc donné lieu à des évolutions dans des contextes très spécifiques.

La plus célèbre étant le framework Scrum, qui permet également de faire intervenir plus facilement les gens du « business » et du « métier », en donnant un cadre plus « généraliste » : Scrum est lui aussi orienté produit, mais sans distinction particulière. Même s'il est très utilisé pour le logiciel.

Framework

Scrum est volontairement incomplet ! Il souhaite laisser la possibilité aux équipes de venir le compléter avec leurs propres process, leurs propres outils, afin qu'ils développent leur propre « méthode » interne. Il pose juste « le cadre » à respecter.

Conséquence

Une mauvaise compréhension du cadre Scrum peut être une des conséquences de sa mauvaise application. Une méthode implique précisément tous les outils, les process, les étapes à suivre à la lettre. Scrum n'a rien de tout cela.

Modèle projet

Cette façon de procéder n'est pas nouvelle : c'est la même chose pour le modèle Waterfall/Cascade. Ce sont des modèles d'approche projet qui permettent de découper en grandes étapes le chemin à suivre. Mais il n'indique pas les outils ou les process exacts ! Les approches prédictives et adaptatives ne sont pas des méthodes en tant que telles.

Méthode en cascade

MERISE est une méthode d'analyse, de conception et de gestion de projet informatique, qui s'appuie sur le découpage en cascade. Mais avec une série d'outils, de documents, ou de process à mettre en place, et à respecter. On peut la qualifier de réelle méthode en cascade, que l'on peut suivre de A jusqu'à Z afin de réaliser son projet informatique.

Bonnes pratiques

De même pour les bonnes pratiques telles que le PMBoK en gestion de projet. Elles donnent des outils ou des conseils pour piloter un projet, mais ne peuvent pas être suivies à la lettre pour pouvoir en gérer un.

Suivre un cadre

Scrum se base sur l'Agile (qui n'est donc pas une méthode non plus) afin de mettre en place un cadre qu'il faut respecter, afin d'optimiser les chances d'avoir une organisation Agile. Mais qui laisse l'équipe s'auto-organiser et s'autogérer pour choisir les outils, ou les process de développement à mettre en place. Comme le Waterfall avec MERISE.

Le Scrum Guide

Porté par 2 des 17 créateurs du manifeste Agile, Ken Schwaber et Jeff Sutherland, Scrum s'est vu également officialisé avec le Scrum Guide et fournit une structure pour la planification, la livraison et l'amélioration continue de produits.

Objectif du Scrum Guide

Il est basé sur des valeurs, des rôles, des événements, des artefacts et des règles pour aider les équipes dans leur travail collaboratif. Il se base sur l'empirisme, le Lean et l'autoorganisation pour cela.

Comme Agile, Scrum base l'ensemble de son cadre sur 3 piliers, et 5 valeurs dites F.O.R.C.E.

3 piliers et 5 valeurs

Adaptation

Focus

Transparence

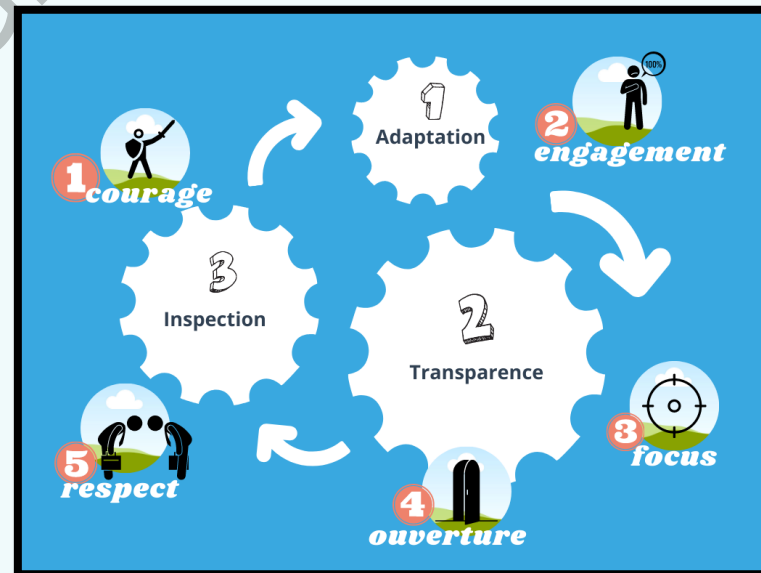
Ouverture

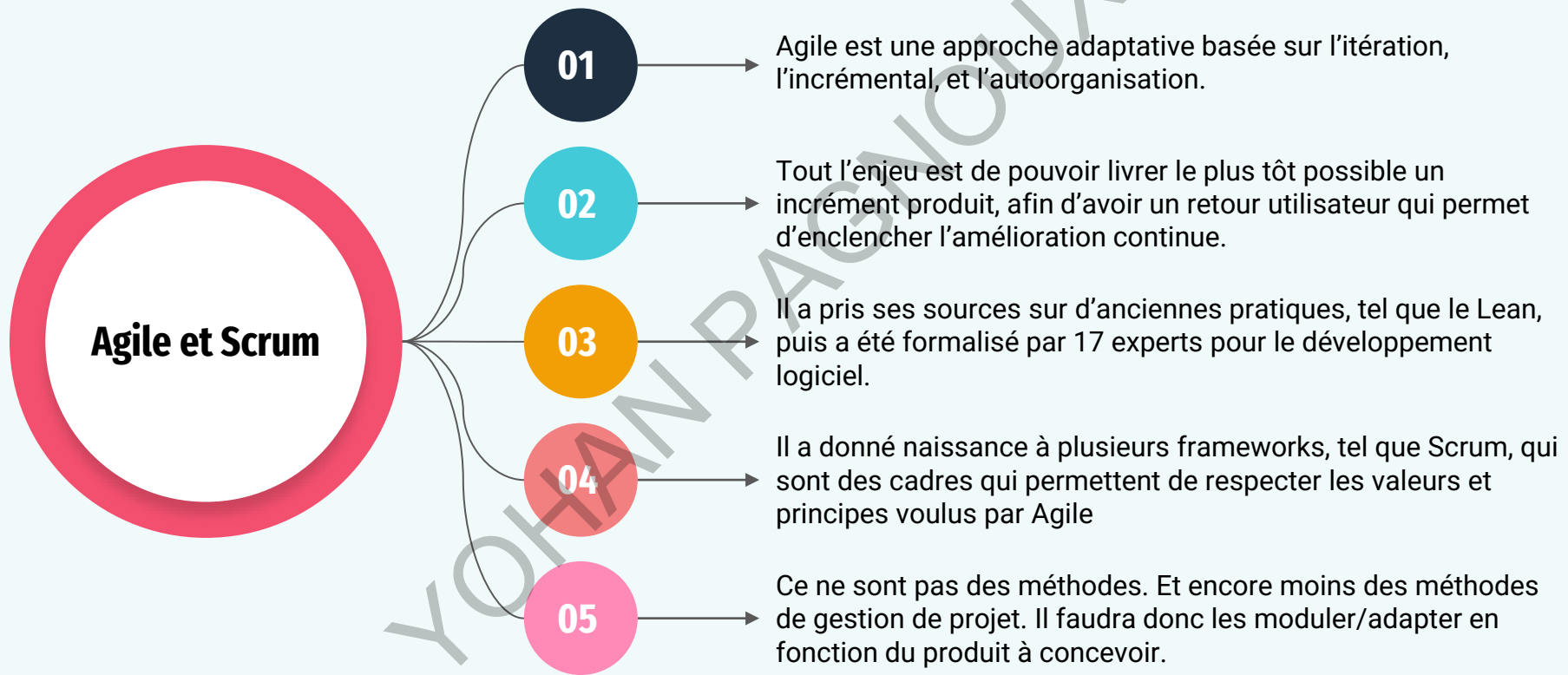
Respect

Inspection

Courage

Engagement





Équipe SCRUM (3 à 10 membres)



**Product
Owner**



Scrum Master



**Équipe de
développement
(3 à 8 devs maximum)**



**Client et/ou
utilisateur**

Le client/utilisateur ne fait pas partie, en tant que tel, de l'équipe SCRUM, mais il est essentiel dans le processus de développement de la solution. La taille maximale est de 10 membres (PO et SM inclus). Et il n'y a pas de chef de projet en Scrum : les responsabilités d'un CdP classique sont dispersées entre tous les membres de l'équipe.

Dans Scrum, ce qui est le plus important, ce sont les gens !



C'est également la première valeur du Manifeste agile : « Les individus et leurs interactions sont plus importants que les processus et les outils ».

Le contrôle fait place à la confiance, et l'autoorganisation remplace toute notion de management hiérarchique.

Le management ne disparaît pas malgré tout, mais devient un élément collectif.

En Scrum, il n'y a pas de chef qui dicte le tempo, et qui attribue les tâches arbitrairement.

Les « développeurs »

- Ils réalisent techniquement le produit
- Ils choisissent, collectivement, les items à développer.
- Chaque développeur possède ses propres compétences.
- Un développeur peut être lead technique et guider les autres.
- L'équipe de « dévs » s'appuie sur le Daily meeting pour s'organiser.
- Ils ne restent pas isolés : ils doivent échanger avec les utilisateurs.
- Ils définissent le Sprint Backlog, le Sprint Planning, et le Sprint Goal.

Le Scrum Master

- Ce n'est pas un chef de projet, et il n'a aucun pouvoir hiérarchique.
- Il motive l'équipe et les aide à s'auto-organiser. Il reste à leur contact.
- Il fait respecter le cadre Scrum.
- Il guide, forme, soutien, coach.
- Il gère les conflits internes.
- Son objectif principal est de faire disparaître les « obstacles » pouvant intervenir durant un sprint.
- Il aide le PO à la bonne gestion du backlog en lui suggérant des outils.
- Il ne fait pas de suivi individuel !

Le Product Owner

- Ce n'est pas un chef de projet ni un supérieur hiérarchique également.
- C'est un « rôle » spécifique à Scrum.
- Il est responsable de la valeur du produit et sa vision globale (Product Goal).
- Il organise, affine, et priorise les items du backlog.
- Il s'occupe de la stratégie de livraison (date de release par exemple).
- Il est fonctionnel et technique.
- C'est le « traducteur » attitré entre les gens du métier, et l'équipe de devs.
- Le PO est une seule personne !



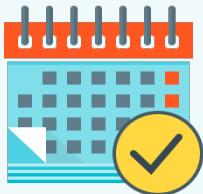
Les intervenants extérieurs à l'équipe Scrum

Client : Commanditaire du projet. Participe aux feedbacks du produit et à son amélioration continue. Même s'il est celui qui « met l'argent sur la table », le client n'a pas à venir décider de l'organisation interne de l'équipe Scrum. Mais il est un soutien, financier principalement, pour la réalisation du produit.

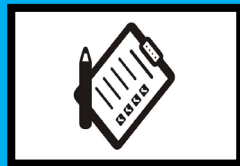
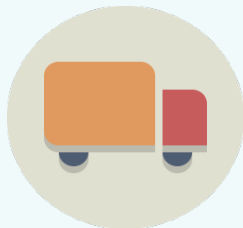
L'utilisateur final : celui qui va utiliser le produit. Il participe également aux feedbacks durant la review. Un représentant des utilisateurs peut être choisi. Et il (ou ils) doit être disponible autant que possible durant les Sprints afin d'aider l'équipe à réaliser le produit, ou à affiner les items du Product Backlog.

L'expert externe : il se peut qu'une équipe Scrum n'ait pas toutes les compétences nécessaires. Elle peut faire appel ponctuellement à un expert externe, qui a ces compétences, durant un ou plusieurs Sprints. L'objectif est aussi d'apprendre de lui, pour ne plus devoir en dépendre lors des prochains Sprints.

Les parties prenantes : le sponsor, les différents fournisseurs, le ou les sous-traitants, les partenaires (technologiques, politiques, régionaux, organismes publics, etc.), le service support, les différents consultants (coach ou formateurs en Scrum, etc.)



Développement en
itération (Sprint)



Identifier le besoin
client



Conception et livraison
de la solution

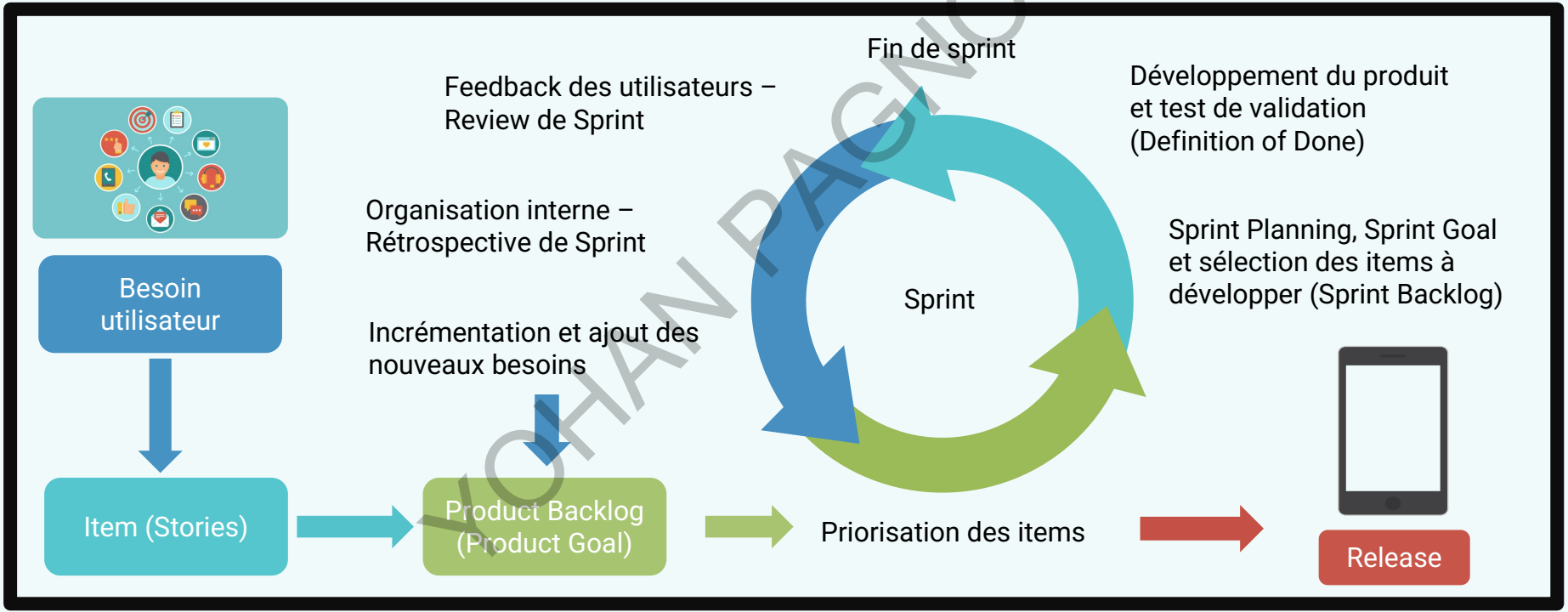


Feedback et
amélioration continue

Livrer un produit opérationnel à chaque fin d'itération (incrément)
si possible, voir pendant une itération.

3 artefacts : Product Backlog (Product goal), Sprint Backlog (Sprint Goal), et Incrément (Definition of Done).
Un artefact représente un travail ou une valeur à atteindre. Il permet un consensus entre chaque membre de l'équipe.

5 événements : le Sprint, le Sprint Planning, le Daily Scrum, la Sprint Review, et la Sprint Retrospective.
Chaque événement dans Scrum est une occasion formelle pour inspecter et adapter les artefacts Scrum. Le Sprint en lui-même englobe le reste des événements. Cela permet de créer de la régularité, sans nécessiter de rajout, et afin d'enclencher des itérations efficaces.



Il est défini par une Timebox

- ✓ Entre une semaine et un mois
- ✓ Avec une planification (Quoi livrer et comment travailler ?)

Le Sprint permet un budget fixe

- ✓ Le nombre d'itérations
- ✓ Le nombre d'incréments
- ✓ Budget pour la mise en place

Un Sprint reste le même

- ✓ Un Sprint commencé doit être fini (sauf décision du PO si le Sprint Goal devient obsolète)

Le Sprint est une itération

- ✓ Il se répète
- ✓ Il y a un objectif (Sprint Goal)

Le Sprint zéro (Premier Sprint)

- ✓ Ce n'est pas un Sprint classique
- ✓ Il sert pour s'organiser et commencer à produire

L'incrément

- ✓ Après la Review
- ✓ Pendant le Sprint

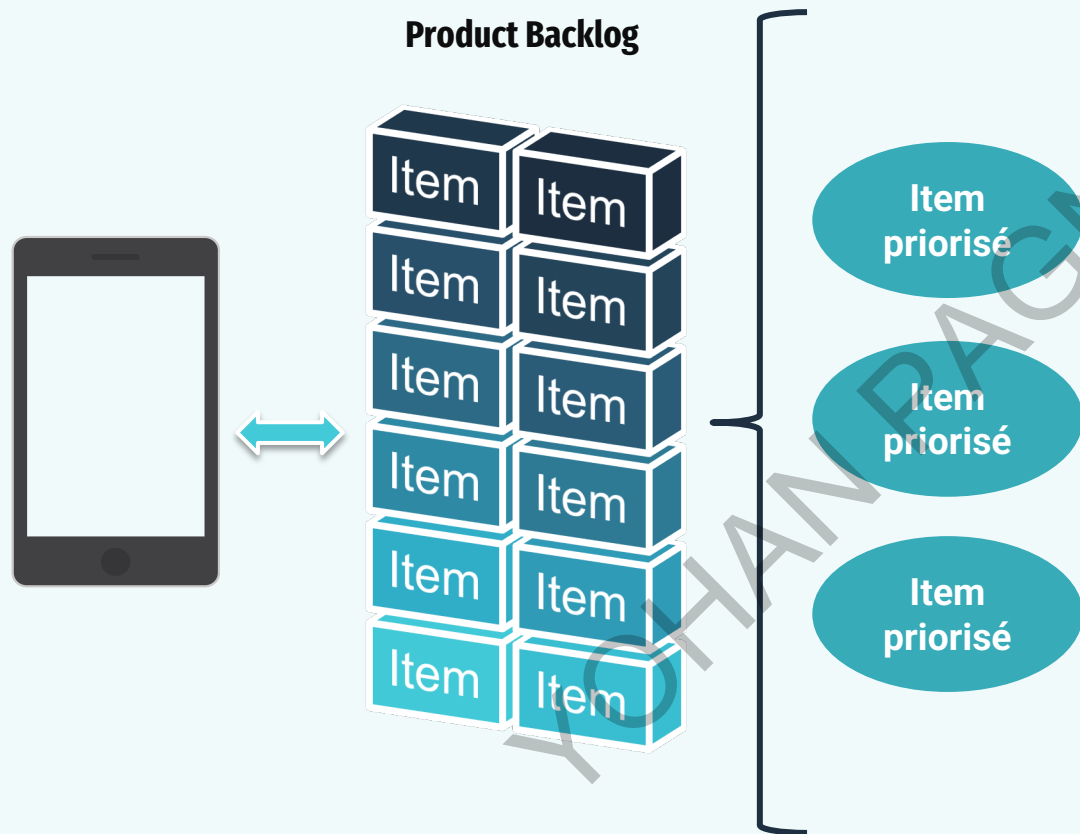
Il y a des jalons en fin de Sprint

- ✓ La Review
- ✓ La Retro

Après plusieurs Sprints : la Release

- ✓ Période de temps plus longue composée de plusieurs Sprint
- ✓ Livraison d'un produit complet





Chaque produit doit avoir son propre objectif (Product Goal) défini par le PO. Il traduit le résultat à atteindre.

Le Product Backlog est une liste ordonnée et émergente de ce qu'il est nécessaire de faire pour atteindre cet objectif.

C'est l'unique source du travail entrepris par la Scrum Team. Et dont le PO en est le responsable.

On peut le comparer à un cahier des charges, en plus léger (focus produit), et qui va constamment évoluer.

L'équipe de développement va devoir traiter les items priorisés, et les réaliser durant le Sprint.

Lorsqu'un item est priorisé et est prêt à être développé, il est transféré dans le Sprint Backlog.

Le Product Backlog est l'outil de TOUTE l'équipe. L'équipe entière décide ce que seront les « items ».

La technique des Users Stories (issue d'XP) est la plus utilisée, mais elle n'est pas obligatoire.

Les items qui seront transformés en fonctionnalités sont des incréments, et doivent être livrés en fin, ou pendant le sprint.

Découpage des Stories

En tant que [persona]

je [souhaite que]

[afin de]

[Persona] : Exprimer de l'empathie. Comprendre comment cette personne travaille, pense et ce qu'elle ressent
[souhaite que] : Décrire le service que va rendre la fonctionnalité, son objectif.
[afin de] : Quel est le principal problème à résoudre

Exemple

En tant que comptable

je souhaite organiser mes bilans par catégorie et par priorité

afin de les classer dans un dossier sécurisé et isolé des autres dossiers.

La définition de fini pour une story est une checklist dont on coche les cases, lorsque les objectifs voulus sont atteints. Il faut qu'elle passe avec succès ses conditions d'acceptation pour être considérée comme un incrément, puis être livrée en fin de Sprint.

Les critères sont décidés dès le Sprint Zéro. Il faut que la DoD soit décidée en équipe, et non pas uniquement par le PO. Il peut ne pas être toujours disponible pour vérifier les conditions d'acceptations.

La feature est un service ou une fonction du produit qui est clair pour les parties prenantes ; et qui va ensuite se décomposer en plusieurs stories. On essaye de référencer les features ailleurs que dans le backlog. Ce sont les users stories, qui découlent des features, qui vont venir fournir le backlog.

Voyager à l'étranger

Epic Story - L'Epic est une story qui est encore trop grosse et trop peu détaillée pour pouvoir être réalisée efficacement lors d'un sprint. Il va donc falloir la décomposer, l'affiner, avant de la considérer suffisamment petite pour qu'elle soit réalisable lors d'un sprint.

Réserver un billet

Payer un billet

Stories (User ou autres) Une user story est un petit morceau de fonctionnalité du produit et qui peut être développé durant un sprint.

Également, un sprint peut servir à l'équipe de développement pour ajouter une fonctionnalité utile uniquement pour l'équipe, régler la dette technique ou corriger un bug.

En tant que client, je souhaite réserver un billet directement depuis mon compte afin de le recevoir par mail au format PDF.

En tant que client, je souhaite recevoir une notification depuis l'application afin de connaître le statut de mon vol.

En tant que client, je souhaite payer par carte bleu Visa, en 3x sans frais, afin d'échelonner mon paiement.

En tant que développeur, je souhaite avoir une notification afin de suivre les incidents résolus.

Le Product Owner est le principal affineur. Il affine parfois seul, et durant le Sprint. Mais il est souhaitable que toute l'équipe Scrum participe.

Il y a deux possibilités pour l'organisation des réunions d'affinage : un rythme régulier (tous les mercredis) ou sur demande (lorsque le Scrum Master constate qu'il y a un risque potentiel d'ambiguïté, ou une pénurie d'US). Évidemment, on peut mixer les deux types de réunions.

Ce sont les User Stories. Elles s'adressent avant tout aux parties prenantes et servent à fournir de la valeur, et décrivent les fonctionnalités et les améliorations du produit.

Story fonctionnelle

Ce sont les problèmes ou les erreurs qui doivent être corrigés dans le produit. Ils sont souvent appelés "tickets de bug" et sont prioritaires en fonction de leur gravité.

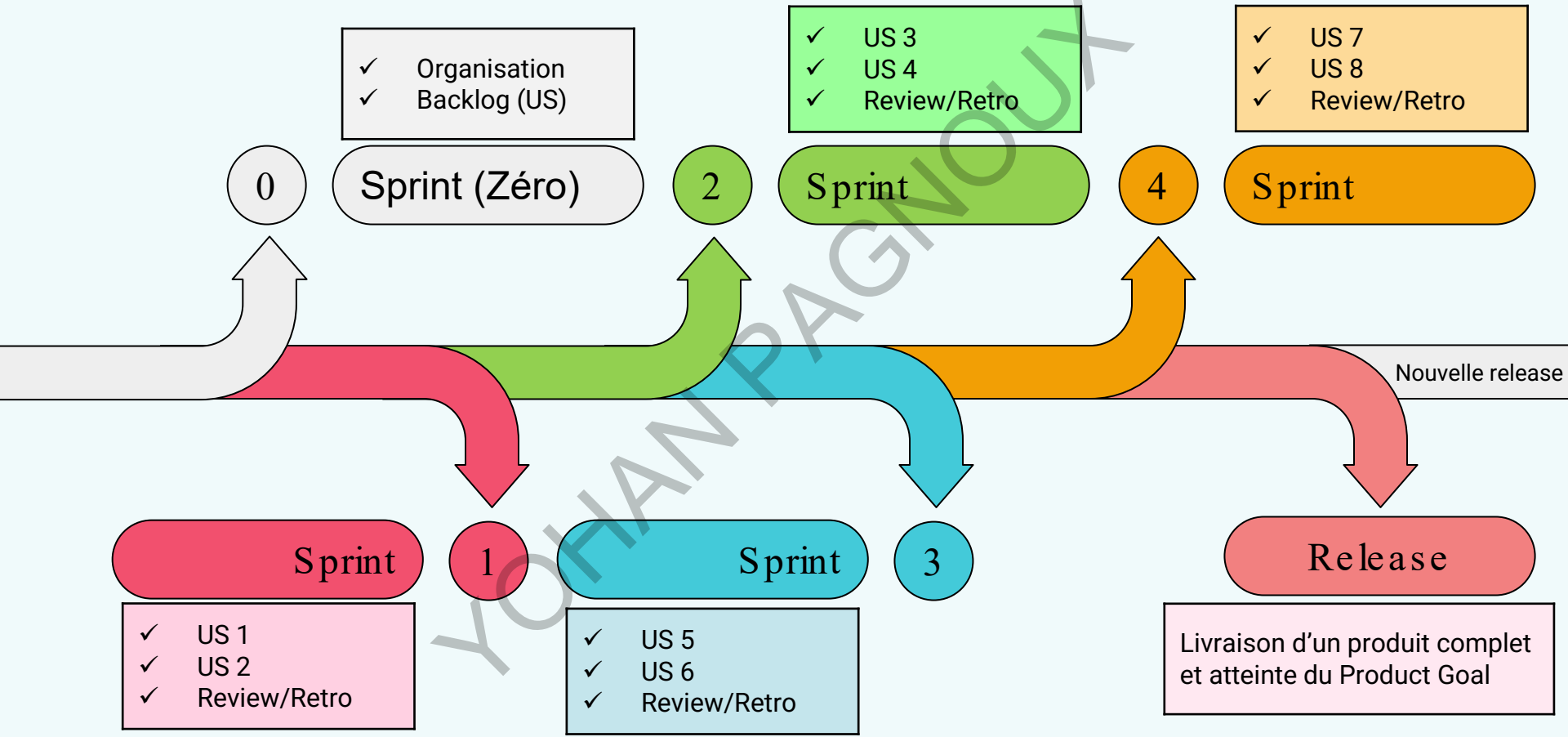
Correction de bug

Travail réalisé par l'équipe et non visible des parties prenantes. Ce sont les améliorations de l'infrastructure, des performances, de la sécurité, etc.
Exemple : étudier puis installer une solution de stockage

Story technique

La dette technique fait référence aux choix de conception et de développement qui ont été faits dans le passé et qui peuvent entraîner des coûts supplémentaires à long terme et rendre le produit moins maintenable. Elle peut inclure des éléments tels que le code obsolète, le manque de documentation, la mauvaise qualité du code.

Remboursement de la dette technique



Roadmap d'une release

Le Daily meeting

Rendez-vous qui a lieu tous les jours. Il donne le rythme d'exécution du Sprint. Le Daily-meeting n'est pas un compte rendu, mais un moment de partage entre développeurs afin d'optimiser l'efficacité de l'équipe.

Le Scrum Master n'en est pas l'animateur officiel. Il participe pour aider les devs à faire disparaître les obstacles. Mais c'est un moment dédié aux devs. Le PO n'est pas obligé d'y participer, mais sa présence est (vivement) recommandée.

La Review de Sprint

La review est l'événement où l'équipe Scrum se réunit avec les parties prenantes pour présenter les fruits de son travail, et voir s'il répond au besoin, pour ensuite récupérer le retour des utilisateurs et enclencher une amélioration continue du produit. C'est un moment de partage avec eux, et ne doit pas se limiter à la démonstration de l'incrément.

À chaque sprint, un incrément est produit, et cela, dès le premier Sprint. On présente ce qu'on a obtenu lors de la review, et l'ensemble de l'équipe Scrum est convié. C'est le Product Owner qui anime les échanges, mais tout le monde participe. Pour un Sprint d'un mois, la durée recommandée pour la Review est d'environ 2 à 3 heures.

La Rétrospective de Sprint

Lors d'un projet classique, il est nécessaire de faire un bilan du projet lorsque celui-ci touche à sa fin. Avec Scrum, on n'attend pas la fin pour faire un bilan : la « rétro » a lieu à chaque fin de Sprint. Et cela, dès le premier Sprint. Elle est limitée dans le temps à un maximum de 2 à 3 heures pour un Sprint d'un mois. Et a lieu le même jour que la review.

La rétrospective permet l'amélioration de processus, l'organisation, et la définition of done. Elle discute des points positifs et négatifs de l'organisation de travail. La présence du PO n'est pas forcément obligatoire, mais encouragée. S'il y a des problèmes pour la compréhension des items du backlog, la review lui permettra de les comprendre.

Focus sur le produit, mais pas sur le « projet »

Il n'incorpore pas naturellement des notions importantes pour la gestion du portefeuille de projet de l'entreprise telles que la gestion des risques, des délais, ou du budget. Il faut donc trouver un équilibre entre le respect de ces paramètres et les avantages qu'il procure pour la conception du produit.

Idéal pour les petites équipes

Scrum limite à 10 personnes, car, au-delà, la communication ou l'auto-organisation peuvent être défaillantes. Les projets avec plusieurs livrables complexes nécessitant des équipes plus conséquentes ne pourront pas bénéficier des bons aspects des approches Agiles.

Ne peut pas s'adapter à tout type de produit

Agile, c'est de la conception de logiciels. Bien que son utilisation ait pu s'étendre à d'autres domaines, il peut ne pas être adapté à tous types de produits. Si l'objectif est d'avoir un produit finalisé ou si le besoin est suffisamment clair, inutile d'adopter cette approche.

Attention à son utilisation

Certaines équipes adoptent partiellement l'approche Agile, comme Scrum, en choisissant ce qui leur convient, mais en négligeant le reste. Cependant, il FAUT respecter les règles et les valeurs recommandées des différentes approches pour éviter des problèmes graves lors de la conception du produit.

Implication constante des parties prenantes

Une participation continue et proactive des parties prenantes est FONDAMENTALE. Si elles ne peuvent pas dédier suffisamment de temps, cela peut entraver le processus, et rendre complètement caduque son utilisation qui est, avant tout, basée sur le rapprochement entre les personnes.

Ne peut pas convenir à tous

Les itérations courtes et fréquentes peuvent entraîner une pression constante sur l'équipe, ce qui peut mener à du surmenage. De plus, si l'équipe manque de compétences spécifiques et a besoin d'un cadre strict, cela peut compromettre son utilisation et la qualité du produit à livrer.

Adapté à des projets plus grands et plus complexes, SAFe est un framework qui propose d'appliquer une approche Agile à l'ensemble d'une entreprise



Itérations



Approche Lean



Pensée systémique



Portefeuilles de
programmes et projets

Il y a 4 niveaux d'application de SAFe dans l'Organisation.
Chaque niveau propose des rôles et des objectifs différents :



Essential



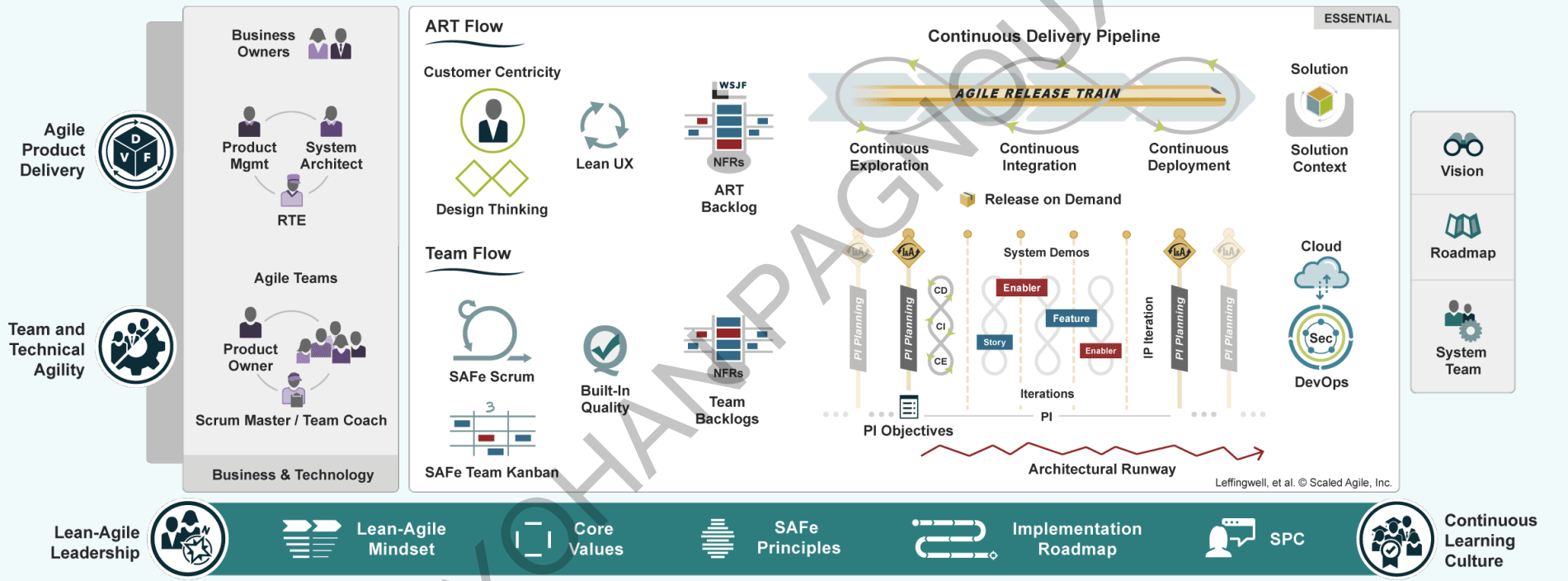
Large Solution

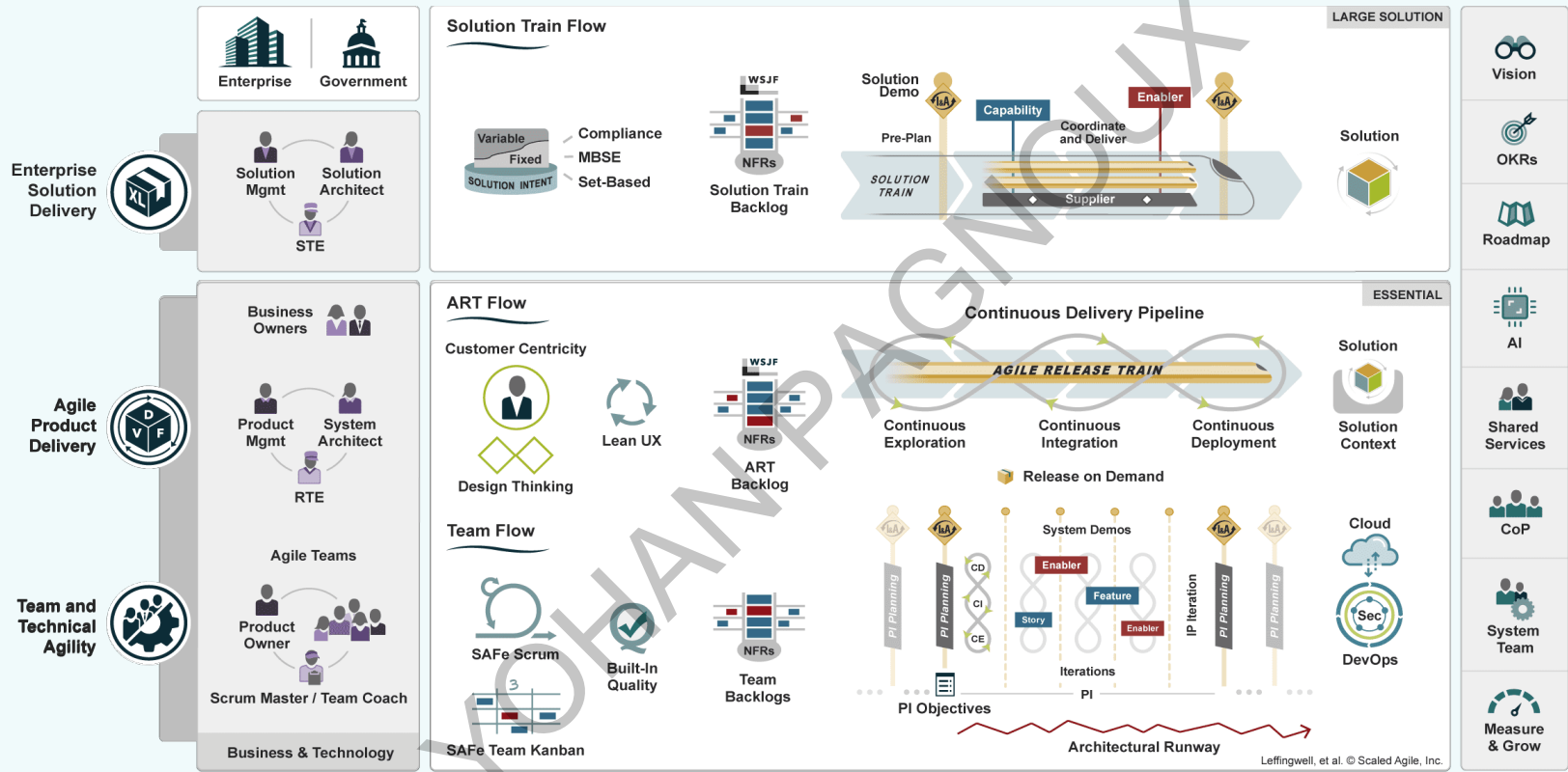


Portfolio

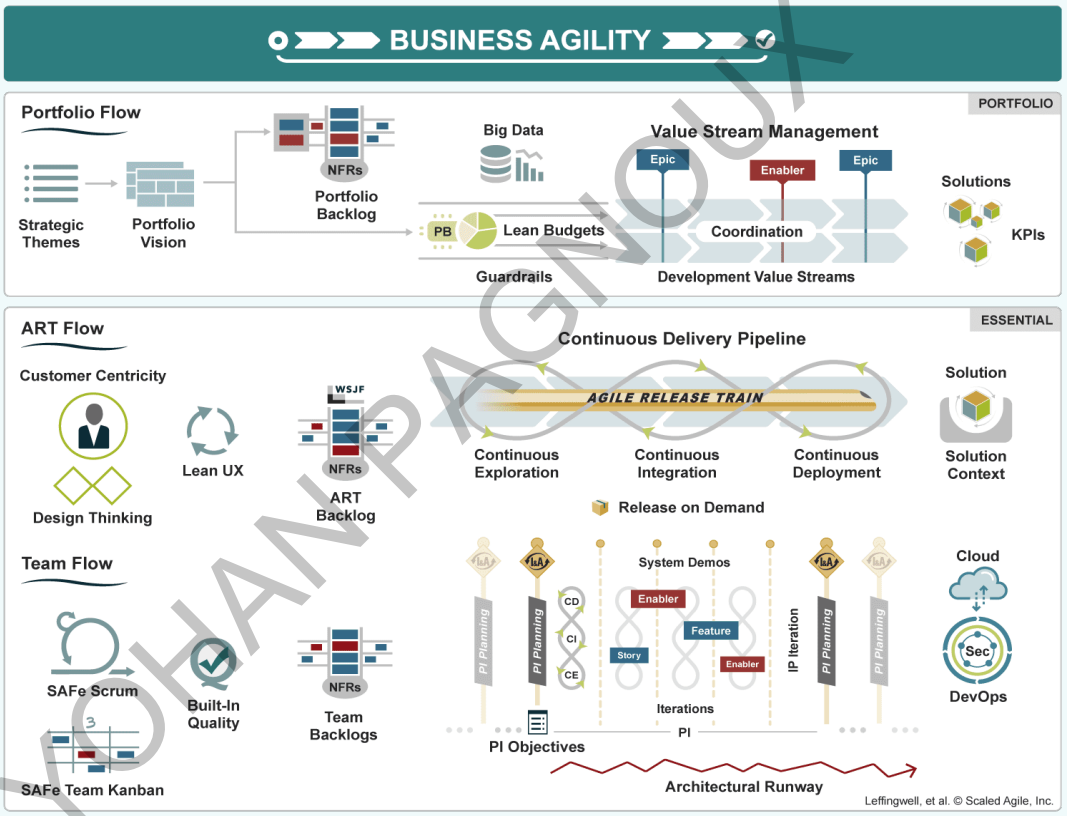
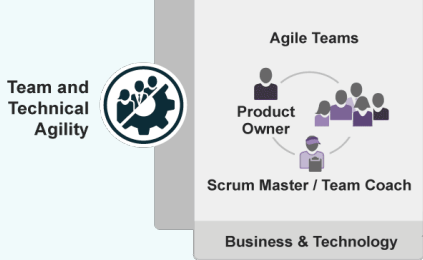
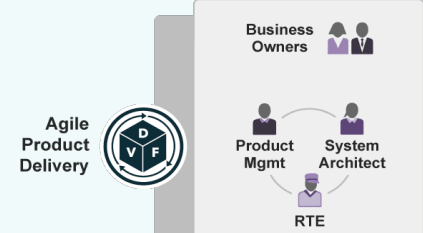
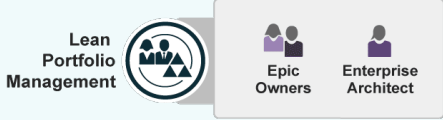


Full SAFe

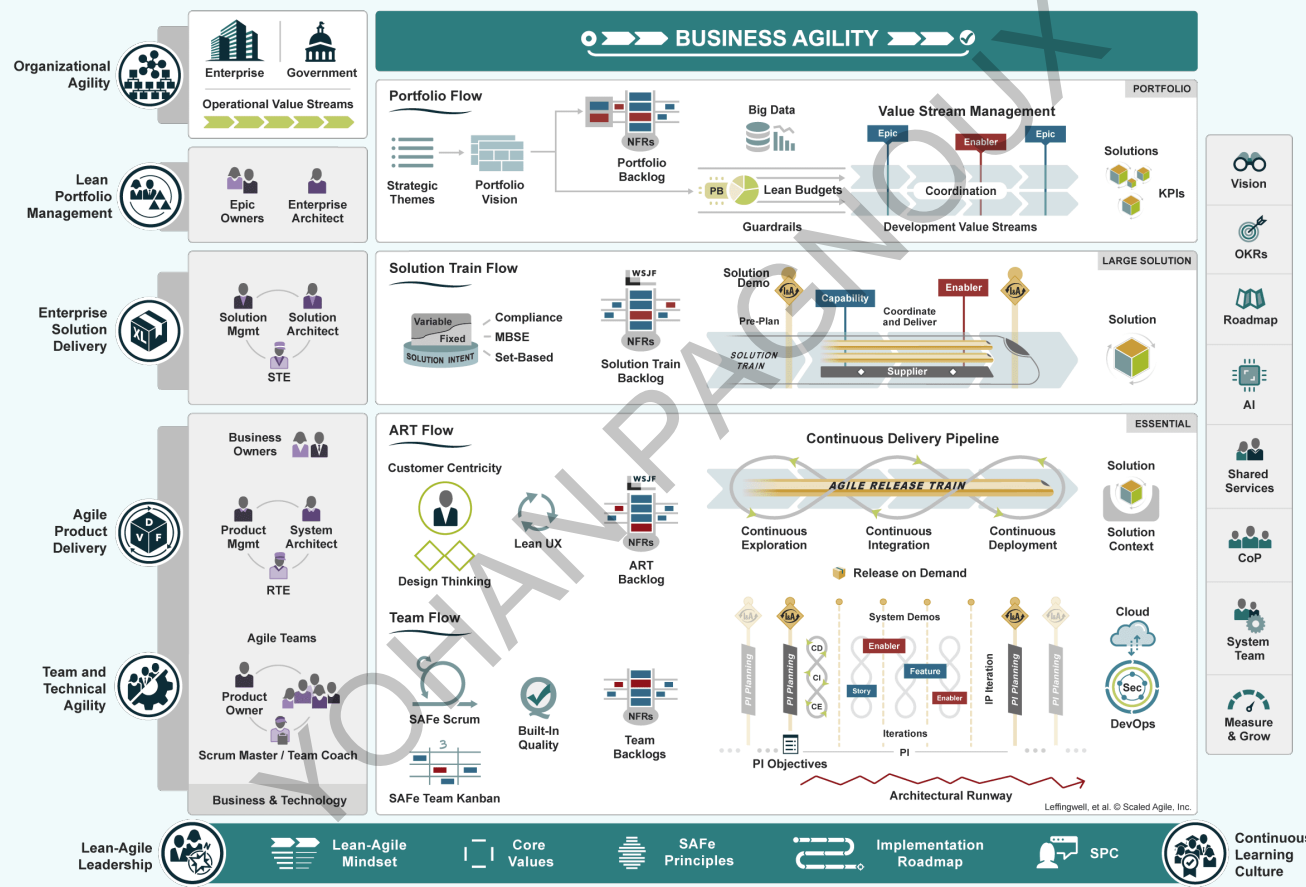




Leffingwell, et al. © Scaled Agile, Inc.



- Vision
- OKRs
- Roadmap
- AI
- Shared Services
- CoP
- System Team
- Measure & Grow



Scrum c'est :

Un framework qui permet d'installer un cadre pour travailler avec une approche Agile (adaptative et incrémental).

Il a pour objectif :

D'installer un cadre de travail pour travailler avec une approche Agile. Il est naturellement incomplet pour laisser le plus de liberté possible afin que l'équipe s'auto-organise, gagne en expérience, pour devenir plus efficace grâce à l'empirisme. On utilise un Product Backlog, composé d'items priorisés, pour ensuite développer les incréments du produit (les fonctionnalités).

Il diffère des approches classiques

- Il n'y a pas de hiérarchie
- Les développeurs travaillent aussi sur la conception, et non pas que sur la réalisation du produit
- L'objectif reste la conception d'un produit, et non pas d'organiser la gestion d'un projet
- Il est fait pour les petites équipes (une dizaine en moyenne)
- C'est un changement radical pour une culture d'entreprise.

3 artefacts

Product Backlog, Sprint Backlog, Incrément

5 événements

Sprint, Sprint Planning, Daily Meeting, Sprint Review et Sprint Retro

Scrum est composé de :

Développeurs

Réalise techniquement le produit, mais participe à toutes les activités de conception et d'organisation de l'équipe.

Un Scrum Master

Leader/serviteur de l'équipe, il aide chaque membre à faire disparaître les obstacles.

Un Product Owner

Responsable de la valeur dégagée par le produit, il priorise les items et aide l'équipe à la compréhension du besoin des utilisateurs.

Parties prenantes

Client et utilisateur : ceux qui ont un intérêt d'usage ou financier par le produit créé.
Expert : Renfort qui aide l'équipe lors des Sprints.
Fournisseurs, sous-traitant, partenaires, etc.