

CHAPITRE 1

SAFe (Scaled Agile Framework)

Une carte interactive présentant les divers éléments du framework est accessible <https://scaledagileframework.com/>

Définition

Le framework SAFe est un framework Lean-Agile permettant de faire travailler plusieurs équipes Scrum ensemble au sein d'une même structure.

On y retrouve de nombreuses pratiques dans SAFe : le Scrum, pratiques du Lean (dont le Kaizen), concept de Product Development flow, le kanban, l'XP, le devops (intégration continue, livraison continue...), le lean startup et l'agile UX.

SAFe 5 for Lean Enterprise

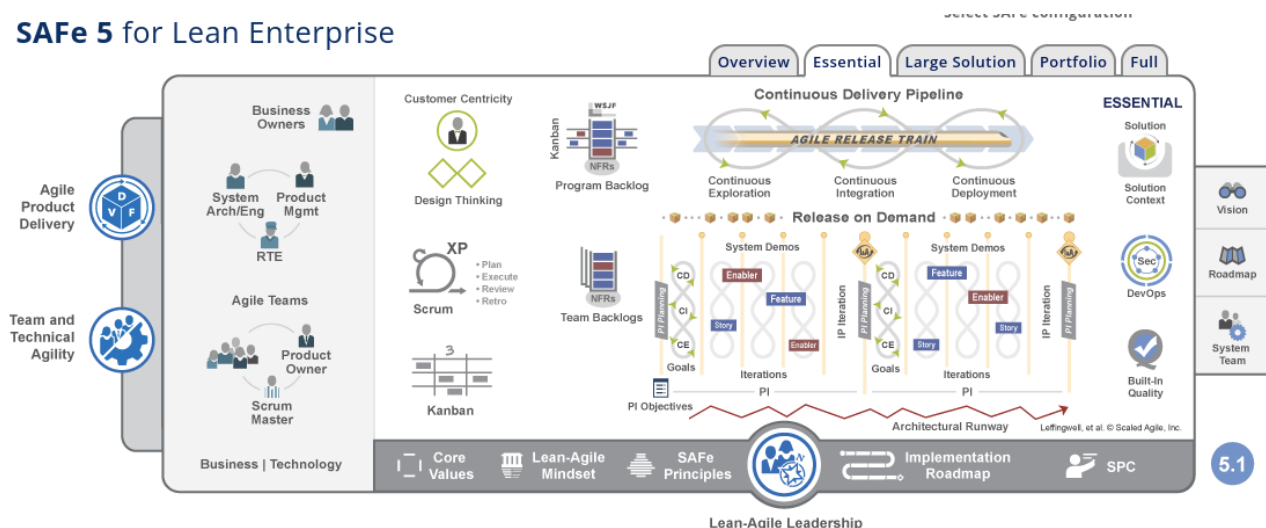


FIGURE 1.1 – SAFe v5 essentiel

Lorsqu'on travaille sur de gros produits il est parfois nécessaire de constituer plusieurs équipes. La mise à l'échelle des équipes (e.g Scrum) peut être faite en exploitant le framework SAFe.

Affirmation

Il faut aligner et coordonner nos équipes Agile. SAFe nous propose une solution pour arriver à cela.

Dans une grosse structure, il serait insensé que chaque équipe développe ce qu'elle veut. SAFe aide les équipes à comprendre la situation dans son ensemble et à s'aligner sur les objectifs de l'entreprise.

1 Configurations SAFe

Le SAFe se structure autour d'un schéma d'organisation spécifique qui définit comment les différents niveaux hiérarchiques et fonctionnels interagissent dans le cadre des projets agiles.

Suivant la taille de la solution, il existe 4 configurations différentes :

1. Essential SAFe : qui définit les rôles, cérémonies et artefacts minimaux pour réussir une transformation à l'échelle.
2. Large Solution SAFe : Lorsqu'une solution ne peut pas être réalisée par une seule *Agile Release Train* il faut alors en coordonner plusieurs.
3. Portfolio SAFe : Maintenant que les équipes informatiques sont Agile il faut que toute la hiérarchie entame une transformation. On améliore toute la *chaîne de valeur*.
4. Full SAFe : combine les 3 niveaux

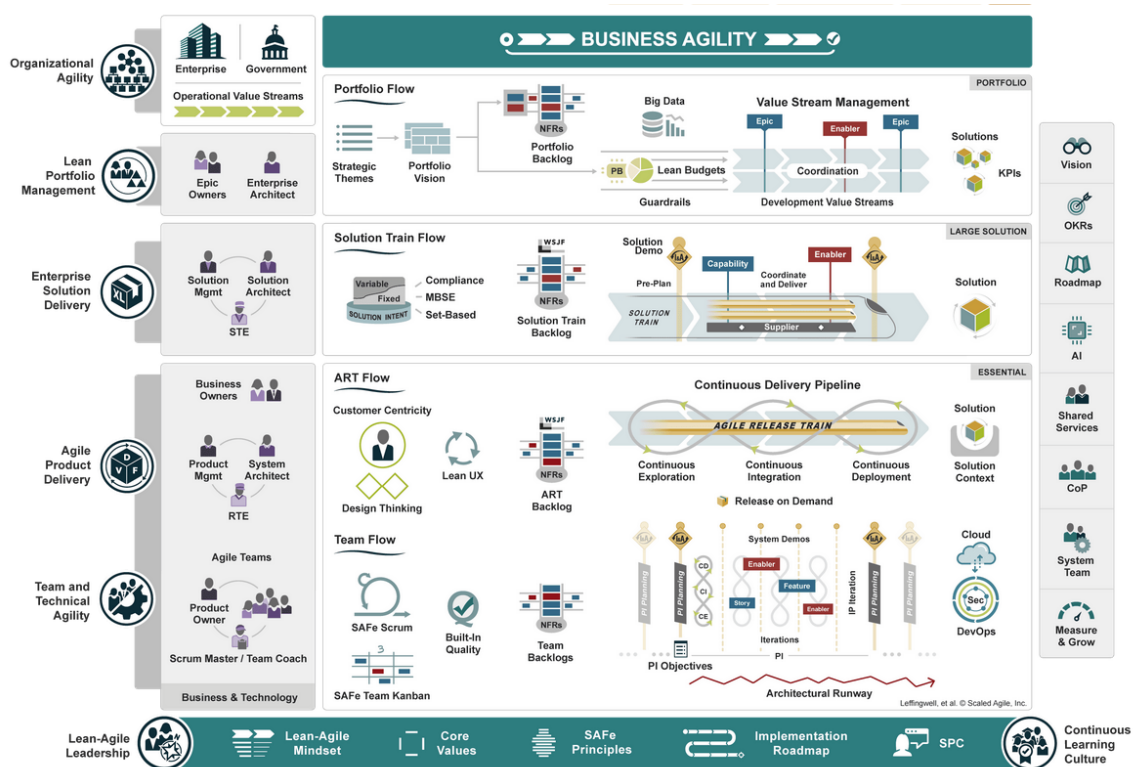


FIGURE 1.2 – Full SAFe

Note : Dans les sections suivantes nous ne décrivons que le premier niveau Essential SAFe

2 Agile Release Train (ART)

Définition

Représente l'ensemble des équipes Agile qui vont travailler ensemble.

L'objectif de l'ART est de faire travailler tout le monde ensemble. Cela représente entre 50 et 125 personnes réparties dans des équipes agiles de 5 à 11 personnes.

Chaque équipe Agile dispose de son Product Owner et Facilitateur Agile (e.g. Scrum Master) et des développeurs. L'ensemble des équipes est coordonné par un RTE, un Product Manager et un System Architect.

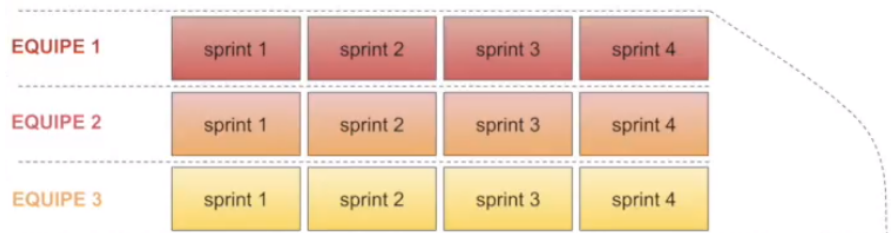


FIGURE 1.3 – Le train composé de n équipes rythmées par des sprints

L'Agile Release Train est donc composée de plusieurs équipes qui vont suivre la même cadence. Souvent 4 itérations de 2 semaines. Toutes les deux semaines, chaque équipe délivre son Incrément et au bout de la 4e itération l'ensemble des Incréments sont intégrés ensemble pour former le *Product Increment*.

2.1 Niveaux

Dans l'ART nous avons deux niveaux

- Le niveau des équipes qui vont délivrer des incréments lors de chaque itération (PO, SM, devs)
- Puis le niveau *programme* qui coordonne toutes les équipes du train via un *backlog program* commun à tout le train. En parallèle chaque équipe à son propre *backlog d'équipe* (RTE, PM, System Architect)

2.2 Coordonner via l'ART

En mettant en place un ART on souhaite coordonner nos équipes. Par conséquent :

1. Au départ, on va tous partir ensemble en faisant un *PI Planning*. C'est une grosse planification permettant de s'aligner sur le travail à réaliser pour les 4 prochains Sprint.
2. Chaque équipe Scrum vont par la suite faire leur sprint planning, sprint review, sprint rétrospective au même moment.
3. A chaque fin de sprint, nous allons faire un *System Demo* où on présente l'ensemble de ce qui a été réalisé sur le sprint par l'ensemble des équipes. Comme avec scrum, l'objectif est d'obtenir des feedback rapidement.

Note : Au bout de deux semaines, l'incrément livré correspond au travail réalisé par l'ensemble des équipes Agile.

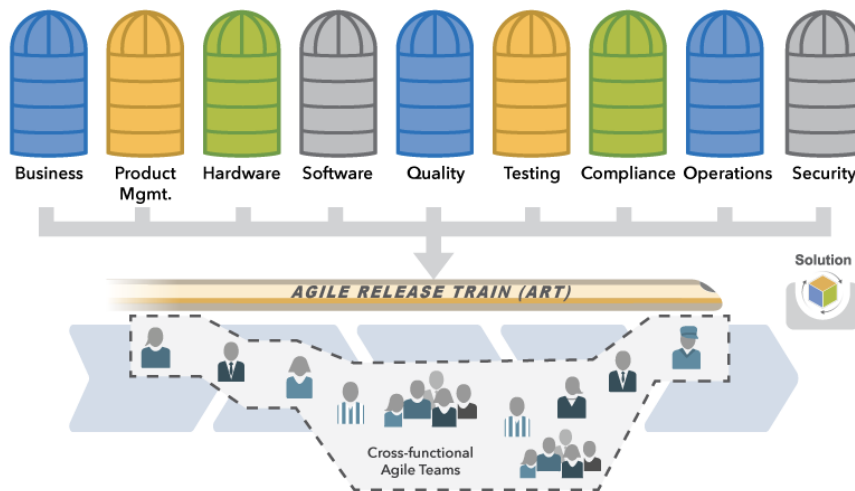
2.3 Organiser sur la valeur

Affirmation

Il est nécessaire d'avoir l'ensemble des compétences nécessaire dans notre ART pour la bonne réalisation du produit.

Dans les grosses organisations, il n'est pas rare de voir les développeurs travailler avec les développeurs, les testeurs avec les testeurs, les architectes et les ingénieurs système travaillent ensemble, etc ...

L'ART propose de briser les silos pour construire des équipes pluridisciplinaires qui est optimisé pour faciliter le flux de valeur depuis l'idéation jusqu'au déploiement et à la diffusion, et jusqu'à l'exploitation. L'équipe dispose de toutes les personnes et de tout ce dont elle a besoin pour définir, fournir et exploiter des solutions. Elle devient auto-organisée. Nous avons donc de la valeur qui circule plus rapidement.



2.4 Rôles

Affirmation

Afin de faciliter l'alignement des différentes équipes, SAFe propose différents rôles.

Les rôles suivants facilitent la mise en place de l'ART, ils se placent au-dessus des équipes scrum afin de les coordonner :

- *Release Train Engineer* : facilite l'exécution du programme, l'élimination des obstacles, la gestion des risques et des dépendances, et l'amélioration continue ("Super Scrum Master")
- *Product Management* : est le responsable du produit. Il travaille avec le client et les POs pour comprendre et communiquer sur leurs besoins.
- *System Architect/Engineering* : une personne ou une équipe qui définissent l'architecture globale du système. Il définit les exigences non fonctionnelles, les éléments du système, les sous-systèmes, etc ...

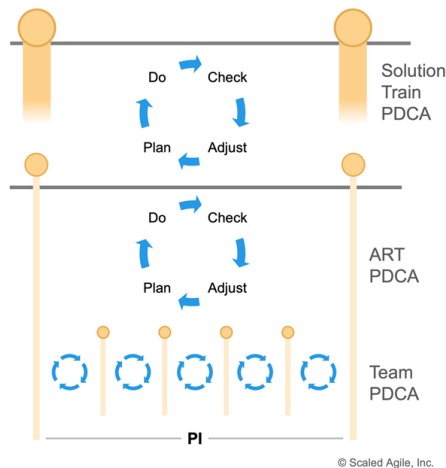
3 Cadence

3.1 Program Increment

Définition

Le Program Increment (PI) d'un ART est le résultat d'une itération de 8 à 12 semaines

SAFe est composé de deux niveaux de cycles imbriqués. Un cycle macro *Program Increment* de 10 semaines composées de plusieurs itérations de deux semaines.



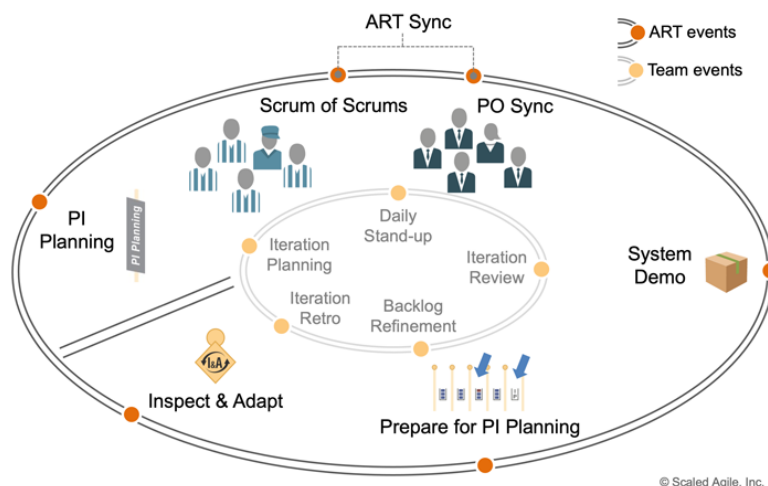
SAFe divise le temps de développement en plusieurs séries d'itération au sein d'un même PI.

Chaque cycle PI suit les évènements suivants :

- Planifier le cycle via me *PI Planning*
- Faire : PI exécution
- Check : via le *System Demo*
- Adjust : via le *Inspect and Adapt*

3.2 Exécution de la Program Increment Iteration

Sur la figure ci-dessous on remarque que l'équipe va à la fois faire des réunions pour gérer l'ART puis chaque équipe Agile vont faire leurs propres cérémonies en interne.



3.3 PI Planning

Définition

C'est la planification du Program Increment, l'équipe va planifier, construire, valider la prochaine Product Increment Itération. On obtient le PI Objective

Le rôle du PI Planning est de faire le pont entre le niveau *programme* et le niveau *équipe*. Pour ce faire, le RTE, le PM, le System Architect ainsi que les SM, PO et développeurs nécessaires au bon déroulement de la *PI Itération* vont se réunir pendant 2 jours pour se donner les objectifs, se coordonner et s'aligner.

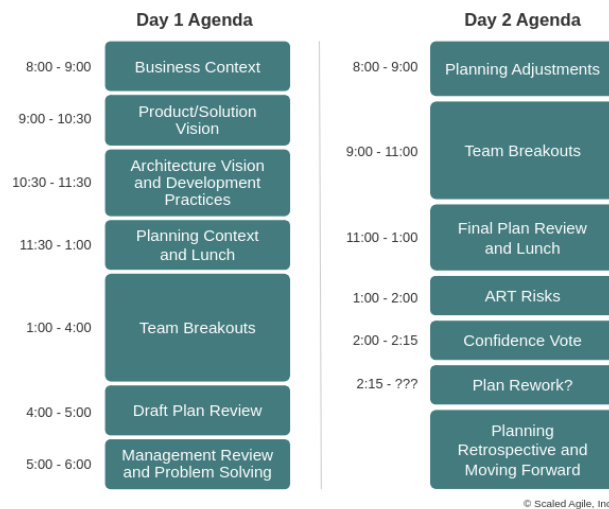


FIGURE 1.4 – PI Planning sur deux jours

Réduire les dépendances

Affirmation

Lors du PI Planning nous allons identifier des dépendances entre les équipes. Et nous devons les réduire au maximum.

On ne doit pas chercher à synchroniser les dépendances, mais bien à faciliter la réallocation des éléments de travail de chaque équipe. Ainsi pour chaque dépendance identifiée on va se demander comment l'éliminer ou la minimiser.

Les équipes décident

Affirmation

Durant le PI Planning les équipes définissent le rythme ainsi que le niveau de qualité.

Ce n'est pas à l'équipe *programme* de décider la charge de travail des équipes. Dans le cas d'une charge de travail trop élevée, la qualité du produit sera négativement impactée.