

# L'ARCHITECTURE D'ENTREPRISE AU CŒUR DE LA TRANSFORMATION AGILE



Janvier 2019

LIVRE BLANC

## *Rédacteurs*

*Nicolas CHEVALIER  
GLUENDO*

*Patrice FRICARD  
SOCIETE GENERALE*

*Pierre Frédéric ROUBERTIES  
CENTRALESUPELEC EDUCATION*

## Table des matières

<b>1. .... Pourquoi un livre blanc sur l'Architecture d'Entreprise et l'agile à l'échelle ?</b>	<b>4</b>
<b>2.     <i>Méthodologie de travail</i></b>	<b>6</b>
<b>3.     <i>Panorama des cadres d'agile à l'échelle</i></b>	<b>7</b>
<b>3.1   Panorama des cadres d'agile@scale</b>	<b>7</b>
<b>3.2   Un peu d'histoire</b>	<b>7</b>
3.2.1   Émergence de l'agile dans les années 90 en réaction aux approches classiques jugées trop lourdes	7
3.2.2   Généralisation avec la transition numérique à partir de 2007	7
3.2.3   Nécessité de s'organiser pour passer l'agile à l'échelle à partir des années 2010	9
<b>3.3   Présentation des principaux cadres d'agile@scale</b>	<b>10</b>
3.3.1   Scrum of Scrums / Nexus	10
3.3.2   Large Scale Scrum (LeSS)	11
3.3.3   Spotify	13
3.3.4   Scaled Agile Framework (SAFe® 4.6)	15
3.3.5   Disciplined Agile (DA 3.0)	17
3.3.6   Autres cadres	21
3.3.7   Positionnement relatif des différents cadres d'agile@scale	22
<b>4.     <i>Les approches d'Architecture agiles</i></b>	<b>25</b>
<b>4.1   La place de l'architecture dans les cadres d'agilité à l'échelle</b>	<b>25</b>
4.1.1   Relation de Nexus à l'architecture	25
4.1.2   Relation de LeSS à l'architecture	25
4.1.3   Relation de Spotify à l'architecture	26
4.1.4   Relation de SAFe® à l'architecture	26
4.1.5   Relation de Disciplined Agile à l'architecture	27
4.1.6   Analyse comparative et leçons à tirer	28
<b>4.2   L'approche « Continuous Architecture »</b>	<b>29</b>
<b>4.3   L'approche « Evolutionary architecture »</b>	<b>30</b>
<b>4.4   L'approche Open group</b>	<b>33</b>
<b>4.5   L'approche Collaborative Enterprise Architecture (Bente)</b>	<b>35</b>
<b>4.6   L'approche Architecture Flexible</b>	<b>37</b>
<b>5.     <i>Des changements majeurs pour l'AE</i></b>	<b>39</b>
<b>5.1   De nouveaux espaces de jeu du fait de la transformation digitale</b>	<b>39</b>
<b>5.2   Risques et opportunités de l'agile@scale</b>	<b>40</b>
5.2.1   Principaux risques pour l'Architecture d'Entreprise	40
5.2.2   Principales opportunités pour l'Architecture d'Entreprise	41
<b>6.     <i>Quelle réponse de l'Architecture d'Entrepris</i></b>	<b>42</b>
<b>6.1   Soutenir l'agile@scale</b>	<b>42</b>

<b>6.2 Gagner le défi de la confiance .....</b>	<b>47</b>
<b>6.3 Revoir son cadre d'activité .....</b>	<b>51</b>
<b>6.4 Basculer d'une logique projet à une logique produit .....</b>	<b>60</b>
6.4.1 La phase de cadrage (Framing) .....	60
6.4.2 L'exécution .....	61
<b>6.5 Faire évoluer ses méthodes et livrables .....</b>	<b>62</b>
6.5.1 Appréhender la stratégie : Dynamique d'un monde VUCA .....	62
6.5.2 Le modèle Cynefin : un outil d'aide à la décision .....	65
6.5.3 Exemple de nouveaux outils .....	65
<b>7. Réussir la transition .....</b>	<b>67</b>
7.1 Les grandes étapes du passage à l'agile .....	67
7.2 Les modèles top-down et bottom-up.....	68
7.3 Transformer les cellules d'Architecture.....	69
7.4 Evolution du rôle des architectes.....	70
<b>Conclusion .....</b>	<b>72</b>
<b>Annexe : Enquête conduite en décembre 2018.....</b>	<b>74</b>
<b>Bibliographie et référence.....</b>	<b>86</b>

## Le Club



Exigences métiers, entreprise numérique, transformation du SI, maîtrise des données, opportunités technologiques, ... Le Club Urba-EA aborde les différents sujets liés à l'Architecture d'Entreprise (ou «Enterprise Architecture, EA»). Le Club Urba-EA est une association inter-entreprises, créée en 2000 par AXA, FNAC, ORESYS, RATP, SUEZ Lyonnaise des Eaux. Le Club est une association professionnelle à but non lucratif.

Transformation numérique et Architecture d'Entreprise



Les 3 domaines cibles de l'Architecture d'Entreprise

### Les actions menées par le club

#### Les Mardis de l'Urba-EA organisés en partenariat avec le CIGREF

Rendez-vous mensuels thématiques sur les tendances en architecture et en gouvernance du Système d'Information, sur la base d'exposés d'entreprises, d'experts et de fournisseurs.

Exemples de sujet :

- Architecture d'Entreprise et Maîtrise de l'information
- Quelles architectures pour le Big Data
- Sécurité du SI et Architecture d'Entreprise
- Apports & réalités du TOGAF- Open group
- Projets agiles et DevOps : accélérer la transformation du SI...

#### Les projets d'échanges et de capitalisation

Menés en groupes de travail, ils sont l'occasion de partager les expériences entre les entreprises et de capitaliser des best-practices.

Exemples de sujet :

- Dimensions économiques de l'EA
- Clés et pratiques de travail de l'AE avec les Métiers
- Référentiels de données d'entreprise - MDM
- Vers un SI hybride : AE et solutions Cloud
- Gouvernance du Patrimoine Applicatif, ...



1. S'échanger des informations entre professionnels en charge de l'évolution des systèmes d'information.



2. Partager les expériences et capitaliser les savoir-faire.



3. Promouvoir les démarches auprès des entreprises (Directeurs de SI, Maîtres d'ouvrage, Management, ...) ainsi qu'auprès de l'enseignement supérieur

Rejoignez les **80** entreprises adhérentes et plus de **100** membres actifs sur [urba-ea.org](http://urba-ea.org)



Dans le prolongement des travaux du Centre d'Excellence en Architecture d'Entreprise (CEISAR) de l'Ecole Centrale Paris, CentraleSupélec Executive Education propose des formations à l'architecture pour les professionnels en activité, dont le Mastère Spécialisé Architecte des Système d'Information, Option Architecture d'Entreprise. En effet, l'architecture des SI occupe historiquement une place centrale dans son offre de formation sur l'IT et le Digital. Soucieux de faire progresser l'état de l'art des pratiques d'architecture, CentraleSupélec est heureux de pouvoir contribuer avec des organisations professionnelles à des livres blancs sur des thématiques au cœur des préoccupations actuelles des acteurs du domaine, comme la relation entre l'architecture et l'agilité.

# 1. Pourquoi un livre blanc sur l'Architecture d'Entreprise et l'agile à l'échelle ?

---

La décennie qui s'achève a vu émerger des évolutions majeures dans l'environnement des entreprises :

- L'accélération des **innovations technologiques** a rendu possible un nouveau monde où le digital est omniprésent, modifiant les équilibres stratégiques dans tous les secteurs d'activité et obligeant les acteurs historiques à se remettre en cause face à la disruption de nouveaux entrants
- Des **évolutions sociologiques** importantes se sont opérées : sous l'impulsion des géants du numérique, de nouveaux modes d'expression et de relations entre les êtres humains sont apparus, transformant à jamais nos sociétés (Web 2.0, l'âge de la multitude, économie collaborative, démocratie participative ...)

Dans ce contexte, quelques facteurs essentiels sont à prendre en compte pour les entreprises :

- Le **client** devient central : l'expérience utilisateur doit être pensée pour répondre à ses besoins intimes. Cela concerne le client final, le consommateur, l'utilisateur des produits et services de l'entreprise mais aussi les employés, qui sont les clients internes de l'organisation.
- **L'innovation** est une absolue nécessité dans un environnement de plus en plus concurrentiel et l'entreprise doit se transformer pour créer les conditions de la réinvention permanente de ses offres et de ses activités.
- **L'ouverture** vers l'extérieur devient la règle. L'entreprise se rapproche de ses clients et de ses partenaires et devient un maillon d'un **écosystème**.
- L'exigence de **vitesse** et de **réactivité** est de plus en plus forte : dans un environnement mondialisé où la concurrence est de plus en plus acerbée, chaque entreprise doit savoir agir et réagir vite pour développer son avantage concurrentiel ou limiter celui de ses concurrents.
- **L'incertitude** devient la norme : les potentialités offertes par les nouvelles technologies à un rythme accéléré rendent le futur difficilement prévisible. L'entreprise doit apprendre à gérer cette incertitude et à tirer profit de toutes les opportunités tout en limitant les risques.
- L'exigence de vitesse et l'incertitude poussent l'entreprise à développer une capacité **d'adaptation** en continu
- L'explosion des opportunités portées par les technologies et leur accélération génère également plus de **complexité** que jamais.



Les défis auxquels sont confrontés les entreprises et organisations nécessitent d'une part de concentrer les énergies sur le développement de la valeur, et d'autre part d'accélérer les capacités d'actions-réactions. C'est à ce double titre que la transformation « agile » est un élément central de réponse à l'impératif de vélocité.

Dans la continuité des approches de qualité totale (PDCA, Lean Management, ...), les démarches agiles mettent en œuvre une boucle de feed-back qui permet de **délivrer plus vite** tout en traitant **l'incertitude** : **l'équipe apprend** grâce au retour du client à chaque itération, limitant ainsi les risques métier et techniques et convergeant vers le produit qui apporte le **plus de valeur**.

Il n'est donc pas étonnant que l'agile devienne progressivement la norme en gestion de projet innovant, bien au-delà des frontières des équipes de développement où elles sont restées longtemps cantonnées.

Cette généralisation de l'agile à un grand nombre d'équipes projets a des conséquences importantes. Elle implique de :

- Repenser les activités de définition de la stratégie, sa déclinaison dans un portefeuille de projets, et la coordination des équipes de mise en œuvre
- Mettre en place un grand programme de changement pour que l'organisation comprenne et intègre la culture de l'agile et performe dans sa mise en œuvre à grande échelle.

La révolution agile remet en cause beaucoup de réflexes en place dans les entreprises depuis des décennies et bouscule l'organisation, les processus et les métiers historiques.

L'architecture d'entreprise n'échappe pas à la règle. Perçue par certains comme une activité de planification stratégique centrée sur le long terme, lourde à mettre en œuvre, sans garanties de résultats rapides, elle est parfois considérée comme inopérante et donc inutile dans un environnement totalement incertain.

Pourtant, ses objectifs d'alignement sur une vision stratégique, de rationalisation, de maîtrise de la complexité, et de mise en commun de ressources partagées paraissent plus que jamais pertinents dans l'environnement actuel.

Dans un article de McKinsey d'août 2016 " How Enterprise Architects can help ensure success with digital transformations " <sup>1</sup>, les auteurs démontrent que la transformation numérique augmente sensiblement la complexité de l'architecture. Ils mettent également en exergue que la démarche d'architecture apparaît plus que nécessaire, à condition de comprendre la philosophie de l'agile, d'adapter les compétences, l'organisation et les pratiques des architectes à ce nouvel environnement.

Les approches agiles déployées à grande échelle, combinées avec une démarche d'architecture rénovée doivent permettre de reconfigurer et d'aligner les activités de transformation de l'entreprise pour répondre aux défis de notre temps.

Ce livre blanc vous propose :

1. d'explorer l'état de l'art
  - a. des cadres d'agile à l'échelle<sup>2</sup> et la place qu'ils donnent à l'architecture
  - b. des démarches d'architecture d'entreprise agile développées par des experts telles que « Continuous Architecture », « Evolutionary Architecture », « Collaborative EA » ...
2. de bénéficier de l'expérience d'entreprises témoins qui ont bien voulu partager les leçons de leurs projet de mise en œuvre de l'agile, ainsi que les résultats de notre enquête sur les relations entre l'agile à l'échelle et l'architecture
3. de découvrir quelques facteurs clés de succès dans la mise en œuvre de pratiques d'architecture d'entreprise agile et
4. d'introduire le programme de transformation qu'il faut engager pour y parvenir (et en particulier des focus sur la posture, les compétences, l'organisation, les pratiques et les livrables)

---

<sup>1</sup> <https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/how-enterprise-architects-can-help-ensure-success-with-digital-transformations>.

<sup>2</sup> Nous utiliserons dans ce document le terme « agile@scale » communément utilisé pour désigner l'agile à l'échelle

## 2. Méthodologie de travail

Notre approche pour écrire ce livre blanc s'est articulée en quatre étapes :

1. En externe : la prise en compte de l'état de l'art très riche sur le thème de l'agile à l'échelle et de l'architecture. De nombreux travaux sont disponibles dans des articles, des livres, des cadres méthodologiques... et nous avons donc souhaité en faire une synthèse pour nourrir notre groupe de travail.
2. En interne : Les travaux groupe de travail « Agile à l'échelle et AE » du club, reposant sur des réunions de réflexion, des échanges avec les membres du club lors des « mardis de l'Urba » et des ateliers d'explorations organisés avec des entreprises témoins, que nous remercions (Pôle Emploi, Ministère de l'Education nationale et de la recherche, Ventes-privées.com,...) ainsi que la mise au point d'une enquête en ligne (dont les résultats sont présentés en annexe).
3. En collaboration avec le CIGREF : L'organisation d'un colloque le 13 décembre 2018 avec pour titre « Architecture et Agile : Oui, c'est possible » qui aura permis de présenter les premiers résultats de nos travaux, leur cohérence et complémentarité avec les travaux du CIGREF sur l'agilité, les nouvelles architectures. Le colloque a aussi été l'occasion de bénéficier de retours d'expériences complémentaires d'AMADEUS, de la Société Générale, ...
4. La synthèse des points clés émergents de ces travaux dans ce livre blanc

Animation du groupe de travail du club urba 2018	
Nicolas CHEVALIER	GLUENDO
Bruno GUENODEN	COVEA
Inscrits aux groupes de travail 2018	
Valérie PENAUD	POLE EMPLOI
Claire ROBIEUX	ACCOR
Alexandre LARROUMET	AG2R LA MONDIALE
Rodriguez ANTONIO	GROUPEAGRICA
Philippe HANOT	AIRBUS
Frederic PIERRON	LA POSTE
MARC ESCUYER	BANQUE DE FRANCE
OLIVIER GOUPIL	BNPP
JACQUELINE SUAU	CARREFOUR
Corinne GARNERONE	CEA
Regis HEURDIER	CNAV
Patrice FRICARD	SOCIETE GENERALE
Pierre-Frédéric ROUBERTIES	CENTRALESUPELEC
André LISSARRAGUE	CNRS
Marylene LE MONNIER	EDF
Nadzeya OULE	EDF
Jean-François PEREZ	INSEE
Chrystelle SALLE	LA POSTE
Nicolas CHAIGNEAU	Ministère de l'éducation
Catherine BLANZAT	CNRS
Eric BELOUET	MNH
Martine PION	RADIO FRANCE
Olivier CONSTANT	RHAPSODIES

## 3. Panorama des cadres d'agile à l'échelle

---

### 3.1 Panorama des cadres d'agile@scale

**Comment les entreprises peuvent-elles passer leurs démarches agile@scale en s'appuyant sur les cadres disponibles sur le marché ?**

### 3.2 Un peu d'histoire

Afin de comprendre comment nous en sommes arrivés à la généralisation de l'agile@scale, il est intéressant d'analyser comment les démarches agiles sont nées et se sont développées.

#### 3.2.1 Émergence de l'agile dans les années 90 en réaction aux approches classiques jugées trop lourdes

Alors qu'on a trace de développement incrémental dès la fin des années 50 puis de développement logiciel adaptatif dans les années 70 aux États-Unis, l'émergence réelle de méthodes de développement incrémental est plutôt située dans les années 90 : d'abord avec les approches RAD (1991) puis « Unified Process » (UP) et DSDM (1994). SCRUM est mentionné pour la première fois par ses auteurs, Ken Schwaber et Jeff Sutherland lors de la conférence sur la programmation orientée Objets OOPSLA en 1995.

L'agile n'est pas une création spécifique du monde du logiciel puisque des approches de conception basées sur la collaboration simultanée de tous les contributeurs dans une équipe autonome et co-localisée existent déjà dans le monde de l'aéronautique sous le terme d'**ingénierie concourante**, comme à l'ESA <sup>3</sup>(Concurrent Design Facility).

L'agile s'est développé en réaction aux problèmes rencontrés avec les méthodes séquentielles classiques : lourdeur de la documentation qui n'est pas alignée avec le produit, projet trop long par rapport aux attentes des métiers, livrable qui ne correspond pas aux besoins métier, effet tunnel (longue durée et manque de feed-back entre l'expression de besoins et la validation du produit fini, pouvant induire la non satisfaction du besoin, soit qu'il ait été mal compris, soit qu'il ait évolué pendant le projet) ...

En 2001, les principaux promoteurs des approches de développement logiciel agile se regroupent pour produire le **Manifeste Agile**. A partir de cette date, les approches agiles se structurent progressivement et se rationalisent principalement autour de SCRUM, XP et Kanban. L'agile est surtout utilisé par les éditeurs de logiciels, industriels de l'électronique et des systèmes embarqués et les sociétés de services informatiques. Dans les entreprises utilisatrices, son usage reste assez confidentiel, centré sur des projets de taille modeste, et pratiqué à l'intérieur des directions informatiques dans les équipes de développement logiciel. L'agile est aussi largement utilisé par les startups de la bulle internet au tournant des années 2000 mais la vraie révolution numérique n'est pas encore là ...

#### 3.2.2 Généralisation avec la transition numérique à partir de 2007

Avec l'apparition progressive des géants du numérique et le cortège des startups du digital à leur suite, les approches agiles deviennent de plus en plus la norme dans cet environnement. Dans le même temps, des « serial entrepreneurs » comme Eric Ries avec sa méthode **Lean Startup**, vont faire le lien avec les approches d'**amélioration continue**, issues des démarches de qualité totale et de **Lean Management**. Dans un environnement très incertain, où la capacité d'innovation et la vitesse sont essentielles, la **boucle de feed-back** est l'alliée de l'entrepreneur : il va pouvoir définir et développer son produit de manière

---

<sup>3</sup> European Space Agency



incrémentale en s'appuyant sur les retours de ses clients. Le **client** est en effet au centre de toutes les attentions, comme le montre la généralisation des démarches de **Design Thinking**, elles aussi très marquées par une démarche itérative de découverte du « problème » du client et de conception de prototypes de solutions.

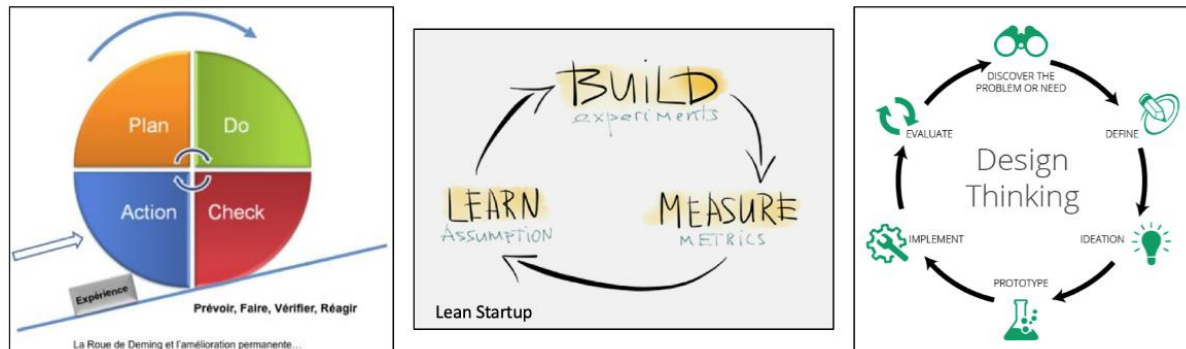


Figure 1. La boucle de feed-back est centrale dans les différentes démarches

La transition numérique amène la **généralisation des approches agiles** :

- Les technologies de l'information deviennent centrales dans la stratégie des entreprises. Le logiciel est au cœur de ces nouveaux projets, ce qui expose de plus en plus d'acteurs de l'entreprise à ces approches au contact des équipes de développement
- Le rapprochement des approches agiles IT, avec le Design Thinking et Lean Startup, qui partagent le même ADN de l'amélioration continue va également favoriser le déploiement de l'agile
- Les transformations sociétales et technologiques actuellement à l'œuvre poussent les entreprises à revoir leur culture et leurs modes de management : le concept d'« entreprise agile » émerge, application beaucoup plus profonde (que le développement de produit) de la philosophie agile à toute l'entreprise

Ainsi on constate que les entreprises adoptent de manière exponentielle les approches agiles et que cette adoption généralisée est encore assez récente :

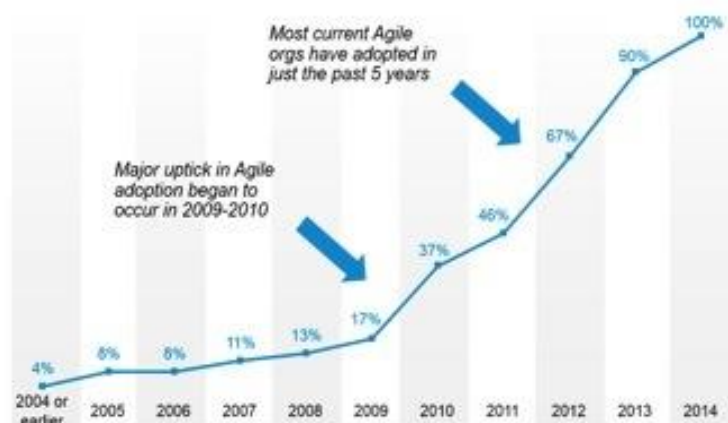


Figure 2. Enquête HP 2015 sur 601 professionnels de l'IT aux US

### 3.2.3 Nécessité de s'organiser pour passer l'agile à l'échelle à partir des années 2010

Les approches agiles ont fait leurs preuves chez les « pure players » du numérique, sur l'ensemble de leurs produits, ce qui peut impliquer des centaines, voire des milliers de développeurs. Dans les entreprises utilisatrices également, les projets candidats aux approches agiles se multiplient et la question de coordonner les équipes agiles se pose.

Les méthodes agiles comme SCRUM insistent sur l'importance d'équipes autonomes, pluridisciplinaires, auto-organisées et si possible co-localisées pour favoriser la communication et la collaboration avec des outils de management visuel. Elles semblent donc plutôt adaptées à des projets à taille humaine, pouvant être menés à bien par une équipe de 5 à 10 personnes (« two-pizzas team »). On peut donc s'interroger sur le bien-fondé de vouloir généraliser ces approches sur un grand nombre d'acteurs. Il va falloir ajouter des couches de coordination et de pilotage d'un portefeuille d'activités. Les tenants de l'agile y voient un risque d'une gouvernance trop lourde et une perte de la proximité qui permet la collaboration informelle au sein de l'équipe agile. Cependant, quelques experts, capitalisant sur leur expérience, développent progressivement des « cadres » **d'agile@scale, ensemble de bonnes pratiques, de principes et de recettes pour organiser un ensemble d'équipes, tout en conservant la philosophie agile.**

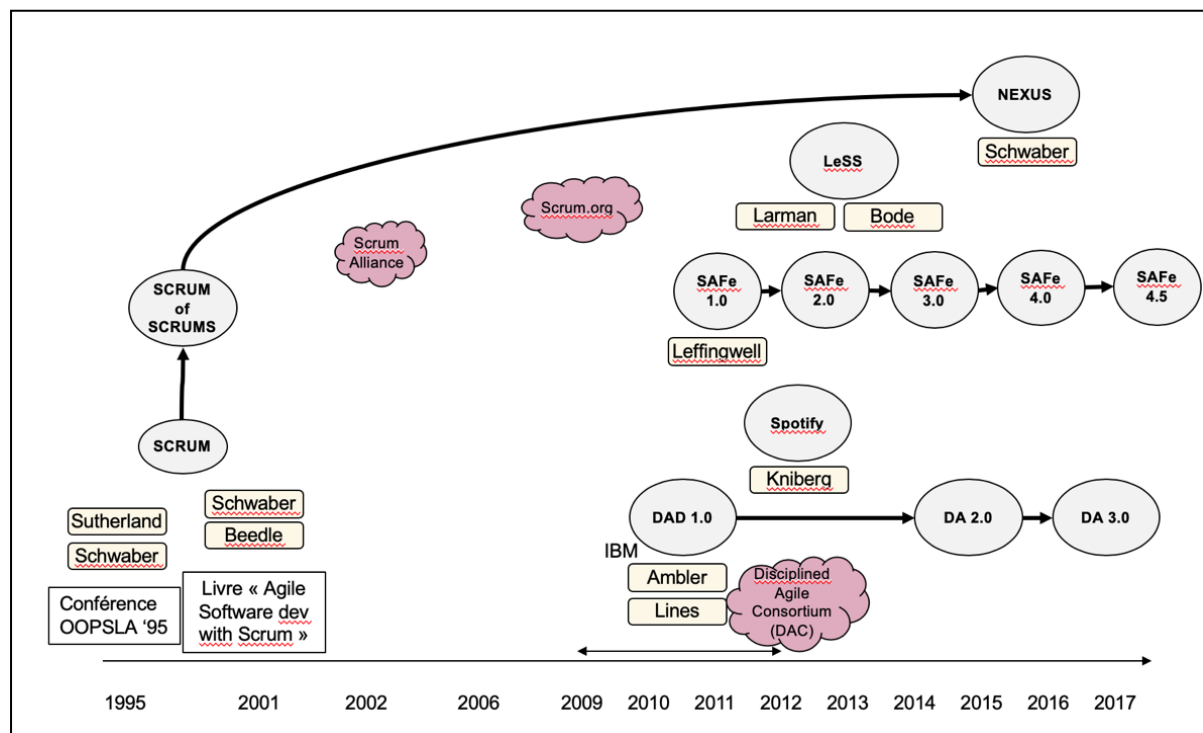


Figure 3. Historique des cadres d'agile@scale

Dans la suite de ce document, nous nous concentrerons sur les cadres suivants :

- Scrum of Scrums, Nexus
- Large Scale Scrum (LeSS)
- Spotify
- Scaled Agile Framework (SAFe® 4.6)
- Disciplined Agile (DA 3.0)
- Les cadres d'agile@scale émergents (OpenSpace Agility, FAST, RAGE...)

Nous présenterons les principes fondamentaux de ces cadres et analyserons plus spécifiquement leur relation aux activités d'architecture.

### 3.3 Présentation des principaux cadres d'agile@scale

#### 3.3.1 Scrum of Scrums / Nexus

- *Historique*

Bien que développée dès 1995, l'approche SCRUM ne fut pas décrite formellement avant 2001 dans le livre de M. Beedle et K. Schwaber « Agile Software Development with SCRUM » et surtout à partir de 2011 dans « The SCRUM guide » de K. Schwaber et J. Sutherland, véritable bible de SCRUM. Ce document d'une vingtaine de pages, qui se veut léger comme SCRUM, n'explique pas comment passer SCRUM à l'échelle. Pourtant, en 2001 J. Sutherland documente dans l'article « Agile can scale : Inventing and Reinventing SCRUM in five companies » son expérience d'agile@scale dès 1996, avec une technique simple de Scrum of Scrums, c'est à dire d'équipes Scrum de coordination composées de représentants des équipes Scrum de base.

En 2015, K. Schwaber, pilier de Scrum.org, publie « the Nexus Guide » sous la pression de l'émergence d'autres cadres d'agile@scale. Il souhaite augmenter Scrum de manière minimale, en respectant ses principes de base tout en le passant à l'échelle. Le cadre Nexus (« Nexus Framework » comme ses auteurs le nomment eux-mêmes), est dans la continuité de Scrum of Scrums, qu'il précise et complète.

- *Philosophie*

L'objectif de cette approche est de rester le plus simple et le plus proche possible de l'approche SCRUM . On cherche à conserver une démarche Scrum pure, à limiter la complexité au passage à l'échelle, en faisant confiance au maximum aux équipes agiles. Passer à l'échelle nécessite de mettre en place un niveau de coordination entre les équipes. On cherche ici à limiter la surcharge engendrée par ce niveau de coordination et à tout faire pour ne pas freiner ou gêner les équipes agiles dans leur progression. Les tenants de cette approche critiquent fortement les cadres qui prescrivent trop de rôles ou de pratiques, jugés trop structurants et donc à l'encontre de la philosophie agile. Par ailleurs, en restant très proche de Scrum, la méthode cherche à limiter la courbe d'apprentissage nécessaire pour passer à l'échelle dans les équipes qui maîtrisent déjà Scrum.

- *Le contenu en deux mots*

**Scrum of scrums** (aussi appelé « Meta Scrum ») consiste simplement à ajouter une équipe Scrum de coordination des équipes Scrum de base. Un représentant de chaque équipe participera aux cérémonies de l'équipe Scrum de scrums. Le Stand Up Meeting est moins fréquent : plutôt hebdomadaire que quotidien, et se concentre sur les questions de coordination entre les équipes. S'il y a beaucoup d'équipes à coordonner on peut aussi continuer à passer en l'échelle en créant un niveau supplémentaire chargé de la coordination (scrum de scrums de scrums). La seule connaissance de Scrum suffit pour appliquer cette approche.

**Nexus**, formalise davantage l'usage de Scrum à l'échelle. **Nexus a pour objectif de coordonner entre 3 et 9 équipes Scrum travaillant sur un seul produit** (donc un seul Backlog Produit). Le backlog de produit est affiné pour permettre la distribution des items sur les différentes équipes lors du « Nexus Sprint

Planning », qui réunit des représentants des différentes équipes. Les objectifs des différentes équipes sont alignés et les dépendances sont transparentes. Le rôle de la « Nexus Integration Team » est créé pour s'assurer que le produit final est conforme aux attentes. Comme une équipe Scrum, elle est composée du Product Owner, d'un Scrum Master et de membres qui représentent les différentes équipes. Son travail est centré sur la coordination des équipes. Elle réalise des mêlées quotidiennes sur ce thème et contrôle la qualité du livrable final dans une « Nexus Sprint Review », et la bonne coordination des équipes dans une « Nexus Sprint Retrospective ».

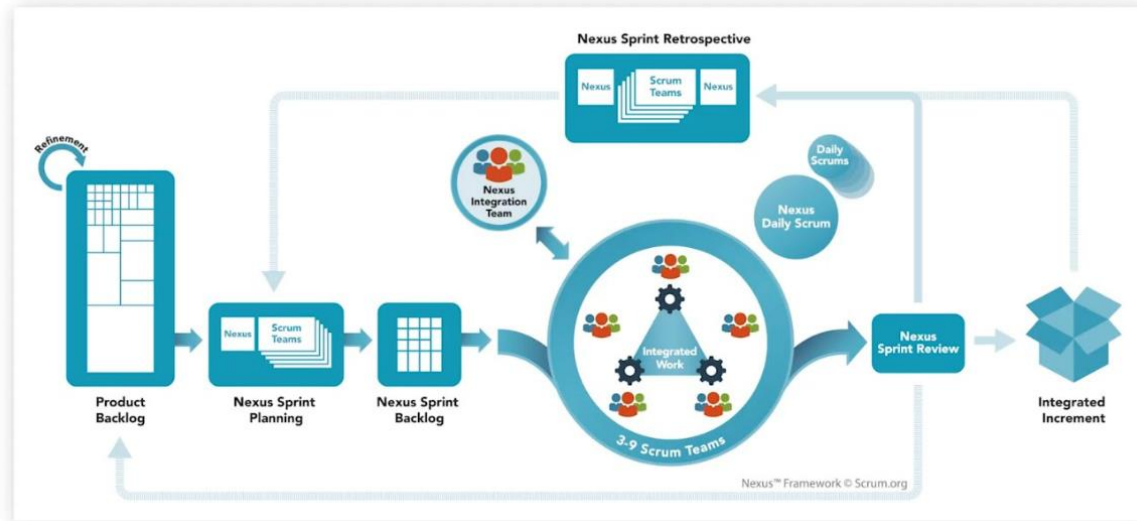


Figure 4. Nexus Framework - extrait de "The Nexus Guide", 2018

### 3.3.2 Large Scale Scrum (LeSS)

- *Historique*

Large Scale Scrum (LeSS) est un cadre d'agile@scale développé par Craig Larman et Bas Vodde à partir de 2005, puis évoqué dans le livre « Scaling Lean and Agile Development » en 2009. Le cadre est décrit de manière plus précise en 2015 dans « Large-Scale Scrum : More with LeSS » des mêmes auteurs. En tant qu'experts de l'agile, les deux auteurs ont mis en œuvre leur approche dans de nombreux contextes depuis 2005 : Telecoms, Finance, Marketing, jeux en ligne, offshoring ...

- *Philosophie*

La philosophie de LeSS est proche de celle de Nexus : très inspiré du Lean, l'objectif est de limiter au maximum la surcharge due à la coordination de l'agile@scale et de rester proche de l'esprit Scrum, pour capitaliser sur la connaissance de Scrum dans les équipes et accélérer la courbe d'apprentissage. Le cadre se veut peu prescriptif et réutilise au maximum les principes de Scrum.

- *Le contenu en deux mots*

Comme l'indique les auteurs : « LeSS est Scrum appliqué à de nombreuses équipes travaillant ensemble sur un produit. » Le périmètre d'application de LeSS est donc semblable à celui de Nexus. Néanmoins, la version étendue de LeSS, LeSS Huge permet de traiter de plus gros projets, ce qui n'est pas prévu dans Nexus. LeSS contient en fait deux cadres : LeSS (pour 3 à 8 équipes) et LeSS Huge pour plus de 8 équipes. Ces cadres restent adaptés à des projets qui produisent un seul produit, avec un seul backlog, un seul Product Owner, et la même durée de sprint pour toutes les équipes.

LeSS repose sur quelques principes : L'approche systémique (**Systems Thinking**), la pensée Lean (**Lean Thinking**), la transparence, faire plus avec moins, une vision 360° du produit, le centrage client, l'amélioration continue vers la perfection, le contrôle empirique de processus, la théorie de la file d'attente.

Le cadre prévoit deux sessions de planification de sprint :

- « **Sprint Planning One** » : une réunion de planification avec quelques représentants de toutes les équipes, puis
- « **Sprint Planning Two** » : une session de planification pour définir le backlog de sprint de chaque équipe, équipe par équipe, ou entre plusieurs équipes si elles ont des dépendances fortes

Une coordination forte existe entre les équipes par l'intermédiaire d'une plateforme d'intégration continue : Toutes les équipes travaillent sur cette plateforme et intègrent leurs développements régulièrement en faisant attention à documenter le code et à travailler ensemble sur les points qui nécessitent de la coordination.

En fin de sprint ont lieu la revue de sprint (globale), les rétrospectives d'équipes (séparées), puis la rétrospective globale, avec quelques représentants de chaque équipe, centrée sur la coordination entre les équipes.

L'affinage du backlog produit peut également faire l'objet de réunions globales régulières.

L'organisation générale est alignée sur la structure du produit avec des groupes produit. Les « communautés de pratiques » permettent des échanges transversaux.

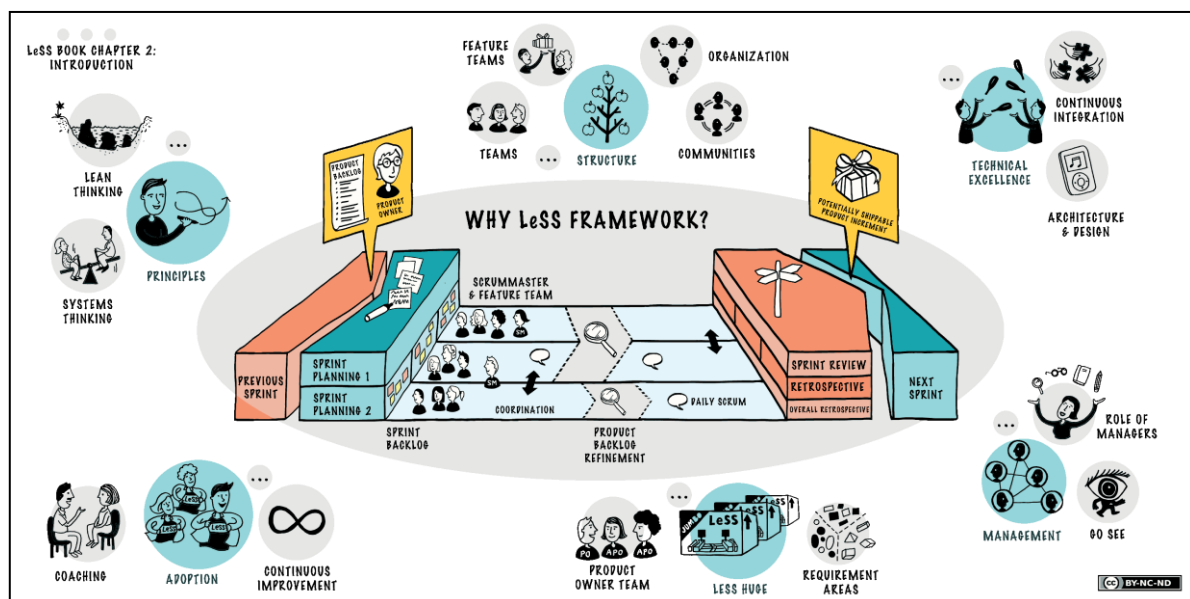


Figure 5. Vue générale de LeSS

Lorsque le projet dépasse une certaine taille (8 équipes), il est recommandé d'utiliser le cadre LeSS Huge. Le projet est subdivisé en domaines fonctionnels (« Requirement Areas ») de 3 à 8 équipes, centrés sur les besoins du client. Le principe de travailler sur un seul produit, intégré en continu, un seul backlog (dans lequel les items sont répartis sur les domaines fonctionnels) et un seul rythme d'itération, reste valable.

De nouveaux rôles apparaissent :

- Area Product Owner : le PO du domaine fonctionnel. Les APOs travaillent en équipe sous la coordination du PO global pour affiner le backlog de produit
- Area Feature Teams : ce sont les équipes qui travaillent avec l'Area product Owner, comme si on appliquait le framework LeSS au domaine fonctionnel

Les planifications des sprints des différents domaines ont lieu en parallèle.

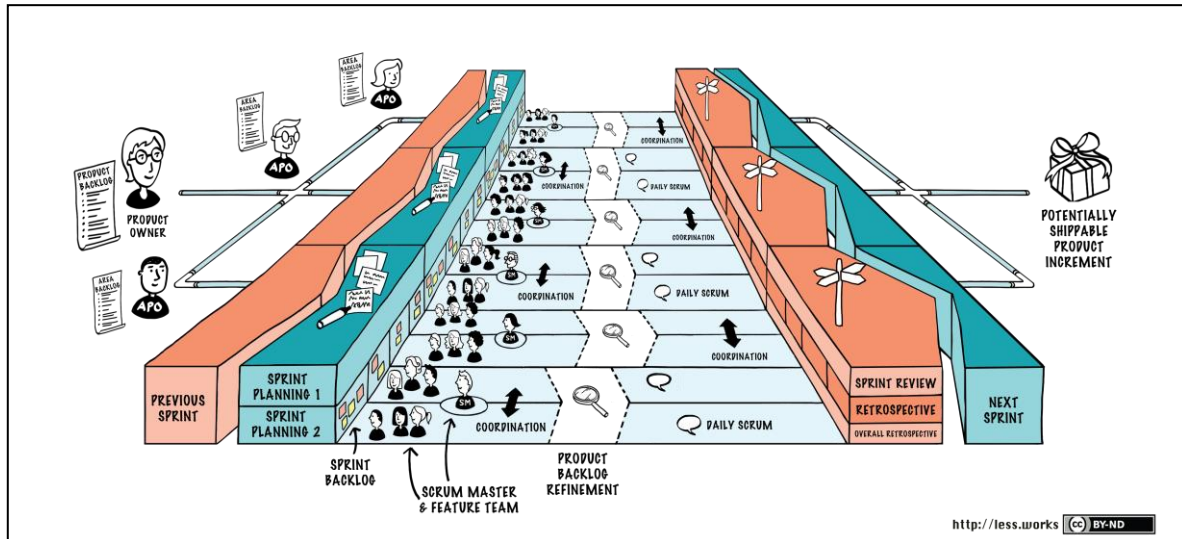


Figure 6. Vue générale de LeSS Huge

### 3.3.3 Spotify

- *Historique*

Après avoir utilisé avec succès Scrum depuis son lancement en 2008, les équipes de Spotify grandirent beaucoup et identifièrent un besoin de passer cette approche agile@scale (tout en se libérant de certaines contraintes de Scrum). C'est ainsi que naquit progressivement la culture agile de Spotify, formalisée en 2014 dans un blog et une vidéo de Henrik Kniberg, coach agile ayant accompagné cette transformation : « Spotify Engineering Culture ».

- *Philosophie*

Car il s'agit bien d'une culture d'entreprise, d'un état d'esprit, d'une posture agile, représentée par quelques principes d'organisation et de comportement, plus que d'un cadre prescriptif avec des rôles, des cérémonies, des livrables et des pratiques... L'objectif est de rester le plus agile possible et de limiter au maximum la bureaucratie et les processus de contrôle.

La culture est essentielle pour Spotify. C'est pourquoi, des rôles dédiés existent pour diffuser la culture, de l'intégration des nouveaux entrants, jusqu'à l'accompagnement des équipes les plus expérimentées.

- *Le contenu en deux mots*

La culture Spotify met en avant :

- **L'autonomie** des équipes, auto-organisées et pluridisciplinaires (en général moins de 8 personnes), avec une mission précise et responsable de son produit de A à Z (de la conception à l'exploitation). On n'emploie pas ici le terme de « **feature team** » mais de « **autonomous squad** ». Une squad est totalement autonome (ce qui produit de la motivation et de la vitesse), à partir du moment où elle est alignée avec la stratégie produit et qu'elle respecte les objectifs négociés et la mission supérieure de l'entreprise (« *loosely coupled, but tightly aligned squads* »).



- Le **partage** de bonnes pratiques, plutôt que la standardisation
- Une **architecture** en microservices, où chaque équipe est responsable de ses propres systèmes, mais ouvre son code aux autres équipes, dans un modèle open source interne
- Le **respect** mutuel entre les collègues et la volonté d'apporter son aide quand nécessaire
- Une organisation en **communautés** fluides plutôt qu'un organigramme rigide
- Des **releases petites et fréquentes** (donc automatisées autant que possible) ; les équipes peuvent les gérer de manière découplée grâce à l'architecture de l'application. Par ailleurs, elles sont autonomes pour décider quand une fonctionnalité doit aller en production. Elles se synchronisent grâce aux **trains de release** (« release trains »)
- La **confiance** est plus importante que le contrôle
- Une **erreur** est l'occasion **d'apprendre** : Tout le monde commet des erreurs à un moment ou à un autre ; il faut échouer vite pour apprendre vite. D'où l'intérêt des rétrospectives et des post-mortems <sup>4</sup>. L'impact des erreurs est limité par des déploiements graduels, petits et fréquents.
- Product Management : le développement de fonctionnalités est centré sur un impact métier clairement identifié et mesuré, un bénéfice pour le client final. Il passe par la création d'un « Minimum Viable Product » (**MVP**<sup>5</sup>, approche **Lean Startup**), première version basique du produit, puis une boucle de feed-back pour améliorer le produit jusqu'à ce que l'impact désiré soit atteint.
- **L'innovation** est prioritaire sur la prédictibilité. Tous les collaborateurs consacrent 10% de leur temps sur des activités d'innovation (« Hack Time »).
- Prendre des décisions factuelles basées sur des expérimentations concrètes plutôt que sur des a priori (« **data-driven decisions** »)
- **Eviter de faire des gros projets**, car ils représentent un gros risque ; chercher à les subdiviser en petits projets. Si ce n'est pas possible, adopter le Visual Management et des points de synchronisation fréquents pour garder l'alignement des équipes.
- L'amélioration continue et la **réduction des gâchis** sont essentiels : de nombreuses techniques existent (comme le Toyota Kata <sup>6</sup>) pour apporter des solutions à ce qui ne marche pas. C'est un souci permanent des équipes.

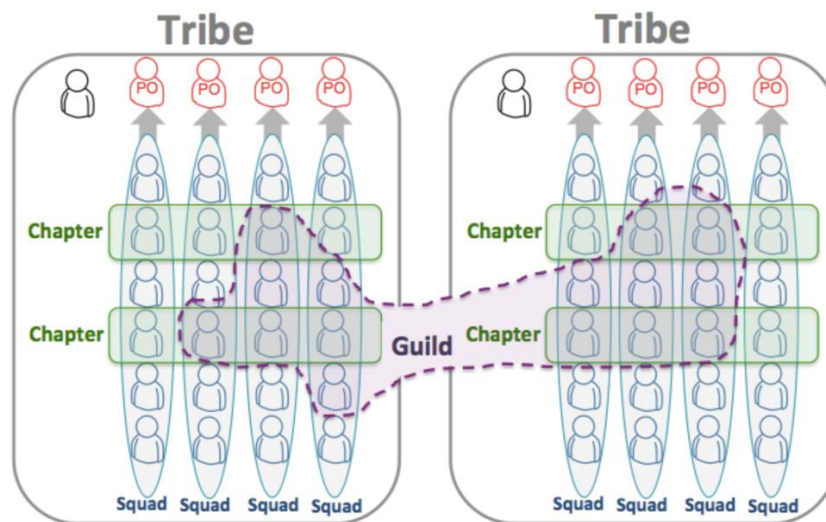


Figure 7. Organisation de Spotify

<sup>4</sup> Post Mortem : Analyse réalisée après un événement : fin d'un sprint, mise en production, incident, ...

<sup>5</sup> MVP Minimum Valuable Product

<sup>6</sup> « Toyota Kata », Mike Rother, McGraw-Hill, 2009

Avoir plus de 50 squads sur 4 lieux impliquent un minimum d'organisation. L'entreprise regroupe les squads en **tribus** (« tribes »). Une tribu correspond souvent à un produit ou un domaine fonctionnel. Les squads d'une tribu ont besoin d'échanger fréquemment et sont situés dans un même lieu.

Au sein d'une tribu, on trouve une organisation matricielle :

- Les squads sont focalisées sur la production et la qualité, autour du PO
- Les **chapitres** (chapters) sont des zones de compétence ou métier (développement, test, ...). Le Chapter Lead est responsable de développer les compétences de son équipe pour leur permettre de progresser et d'évoluer d'une squad à une autre sans changer de manager.
- Une communication informelle est établie entre les équipes par l'intermédiaire des **guildes** (« Guilds »), qui sont des communautés d'intérêt transverses aux tribus. Les guildes permettent le développement des compétences et échangent par voie électronique et dans des conférences biannuelles. Chacun peut entrer et sortir d'une guilda quand il le souhaite, quelle que soit sa fonction. Pour Spotify, le fonctionnement en communauté est plus important qu'une structure organisationnelle forte.

### 3.3.4 Scaled Agile Framework (SAFe® 4.6)

- *Historique*

Scaled Agile Framework (SAFe®) est un cadre d'agile@scale développé par Dean Leffingwell, un consultant expert en agilité depuis les années 2000. Après avoir été Senior Vice President chez Rational Software, auteur de plusieurs livres sur le lean et l'agile, il a travaillé dans plusieurs startups à développer l'agile@scale. Le cadre est public mais reste la propriété de Scaled Agile Inc, qui développe également des contenus de formation et de certification (Certified SAFe Agilist, Certified SAFe Practitioner, ...). Il est alimenté principalement par une dizaine de contributeurs (et une communauté) depuis sa création en 2011 et arrive à maturité en 2018 avec sa version 4 (4.6 en octobre 2018).

- *Philosophie*

SAFe® est un cadre très complet (et donc prescriptif) mariant différentes disciplines pour permettre de créer des produits complexes de qualité pour le bénéfice de l'entreprise et des équipes. Il est fondé sur la synthèse des approches Lean, Agile et DevOps, de nombreux retours d'expérience de terrain, la création d'une communauté de pratiquant pour stimuler son évolution, et un ensemble de partenaires (pour la formation et la certification). SAFe® est une marque déposée de Scale Agile Inc, qui assure le développement du cadre et tire ses revenus des formations, certifications et missions d'accompagnement. Le contenu du cadre SAFe® est librement accessible en ligne :

<http://www.scaledagileframework.com>.



- *Le contenu en deux mots*

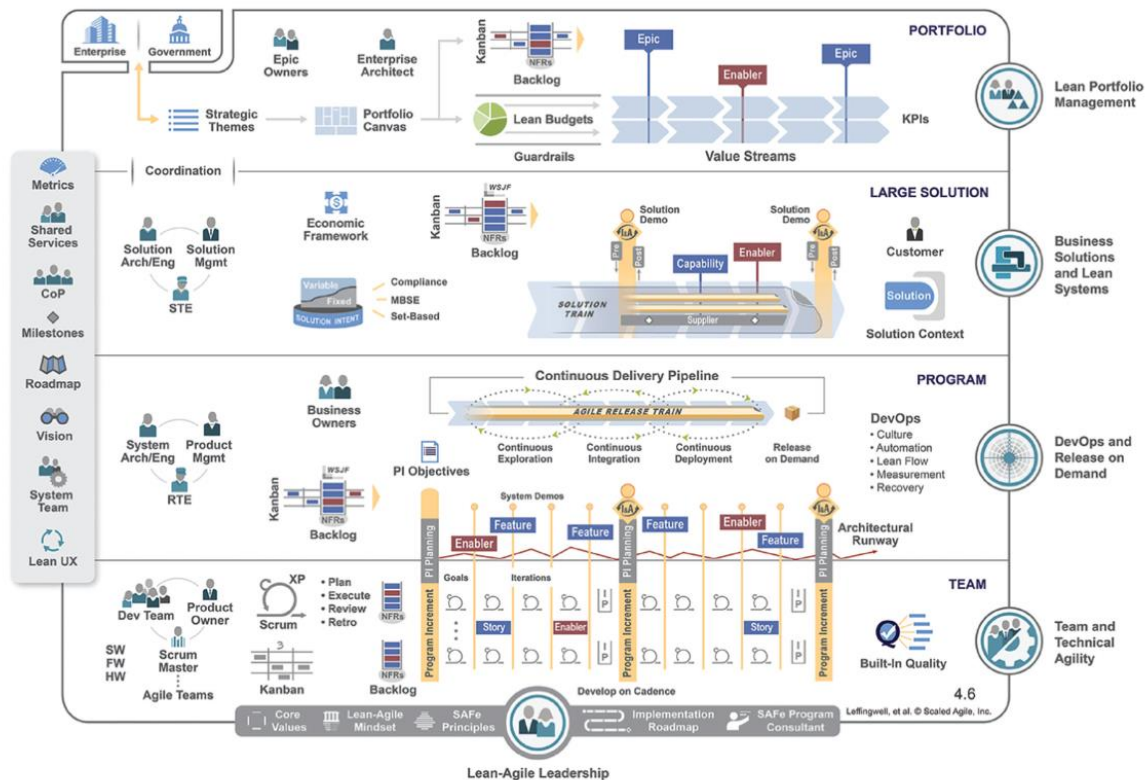


Figure 8 Vue générale de SAFe® 4.6

SAFe® définit 4 niveaux :

- Le niveau « Equipe » (**Team**) où on retrouve les pratiques Lean, Agile, DevOps du marché pour la livraison des produits d'une seule équipe
- Le niveau « programme » (**Program**) où on intègre le travail de 5 à 12 équipes sur un même produit dans un train de release (Agile Release Train)
- Le niveau « Grande Solution » (**Large Solution**) où on intègre plusieurs trains de release pour bâtir une grande solution
- Le niveau « Portefeuille » (**Portfolio**) où on adresse la planification stratégique et la définition des investissements

Suivant les besoins de l'entreprise, SAFe® peut être utilisé en 4 configurations :

- **Essential** Configuration : Team + Program
- **Portfolio** Configuration : Team + Program + Portfolio
- **Large Solution** Configuration : Team + Program + Large Solution
- **Full** Configuration : Team + Program + Large Solution + Portfolio

SAFe® introduit 5 compétences clés :

- **Lean-Agile Leadership** : les décideurs doivent mettre en pratiques les valeurs du Lean et de l'Agile et créer un environnement favorable pour les équipes pour rendre possible la transformation Lean-Agile de l'entreprise
- **Team and Technical Agility** : Les équipes agiles maîtrisent les pratiques de base de l'agilité (Scrum et Kanban, mais aussi les techniques d'ingénierie agile XP, BDD, TDD...) et savent également travailler de manière coordonnée dans un train de release

- **DevOps and Release on Demand** : Toute l'organisation (Métier, Dev, Ops, Sécurité ...) est focalisée sur la valeur pour le client et travaille pour être en mesure de délivrer la valeur attendue par le marché au moment voulu. Tous les acteurs sont organisés dans un train de release et se coordonnent pour être en mesure de délivrer à tout moment grâce à un processus de livraison continue en appliquant les 5 concepts de DevOps **CALMR** (Culture, Automation, Lean flow, Measurement, Recovery)
- **Business Solutions and Lean Systems Engineering** : Pour les "grandes" solutions, SAFe® propose 8 pratiques d'ingénierie :
  - Les composantes de la solution sont bâties par des trains de release (ART)
  - Les composantes de la solution sont intégrées dans un train de solution (Solution Train)
  - Capturer et raffiner progressivement les caractéristiques essentielles de la solution dans un « Solution Intent Repository », pour coordonner les équipes et les garder alignées sur la vision
  - Utiliser plusieurs horizons de planification (fin à court terme et macro à plus long terme)
  - Penser une architecture « scalable », modulaire, réutilisable, maintenable en gardant un équilibre entre architecture intentionnelle et émergente
  - Appliquer l'intégration continue avec discernement, en arbitrant entre le coût et les bénéfices d'intégrer souvent
  - Intégrer les contraintes de conformité réglementaire dans les activités quotidiennes de l'équipe
- **Lean Portfolio Management** : Pour relier les produits à la stratégie d'entreprise, le cadre propose de gérer des thèmes d'investissements et des budgets « Lean ». Les thèmes sont gérés dans un backlog revu régulièrement dans une approche Kanban. Le niveau « Portfolio » doit coordonner les différentes solutions où les différents trains en gardant une vision globale, tout en maintenant une décentralisation maximale des décisions vers ces niveaux.

### 3.3.5 Disciplined Agile (DA 3.0)

#### • *Historique*

Initialement développé chez IBM Rational entre 2009 et 2012 sous le nom de Disciplined Agile Delivery (DAD), ce cadre est passé sous la responsabilité du Disciplined Agile Consortium en 2012 et sa version 1.0 a été publiée dans le livre « Disciplined Agile Delivery ». Par la suite, ces auteurs ont souhaité élargir le périmètre du cadre pour traiter tout le cycle de vie des solutions : non seulement le développement mais également l'initialisation et l'exploitation des produits. Le cadre est maintenu au Canada par Disciplined Agile Inc. sous la responsabilité des experts à l'origine du cadre S. Ambler et M. Lines. L'entreprise propose une offre d'audit et d'accompagnement ainsi qu'un référentiel de certification. A partir de la version 2 parue en Aout 2015, le cadre s'appelle Disciplined Agile (DA) 2.0, puis DA 3.0 en Août 2017.

#### • *Philosophie*

Disciplined Agile est une boîte à outil qui rassemble un ensemble de pratiques et propose de choisir l'approche adaptée à votre contexte/projet. Il fait la synthèse et met en perspective d'autres méthodologies comme Agile Modeling, Scrum, Kanban, Lean, Lean Startup. Partant du constat que les approches agiles dans les projets/programmes ne sont pas efficaces si le reste de l'entreprise n'est pas agile, son objectif est également de généraliser l'agile à toute l'entreprise. C'est un cadre global qui intègre toutes les bonnes pratiques à l'état de l'art pour gérer un SI intégré dans l'entreprise, au service des métiers.

- *Le contenu en deux mots*

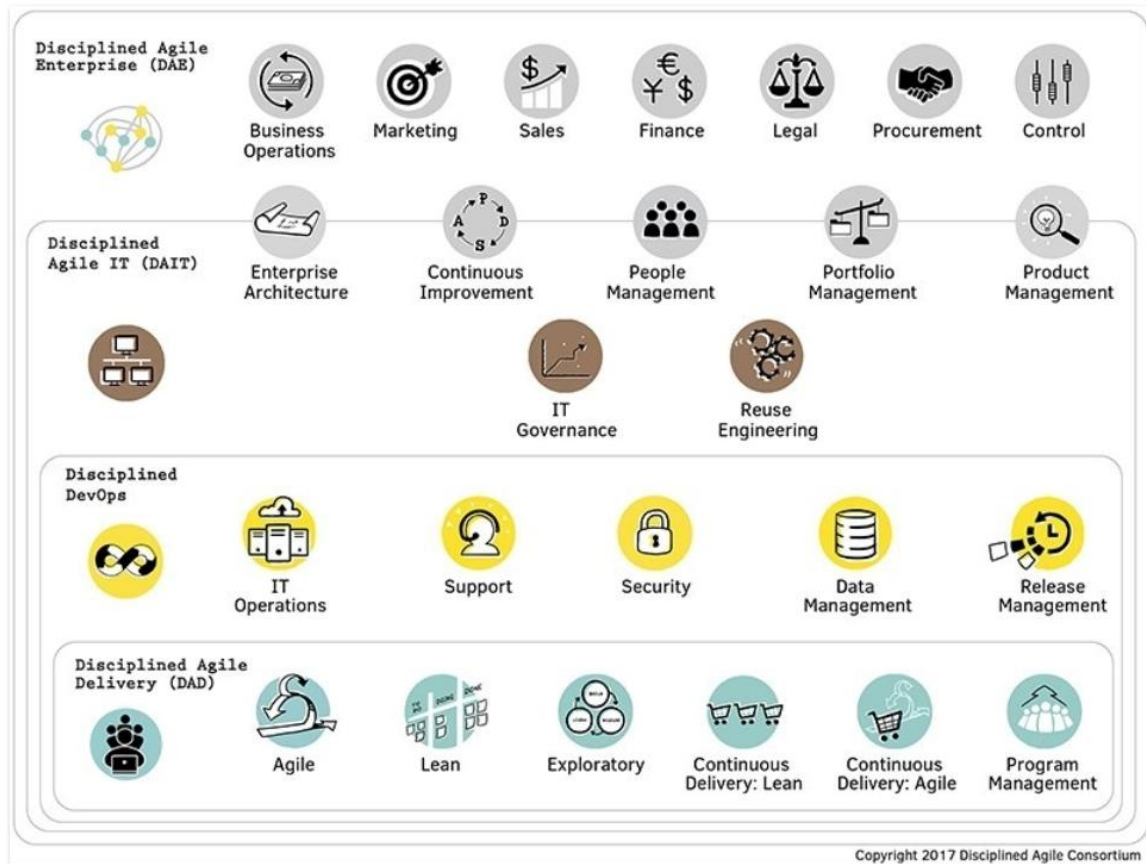


Figure 9. Vue générale de DA 3.0 - L'agile étendu à l'entreprise

DA définit quatre niveaux ou couches :

- **Disciplined Agile Delivery (DAD)** : Le cœur est le processus de « delivery » des produits, pour lequel on peut opter pour différentes options : Scrum, Lean, Lean Startup ...
- **Disciplined DevOps** : Le produit doit également être pensé par rapport à des contraintes d'exploitabilité, de support, de sécurité, de gestion des releases. DAD préconise une approche intégrée Développement / Opérations / Sécurité (DevSecOps).
- **Disciplined Agile IT** : inclut les activités de planification stratégique (gestion de portefeuille, gestion de programme, architecture d'entreprise) et la gestion des compétences et du développement de l'équipe par l'amélioration continue
- **Disciplined Agile Enterprise** : définit ce qu'est une entreprise agile et comment elle peut gagner dans un environnement soumis à la disruption digitale

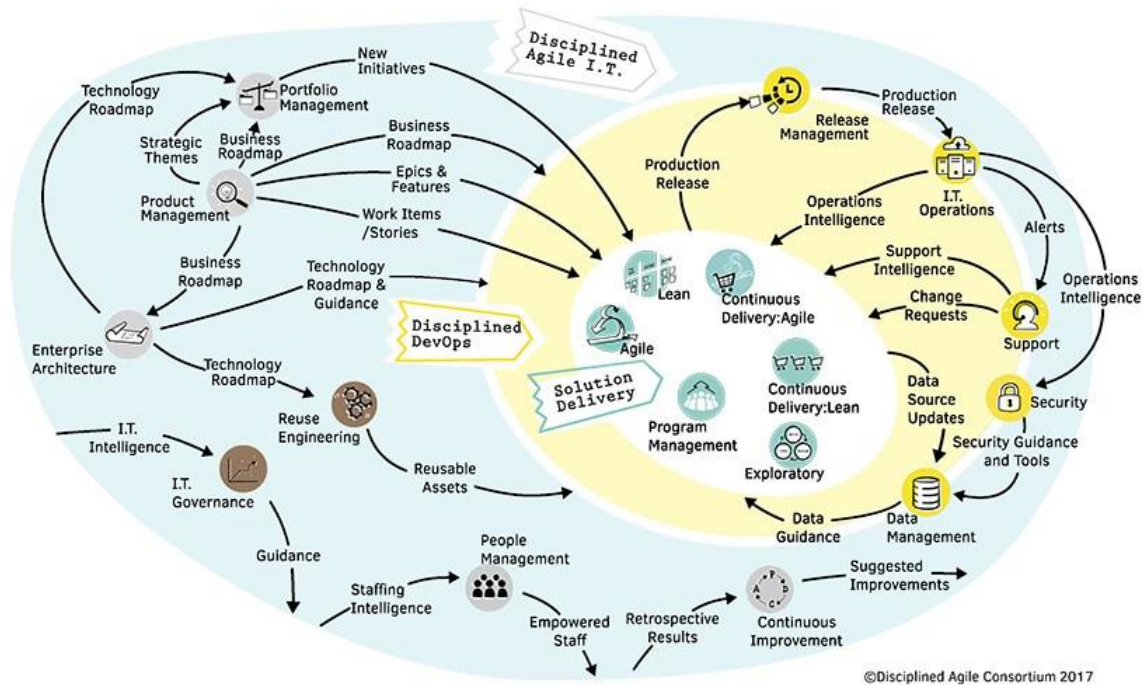
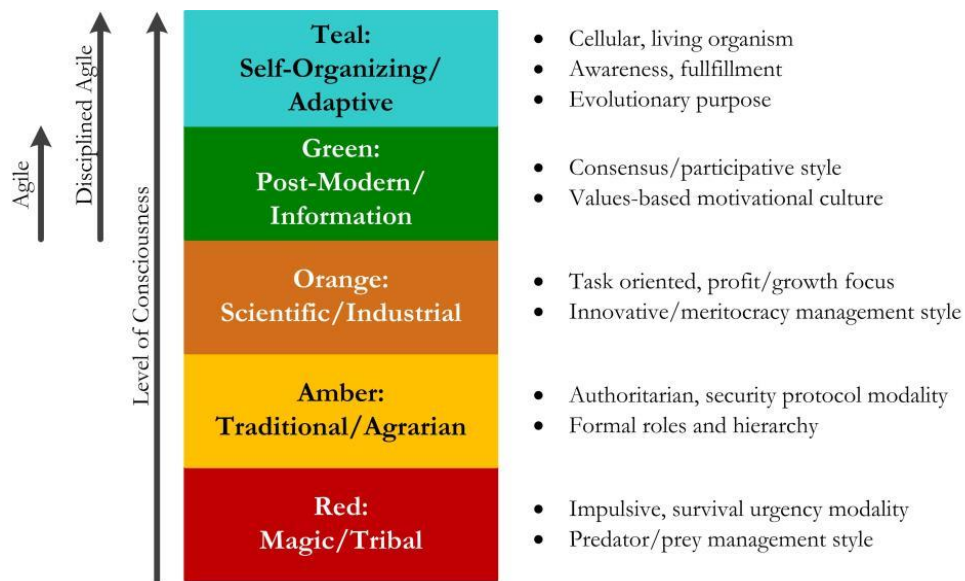


Figure 10. Interaction entre les différents niveaux de DA 3.0

Adapted from *Reinventing Organizations* by Frederic Laloux

2017 © Disciplined Agile Consortium

Figure 11. Différents types de culture et d'organisation d'entreprise

Disciplined Agile définit un ensemble de principes (<https://www.disciplinedagiledelivery.com/principles/>) et de rôles primaires et secondaires :

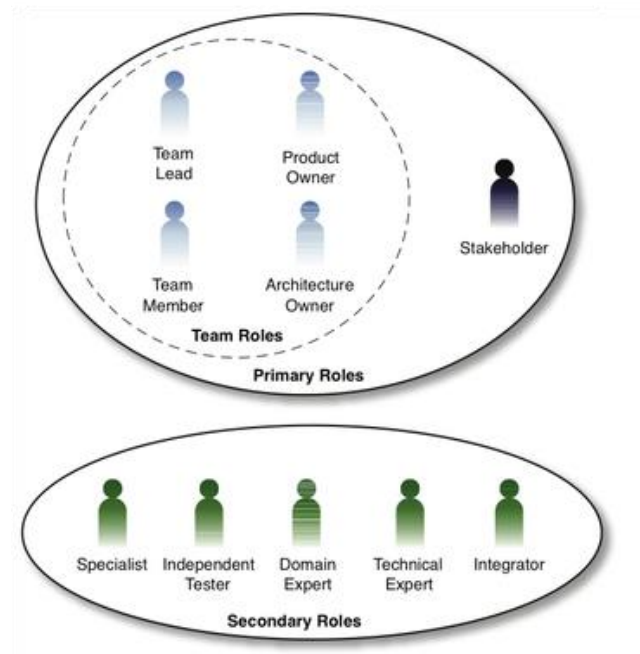
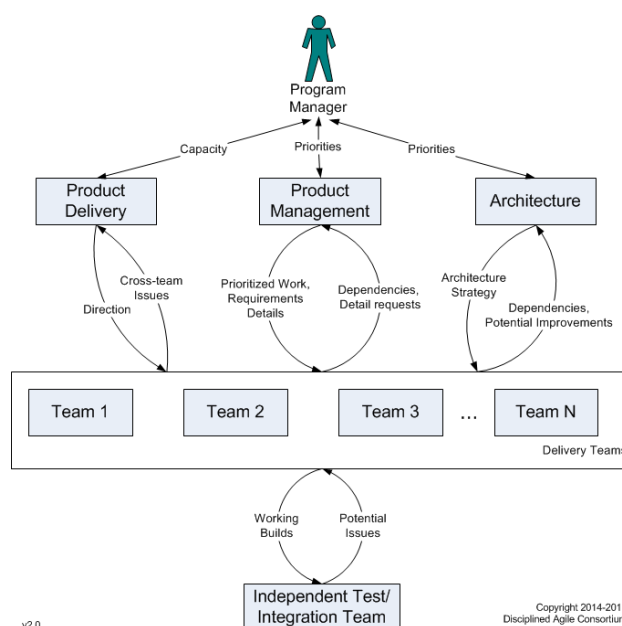


Figure 12. Les rôles dans Disciplined Agile

On note qu'un des membres de l'équipe agile doit porter la responsabilité de l'architecture : cela peut être le Team lead, un des membres de l'équipe ou un architecte dédié. Il est généralement l'animateur de la réflexion autour de l'architecture dans l'équipe, en contact direct avec l'équipe d'architecture d'entreprise, servant à la fois à promouvoir les standards auprès de l'équipe et à remonter la vision de l'équipe auprès de l'équipe d'architecture.

Concernant le passage à l'échelle, DA propose différentes approches suivant la taille de l'équipe (et également d'autres critères comme la répartition géographique et organisationnelle, l'impact de la conformité réglementaire, le niveau de complexité métier et technique). Pour les grandes équipes (plus de 35 p.), DA conseille la mise en place de rôle de coordination : Directeur de programme, équipe produit, équipe architecture...



v2.0

Copyright 2014-2015  
Disciplined Agile Consortium

Figure 13. Interaction entre les rôles de Disciplined Agile



### 3.3.6 Autres cadres

Notre groupe de travail a également identifié d'autres cadres méthodologiques relatif à l'agile@scale. Sans vouloir les développer, nous les mentionnons pour référence :

- **OpenSpace Agility** (<http://openspaceagility.com/>) est une approche pour déployer l'agile sur un grand nombre de personnes. Elle se base sur l'Open Space Technology, une technique d'animation pour faire collaborer un grand nombre personnes autour d'un objectif commun. Une réunion « Open Space », avec tous les participants des différentes équipes, est organisée à fréquence régulière pour définir les prochaines étapes et faire le bilan de l'étape précédente.

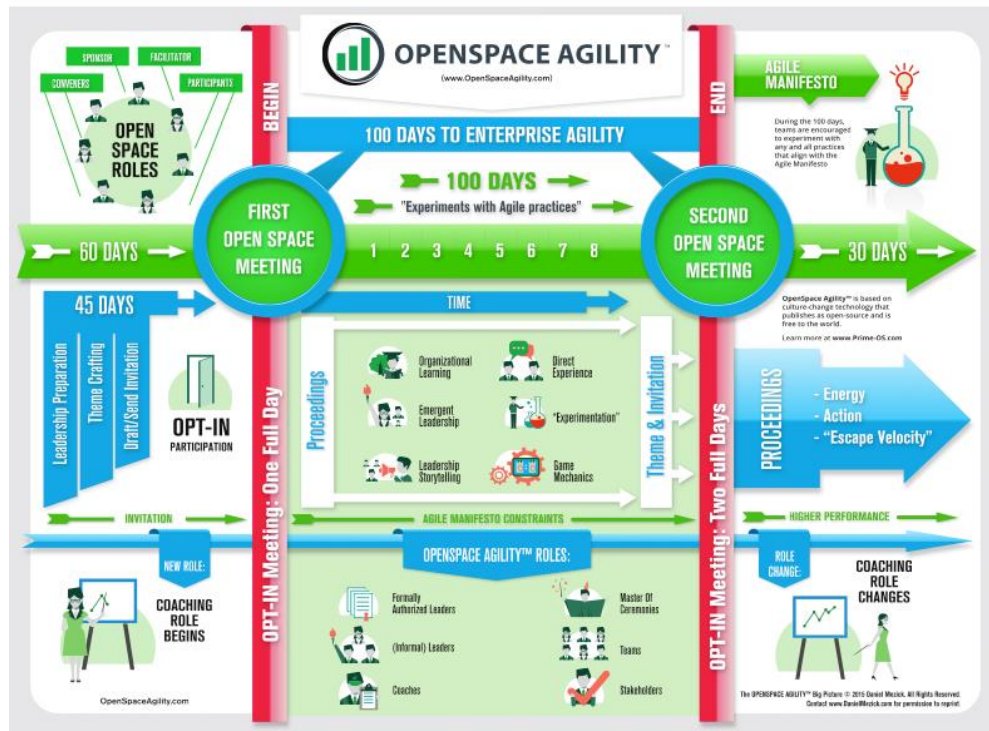


Figure 14. Vue générale d'OpenSpace Agility

- **Fast Agile Scaling Technology (FAST™)** (voir <http://www.fast-agile.com/>) est un cadre d'agile@scale dynamique : les équipes peuvent se recomposer à chaque itération. L'ensemble des participants (au maximum 150) compose une tribu, qui se réunit régulièrement en utilisant la technique de l'Open Space Technology. En fonction des priorités définies par la tribu, les participants se répartissent en équipes (swarms) pour l'itération. FAST s'appuie sur des pratiques éprouvées : XP, Lean Software Development, modern Agile (<http://modernagile.org/>) et quelques livrables : Release Map, Feature Trees, Decision Log, Tribe Agreements (les règles de vie de la tribu)
- **cPrime RAGE (Recipes for Agile Governance in the Enterprise)** (<https://www.cprime.com/rage/>) introduit la notion de gouvernance agile, c'est-à-dire la capacité à prendre des décisions de manière agile à tout niveau (projet, programme, portefeuille...). Ce référentiel identifie les éléments à mettre en place pour définir votre gouvernance agile : des rôles, des cérémonies, des livrables, des métriques et des points (moments) de décision. Il permet donc de définir son propre cadre, en s'inspirant des cadres du marché mais en prenant aussi en compte ses propres spécificités.

- **ScALeD** (<http://scaledprinciples.org/>) est une synthèse d'un ensemble de principes bien connus en agile et essentiels pour passer à l'échelle :
  - être centré sur la valeur pour le client et produire de petits incréments,
  - composer de petites équipes pluridisciplinaires et responsabilisées,
  - promouvoir l'optimisation globale par une architecture modulaire mais aussi une organisation qui favorise l'échange d'informations, la communication directe, des objectifs communs et partagés, avec des synchronisations fréquentes
  - Avoir des managers qui fixent des objectifs clairs et facilitent le travail des équipes (servant leaders), décentralisent la prise de décision et promeuvent le changement
  - Adopter l'amélioration continue : Inspecter et adapter le produit, le processus et l'organisation souvent, au niveau des équipes elles-mêmes
- **XSCALE** (<https://xscalealliance.org/> et <https://xscale.wiki/>) est un cadre pour construire des organisations agiles. Il est fondé sur quelques principes (eXponential, Simple, Continuous, Autonomous Learning Ecosystems), des structures (squad, chapitre et guild, qui n'est pas sans rappeler Spotify), des rôles (en particuliers pour les managers et les coaches), des pratiques et des métriques.

### 3.3.7 Positionnement relatif des différents cadres d'agile@scale

Plusieurs enquêtes permettent d'établir le niveau d'adoption des principaux cadres d'agile@scale :

- Le "Gartner Market Guide for Enterprise Agile Frameworks 2016" montre un intérêt fort du marché pour SAFe®

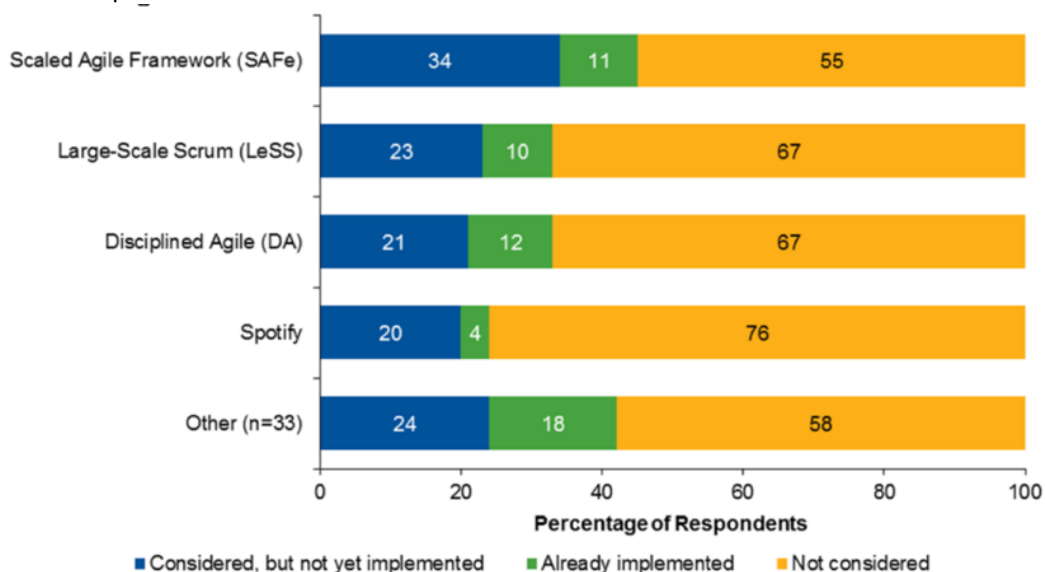


Figure 15. Taux des différents cadres en 2016 selon Gartner

- VersionOne « 12th annual report on the State of Agile » 2018 confirme l'émergence de SAFe® :

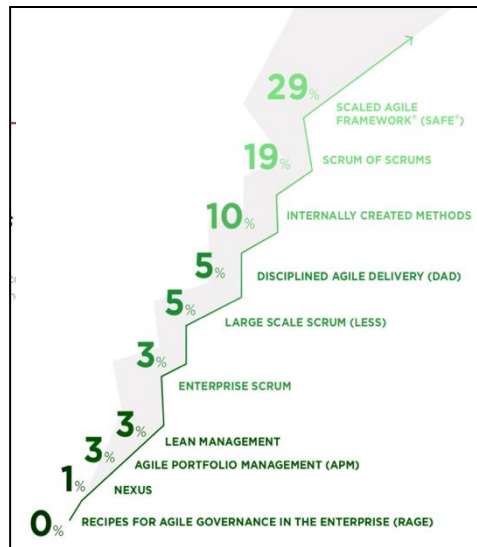


Figure 16. Taux de citation des cadres d'agile@scale par les répondants à l'enquête VersionOne 2018

- cPrime « Scaling Agile Report 2017 » (5000 p. interrogées) donnent une tendance similaire :

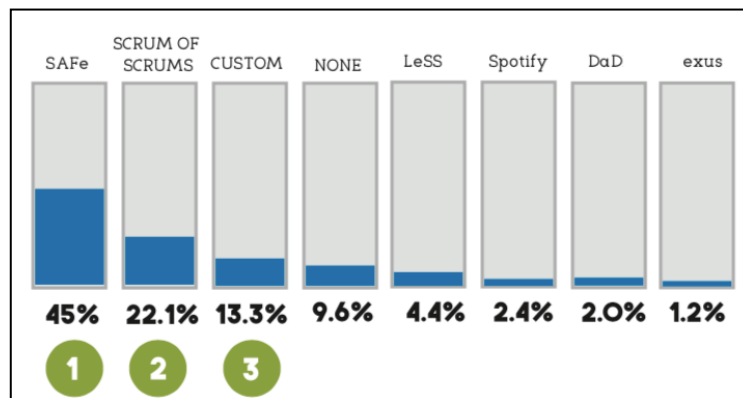


Figure 17. Choix préférentiel des répondants de l'enquête cPrime 2017

Par ailleurs, une analyse comparative des différents cadres d'agile@scale établie par des experts de l'agilité est disponible ici : <http://www.agilescaling.org/>

Les auteurs ont évalué différents cadres sur de nombreux critères comme la popularité, le niveau de flexibilité, la complétude, la disponibilité d'expertise ou de formations, ...ce qui permet de mettre en perspective les différents cadres du marché. A titre d'exemple, voici un extrait de la matrice ASK :



Aspects / Criteria	Scrum-of-Scrums (SoS) PO meta-scrum	Large Scale Scrum (LeSS) Larman/Vodde	Scaled Agile Framework (SAFe) Leffingwell	Disciplined Agile Delivery (DAD) + Agility at Scale Ambler/Lines	Spotify "model" (Tribes, Squads, Chapters & Guilds) Kniberg	Nexus / Scaled Professional Scrum Scrum.org
Description	An important mechanism that may be enough for smaller organizations but is not a full scaling approach	Larman / Vodde model as documented in "Scaling Lean & Agile Development"	The method documented by Dean Leffingwell and Scaled Agile, Inc.	Scott Ambler model documented in the book "Disciplined Agile Delivery"	The method used at Spotify, featuring Squads, Tribes, Chapters & Guilds.	Scrum-based scaling with an "exoskeleton" called Nexus plus over 40 practices
Web Link	<a href="http://guide.agilealliance.org/guide/scrum-of-scrums.html">guide.agilealliance.org/guide/scrum-of-scrums.html</a>	<a href="http://less.works/">less.works/</a>	<a href="http://scaledagileframework.com/">scaledagileframework.com/</a>	<a href="http://disciplinedagiledelivery.com/">disciplinedagiledelivery.com/</a>	<a href="http://scaling-agile-spotify-11.pdf">scaling-agile-spotify-11.pdf</a>	<a href="http://www.scrum.org/Resources/The-Nexus-Guide">www.scrum.org/Resources/The-Nexus-Guide</a>
Completeness of coverage of "levels", including:	Low	Medium	High	High	Medium	Medium
Portfolio	Low	Medium*	Medium	Medium	Low	Medium
Program Structure	Low	Medium*	High	High	Low	Medium
Inter-team Coordination	Medium	High	High	High	High	Medium
Team Level	Medium	Medium	High	High	High	High
Tech Practices	Low	Medium	Medium	High	Medium	Medium
Popularity / Adoption (new/growing (low) vs. established/leader (high))	High	Medium	High	Low	Low	Low
Flexibility / Emergence: Prescriptive (low) vs. emergent (high)?	High	High	Low	Medium	High	Medium
Typical Cost to Implement	Low	Low	High	Medium	Low	
Availability of Details & Support	Low	Medium	High	Medium	Low	Low

Cette étude corrobore le résultat des enquêtes mentionnées ci-dessus : SAFe® est le cadre le plus déployé et le plus structurant. LeSS et Nexus sont des cadres beaucoup plus légers et proches de Scrum, et finalement assez peu déployés à grande échelle. DAD reste confidentiel, bien qu'un des cadres les plus complets.

## 4. Les approches d'Architecture agiles

### 4.1 La place de l'architecture dans les cadres d'agilité à l'échelle

Tout projet nécessite un travail d'architecture (au sens design/conception). Ce qui change c'est :

- **Combien** : faut-il documenter une architecture très fouillée, par exemple en deux itérations avec d'abord une conception générale (high level design) puis une conception détaillée (low level design) ?
- **Qui** le fait : le travail de conception est-il l'apanage d'une équipe d'architecture ? ou doit-il être conduit par l'équipe agile elle-même ? ou un mix des deux ? et avec quel périmètre de responsabilité ?
- **Quand** : l'agile introduit le débat entre l'architecture **intentionnelle** (définie entièrement avant le développement, comme dans les approches classiques) et l'architecture **émergente** (définie en cours de développement et qui émerge donc du travail de l'équipe). Les puristes de l'agile estiment que passer trop de temps sur l'architecture a priori est contre-productif, car ce design sera remis en cause au moment du développement (danger du « Big Design Up Front »). Les tenants de l'architecture intentionnelle mettent en avant qu'il paraît difficile de construire un système qui saura faire face à des changements importants au cours de nombreuses itérations sans avoir pensé un minimum sa structure générale avant de commencer son développement...

Les cadres agiles puristes, peu prescriptifs, restent plutôt sur un rôle proéminent de l'équipe agile et une architecture émergente (SoS, Nexus, LeSS).

Les cadres plus complets (DA, SAFe) reconnaissent davantage les rôles transverses comme l'architecture.

#### 4.1.1 Relation de Nexus à l'architecture

Aucun rôle d'architecture n'est spécifiquement défini dans le cadre Nexus. Des standards architecturaux peuvent exister ; il est alors de la responsabilité de l'équipe d'intégration de les promouvoir auprès des équipes agiles, comme l'indique le « Nexus Guide » :

*"The Nexus Integration Team coaches the individual Scrum Teams on the necessary development, infrastructural, or architectural standards required by the organization to ensure the development of quality Integrated Increments."*

Comme pour Scrum, Nexus laisse l'autonomie et la responsabilité du design et de la réalisation du produit aux équipes ; l'architecture est donc plutôt **émergente** (constatée et améliorée a posteriori) qu'intentionnelle (développée en amont de la réalisation).

#### 4.1.2 Relation de LeSS à l'architecture

Pour les auteurs :

- L'architecture du système se trouve dans le code ; elle évolue au fur et à mesure du développement.
- L'acte de conception est donc exercé par les programmeurs, qui doivent le maîtriser.
- Le rôle d'architecte dans les entreprises est trop souvent décorrélé du code ; on aboutit à des architectes qui n'ont plus le contact du terrain et agissent dans leur tour d'ivoire.
- Il faut préférer des « *Master programmer Architects* » plutôt que des « *Powerpoints Architects* »

Et dans une keynote de 2011, Craig Larman explique : *"Architecture is a bad metaphor. We don't construct our software like a building, we grow it like a garden."*

Les auteurs sont donc clairement dans une préférence pour l'architecture **émergente** plutôt que l'architecture intentionnelle et préconisent de limiter au maximum l'existence d'un groupe d'architectes déconnectés des équipes de développement.

### 4.1.3 Relation de Spotify à l'architecture

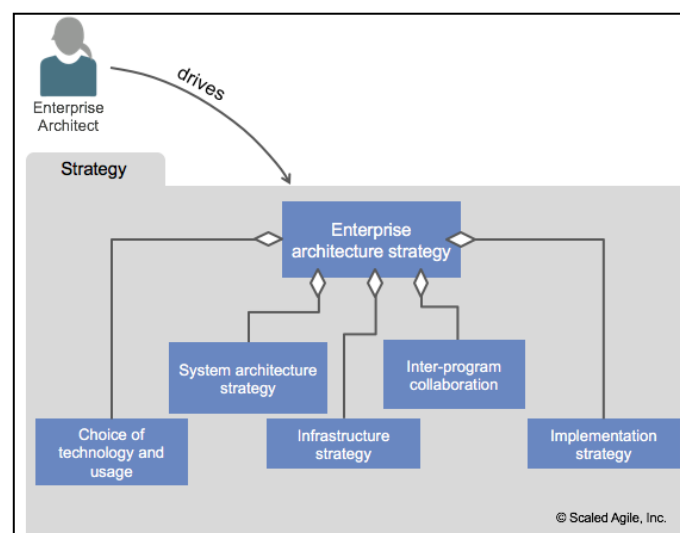
Dans la culture Spotify, un maximum d'autonomie est donné à l'équipe agile, ce qui laisse peu de place aux fonctions transverses comme l'architecture. L'architecture est principalement du domaine de responsabilité de l'équipe et est donc **émergente**. Néanmoins, on peut citer cette phrase qui indique bien qu'un **minimum de transversalité est assuré** : « *We have a **chief architect** role, a person who coordinates work on high-level architectural issues that cut across multiple systems. He reviews development of new systems to make sure they avoid common mistakes, and that they are aligned with our architectural vision. The feedback is always just suggestions and input - the decision for the final design of the system still lies with the squad building it.* »

Les standards architecture adoptés par l'entreprise émergent plutôt en **bottom-up** (on généralise ce qui a marché dans une ou plusieurs équipes), qu'en top-down, imposé par une équipe d'architecture.

### 4.1.4 Relation de SAFe® à l'architecture

SAFe® étant un cadre basé sur une approche Lean-Agile, il recommande de décentraliser autant de décisions d'architecture que possible au niveau des équipes. Néanmoins, il reconnaît que certaines décisions d'architecture structurantes doivent être centralisées et définit plusieurs rôles d'architecture sur ses différents niveaux :

- « Solution Architect » au niveau Large Solution ou Program
- « Enterprise Architect » au niveau Porfolio



Les architectes interviennent sur la stratégie technique, la conception générale et les scénarios d'architecture (et leur impact économique), la décomposition des systèmes complexes en sous-systèmes et en composants, la définition de leurs interfaces, l'identification des exigences non fonctionnelles (NFRs) et l'animation de la prise de décision de design dans les équipes.

La fonction d'architecture est en support des équipes agiles et facilite les décisions d'architecture de manière collaborative. Le cadre cherche un équilibre entre architecture intentionnelle et émergente et

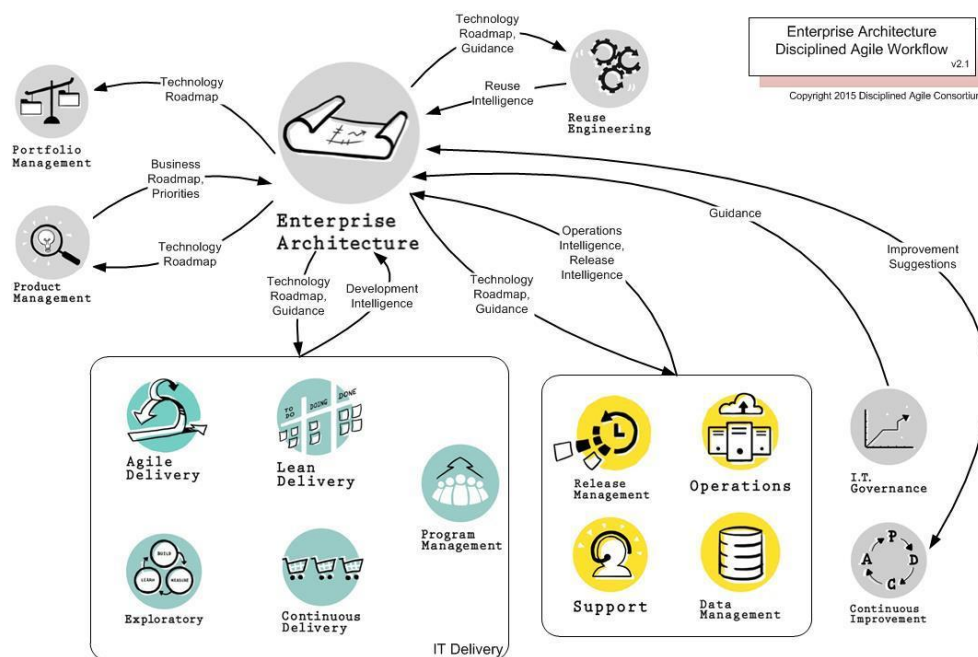
cite James Coplien : « *While we must acknowledge emergence in design and system development, a little planning can avoid much waste.* » (<https://www.scaledagileframework.com/agile-architecture/>)

#### 4.1.5 Relation de Disciplined Agile à l'architecture

DA définit explicitement un rôle d' « Architecture Owner », qui doit être assuré au sein de chaque équipe (souvent par le « Team Lead »). Inspiré du cadre « Agile Modeling », DA préconise un minimum d'architecture intentionnelle (Initial Agile Architecture Modeling) pour limiter les risques techniques, réduire le temps de développement, améliorer la communication autour du produit, et permettre le passage à l'échelle en subdivisant le systèmes en sous-systèmes avec des interfaces claires.

A l'échelle, un « Chief Architecture Owner » assure la coordination de l'équipe des « Architecture owners » des différentes équipes.

Au niveau « Disciplined Agile IT », DA intègre explicitement le processus d'Architecture d'entreprise et définit ses interactions avec les autres processus, dont les processus de delivery des équipes agiles :



Il détaille ses activités, très proches des activités classiques de l'architecture d'entreprise telle que définie dans les grands cadres comme TOGAF :

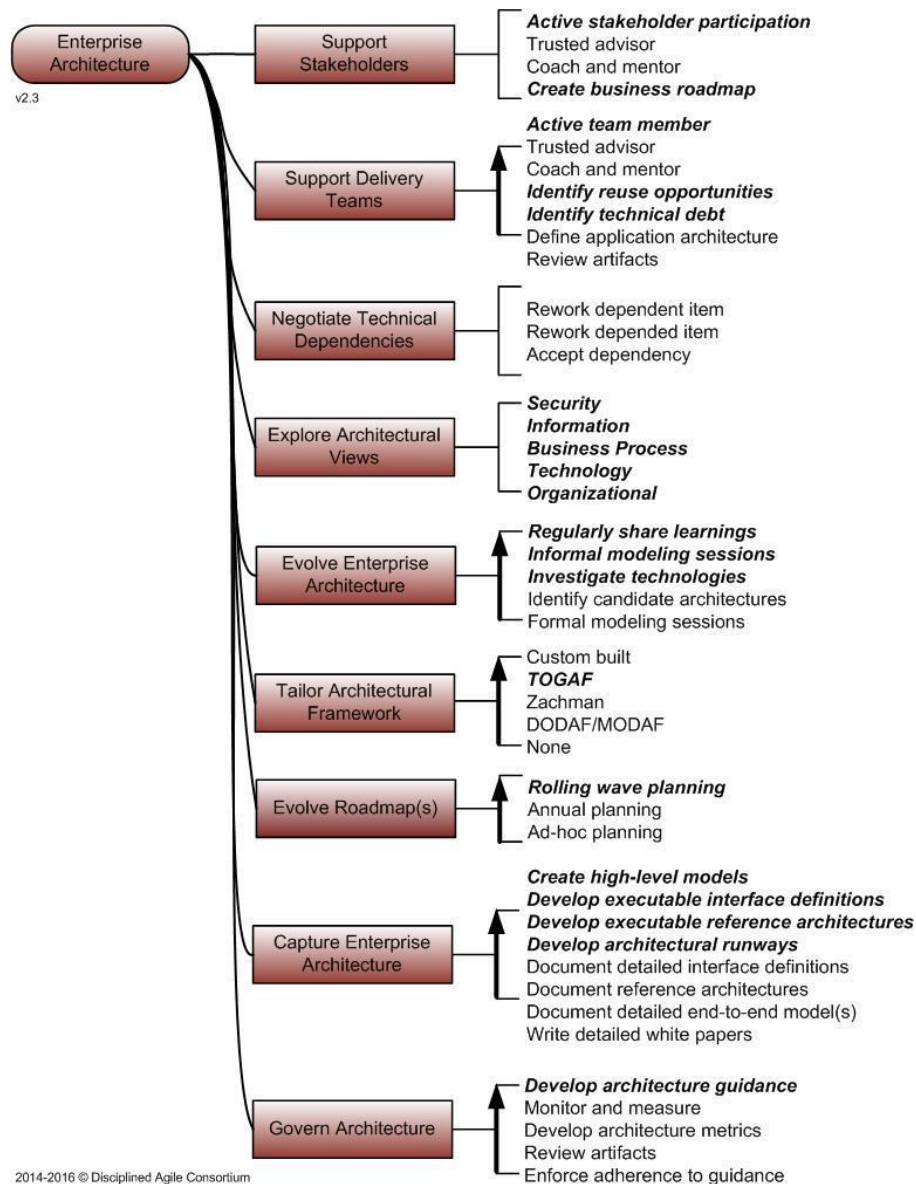


Figure 18. Activités de l'Architecture d'Entreprise selon Disciplined Agile

Comme SAFe®, DA préconise un fonctionnement proche du terrain en collaboration étroite avec les équipes. Il définit la posture de l'architecte d'entreprise agile (<https://www.disciplinedagiledelivery.com/agile-enterprise-architect-mindset/>) comme collaborative, orientée apprentissage, dans le partage, orientée métier, pluridisciplinaire, itérative, proche du terrain, pragmatique, prônant la réutilisation, et soucieuse de la dette technique.

#### 4.1.6 Analyse comparative et leçons à tirer

**Nexus et LeSS** semblent appropriés aux équipes maîtrisant Scrum et/ou très matures. Ils permettent de passer à l'échelle à moindre coût en limitant l'investissement initial. Mais ces cadres restent centrés sur le développement d'un seul produit complexe (un seul backlog) et ne développent pas les activités liées à la gestion d'un portefeuille de solutions.

**SAFe® et DA** sont les cadres les plus aboutis permettant de gérer les cas d'usage les plus complexes, tout en mariant les pratiques de l'agile et les activités classiques de l'IT ; ils apparaissent donc bien adaptés pour des grands comptes souhaitant déployer l'agile dans un contexte existant.

**Spotify** occupe une place à part : c'est un cadre peu prescriptif qui décrit plutôt une philosophie ; il est donc très approprié pour des équipes plus matures en agile, même si son modèle d'organisation est couramment repris y compris dans des entreprises qui débutent sur l'agile@scale.

Les cadres comme **Open Space Agility** ou **XSCALE** sont moins directement liés au développement de solutions logicielles et proposent des modes d'organisation innovants pour tout type de projet : ils ouvrent la voie au développement de l'entreprise agile, au delà des frontières de l'IT.

### Retours des explorations



*Sur le terrain, la plupart des entreprises développent leur propre approche, en s'inspirant de plusieurs cadres (ce qui ressort dans les études du Gartner et de cPrime citées plus haut) : la combinaison SAFe® + Spotify est souvent citée.*

Le développement d'une approche d'agile@scale est un **processus long et complexe** : l'entreprise doit monter progressivement en maturité et ne peut pas espérer atteindre un niveau d'efficacité maximal avant plusieurs années. Le processus de transition sera étudié au chapitre « La phase de transition ».

## 4.2 L'approche « Continuous Architecture »

“Continuous Architecture : Sustainable Architecture in an Agile and Cloud-Centric World”, Murat Erder, Pierre Pureur, Elsevier 2016

Les auteurs constatent que le fossé entre les architectes et les équipes de développement se creuse car l'agile pousse à développer toujours plus vite et en continu (Continuous Delivery et DevOps). Les pratiques doivent donc évoluer pour combler ce fossé.

6 principes sont mis en avant :

- Architecturer des produits, pas des livrables de projets (permet d'être mieux aligné sur le métier tout en facilitant la réutilisation de patterns)
- Se concentrer sur les caractéristiques architecturales (« Quality Attributes » : Performance, Robustesse, Évolutivité, Exploitabilité, Maintenabilité...), pas les exigences fonctionnelles
- Retarder les décisions de conception jusqu'à ce qu'elles soient absolument nécessaires
- Architecturer pour le changement : Tirer profit de petit composants faiblement couplés
- Définir une architecture pour le développement mais aussi pour (faciliter) les tests et le déploiement
- Définir l'organisation des équipes après le découpage du système (voir la loi de Conway)

Puis les auteurs introduisent des techniques et des outils pour délivrer l'architecture en continu :

- Exigences
  - Modéliser le métier à haut niveau avec des diagrammes de chaîne de valeur (Value Chains) – Des cartes de capacités Métier (Business Capability maps) peuvent aussi être utilisées à condition de ne pas sombrer dans les détails

- Le détail des chaînes de valeur peut être décrits dans les « user stories », qu'on regroupe dans des domaines (« user story themes »), des cas d'usage ou dans des « Features » (comme dans SAFe®)
- Pour les exigences non fonctionnelles, utiliser la technique des « Utility Trees » de la méthode ATAM (Architecture Trade-Off Analysis Method) et des scénarios d'architecture
- Les exigences sont capturées et affinées tout au long du processus projet, en juste à temps, dans des workshops ou des interviews des parties prenantes
- Les exigences fonctionnelles et non fonctionnelles et les scénarios d'architecture doivent être priorisés
- Conception
  - Il est essentiel de lister et de relier les décisions d'architecture puis de les documenter quand elles sont prises – le plus tard possible - (leur contexte, contraintes et leurs justifications)
  - La liste des décisions d'architecture constitue un backlog d'architecture ; les décisions devront être priorisés et traitées au bon niveau (dans l'équipe, ou à un niveau transverse)
  - Une technique utile pour prioriser les décisions d'architecture est la technique QFD de Six Sigma (EvolutionQuality Function Deployment) qui relie les exigences aux objectifs/bénéfices attendus par le métier
  - Développer des architectures de transition
    - Documenter la situation de départ
    - Concevoir la situation cible, à un haut niveau
    - Développer des situations intermédiaires et détailler les architectures de transition correspondantes. On parlera de « plateaux » d'architecture : chaque plateau introduit un changement d'architecture important et donne une base stables pour des évolutions fonctionnelles, sur une période d'environ 6 mois.
- Livraison
  - Mettre en place une démarche DevOps aboutie en incluant dans l'équipe toutes les compétences nécessaires pour le développement, les tests et l'exploitation et en travaillant sur :
    - Le feedback et la supervision en continu
    - L'intégration continue
    - Les releases et le déploiement continus
    - Les tests en continu
    - L'automatisation grâce au cloud
- Validation
  - A priori : L'architecture peut être validée par un mécanisme de « peer reviews », à fréquence régulière pour contrôler et tester les décisions d'architecture
  - A posteriori : L'approche peut être outillée pour automatiser des tests de charge, de revue de qualité de code etc ....

La Continuous Architecture consiste donc à faire de l'architecture en continu au plus près des équipes, à chaque étape du cycle de vie du produit, en évitant de produire trop de design trop tôt et en utilisant des techniques éprouvées permettant de prioriser les sujets dans un backlog d'architecture.

### 4.3 L'approche « Evolutionary architecture »

*“Building Evolutionary Architectures : Support Constant Change”*, Neal Ford, Rebecca Parsons & Patrick Kua, O'Reilly, 2017

Les auteurs tentent de définir ce qu'est l'architecture (plus précisément l'architecture logicielle) et comment cette discipline évolue dans le contexte actuel marqué par l'incertitude et l'exigence d'agilité. Tout projet doit prendre en compte des exigences fonctionnelles et non fonctionnelles. L'architecture



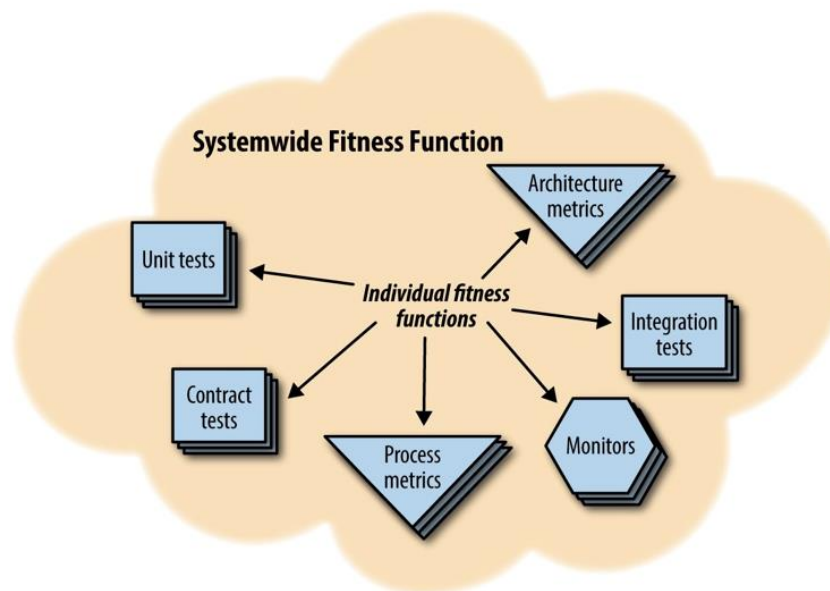
traite surtout les **exigences non fonctionnelles**, les capacités en « -ilité », exploitabilité, maintenabilité, utilisabilité ... Dans un contexte où tout change tout le temps, il est clé de prendre en compte **l'évolutivité** de l'architecture, sa capacité à prendre en compte le changement, à s'adapter facilement. L'architecture est pensée comme étant continue, sans état final, évoluant en permanence mais protégeant ses qualités essentielles pour ne pas se dégrader avec le temps.

L'architecture évolutive (« **Evolutionary Architecture** ») est définie comme étant capable de supporter des **changements incrémentaux**, mais ces changements sont **cadrés** et prennent en compte **plusieurs dimensions**.

Toute architecture répond à un but. Les changements sont cadrés pour s'assurer que les caractéristiques essentielles de l'architecture pour remplir son but sont protégées lors des changements. Pour cela, les auteurs introduisent la notion de **fonction objectif** (« **fitness function** »), qui fournit un moyen de mesurer quantitativement la qualité de l'architecture produite. Quand la fonction objectif est optimisée, la caractéristique architecturale qu'elle représente est remplie.

Une fonction objectif fournit une validation objective d'une caractéristique architecturale. Les fonctions objectifs s'implémentent sous forme de tests manuels ou automatisés, des relevés de métrologie ... pour mesurer la qualité du code, le respect de standards, le niveau de performance.

On peut analyser une architecture suivant différentes perspectives et trouver différentes caractéristiques qu'on souhaite protéger. Il est donc naturel de considérer différentes **dimensions** (métier, technique, données, organisationnelle, financière...) et de définir plusieurs fonctions objectifs primaires correspondantes, ainsi qu'une fonction objectif globale qui combine ces fonctions primaires, avec des poids relatifs, qui montrent les arbitrages faits entre les caractéristiques architecturales.



Les fonctions objectifs appartiennent à différentes **catégories**. Elles peuvent être discrètes ou continues, statiques ou dynamiques, automatisées ou manuelles, temporelles ou non, intentionnelles ou émergentes, spécifiques d'un domaine ou non. Elles sont priorisées en 3 niveaux : « clé » (essentiels pour les choix d'architecture), « importantes » et « sans impact » (sur les choix d'architecture). Elles doivent être revues régulièrement et permettent de connaître à tout moment l'état de santé de notre architecture.



Les auteurs promeuvent l'utilisation d'usine de développement et de déploiement de logiciel automatisés pour faciliter le **changement incrémental**, tout en testant en continu les caractéristiques architecturales en ayant automatisé les fonctions objectifs dans le processus.

Un aspect essentiel de l'architecture est le niveau de **cohérence** et de **couplage** entre les composants du système. Comme les autres caractéristiques de l'architecture, ils doivent être testés et vérifiés avec des fonctions objectifs. La **modularité** est une caractéristique recherchée, qui pose la question de la granularité des composants pour qu'ils soient réutilisables. On parle ici de **quantum architectural**, i.e. la plus petite taille d'un composant cohérent, et déployable indépendamment. Pour une application monolithique, le quantum est l'application complète. Dans une architecture de microservices, c'est le microservice. Il existe différents styles d'architecture (monolithes non structurés, architectures en couche, monolithes modulaires, architectures extensibles par plug-ins dites microkernel, architectures orientées événements, les architectures orientées services, à base de bus ou de microservices, les architectures « serverless ») avec des tailles de quantum différentes, et donc une évolutivité différente.

L'architecture évolutive concerne également l'architecture des **bases de données**. Les schémas des bases de données relationnelles doivent être pensés pour pouvoir supporter le changement incrémental. La complexité des schémas augmente souvent avec le temps, il faut donc ne pas hésiter à les nettoyer régulièrement. Il faut également éviter de partager la même base de données entre plusieurs applications car cela crée du couplage fort et limite les possibilités de changement. Les **transactions** sont aussi un facteur de couplage fort, qu'il peut être difficile de traiter avec certains styles architecturaux (comme les microservices).

Les **progiciels** sont un défi pour l'architecture évolutive car le changement incrémental est difficile (pas d'automatisation des tests et du déploiement), le couplage est fort et la définition de fonctions objectifs est ardue.

Enfin les auteurs délivrent quelques **bonnes pratiques** pour construire des architectures évolutives comme :

- Supprimer la variabilité inutile,
- Rendre les décisions réversibles,
- Préférer l'évolutivité à la prédictibilité,
- Mettre en place des barrières « anti-corruption »,
- Construire des prototypes jetables, se prémunir des dépendances externes, ...

Ils introduisent également quelques **anti-patterns** :

- La dépendance à un fournisseur principal (« vendor king ») – surtout pour les progiciels -,
- La non maîtrise des couches inférieures sur lesquelles on construit son architecture et les incidents qu'elles peuvent introduire (« Leaky abstractions »)
- La limite des progiciels et L4G qui ne couvrent facilement que 90% des besoins mais ne peuvent couvrir le reste (« Last 10% trap »)
- Les abus de la réutilisation : plus le code est générique, moins il est utilisable ....
- L'abus des technologies à la mode : bien pour le CV, moins bien pour le problème à résoudre...
- Une gouvernance trop normative
- Des cycles de release trop lents
- Des cycles de planification trop longs : les décisions d'investissement doivent pouvoir être revues fréquemment et ne pas engendrer trop d'architecture en amont

Enfin, l'architecture évolutive dépend de l'organisation choisie. Elle est plus facile à mettre en œuvre dans une organisation avec des équipes multidisciplinaires, par domaine métier, avec une philosophie centrée « produit » plutôt que « projet », conscientes de leur environnement (avec des points d'intégration définis par des « consumer driven contracts »). L'architecture évolutive dépend également de la culture d'entreprise, de sa maturité et de son approche du changement, de sa propension à mesurer les résultats par des métriques, sa volonté d'expérimenter pour apprendre, ...

## 4.4 L'approche Open group

*Agile Architecture in the Digital Age, July 2018*<sup>7</sup>

Le livre blanc de l'Open Group a pour objectif de poser le problème de l'architecture dans un environnement agile et de proposer un plan de travail. Le framework d'architecture agile proposé est en cours de développement et une première version est attendue pour Juin 2019.

Au-delà de ces travaux, un référentiel de bonnes pratiques a été établie : *The Digital Practitioner Body of Knowledge*<sup>TM8</sup>

Alors que l'architecture est parfois perçue comme un anti-pattern dans le déploiement de l'agile à l'échelle, ce livre blanc démontre que l'architecture est essentielle pour traiter un certain nombre de problèmes qui ne relèvent pas uniquement de l'organisation de l'équipe mais surtout de la structure du produit.

Il apparaît fondamental que :

- Les architectes d'entreprises travaillent à la modularisation des monolithes du SI
- Les pratiques d'architecture évoluent pour être compatible avec les démarches agiles
- Le corpus de connaissances des architectes soit mis à jour à l'heure du digital
- La gouvernance de l'architecture soit revue avec le passage de grands programmes à de nombreuses équipes autonomes

**Autonomie et couplage lâche** sont intimement liés : Pour être efficaces, les équipes agiles doivent être autonomes, ce qui n'est possible que si les produits sur lesquels elles travaillent sont faiblement couplés. Pour atteindre ce couplage lâche, une approche classique est la décomposition du produit en couches, mais Eric Evans a démontré qu'il était plus pertinent de découper par domaine fonctionnel : c'est le **Domain Driven Design** (DDD). Cela permet de définir des services avec un niveau élevé de cohérence interne et un couplage lâche entre eux. Les services peuvent être développés et testés indépendamment, ce qui permet d'accélérer le déploiement continu.

Mais l'autonomie des équipes doit être contrebalancée par un **alignement** efficace (même constat que Spotify) : une vision stratégique claire (le pourquoi) et de l'autonomie sur la mise en œuvre (le comment). La relation entre les décideurs et les équipes de type « command and control » ne fonctionnent pas avec des équipes autonomes. Dans une organisation agile, c'est la culture et les valeurs qui lient les acteurs et non pas des règles et des procédures venus d'en haut. Les architectes doivent donc intégrer aussi ce changement dans leurs relations aux équipes. La gouvernance de l'architecture doit évoluer.

---

<sup>7</sup> <https://publications.opengroup.org/w186>

<sup>8</sup> <http://pubs.opengroup.org/dpbok/snapshot/>

La définition de la stratégie, du business model et des produits devient également itérative avec des approches comme Lean Startup, où les équipes développent et testent des nouveaux produits et itèrent en fonction des résultats. De nouveaux business models émergent dont le modèle des « plateformes », qui révolutionnent tous les secteurs d'activité. Les plateformes délivrent de la valeur plus vite pour les clients qui sont impliqués directement et sont aptes à croître et à évoluer beaucoup et rapidement.

De nouveaux patterns d'architecture logicielle et de nouvelles technologies sont apparus avec l'émergence des géants du Web. Ils doivent être pris en compte dans le corpus des connaissances des architectes d'entreprise :

- Les microservices et les APIs
- Les nouveaux modèles de traitement analytiques comme MapReduce
- Des librairies open source d'intelligence artificielle comme Tensorflow ou OpenFace
- Une nouvelle vision des architectures distribuées basées sur du « commodity hardware », à la scalabilité horizontale
- Le théorème de CAP
- Les objets à état immutable
- La programmation fonctionnelle
- Les sagas pour gérer les transactions dans les architectures de microservices
- Les bases de données NoSQL
- L'analytique en temps réel (et non plus en batch) par le traitement de flux d'évènements
- L'automatisation de l'infrastructure (« Infrastructure as Code ») et DevOps
- Les applications Cloud Natives et leurs 12 facteurs

Ces tendances de fond justifient de repenser le rôle et la gouvernance de l'architecture dans le « **Agile Architecture Framework** » (AAF).

L'architecte doit mettre à jour ses connaissances pour rester crédible auprès des équipes et agir comme un facilitateur et un mentor. Sa valeur réside dans :

- Une vision globale de l'entreprise sur une page
- La diffusion d'une culture de développement basée sur le partage et le travail collaboratif
- La facilitation de l'émergence de standards issus des équipes
- La mesure automatisée de la conformité
- Une gouvernance de l'architecture adaptée

Le processus architectural ne donne plus lieu à un travail massif avant le projet (« Big Up Front Design ») mais à une « **Minimum Viable Architecture** » (MVA) qui sera affinée par la suite.

On peut également s'inspirer du concept de décision d'architecture de type 1 et 2 (utilisés chez Amazon et Netflix) : une décision de type 1 est irréversible et nécessite une gouvernance stricte. Une décision de type 2 est réversible et devrait être prise rapidement par les équipes sur le terrain (guidées par une vision globale de l'architecture).

Le développement du cadre AAF sera conduit sur 3 axes :

- Autonomie, isolation et alignement (rouge)
- Processus et rôle de l'architecture (bleu)
- Corpus de connaissances de l'architecture (vert)

<b>AAF.E-01</b> <b>Loosely-Coupled Systems &amp; Organizations</b> <ul style="list-style-type: none"> <li>How to architect loosely-coupled systems?</li> <li>How to architect modular organizations composed of autonomous teams?</li> <li>How to design organizations that produce modular architectures?</li> <li>How to refactor highly-coupled and monolithic systems?</li> </ul>	<b>AAF.E-02</b> <b>Business Architecture Patterns</b> <ul style="list-style-type: none"> <li>How to innovate business and operating models?</li> <li>Which business strategy concepts can help align the enterprise: vision, mission, purpose, ...?</li> <li>How to decompose the business into modular operating units?</li> <li>How to deploy the strategy in a non-command-and-control way?</li> </ul>	<b>AAF.E-03</b> <b>Aligned Organizations &amp; Systems</b> <ul style="list-style-type: none"> <li>How to preserve local autonomy while enforcing global alignment?</li> <li>Which organizational model and culture changes are required?</li> <li>Which governance model will keep organizations and systems aligned while preserving autonomy?</li> <li>How to enable services interoperability and composability?</li> </ul>	<b>AAF.E-04</b> <b>Software Architecture Patterns</b> <ul style="list-style-type: none"> <li>How to architect highly distributed software systems that are: <ul style="list-style-type: none"> <li>Responsive to user requests</li> <li>React to variable load conditions</li> <li>Remain available?</li> </ul> </li> <li>How to leverage big and fast data architecture patterns?</li> <li>What is the impact of AI/ML on system architecture?</li> </ul>
<b>AAF.E-05</b> <b>Minimum Viable Architecture</b> <ul style="list-style-type: none"> <li>How much architecture work should be done up-front for the next agile iteration?</li> <li>How should architecture decisions be made and validated?</li> <li>How should MVA influence/impact agile teams?</li> <li>How to align MVA with MVP?</li> </ul>	<b>AAF.E-06</b> <b>Evolvable Architecture</b> <ul style="list-style-type: none"> <li>Which architecture practices and patterns will facilitate future change?</li> <li>How to anticipate change and avoid unnecessary complexity?</li> <li>How to prevent the architecture from gradually degrading over time?</li> </ul>	<b>AAF.E-07</b> <b>Maturity Model</b> <ul style="list-style-type: none"> <li>How many maturity levels?</li> <li>How to define maturity levels?</li> <li>How to assess the enterprise's maturity level?</li> <li>What are the pre-conditions of a successful move to the next maturity level?</li> <li>What are the key success factors?</li> </ul>	<b>AAF.E-08</b> <b>Architect's Role &amp; Responsibilities</b> <ul style="list-style-type: none"> <li>What is the architect's role as a squad member?</li> <li>Should the architect become an "über" developer?</li> <li>What is the architect's role as guardian and defender of the overall system's coherence?</li> </ul>
<b>AAF.E-09</b> <b>The Agile Architect's Competencies &amp; Skills</b> <ul style="list-style-type: none"> <li>What core set of competencies and skills should all architects possess?</li> <li>Which soft skills are needed to lead and facilitate team collaboration?</li> <li>Toward a "T-shaped" full-stack profile that includes software development skills?</li> </ul>	<b>AAF.E-10</b> <b>Domain-Driven Design &amp; Event-Driven Architecture</b> <ul style="list-style-type: none"> <li>How to identify contexts and aggregates using event storming?</li> <li>How to draw context maps using the DDD strategy patterns?</li> <li>How to protect application code from future technical debt?</li> </ul>	<b>AAF.E-11</b> <b>Data &amp; Information Modeling</b> <ul style="list-style-type: none"> <li>How do we evolve data/information modeling techniques to cater for big and fast data technology?</li> <li>How to handle data when using the μ-services architecture style?</li> <li>What are the impacts of real-time streaming analytics on system architecture?</li> </ul>	<b>AAF.E-12</b> <b>Complex Systems Modeling</b> <ul style="list-style-type: none"> <li>How to model and steer the evolution of complex adaptive systems?</li> <li>How to use a Design Structure Matrix (DSM) to reveal both hierarchical ordering and cyclic groups within a complex technical system?</li> </ul>

Figure 19. Vue générale de la proposition d'Agile Architecture Framework

## 4.5 L'approche Collaborative Enterprise Architecture (Bente)

"Collaborative Enterprise Architecture : Enriching EA with Lean, Agile and Enterprise 2.0 practices", Stefan Bente, Uwe Bombosch, Shailandra Langade, Elsevier 2012

Selon les auteurs, le rôle de l'architecture d'Entreprise est de :

- Maîtriser la complexité des SI
- Aligner les SI sur le métier et garantir qu'ils créent de la valeur

Hélas, trop souvent elle ne parvient pas à démontrer au métier les bénéfices qu'elle peut apporter. On peut analyser sa performance sur 4 axes :

- **Perspective** : L'AE est-elle une vue d'avion ? ou au contraire au cœur des projets ?
- **Gouvernance** : rigide sur la conformité aux règles d'architecture ? ou plutôt tolérante ?
- **Stratégie** : Travaille-t-on sur un plan stratégique à long terme ? ou pas du tout ?
- **Transformation** : Quelle est la vitesse de changement dans le patrimoine SI, rapide ou lent ?

Les deux extrêmes de chaque dimension sont des anti-patterns. On recherchera donc un juste milieu et la situation de l'AE sera représentée sur un tableau de bord qui reprend ces 4 dimensions.

Afin d'améliorer les pratiques de l'AE, il est intéressant d'incorporer les bonnes pratiques du **Lean**, de **l'agile** et de **l'entreprise 2.0**. Les trois lignes directrices de l'AE collaborative sont :

- **Établir un ensemble « lean » de processus et de règles** ... plutôt que de surcharger les parties prenantes de processus bureaucratiques et de livrables inutiles
- **Adopter des techniques de résolution de problème évolutives** ... plutôt que de figer dans le détail la cible dans un modèle rigide
- **Promouvoir et animer la participation ouverte** ... plutôt que de s'appuyer uniquement sur des experts et une sagesse venue d'en haut

A partir du Lean et de l'agile, les auteurs proposent trois blocs de base pour l'AE collaborative :

- **Se débarrasser du gaspillage en simplifiant les processus d'architecture**
  - Les 7 gaspillages de l'AE sont : les livrables partiellement terminés, la sur-architecture (trop détaillée), des processus redondants, trop d'interfaces entre les acteurs, le multitâche, les délais pour obtenir l'information, les livrables contiennent des erreurs
- **Impliquer toutes les parties prenantes dans des itérations d'architecture**
  - L'architecture est développée progressivement au fur et à mesure des itérations
  - L'équipe d'architecture peut être une équipe agile autonome ou intégrée à l'équipe agile qui implémente la solution
  - Quand il y a plusieurs équipes agiles (agile@scale), il peut y avoir une équipe agile d'architecture qui regroupe les architectes des différentes équipes
- **Développer l'architecture itérativement grâce à un Kanban d'architecture**
  - On peut définir un backlog d'exigences qui seront progressivement raffinées suivant leur niveau et l'horizon de temps

Enterprise Level	Planning Horizon	Architecture Level According to TOGAF	Requirement Level in Agile
Business and IT strategy	Vision	Enterprise strategic architecture	Theme
Project portfolio	Road map	Segment architecture	Epic
Program	Release	Capability architecture	Feature
Project	Milestone/ iteration		User story

- Les phases A à H de l'ADM de TOGAF peuvent servir pour définir les différentes étapes du Kanban de l'AE

A partir des concepts de l'entreprise 2.0, trois nouveaux blocs de bases pour l'AE sont ajoutés :

- **Participation à la connaissance**
  - La formalisation de la connaissance (modèles de données, cartographies, ...) doit mobiliser tous les acteurs des SI et être partagée
  - Le rôle de l'architecte est dans la facilitation et la modération
- **Participation aux décisions**
  - Une plateforme collaborative permet de prendre en compte les avis des équipes sur la situation actuelle et les opportunités futures, ce qui peut éclairer les décisions d'architecture
- **Participation à la transformation**
  - Les équipes sont invitées à identifier les composants qu'elles réutilisent, documenter les nouveaux composants qu'elles créent ou les changements réalisés sur des composants existants en production, le tout étant relié sur une vue d'ensemble fournie par l'AE. Cela permet de développer la réutilisation et de capitaliser sur les problèmes rencontrés.

La mise en œuvre d'une architecture d'entreprise collaborative nécessite un programme de conduite du changement approprié, débutant par une analyse de l'attitude de l'organisation face au changement, l'identification des raisons de changer, la définition d'une vision et l'implication de tous les acteurs dans la transformation.

## 4.6 L'approche Architecture Flexible

*René Mandel - De la stratégie business aux systèmes d'information : l'entreprise et son écosystème- Dunod- 2006*<sup>9</sup>

Le concept d'agilité donne une capacité déterminante aux projets. La **flexibilité** est un concept différent qui s'applique au résultat des projets : le produit, ou l'ensemble des produits qui sont intégrés, y compris ceux issus des anciens projets.

**La flexibilité est une capacité d'adaptation de cet ensemble de produits**, face à des évolutions. Ces évolutions, de toutes natures, peuvent être imprévues, imprévisibles, voulues, anticipées, aléatoires, en rupture, ...

En regard de chaque évolution, dans la durée, un observateur objectif constate une flexibilité qui peut être :

- Nulle, par exemple pour des raisons technologiques (la technologie n'est pas disponible pour le produit), ou pour des raisons socio-techniques : le personnel n'est pas formé dans les délais, les utilisateurs refusent la rupture, ...
- Totale : il suffit de modifier un paramètre, couramment pratiqué, testé, déployé,
- Et toutes sortes de situations intermédiaires sont possibles, dans lesquelles un projet agile a du sens.

L'Architecture d'Entreprise ambitionne une vision globale, transcendant les systèmes, elle se préoccupe donc de tous les types de flexibilités, celles qui sont purement technologiques, mais aussi celles qui sont de l'ordre « fonctionnel », ou « métier », voire « business ».

Les capacités technologiques (ubiquité, connectivité généralisées...) s'accroissent, les capacités d'agilité des projets passent à l'échelle : ceci étend le domaine des possibles pour des apports de valeur, **bien au-delà des frontières de l'entreprise**.

En parallèle la flexibilité du produit s'améliore, sur certains aspects. Mais il n'y a pas de certitude que cet accroissement de flexibilité se porte sur tous les cas d'adaptation qui apparaîtront comme nécessaires au cours du temps.

L'expérience du passé démontre **qu'une dette d'urbanisme se crée naturellement**, sauf effort d'anticipation. Et même avec de tels efforts, il y a un pari qui n'est pas toujours gagné. Ce phénomène est bien connu pour les grands projets, avec leur important taux d'échec (voir Standish Group Chaos Report), en partie amélioré avec les approches agiles.

La flexibilité résulte pour une part des choix de développement que peuvent faire les projets, et des choix techniques. Mais elle **met en cause les choix d'architecture**. Une mauvaise architecture, même avec les développements les plus agiles, et la technologie la plus performante, aboutira à un produit rigide. Par exemple parce que le même concept est inscrit au cœur du produit sous mille formes incohérentes. La

---

<sup>9</sup> <http://trame-business.fr/mon-installation/index.php/architecture-flexible/>

remise en cohérence d'un tel produit peut, en termes de réingénierie, dépasser largement les coûts de développement initiaux.

Pour le dire autrement, la flexibilité du produit résulte de l'association de ces différentes flexibilités (architecture, applicative, technique, ...), et la rigidité est transitive.

### **L'Architecture Flexible**

L'« Architecture Flexible » permet d'accroître la composante flexible d'une architecture d'entreprise.

**Elle est basée sur une autre notion, historiquement bien connue dans la profession : celle d'invariant.**

De fait, tous les systèmes socio-techniques sont l'objet d'évolutions constantes, parfois extrêmement complexes, comme exposé ci-dessus. Par contre, si on recherche en leur cœur les invariants de toutes ces évolutions, on retrouve toujours les mêmes types d'invariants :

- **Sur les chaînes de valeur** une trame typique, facile à objectiver, en s'abstrayant des processus, des algorithmes et projets SI
- **Sur les données**, un cœur invariant, autour du référencement des principaux objets du monde réel (clients, intervenants, produits, services, composants, réseau, ...)
- **Sur les événements** qui provoquent les transformations et le développement des chaînes de valeur (événements clients, logistique, cycles produits, ..).

L'Architecture Flexible est ainsi un outil à la main de l'Architecte d'Entreprise.

C'est aussi un outil fédérateur pour tous les acteurs, car il se focalise sur ces invariants, dont la définition est à la base du fonctionnement cohérent et harmonieux de l'entreprise, ou de l'écosystème. En ce sens, il contribue à la légitimité des Architectes, car ceux-ci se focalisent sur :

- Une cohérence a minima, non invasive, ouverte à la subsidiarité,
- Une projection sur du long terme par le choix des bons pivots de l'évolution.

Enfin, cette Architecture est « Data Centric », ce qui correspond à l'Etat de l'Art actuel, et, plus fondamentalement, elle est « Event Centric », permettant un alignement aux exigences d'instantanéité et connectivité généralisée actuelle. Le traitement des événements, est un des piliers majeurs de cette méthode avec les patterns de Puits d'événements, et de Tri-datation.



## 5. Des changements majeurs pour l'AE

### 5.1 De nouveaux espaces de jeu du fait de la transformation digitale

La transformation digitale et ses implications sur l'architecture et le métier d'Architecture d'Entreprise ont été abordées dans différents travaux du club URBA EA ces dernières années. Au travers du texte et du schéma ci-dessous extrait de travaux de 2014, nous rappelons un des changements majeurs de la transformation tient à l'extension du champ de l'architecture à d'autres espaces SI et aux questions que cela soulève.

<https://fr.slideshare.net/ClubUrbaEA/en-ea-rapport-slideshare> Architecture d'Entreprise et transformation numérique – Daniel Breton, Nicolas Chevalier, 2014

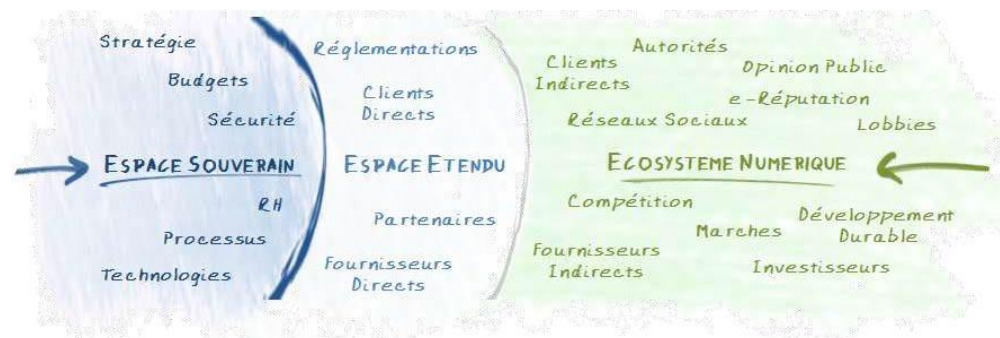
A partir des années 2010, l'ouverture de l'entreprise à l'économie numérique s'accélère.

Les entreprises vivent des changements importants (accélération des innovations, hégémonie de géant du numérique trans-étatiques, changement générationnel, désintermédiation, compétition accrue des pures players du numériques ou de nouveaux entrants en rupture de modèle).

Ils touchent aussi bien leur raison d'être, leur modèle économique, leur stratégie, leur valeur ajoutée, leurs organisations, leurs modes de fonctionnement et bien sur leurs relations à l'écosystème (marchés, clients, fournisseurs, ...). Et dans ce contexte « chahuté », les changements liés aux opportunités du numérique sont un facteur clé de l'évolution des entreprises et de leur écosystème.

En réponse, les entreprises mènent en parallèle des transformations multiples et rapides, de leur environnement, de leur stratégie, de leurs métiers, des relations avec leurs clients.... Les Systèmes d'Information (SI) sont au cœur de ces transformations. Le SI de l'entreprise doit répondre à toutes ces évolutions des activités métiers, mais aussi créer de nouvelles opportunités pour les métiers. Il permet de tirer parti des nouvelles technologies et des nouveaux modes de production (virtualisation, externalisation, cloud...), ou de prendre en compte les nouveaux modes d'accès (mobilité....) et les nouveaux usages (réseaux sociaux...).

L'environnement numérique dans lequel évolue l'entreprise se structure en trois espaces : **l'espace souverain** dont l'entreprise a la maîtrise, **l'espace étendu** dans lequel on trouve les partenaires directs de l'entreprise (clients, fournisseurs...) et **l'écosystème numérique**, un espace ouvert que l'entreprise peut exploiter pour étendre son espace économique ou auquel elle se confronte pour maîtriser les risques de son activité, notamment les risques concurrentiels.



Pour l'entreprise, les enjeux de la transformation numérique sont nombreux : optimisation de la performance opérationnelle, valorisation du patrimoine informationnel..., définition de nouvelles offres, définition de nouveaux champs d'activité, ou même de nouveaux modèles d'affaires pour conquérir de nouveaux marchés ou pour contrer de nouveaux concurrents.

Le SI est au cœur de tous ces enjeux et de nouvelles questions sont soulevées :



- Comment passer d'un outil de productivité tourné vers le SI souverain à un outil de développement du business ouvert sur l'extérieur ?
- Comment devenir plus agile et répondre aux enjeux client, d'innovation, de vélocité, d'incertitude, d'adaptation continue ?

## 5.2 Risques et opportunités de l'agile@scale

Les démarches agiles permettent de délivrer plus vite tout en traitant l'incertitude : l'équipe apprend grâce au retour du client à chaque itération, limitant ainsi les risques métier et techniques et convergeant vers le produit qui apporte le plus de valeur.

Il n'est donc pas étonnant que l'agile devienne progressivement la norme en gestion de projet innovant, bien au-delà des frontières des équipes de développement où elles sont restées longtemps cantonnées.

La généralisation et le déploiement de l'agilité remet en question les cellules d'architecture d'entreprise et le métier d'Architecte d'Entreprise.

Pour le groupe de travail du Club URBA-EA, c'est à la fois un risque et une opportunité.

### 5.2.1 Principaux risques pour l'Architecture d'Entreprise

- *Un décalage de rythme*

L'agilité raccourcit les cycles de décision et de réalisation alors que les travaux d'Architecture d'Entreprise s'inscrivent historiquement dans des cycles longs visant à fixer des cibles et trajectoires jalonnées de paliers stables.



*Comment être plus réactif ? Comment s'inscrire dans ces cycles accélérés ?*

- *Le dictat du court terme*

L'obsession de délivrer de la valeur à chaque sprint peut conduire à privilégier le court terme et à négliger ou ignorer les préconisations d'architecture axées sur la promesse de valeur à plus long terme.



*Comment influencer sur les décisions de court terme ? Comment intégrer les « enablers » et socles d'architecture aux roadmaps agiles ?*

- *Une perte de vision transverse*

L'autonomie des équipes agile risque de conduire à un SI atomisé, sans cohérence et sans vision transverse.



*Quels sont les invariants métiers sur lesquels s'appuyer ? Comment maintenir une vision partagée entre les équipes ? Comment construire un SI cohérent ?*

## 5.2.2 Principales opportunités pour l'Architecture d'Entreprise

- *Des boucles retours rapides*

Les méthodes agiles permettent d'aborder différemment les architectures SI et de construire des routes d'architecture plus innovantes, de travailler dans des logiques d'essai-erreur et de boucles retours vertueuses.



*Comment utiliser « l'agile » pour améliorer les fondations et l'architecture du SI (cf continuous architecture) ?*

- *Des attentes*

Face aux charges élevées de refactoring, aux difficultés d'intégration ou aux problématiques de montée en charge et d'opérabilité, les équipes agiles sont en recherche de solutions et en attente de soutien. L'Architecture d'Entreprise apporte le cadre, les outils et les capacités d'animation du dialogue court terme - long terme indispensable.



*Comment faire face aux attentes élevées et sollicitations nombreuses des équipes ? Comment monter en charge ?*

- *Des sollicitations nouvelles en terme stratégique et RH*

Le positionnement « aux interfaces et en transverse » de l'Architecture d'Entreprise la met en position de jouer un rôle nouveau auprès des décideurs. Au-delà des évolutions SI, l'Architecte d'Entreprise est sollicité dans les réflexions et décisions stratégiques, pour des choix organisationnels, pour définir des responsabilités, pour traiter des blocages, pour être un leader de la transformation managérial et culturel.



*Comment répondre à ces nouvelles sollicitations ? Quelles nouvelles compétences acquérir ? Comment devenir des agents du changement ? Quel profil ?*



*L'architecture émergente fonctionne jusqu'à un certain point car elle peut amener à une :*

- *Réduction de la productivité (peu de réutilisation, reconception et délais excessifs, difficultés d'intégration)*
- *Réduction de la qualité (non respect des exigences non fonctionnelles)*

## 6. Quelle réponse de l'Architecture d'Entrepris

### 6.1 Soutenir l'agile@scale

L'Architecture d'Entreprise est classiquement et historiquement une discipline visant à mieux piloter les transformations du SI. L'Architecture d'Entreprise est essentielle pour éviter le « chaos structurel » pouvant résulter des visions locales et fragmentaires. Elle est essentielle pour porter et traiter les enjeux de :



vente  
privée



#### Partager un outil central unique : La product map

Un livrable central du fonctionnement **Agile@scale** de Vente-privé.com (maintenu par l'équipe d'Architecture d'Entreprise) est la « product map » véritable point de repère des échanges de dialogue et d'aide à la décision pour :

- Structurer et limiter les dépendances entre les produits
- Assurer le partage des backlogs des features entre les équipes
- Identifier les adhérences et positionner les sujets,...

L'Architecture d'Entreprise contribue ainsi à la réussite de l'agile@scale en apportant des réponses aux défis de l'entreprise agile :

- **Agilité stratégique**
- **Vitesse d'exécution**
- **Résilience**
- **Collaboration globale**



## Défi 1 Agile@scale Pivoter

*-Contribuer à l'agilité stratégique-*

### Conduire les études d'impacts

- L'AE étudie les impacts, aide à la projection, construit des simulations, envisage des scénarios possibles
- L'AE définit des cibles et trajectoires
- L'AE conduit les réflexions et identifie les dépendances entre les niveaux métiers, fonctionnels, applicatifs et techniques

### Identifier les opportunités technologiques et business

- L'AE contribue à la veille et à la connaissance des écosystèmes
- L'AE met en perspective le positionnement de l'entreprise dans son écosystème
- L'AE conduit des études prospectives au service des enjeux de l'entreprise
- L'AE décrypte, qualifie les opportunités technologiques

### Innovier

- L'AE propose des solutions innovantes pour répondre aux problématiques et enjeux de l'entreprise
- L'AE définit, structure, évalue les Proofs of Concept ou Proofs of Value permettant de valider les scénarios innovants

vente  
privée



**Confier à des équipes transverses la conduite des études d'impact**



## Défi 2 Agile@scale Agir plus vite

*- Un SI plus flexible et une meilleure usine SI-*

### **Construire un SI plus évolutif et modulaire**

- L'AE renforce l'évolutivité du SI par la réduction de sa complexité
- L'AE procure plus d'agilité par la promotion de solutions modulaires et standardisées

### **Exploiter les ressources internes et externes**

- L'AE identifie et répertorie les ressources internes et externes pertinentes à exploiter
- L'AE facilite l'exploitation des ressources IT à disposition : data, services ou micro-services

### **Développer la qualité et la valeur des ressources (data et services)**

- L'AE définit des plans de valorisations des ressources : programme data, gouvernance et qualité des données, APIisation,...
- L'AE assure le pilotage et/ou est garant du niveau de services des ressources critiques : référentiels entreprise, ...

### **Améliorer la performance des équipes**

- L'AE contribue à une meilleure performance des équipes par l'analyse préalable des options, l'aide à la définition d'orientations claires en ligne avec les enjeux et les contraintes, par la réduction des risques, par l'amélioration continue et la tenue des rétrospectives
- L'AE clarifie les responsabilités et redevabilités entre les équipes
- L'AE réduit les efforts sur l'opération du SI en le rendant plus facile à maintenir



## Défi 3 Agile@scale

### Pérenniser

*- Cohérence et gestion des risques pour un système résilient -*

#### **Etablir des politiques et principes directeurs**

- L'AE établit des principes, règles et standards d'architecture pour le SI en collaboration avec les équipes de sécurité, data, techniques, conformité,...
- L'AE s'assure de la cohérence des évolutions et de l'alignement des équipes en portant des politiques et principes directeurs

#### **Piloter les risques d'architecture**

- L'AE prend en charge l'appréciation des risques d'architecture pour les projets structurants, et la qualification de la conformité des projets à la politique d'architecture de l'entreprise
- L'AE assure la revue de conformité des projets par rapport à l'architecture de référence du SI dans une démarche de risques opérationnels

#### **Promouvoir une architecture de référence du SI**

- L'AE promeut des catalogues de solutions homologuées (technologies, progiciels, patterns, APIs,...)
- L'AE généralise et industrialise les solutions innovantes

#### **Solidifier, optimiser, rationaliser le patrimoine**

- L'AE porte l'amélioration continue, l'optimisation du patrimoine applicatif par la proposition de rationalisations, et par le management des risques d'obsolescence, de dettes techniques ou de dettes d'urbanisation.

#### **Construire des fondations solides**

- L'AE définit des cibles et trajectoires de mise en œuvre des fondations d'informations données (référentiels, systèmes d'échange), des fondations applicatives et des fondations d'infrastructures techniques
- L'AE s'assure de la gouvernance des fondations du SI



## Défi 4 Agile@scale Faciliter la collaboration

*- Un cadre, un état d'esprit et des outils pour mieux collaborer et agir collectivement -*

### Se comprendre

- L'AE apporte un vocabulaire commun et des outils de dialogue entre les acteurs
- L'AE structure la connaissance des produits, ressources et données (définition, localisation, sources, contrats, traçabilité...)
- L'AE fournit une vision d'ensemble adaptée à chaque acteur ou partie prenante

### Collaborer

- L'AE contribue à la définition et au respect des responsabilités des équipes SI
- L'AE facilite la collaboration entre les acteurs des projets SI

### Traiter les blocages

- L'AE a un rôle d'ascenseur dans l'entreprise : l'AE identifie les sources de blocage (architecture mais incompréhension du modèle agile, des modèles d'investissement, des choix organisationnels, des subsidiarités,...) et les traite avec les acteurs pertinents

### Transformer la culture

- L'AE est porteuse des valeurs et principes du Manifeste agile, de la culture agile notamment concernant la prise de risque, le droit à l'erreur et le principe de décision au plus près du terrain.
- L'AE assure la prise en compte des innovations et des décisions d'architectures intentionnelles



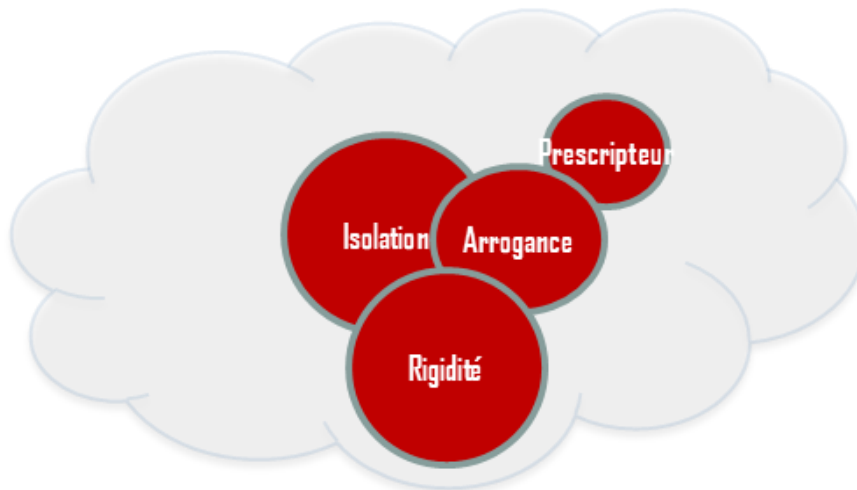
## 6.2 Gagner le défi de la confiance

L'Architecture d'Entreprise a un rôle potentiellement déterminant pour répondre aux défis de l'Agile@scale mais cela nécessite d'être crédible.

**Dans les organisations agiles, la légitimité ne vient pas du positionnement hiérarchique pas mais se gagne sur le terrain.**

Il s'agit maintenant, pour les équipes d'architecture, de structurer et revoir leurs pratiques et leurs postures au quotidien pour éviter de tomber dans des travers de fonctionnement encore malheureusement trop fréquents et incompatibles avec le logiciel « agile ».

### Les postures proscrites



#### # biais d'« Isolation »

L'image de « **Tour d'Ivoire** » est une image dévastatrice pour les cellules d'architecture d'entreprise et les architectes ont le devoir, tout en restant garant des biens communs, de ne pas plus donner corps à cette image forte d'isolation.

Les travaux d'architecture d'entreprise sont par essence collectifs et inclusifs.

Le **partage du « pourquoi »** de l'Architecture d'Entreprise, la **définition collégiale** des principes et règles, l'animation de **communautés de pratiques** d'architecture, la recherche de solutions en **mode inclusif**, l'animation de **rétrospectives** avec les équipes, sont autant d'actions et de marqueurs forts de l'intégration de l'Architecture d'entreprise au sein des organisations.

### # biais d'« Irresponsabilité »

L'image de « **Prescripteur, pas payeur** » est une autre image récurrente catastrophique pour les cellules d'urbanisme.

Les techniques de hiérarchisation des contraintes, de résolution de problèmes et de prise de décisions, les méthodes d'innovation en cas d'injonction contradictoire sont essentielles dans les phases amont de co-responsabilité des décisions.

L'enjeu n'est pas de prescrire et de se dire « j'ai fait mon job ». L'enjeu est bien **d'être acteur de la prise de décision** avec une posture d'ouverture, **d'intégration des différentes contraintes** (budgets, vitesse, ressources et compétences) pour trouver le chemin, **la route d'architecture** la plus pertinente avec toutes les **parties prenantes**.

### # biais d'« Arrogance »

Les « urbanistes architectes d'entreprises », sur un constat de problème d'urbanisation patent, par frustration sur le manque de poids dans la prise de décision peuvent tenir des propos accusateurs de type « **c'est la faute de** » tel service ou projet » si pas urbanisé, ...

C'est un comportement de type « passif-agressif » qui n'est pas compatible avec la fonction d'Architecte d'Entreprise et n'aide pas à l'émergence d'un climat de confiance.

Les architectes doivent travailler **leur assertivité** pour arriver à des compromis acceptables, **ne pas confondre les personnes et leurs rôles**, doivent rester positifs dans le **traitement des problèmes ou l'atteinte des résultats**.

Il s'agit d'être **solidaire des décisions** prises et de traiter les dettes d'urbanisation créées quand il est **possible et opportun** de le faire.

Enfin, la confiance se gagne aussi par la **reconnaissance des erreurs** en Architecture d'Entreprise, par le partage des enseignements issues de leur analyse post-mortem.

### # biais de « Rigidité »

Les urbanistes architectes ont tendance à penser qu'ils sont gardiens du « long terme » et à considérer qu'ils sont porteurs des « décisions structurantes » sur le SI.

Cette posture peut amener à surinvestir sur certaines études amont et sur-considérer l'importance et le côté « définitif » de la décision alors que la vitesse d'évolution du monde digital devrait nous inciter à plus d'humilité et des remises en causes plus régulières de décisions pour ne pas tomber dans le « dogme » rigide d'architecture.

Les « **ça ne se discute pas, un point c'est tout** » ne sont pas acceptables dans des environnements en forte évolutivité.

Au-delà de principes directeurs, il est important de tenir des **logs de décisions** aux différents niveaux pour garder trace des **arguments clefs dans le contexte local**, pour que l'architecte puisse argumenter mais aussi savoir **lancer des révisions** de règles ou de recommandations en cas de **changement de contexte**.

## Les postures de leaders agiles



### # Animateur de communauté

L'Architecture d'Entreprise doit mettre en place les rituels avec et au sein des différentes communautés : Architectes de domaines, techs leads, chefs de projets....



Animation de « guildes » (communautés de pratiques transverses) et de « leagues » (suivi des décisions locales)



Animation de communautés de pratiques orientés savoir-faire (hebdo)

### # Co-décision

L'Architecture d'Entreprise intervient dans un cadre de gouvernance élargie et aidante avec les autres fonctions transverses : sécurité, UX, Architecture technique, juridique, conformité, ...



- Privilégier des circuits courts de décision avec peu d'acteurs
- Ne jamais bloquer une valeur business

### # Souplesse

Retour d'expérience



Passer progressivement d'une logique d'aide bienveillante à une logique de « design authority » au fur et à mesure des phases des cycles projets/produits

### # Humilité

**Retour  
d'expérience**



*S'inscrire dans les rituels des tribus*

**# Leadership**

 **SOCIÉTÉ  
GÉNÉRALE**



*Mettre en place de nouveaux parcours de formations orientés prises de décisions,  
problem solving, leadership,...*

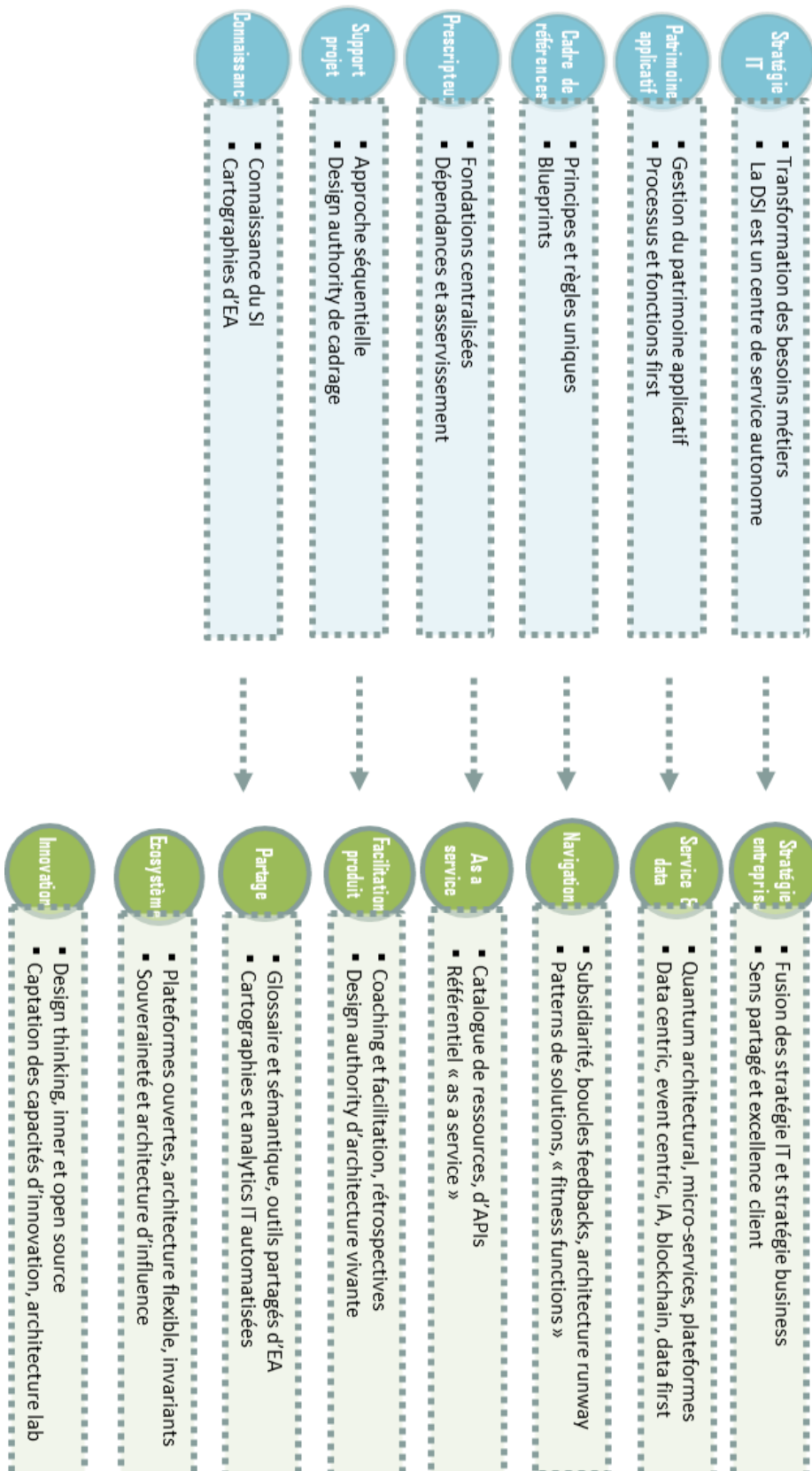
**# Ouverture**

**Retour  
d'expérience**



*Mettre en place des communautés d'architectures ouvertes aux sympathisants*

## 6.3 Revoir son cadre d'activité



## # Stratégie IT => Stratégie entreprise



- Transformation des besoins métiers
- La DSI est un centre de service autonome



- Fusion des stratégie IT et stratégie business
- Sens partagé et excellence client

### Avant :

La stratégie IT était le plus souvent découplée de la stratégie métier. Elle était établie par les DSI avec la « stratégie métier » comme « entrant ». La DSI était gérée comme un centre de service.

### Maintenant :

**L'importance grandissante du SI dans les stratégies d'entreprise mais aussi les impératifs d'agilité conduisent à la fusion des stratégie métiers et IT, à ne plus distinguer les projets métiers et UT**

L'approche est collaborative et non plus séquentielle

- Il s'agit pour l'Architecte d'Entreprise de jouer un rôle dans les phases critiques, les « **pivots** » des **modèles d'affaire ou stratégie**, d'aider à traiter les **situations complexes ou chaotiques** avec des approches appropriées.
- Il s'agit de faire émerger un « **phare** », des résultats à atteindre pour **aligner** les différentes équipes métiers et IT aux différents niveaux : entreprise, portefeuilles, trains, programmes, projets métiers et IT.
- Il s'agit pour l'Architecte d'Entreprise d'aider à **clarifier les problématiques à traiter ou résultats à atteindre** : quel est le « **claim** » **vérifiable à résoudre** ? quelles sont les attentes du client (design thinking,...)
- Il s'agit d'être contributeur et acteur des **jalons clefs de cohérence** entre les équipes agiles (PI SAFE,...)
- Il s'agit d'aider à **définir les organisations** adaptées ( cf loi de Conway selon laquelle les modèles d'organisation influent sur les produits SI)
- Il s'agit enfin d'être porteur des stratégies entreprises, du « **sens partagé** » auprès des équipes.

vente  
privée



*S'ouvrir et s'inspirer des entreprises inspirantes avec des benchmarks, visites avec les C-level,..*

*Définir et partager les invariants métiers*

*Des ambitions claires et partagées en termes de vélocité et de différenciation*

*Une cohérence entre le modèle d'organisation, la culture, les processus et l'IT*

*Pas de distinction entre projets IT et projets business*



*Développer des pre-PI (SAFe) en nombre d'acteurs réduits pour faciliter la transversalité et s'aligner sur les trajectoires*



*Les architectes fournissent de la visibilité sur les besoins futurs pour identifier les orientations*



## # Patrimoine applicatif => Service et data



### Avant :


Les travaux d'urbanisme étaient centrés sur l'évolution d'un patrimoine applicatif optimisé pour soutenir les processus métiers.


### Maintenant :

**L'évolutivité et l'ouverture prime sur la performance.**


**La donnée est le nouvel or noir, les algorithmes les graals**


- Il s'agit pour l'Architecte d'Entreprise de construire des architectures flexibles et évolutives :
  - Découplage, produits, services et micro-services, quantum architectural (« evolutionary architecture »)
  - Plateformes de services, d'automatisation, devops
  - Plateformes d'infrastructure, Cloud,..
- Il s'agit pour l'Architecte d'entreprise de faire évoluer les architectures vers des architectures :
  - Data driven, data centric, event centric
  - Blockchain
- Il s'agit de construire et opérer sur des plateformes ouvertes, de structurer, **définir et rendre disponible les produits** pertinents et valorisables.
- Il s'agit aussi d'aider à la **gouvernance** et valorisation de ses nouveaux assets et par exemple de définir les **plans de valorisation** intrinsèque de la donnée (qualité, disponibilité, ...) ou de valorisation (usages, services, ...).





*Privilégier les plateformes ouvertes pour capter une partie de l'innovation du marché*





*Privilégier les standards au sein d'un domaine (~stack de référence du marché / de l'écosystème)*

*Définir des "chapeaux" en fonction des contextes et des sujets*

*Limiter les dépendances entre les produits / équipes IT*

## # Cadre de référence => Navigation



### Avant :

Les architectes d'entreprise s'assuraient d'établir et de faire respecter, auprès des équipes IT, un cadre régalién de principes et de règles d'architecture.

### Maintenant :

Les travaux de conception sont réalisés par les équipes dans une logique de décentralisation et d'autonomie.

- Il s'agit de positionner les **architectes aux différents niveaux** pour assurer les travaux de **vision**, **d'alignement**, de **structuration**, **d'exigences non fonctionnelle** ou de **tenue des backlogs d'enablers**
- Il s'agit de définir des **architectures de transition** ou paliers stables
- Il s'agit, comme proposée par Gregor Hohpe (<https://martinfowler.com/articles/architect-elevator.html>) de penser son action dans une **logique ascenseur**, d'identifier les blocages aux différents niveaux et **de les traiter**.
- Il s'agit aussi de fournir un **cadre et des points de repères de haut niveau** complétés **d'outils pratiques** pour faciliter le « voyage » et les décisions au travers de **bibliothèques de patterns**.
- Il s'agit de **co-construire les routes d'architecture**
- Il s'agit de **retarder les décisions architecturales le plus tard possible** (seulement si nécessaire)
- Il s'agit de **documenter les prises de décisions (logs de décisions)**
- Il s'agit de **lancer des initiatives exploratoires** en amont des solutions fonctionnelles sur les sujets innovants (prototypage, évaluation d'alternatives), sur les sujets d'architecture eux-même, sur les infrastructures et/ou les sujets de conformité



- Décider plus vite pour échouer plus rapidement
- Savoir où on va dans les 3 mois et où on veut atterrir dans les 3 ans
- Privilégier les règles simples de très bas niveaux
- Cibler la normalisation au sein d'un produit, laisser la liberté sur le reste
- Etablir et diffuser les graphes de dépendances



- Les architectes travaillent en juste à temps avec les équipes sur les composants d'architecture
- L'architecture doit être prête au moment nécessaire, ni trop tôt (car elle sera obsolète lors de son utilisation), ni trop tard
- Sécuriser l'allocation de moyens des enablers techniques par un budget dédié



- Responsabiliser le Product Owner sur l'ensemble des entrants et sortants de l'équipe (y compris les enablers d'architecture)

## #Prescripteur => As a service



### Avant :

Les architectes d'entreprise, au travers de revues de projets, prescrivaient aux équipes :

- Le respect des architectures de référence définies sur base de l'ensemble des besoins et exigences de l'entreprise
- la bonne réutilisation et mutualisation des ressources
- l'asservissement aux fondations et référentiels
- ...

### Maintenant :

**La valeur d'une architecture se mesure notamment à sa capacité à répondre, dans le temps, aux exigences non fonctionnelles majeures du contexte (sécurité, résilience, scalabilité, réactivité,...).**

**L'accessibilité et le niveau de service des ressources (data, services, micro-services) sont les clefs de l'adoption et de la valorisation des services et ressources IT.**

- Il s'agit pour les architectes d'entreprise de cibler et quantifier les **caractéristiques architecturales** essentielles (fitness features en evolutionary architecture) pour un contexte donné et de s'assurer de sa testabilité (automatisée) en continu
- Il s'agit de **s'assurer de l'identification, qualification et adéquation des niveaux de services** des ressources data et APIs
- Il s'agit de **mettre à disposition et promouvoir les ressources**, d'être product owner des catalogues d'APIs,...
- Il peut aussi s'agir **d'opérer (c'est-à-dire d'assurer le pilotage et la réalisation de produits/services) pour des ressources cœurs** (comme les pivots référentiels)

Retour  
d'expérience



*La cellule d'architecture a développé les compétences techniques pour prendre la responsabilité et assurer les évolutions du produit « référentiel pivot client » en mode agile et devops*

## # Support projet => Facilitation produit



### Avant :

Les architectes intervenaient dans les phases de cadrage projet en tant que « design authority ».

Les architectes avaient adoptées des outils et méthodes (TOGAF,...) pour assurer la professionnalisation de leur travaux et argumenter leurs décisions.

### Maintenant :

**La logique produit avec des évolutions régulières oblige à considérer les architectures comme « vivante » avec la valeur et l'excellence client comme guide.**

**Les équipes sont les mieux placées pour faire les bons choix de « design »**

- Il s'agit pour les architectes d'entreprise d'intervenir en **coaching des équipes** pour leur donner d'autres perspectives, rappeler si nécessaire la voix du client, les champs de contraintes et niveaux d'exigences, co-construire des « architecture runway »,...
- Il s'agit d'outiller et former les équipes pour qu'ils puissent **faire par eux-mêmes les bons choix architecturaux et automatiser les tests des exigences non fonctionnelles**
- Il s'agit de mettre en place les organisations et les modes de fonctionnement de **design authority d'autorité d'architecture vivante** avec les pairs et les autres équipes transverses (architecture technique, sécurité, conformité,...) pour s'assurer de la qualité des architectures (fitness functions, ENF,...), du traitement des dettes techniques et dettes d'urbanisation et de la pertinence des choix (feedbacks,...)

vente  
privée



- *Passer de projet à produit avec intégration de la maintenance*
- *Des équipes produits de 6 personnes à privilégier (de 3 à 12 personnes)*
- *Une logique de délégation et de confiance a priori aux équipes sur les choix techniques*
- *Privilégier des équipes produits multidisciplinaire avec limitation des équipes supports transverses*
- *Déployer le plus de SPOF possible pour partager la connaissance et redonder dès que possible*
- *Eviter au maximum les projets « cross-produits »*

Retour  
d'expérience



- *Repousser les décisions structurantes le plus tard possible*
- *Partager largement les outils d'architecture*
- *Avoir des plans de remédiation des dettes techniques et dettes d'urbanisation aux différents niveaux*
- *Passer des principes aux patterns de solutions*
- *Construire des librairies de solutions*

## # Connaissance => Partage



### Avant :

Une part significative de l'énergie des cellules d'Architectes était consacré à la cartographie et la connaissance du SI, souvent pour son propre usage.

### Maintenant :

**L'évolutions des technologies data et digitales ouvrent des champs de possible quant aux outils de gouvernance IT**

**Dans un contexte complexe et agile, il est indispensable d'outiller la collaboration des équipes, de diffuser la connaissance du SI et de développer l'aide à la décision.**

- Il s'agit d'établir **des glossaires** et **lever les ambiguïtés de langage** des différents domaines (Domain Driven Design)
- Il s'agit d'un côté pour l'Architecte d'Entreprise **d'automatiser la collecte de données sur le SI** pour être ancré sur la réalité et limiter l'effort de modélisation de l'existant
- Il s'agit de l'autre côté de **mettre à disposition et d'exploiter** ces données pour aider les études d'impacts, les « what if » et **prises de décisions** aux différents niveaux (stratégie entreprise, étude d'impact, priorisation de backlogs,...)
- Il s'agit de consacrer **un temps conséquent au partage** des travaux d'architecture
- Il s'agit de suivre **en transparence les décisions** aux différents niveaux
- Il s'agit de passer à communiquer, d'informer, **d'animer des communautés de pratiques**,...

vente  
privée



- Être rigoureux sur la documentation produit et la partager
- Partager la connaissance, beaucoup de schéma, de communication
- 50% du temps consacré au support des tech lead (coaching et accompagnement)
- 

e  
pôle emploi



- Traçer et suivre les options dans la « solution intent » développées en collaboration avec les équipes

SOCIÉTÉ  
GÉNÉRALE



- Suivre, outiller et partager en transparence les log de décisions prises aux différents niveaux

## # Innovation



- Design thinking, inner et open source
- Captation des capacités d'innovation, architecture lab

### Avant :

Les Architectes d'Entreprise conduisaient des études prospectives et participaient à la veille technologique de l'entreprise.

### Maintenant :

**L'IT devient un facteur clef de différenciation et de valeur pour les clients**

**L'évolution rapide des solutions et modèles d'architecture (micro-services, cloud, blockchain, big data,...) pose la question de la compétence et de la crédibilité des Architectes d'Entreprise.**

- Il s'agit d'être en capacité technique de **conduire et piloter des expérimentations**, PoCs, PoVs des nouvelles architectures pour mieux les comprendre et cibler leur usage
- Il s'agit de **promouvoir un état d'esprit d'essai-erreur** auprès des équipes, d'inciter au prototypage et au design thinking dans un rôle de leadership méthodologique et de changement de posture.
- Il s'agit aussi de promouvoir **l'innovation sur les sujets d'architecture et les besoins non fonctionnels**



- *Promouvoir l'innovation à tous les niveaux*

### Retour d'expérience



- *Intervenir dès les phases d'idéation*
- *Mettre en place des boucles de retours rapides pour valider les choix d'architecture et gagner en maturité*
- *Développer le potentiel de subversion*



- *Responsabiliser les architectes de niveau solution pour transformer les innovations validées en enablers*



## # Ecosystème



- Plateformes ouvertes, architecture flexible, invariants
- Souveraineté et architecture d'influence

### Avant :

Le SI d'une entreprise visait à outiller et soutenir les processus métiers internes.

### Maintenant :

**Les modèles de plateforme ouverte deviennent les modèles dominants. Les entreprises gagnantes sont celles qui arrivent à capter la valeur et les capacités d'innovation des écosystèmes.**

- Il s'agit pour l'architecte d'entreprise de **comprendre les écosystèmes** en assurant la veille, animant des benchmarks, décryptant les tendances.
- Il s'agit d'animer et promouvoir les **écosystème internes**
- Il s'agit aussi d'être **acteur des projets de partenariat**.
- Il s'agit aussi de **contribuer aux communautés** sectorielles ou communautés technologiques et d'agir ainsi en influenceur au sein des écosystèmes business/IT



*Organiser des pitches d'une minute de tous les projets (hebdo)*



- *Contribuer aux écosystèmes*
- *Jouer en priorité la carte open source*

**Secteur  
bancaire**



- *Conduire des « think tanks d'architecture sectoriels » d*
- *Monter au besoin des consortiums pour opérer des architectures d'écosystème (blockchains, APIs, ...)*

## 6.4 Basculer d'une logique projet à une logique produit

L'architecture se décentralise et se décline à tous les niveaux au même titre que les décisions. Un développeur qui produit une API contribue à l'architecture en faisant un choix de design. Un déploiement d'un container influe sur l'architecture.

Les changements induits par l'agilité s'appliquent alors à l'architecture :

- Vision
- Collaboration
- Orientation sur la valeur
- Expérimentation
- Découpage
- Itération
- Automatisation
- Amélioration Continue

### 6.4.1 La phase de cadrage (Framing)

Traditionnellement les projets étaient initiés par une étude permettant d'identifier l'ensemble des évolutions à apporter au niveau des processus, des liens fonctionnels, des applications, des données, des technologies à utiliser jusqu'à l'infrastructure nécessaire. Afin d'accélérer cette phase de cadrage qui est importante pour projeter l'organisation dans l'exécution, le dossier d'architecture se voit allégé des détails et s'enrichit des notions de valeur et des principes à respecter. C'est un niveau plus macro qui permet aux équipes de prendre les décisions et les technologies adaptées au besoin. Il sert de fil rouge et vise à être utilisé par l'ensemble des acteurs.

C'est un premier découpage qui alimente le « backlog »(liste de tâches) initial.  
Cette analyse se fait de manière **collaborative**.

Les architectes aident à identifier et à prioriser les éléments fondamentaux fonctionnels et techniques.

Par exemple l'utilisation d'une architecture micro-services, comportant de nombreux avantages d'un point de vue de la gestion des changements, s'accompagne entre autres par une évolution de gestion des logs. Lors de la réalisation du premier service avoir une gestion des logs centralisée n'est pas crucial. Par contre dès que vous aurez une centaine de services redondés sur plusieurs sites, ça deviendrait inexploitable sans. Lors de la phase de cadrage, le choix de ce type d'architecture doit intégrer une tâche sur la création de cette capacité.

En sortie du cadrage, nous avons au minimum la vision, la valeur associée, l'analyse de l'existant, les principes à respecter et un « backlog » initial. Afin de se projeter, une tentative de planning peut être inclus dans le dossier d'architecture si vous avez un date butoir (réglementaire par exemple) et une estimation macro.

## 6.4.2 L'exécution

En fonction de l'ambition et des moyens, vous allez avoir d'une à une dizaine d'équipes qui peuvent travailler sur différents sujets autour d'un produit ou d'une chaîne de valeur. Les choix qui vont être fait doivent être documentés par les équipes. La complexité vient du nombre de décisions effectuées via la décentralisation portée par le passage à l'échelle de l'agilité.

Toujours dans cette démarche de collaboration, les architectes restent à **proximité** et contribuent à l'exécution.

- *Démarrage d'une nouvelle itération*

Les équipes agiles vont initier leur backlog. Pour cela ils doivent se projeter sur un besoin métier, sa valeur associée et finalement sur la solution à apporter. Cette solution est présentée au démarrage. L'architecte peut alors soit participer à la définition de la solution si celle-ci est complexe ou simplement à la revue si on retombe sur des pratiques déjà connues.

- *Accompagnement de l'exécution*

**L'automatisation** : Le développement de test d'architecture renforçant les décisions prises dès la compilation du code permet d'assurer que les principes définis sont respectés. Voici des exemples de test d'architecture rencontrés :

- Au niveau du code : utilisation de « archunit » pour vérifier qu'une classe n'a pas de dépendance avec un composant ou une librairie non souhaitée.
- Au niveau composant : entre service utilisation d'approche « consumer driven contract » (<https://martinfowler.com/articles/consumerDrivenContracts.html>) avec « Pact » (<https://docs.pact.io/>)
- Au niveau système : alimentation des référentiels applicatif centraux.

Le but étant de réduire les phases de revue de conformité voir de les supprimer.

**La métrologie** : l'agilité pousse à mesurer la valeur délivrée. Chaque livraison va de pair avec un indicateur défini par l'équipe.

**La transparence** : Toute décision doit être partagée et capturée par un enregistrement. La technique de l'ADR (architecture decision record) préconise d'avoir un stockage au même niveau que le code. Ce document contient la décision avec son contexte et ses conséquences. C'est utile pour les nouveaux arrivants ou autre équipe amenée à modifier un composant en particulier. D'un point de vue externe l'absence de ce type d'information est un mauvais signe. L'architecte de même partage des décisions de plus haut niveau.

**Les fondations** : Les architectes sont amenés à mettre en place des capacités techniques et fonctionnelles globales pour faciliter les transformations. Ces capacités peuvent consister de la création de librairie à la

mise en place de solution de sécurité en passant par des catalogues de service. Le terme « architecture-as-a-service » décrit bien cette nouvelle activité.

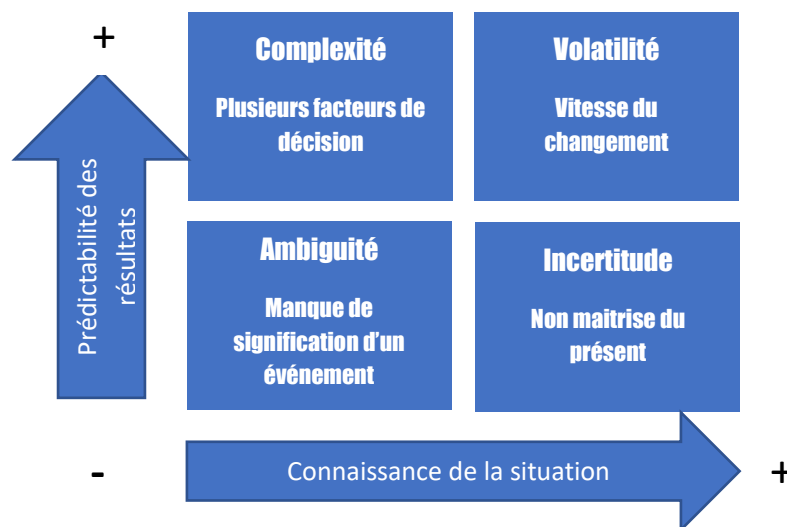
- *Les rétrospectives et les démos*

Ceux sont des cérémonies qui d'un point de l'architecture permettent d'identifier les actions d'amélioration et d'assurer qu'il n'y a pas d'écart majeur. C'est une source d'alimentation des sujets d'architecture via la remontée des points bloquants : Est-ce qu'il faut créer une capacité technique ? Est-ce qu'il y a de nouveaux produits à inclure, un lien avec un autre domaine fonctionnel qui est mal compris ?

## 6.5 Faire évoluer ses méthodes et livrables

### 6.5.1 Appréhender la stratégie : Dynamique d'un monde VUCA

Volatile, incertain (Uncertain), Complexe et Ambiguë : cet acronyme a été introduit par le U.S. Army War College pour caractériser les situations que rencontrent les forces d'interventions sur le terrain. Dans le contexte de la stratégie d'entreprise, nous retrouvons les mêmes éléments. Il est difficile alors de définir des stratégies à 3 ans ou plus.



**Figure 20 La Matrice VUCA**

La numérisation, l'émergence de nouveaux concurrents (start-ups), la mondialisation, les nouveaux interfaces clients sont autant de contraintes qu'il faut prendre en compte.

« En abolissant l'espace et le temps, le digital ouvre des opportunités inédites et tend, de façon corollaire, à une perte de repères. » - Sophie Muffang – Les Echos

- *La vision en réponse à la volatilité*

Il est clé d’avoir le sens, le pourquoi. Les écarts apportés par une forte volatilité sont compensés par un ligne directrice claire. La stratégie de l’entreprise doit être comprise par l’ensemble des équipes et ainsi déclinée à tous les niveaux.

- *La métrologie en réponse à l'incertitude*

La compréhension de l'existant a toujours fait partie des enjeux de l'architecture d'entreprise. Dans l'accélération, la multiplication des systèmes voire la modularisation induite par les nouveaux besoins (ouverture du système), il est vital de collecter le maximum d'information. La définition des bons indicateurs et leur automatisation permet de faciliter la prise de décision. La maintenance de cartographie du système d'information se doit d'être aussi dynamique que la revue de la stratégie.

De même, les revues de conformités, la connaissance des usages et l'identification des faiblesses du système et de l'organisation alimentent la stratégie.

- *Le design thinking à la complexité*

« Le design thinking est une approche de l'innovation et de son management qui se veut une synthèse entre la pensée analytique et la pensée intuitive » - Source Wikipedia

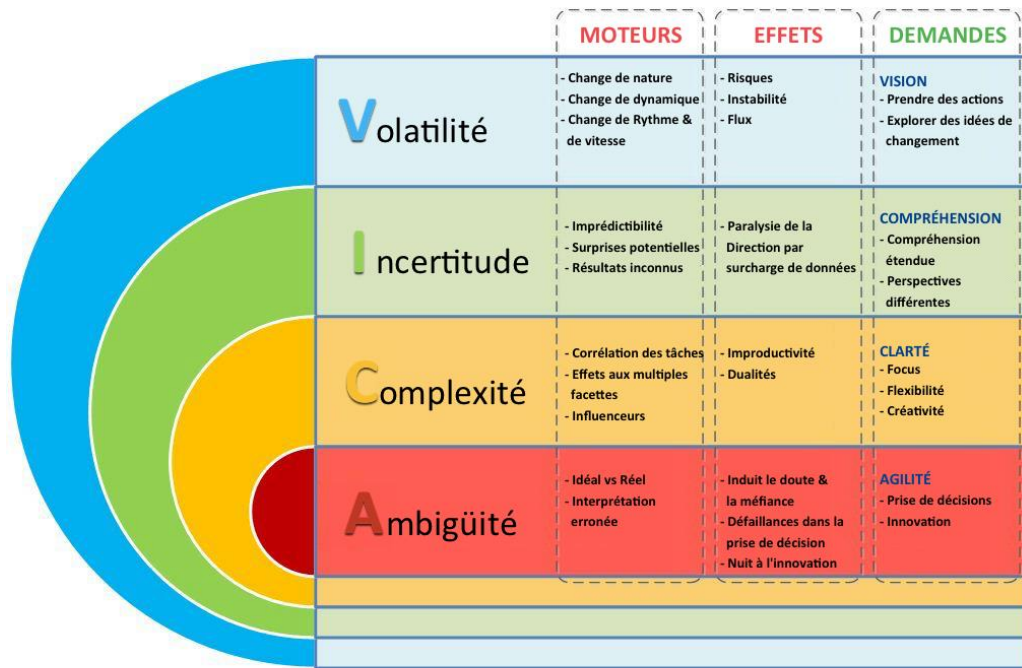
La collaboration est le pilier clé de la définition de la stratégie d'entreprise. L'utilisation de l'approche Design Thinking permet de fédérer l'ensemble des acteurs en proposant un processus en cinq étapes :

- Empathize (Compréhension),
- Define (Définition),
- Ideate (Conception),
- Prototype (Prototypage),
- Test.

Une déclinaison de cette approche est le Design Sprint, créé par Jake Knapp chez Google Ventures, qui donne une méthode très détaillée sur comment exécuter cet exercice.

- *L'agilité en réponse à l'ambiguïté*

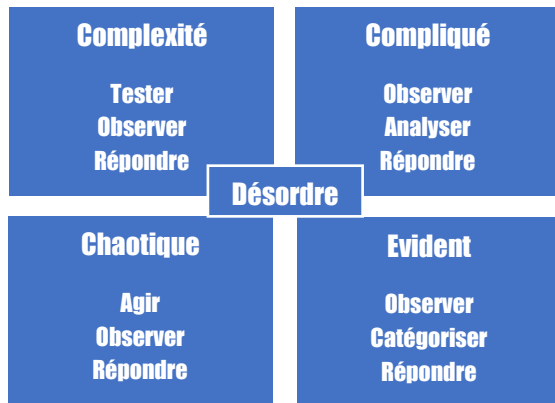
L'agilité est une réponse nécessaire pour répondre à des événements qui n'ont pas de précédent et dont l'issue est non prédictible. Il est alors important d'agir dans un premier temps pour expérimenter et pouvoir obtenir des clarifications. La stratégie de l'entreprise doit alors se faire de manière itérative.

Figure 21 source <https://wikiagile.cesi.fr/index.php?title=VUCA>



## 6.5.2 Le modèle Cynefin : un outil d'aide à la décision

Afin de naviguer dans la définition et l'exécution de la stratégie, le modèle Cynefin (prononcez ke-nav-in) peut servir de boussole pour déterminer le meilleur cap. Créé en 1999 par Dave Snowden puis détaillé avec Cynthia Kurtz, le modèle propose 5 contextes de décision et les stratégies à adopter.



Basé sur les faits/les données, nous pouvons prioriser des actions ou favoriser des pratiques. Il permet d'évaluer la situation pour ainsi prendre la meilleure décision.

## 6.5.3 Exemple de nouveaux outils

Les études, les documents, les livres et des explorations nous ont permis d'identifier des modifications dans les livrables traditionnellement produits ainsi que de nouveaux livrables recommandés.

Ces modifications portent essentiellement sur un découpage des documents et de la revue de la granularité des informations contenues. Il n'est plus attendu d'avoir un dossier d'architecture qui contient l'intégralité des changements à opérer avec leur détail d'exécution. Le dossier d'architecture se concentre sur la vision, la valeur, les principes à appliquer sur le domaine fonctionnel concerné et initie le backlog du produit. Les éléments plus précis sont adressés à chaque itération pour compléter la vision. Ils ne sont plus produits uniquement par les architectes.

Un des enjeux est de maintenir un niveau d'information et une vision transversale dans un environnement très modulaire impliquant une augmentation de la complexité des systèmes informatiques.

### Domain Drivent Design et Event Storming

Le « domain driven design » (DDD) est une approche pour gérer la complexité par la définition d'une vision et d'un langage partagés par toute l'équipe (développeur, représentant métier, architecte, manager...). La modélisation des domaines fonctionnels est une des activités qui permet de gérer l'essentiel de la complexité. Nous pouvons citer certains éléments clés :

- 1- Les bornes du domaine (Bounded context), le périmètre

- 2- Les objets ayant une identité et un tout cohérent (Aggregats)
- 3- Les événements du domaine

Sur ce dernier point, l'utilisation de l'« event storming » permet de capturer efficacement les informations. C'est un atelier de travail assez léger inventé par Alberto Brandolini (<https://www.eventstorming.com/>) basé sur une représentation via des Post-It de l'enchaînement des événements. Il sert aussi bien sur de la modélisation de processus métier que sur la capture de besoin. L'architecte peut être le facilitateur de l'atelier ou bien un participant de par son positionnement d'expert sur certain sujet. Toutes les discussions partent des faits, des événements puis s'intéressent sur l'enchaînement de ceux-ci et leur source. Le but est de bien définir les responsabilités.

### **Un backlog d'architecture**

La capacité d'adaptation est supportée par un découpage des sujets et par une prise de décision juste à temps. Au même titre qu'il y a un backlog au niveau de chaque équipe, une bonne pratique est d'avoir un backlog au niveau du produit ou de l'entité. Les sujets référencés dans ce backlog visent à préparer les prochaines itérations aussi bien sur des évaluations de nouvelles technologies, de nouvelles opportunités métier que sur la gestion de l'obsolescence et de la dette technologique inhérentes à tout système.

Il est initié lors du cadrage puis mis à jour tout le long de l'exécution jusqu'à l'arrêt du projet ou du produit. Une revue régulière est mise en place avec les équipes et les représentants métier afin de prioriser les éléments de ce backlog.

### **Le partage des décisions : « Architecture Decision Record »**

Comme l'architecture devient la responsabilité de tous, il est important de s'assurer que tout le monde partage les décisions prises aux différents niveaux de l'organisation. La notion de journal de décision est déjà existante pour les architectes. Il s'agit là de diffuser la pratique à l'ensemble des équipes. Le meilleur outil identifié pour l'instant est la pratique de l'« architecture decision record ».

C'est un simple fichier texte respectant une structure simple : titre, contexte, décision, statut, conséquence. Ces décisions sont stockées au niveau des décisionnaires. C'est-à-dire qu'une décision prise par les développeurs se retrouve au niveau des gestionnaires de code source et qu'une décision prise au niveau de l'entité sera stockée dans un outil collaboratif utilisable et compréhensible par tous (wiki, gestion documentaire,...).

La pratique de l'ADR est décrite dans le blog de Michael

Nygaard <http://thinkrelevance.com/blog/2011/11/15/documenting-architecture-decisions>

## 7. Réussir la transition

### 7.1 Les grandes étapes du passage à l'agile

Le déploiement de l'agile@scale n'est pas une finalité et s'inscrit dans une trajectoire d'adaptation des organisations et de transformation digitale.

Nous pouvons distinguer 4 états et modèles types de motivation- transformation agile.

#### 1. Le déploiement de l'agile (Scrum,...) au niveau d'une équipe pluridisciplinaire autonome

*Motivations et contexte SI type : Développement des services client : fonctions ou services « front »*

Le rôle des Architectes d'Entreprise va être d'accompagner le couplage avec le SI et de **donner aux équipes les moyens de décider** en redonnant au besoin des perspectives et contraintes plus larges.

#### 2. Le déploiement de l'agile à l'échelle (SAFe, SPOTIFY,...) dans une logique programme, portefeuille ou entreprise

*Motivations et contexte SI type : Trajectoire incrémentale de transformation SI, accélération de la digitalisation : SI bi-modal, ensemble du SI*

Le rôle des Architectes d'Entreprise va être de **repenser l'architecture** pour qu'elle devienne plus modulaire, ouverte et flexible centrée sur la data, d'accompagner la refonte des processus de l'usine SI, de redéfinir en conséquence sur les exigences non fonctionnelles de haut niveau, de gérer et valoriser les ressources SI, de faciliter le dialogue architecture intentionnelle et architecture émergente, d'animer la transversalité, d'animer les communautés, guildes, d'aider les équipes à naviguer dans les choix d'évolution SI, de sécuriser les ressources cœurs en les opérants au besoin.

#### 3. L'atteinte de l'excellence collaborative et la généralisation de nouveaux modèle de management et d'organisation plus adaptative (Entreprise libérée, holacratie,...)

*Motivations type : Revue des valeurs et du sens de l'organisation, recrutement des talents*

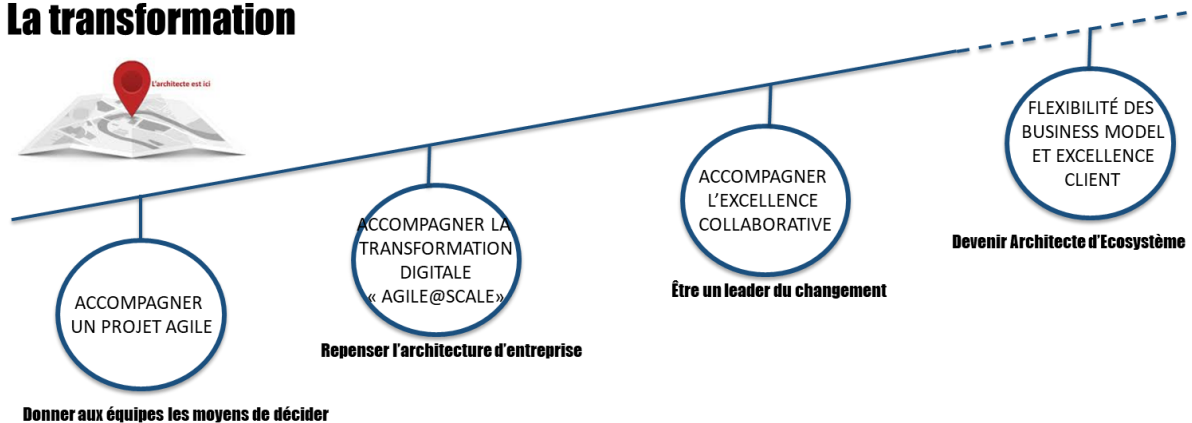
Le rôle des Architectes d'Entreprise va être de **clarifier les rôles et redevabilités sur le SI**, de jouer un rôle leader dans la conduite du changement, d'animer la transversalité, les communautés de pratiques et de mobiliser les parties prenantes sur les décisions à prendre, d'être porteur du sens commun auprès des équipes....

#### 4. La flexibilité accrue des business model et l'excellence client

*Motivations et contexte SI type : Innovation ouverte et captation de valeur*

Le rôle de certains Architectes d'Entreprise va être de devenir des **Architectes d'Ecosystèmes** (interne ou externe), de savoir appréhender la complexité et la volatilité es situations chaotiques, d'aider aux réflexions stratégiques, d'être acteur de l'innovation ouverte et de la mobilisation des talents.

## La transformation



## 7.2 Les modèles top-down et bottom-up

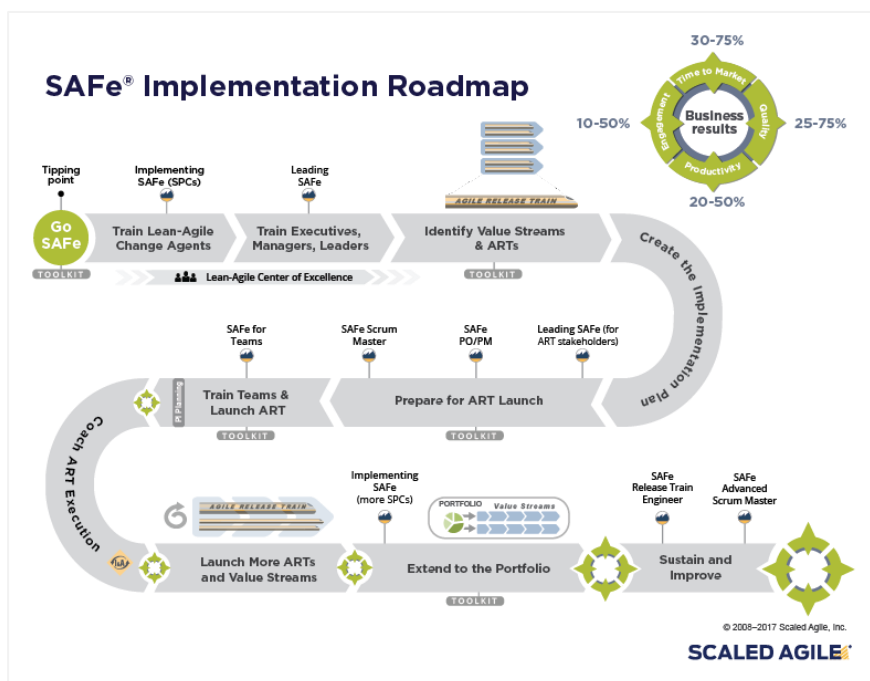
Pour le passage à l'échelle, deux grands modèles émergent :

- Top-Down comme dans SAgile avec un modèle d'implémentation prescriptif
- Bottom-Up qui se base sur une transformation des pratiques

SAFe fournit les grandes lignes directrices pour son adoption structurée autour de 12 points.

L'articulation se fait sur quatre phases sans compter la phase de cadrage :

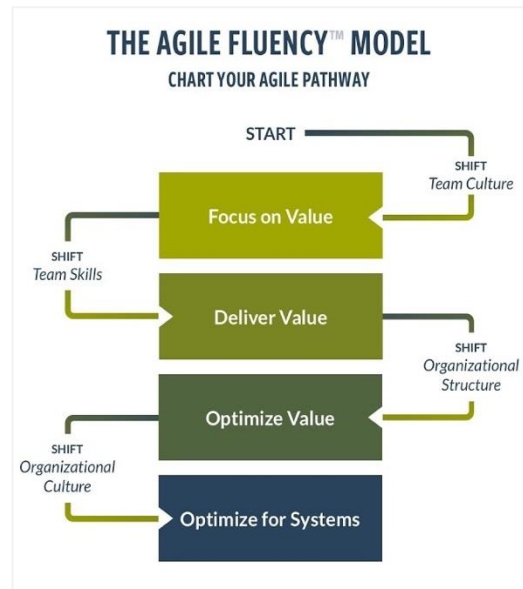
- Préparation de la transformation
- Lancement d'un premier Release Train
- Extension sur d'autres chaînes
- Amélioration continue et maintien des pratiques





- Privilégier un déploiement en mode bottom-up de l'agile au niveau des projets avant de passer à l'échelle en mode top-down
- Conduire des enquêtes de satisfaction auprès des équipes

Le modèle « Agile Fluency » propose dans les premières phases de travailler au niveau de l'équipe mais permet aussi de voir quel niveau est le plus pertinent selon les objectifs que l'entreprise souhaite atteindre.



Nous retrouvons dans le même pattern dans ces conduites des transformations :

- Une phase d'apprentissage formation des équipes et/ou des managers
- Une phase d'extension de pratique
- Une phase d'amélioration continue

C'est résumé dans les pratiques Agile par le Shu-Ha-Ri issu des arts martiaux.

## 7.3 Transformer les cellules d'Architecture

Quels que soient le modèle de transformation et le framework adoptés, l'impact sur la cellule d'architecture présente des éléments communs.

Comme toute transformation, les personnes, les processus et les outils doivent être pris en compte.

Sur les personnes en plus de nouvelles pratiques à intégrer, il faut prendre en compte une évolution des compétences comportementales. Les techniques d'influence, de coaching, de facilitation et de communication doivent être acquises par les architectes :

- Le coaching pour embarquer les équipes sur de nouvelles approches technologiques et les sensibiliser aux enjeux d'architecture (exigences non fonctionnelles)
- L'influence essentiellement sur la priorisation des sujets et la construction de la vision
- La facilitation pour la résolution de problèmes et l'identification des valeurs.

- La communication pour l'animation des communautés et le partage de la vision

Sur les processus, les phases de revues et les points d'interactions avec les équipes doivent être redéfinis. Globalement les phases de cadrage sont réduites dans le temps, les revues sont itératives au niveau de chaque sprint ou groupe de sprint, si une fonctionnalité le nécessite, et les retours sont collectés au niveau des rétrospectives.

La décentralisation des décisions et la co-construction des livrables impliquent que les outils de modélisations et de collecte des informations soient collaboratifs. L'automatisation doit être possible car le nombre de composants est en augmentation et les approches déclaratives sont compliquées à maintenir dans des organisations fragmentées.



#### : Respecter les 7 principes de l'architecture agile SAFe

*Traduction libre des 7 principes par le groupe de travail du Club URBA EA :*

1. La conception émerge, l'architecture est une collaboration
2. Plus le système est complexe, plus la piste d'architecture est longue
3. Construire l'architecture la plus simple qui puisse fonctionner
4. En cas de doute, expérimentez ou modélisez l'architecture
5. Celui/celle qui développe est celui/celle qui teste
6. Pas de monopole de l'innovation
7. Implémentez l'architecture en logique flux

## 7.4 Evolution du rôle des architectes

Comme évoqué dans la transformation de la cellule d'architecture les architectes doivent acquérir de nouvelles compétences et de nouveaux réflexes de par l'évolution de leur rôle.

Une partie du travail de l'architecte est dans une organisation agile déléguée aux équipes ou automatisée. L'architecte se concentrait sur les décisions majeures des projets et veiller à la cohérence du système les d'information.

Dans la phase de transformation, l'architecte doit sensibiliser les équipes aux enjeux de l'architecture. Les équipes doivent prendre conscience de l'écosystème les environnants et ne pas s'intéresser uniquement à la structure des classes de leur programme. L'architecte doit donc coacher les membres des équipes pour qu'ils gagnent en autonomie sans déstabiliser le système existant.

L'image de l'« architect elevator » de Gregor Hohpe propose de nouvelles missions pour les architectes allant de la salle machine en regardant les automatisations de déploiement, les nouvelles infrastructures (cloud) jusqu'au dernier étage de la tour pour expliquer les options d'architecture à la direction. Décrit ainsi, l'architecte est un facilitateur en plus de son domaine d'expertise.

Les équipes d'architectures devront apprendre :

- à interagir avec toutes les **parties prenantes**

- Management : Vision stratégique
- Equipes digitales : Crédibilité et légitimité, influence
- Comment étendre son champ de compétences : stratégie, technologique, méthodologique, ...
- Quelles « Soft skills » : Assertivité, mindset agile, leadership
- à intervenir dans **toutes les phases**, à tout moment, sur tous les sujets
  - Innovation : quel problème résoudre ? quelle opportunité ?
  - Conception : quelle décision prendre (le plus tardivement possible) ?
  - Vie des systèmes : comment organiser les boucles retours ?



Après une phase plus libre, on préfère maintenant respecter les rôles des différents niveaux d'architectes définis dans SAFe

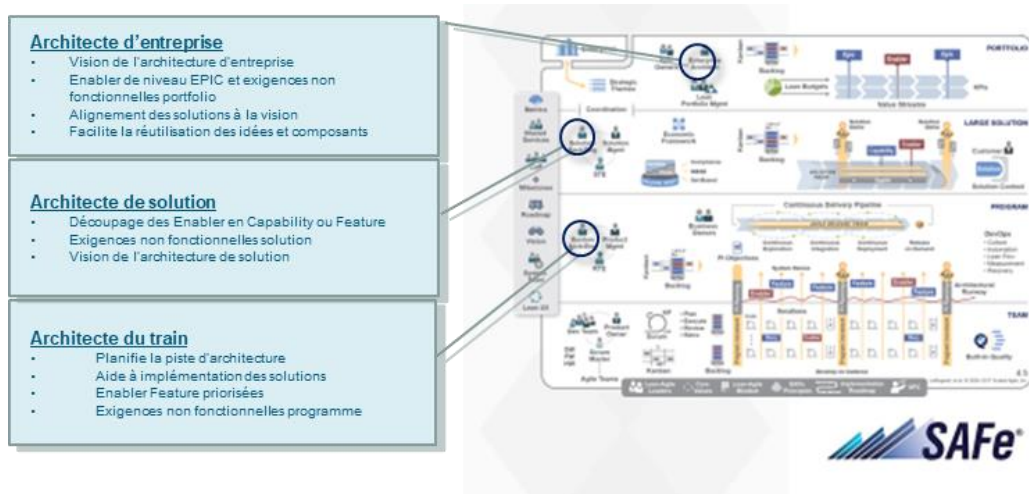


Figure 22 Rôle des architectes dans SAFe



#### Nouveau rôle de l'architecte

- Il incarne l'architecture intentionnel
- Il comprend les besoins métiers présents et futurs
- Il définit la piste d'architecture
- Il porte la vue générale de l'architecture jusqu'aux équipes et encourage l'alignement
- Il aide les équipes à définir et implémenter des solutions adaptées, tout en garantissant la couverture des exigences non fonctionnelles
- Il s'assure que la solution est intégrée et testée fréquemment, y compris les exigences non fonctionnelles

#### Retour d'expérience



- Privilégier des profils techniques avec forte culture agile
- Jouer la carte des profils rares et atypique



## Conclusion

---

- *L'architecture d'entreprise est essentielle*

L'Architecture d'Entreprise apporte des réponses aux défis de :

- **Agilité stratégique**
- **Vitesse d'exécution**
- **Résilience**
- **Collaboration globale**

- *Le défi de la confiance*

La légitimité ne vient pas du positionnement hiérarchique ou du titre mais se gagne sur le terrain.

- *De nouveaux fondamentaux d'architecture SI*

La révolution digitale et le déploiement de l'agile@scale nécessite de revoir les fondamentaux SI, de créer des architectures plus flexibles, vivantes et résilientes, de développer des modèles plateformes,...

- *Une évolution profonde du métier d'architecte d'entreprise*

*Contribuer davantage aux visions et à la stratégie d'entreprise*

- *Clarifier les visions ou les problèmes à résoudre*
- *Faire émerger les chemins de construction*
- *Décrypter les tendances et opportunités technologiques*
- *Aider à naviguer dans un monde Volatile, Incertain, Complexe et Ambiguë*

*Passer d'une gestion du patrimoine applicatif à une valorisation produit*

- *Promouvoir un langage commun et aider à la structuration des produits et des équipes*
- *Penser le SI comme un ensemble de ressources (interne externe) avec la donnée au cœur*
- *Intervenir en continue en tant que design authority d'architecture vivante*
- *Développer des compétences de designer en mettant le client au centre de tout : Valeur à 3 mois*
- *Opérer, en mode service, des socles et référentiels*

*Revoir ses réflexes d'architecte*

- *Se concentrer sur les caractéristiques architecturales*
- *Préférer la flexibilité à la reproductibilité*
- *Retarder les décisions structurantes*
- *Appréhender les ruptures : micro-services, blockchains, cloud, IA, data, networks,...*

*Être un acteur de l'entreprise apprenante*

- *Devenir coah et mentor du changement*
- *Construire les boucles d'interactions pertinentes et aider, en pratique, les équipes à naviguer*
- *Développer la connaissance et les compétences*

*Raisonner écosystème ouvert*

- *Comprendre les chaînes de valeur*
- *Contribuer à l'innovation client et business en captant les capacités d'innovation du marché*
- *Souveraineté et architecture d'influence*

#### *Digitaliser la fonction*

- *Simplifier le processus d'architecture et se débarrasser des gaspillages*
- *Développer des outils collaboratifs de partage et de décision*
- *Automatiser les outils de gouvernance et de pilotage du SI : Algorithmes d'architecture d'entreprise, automatisation des tests d'architecture,...*

- ***L'importance des valeurs et de l'état d'esprit agile***

Au-delà de son apport d'expertise, l'Architectes d'Entreprise doit intégrer et incarner les valeurs agile :

- **Les individus et leurs interactions** plus que les processus et les outils.
- **Un logiciel qui fonctionne** plus qu'une documentation exhaustive.
- **La collaboration avec les clients** plus que la négociation contractuelle.
- **L'adaptation au changement** plus que le suivi d'un plan.

L'architecture est l'affaire de tous, tout le temps.

Une bonne architecture est une architecture qui est comprise et fonctionne.

- ***Les Architectes ont tout pour réussir et réussissent déjà !***

La place donnée à l'architecture dans les frameworks agiles, la littérature abondante sur la transformation des architectures et du métier d'architecte, la reconnaissance et légitimité des architectes que nous avons pu constater lors des explorations dans les entreprises avancées en agile@scale **nous rendent enthousiastes et optimistes sur le devenir de l'Architecture d'Entreprise.**

## REJOIGNEZ LE CLUB URBA EA ET LE GROUPE POUR 2019

L'année 2018 a été une année d'apprentissage et d'échange intense sur l'adaptation de l'Architecture d'Entreprise à l'agile@scale.

**Un grand merci aux membres du Club URBA EA participants et aux entreprises qui nous ont accueillies pour leur esprit d'ouverture « agile ».**

De nombreux sujets soulevés cette année restent à approfondir en 2019. Si vous souhaitez contribuer, participer aux travaux merci de contacter Nicolas Chevalier, consultant Gluendo animateur du thème agile au sein du Club URBA EA.

Coordonnées : [nicolas.chevalier@gluendo.com](mailto:nicolas.chevalier@gluendo.com) - 07 68 91 14 56

## Annexe : Enquête conduite en décembre 2018

Notre enquête a été soumise aux membres du Club Urba et du CIGREF entre les mois d'octobre et de décembre 2018. L'objectif de l'enquête était de :

- Situer le contexte du répondant (entreprise, SI)
- Comprendre le niveau de maturité des pratiques d'agile à l'échelle
- Évaluer l'impact de ces pratiques sur l'architecture
- Identifier les nouvelles pratiques et livrables d'architecture à mettre en œuvre

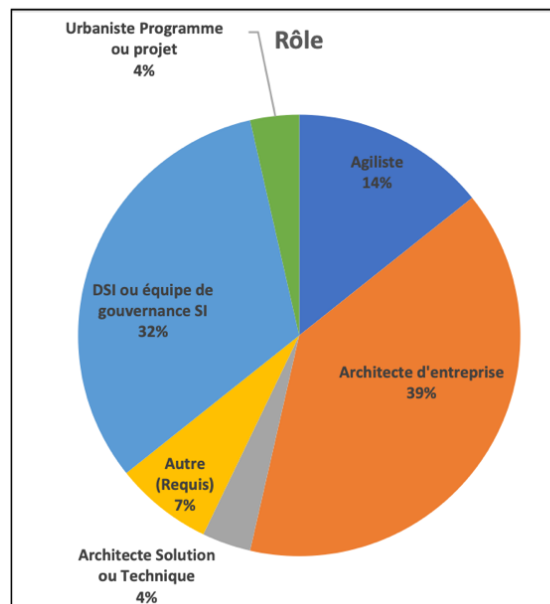
Voici la synthèse des réponses aux questions :

### 1. Quel est votre rôle (titre) dans l'organisation ?

(L'astérisque indique une question à réponse obligatoire) \*

- ☐ DSI ou équipe de gouvernance SI
- ☐ Architecte d'entreprise
- ☐ Urbaniste Programme ou projet
- ☐ Architecte Solution ou Technique
- ☐ Agiliste
- ☐ Autre (Requis)

### 28 répondants



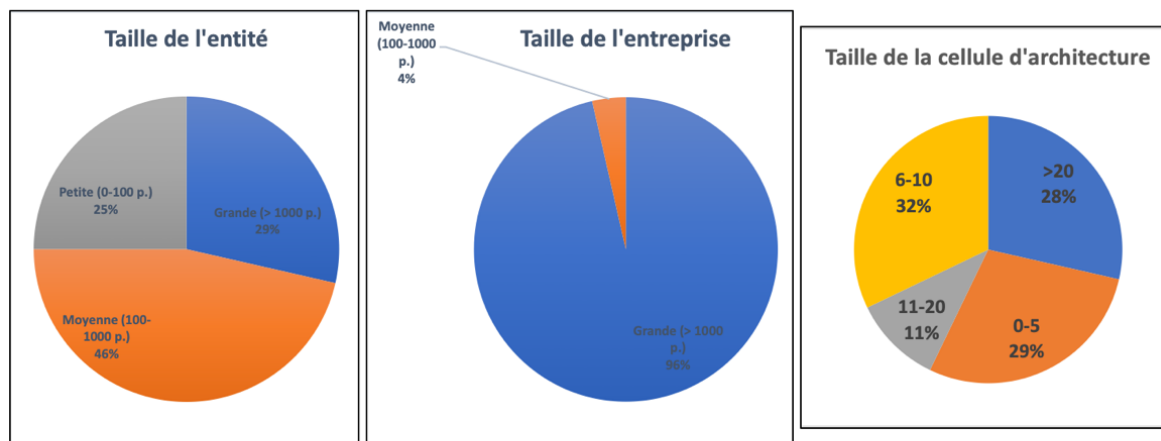
Principalement des architectes d'entreprise et des managers SI

## 2. Comment situez-vous votre organisation sur ces différentes échelles ? \*

	Petite (0-100 p.)	Moyenne (100-1000 p.)	Grande (> 1000 p.)
Taille de votre entité de rattachement	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Taille de l'entreprise	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 3. Taille de la cellule d'architecture (nombre d'architectes) \*

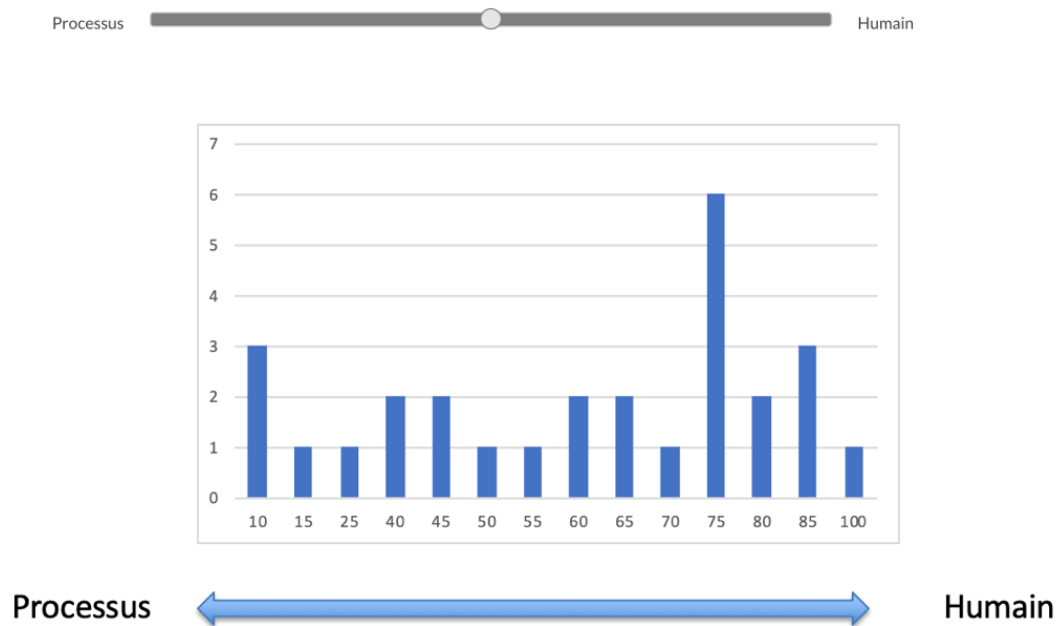
☐ 0-5   ☐ 6-10   ☐ 11-20   ☐ >20



Une grande entreprise, mais une équipe d'architecture à taille humaine

4. Le fonctionnement de votre entité est-il basé sur des procédures bien établies ou plutôt sur l'engagement personnel de chacun ? \*

*Vous devez interagir avec la jauge pour que votre réponse soit valide*



Un fonctionnement de l'organisation plutôt informel

5. Taille de votre SI (nombre d'applications dans le patrimoine SI) \*

☐ 0-50 ☐ 51-100 ☐ 101-500 ☐ > 500

6. Complexité de votre SI \*

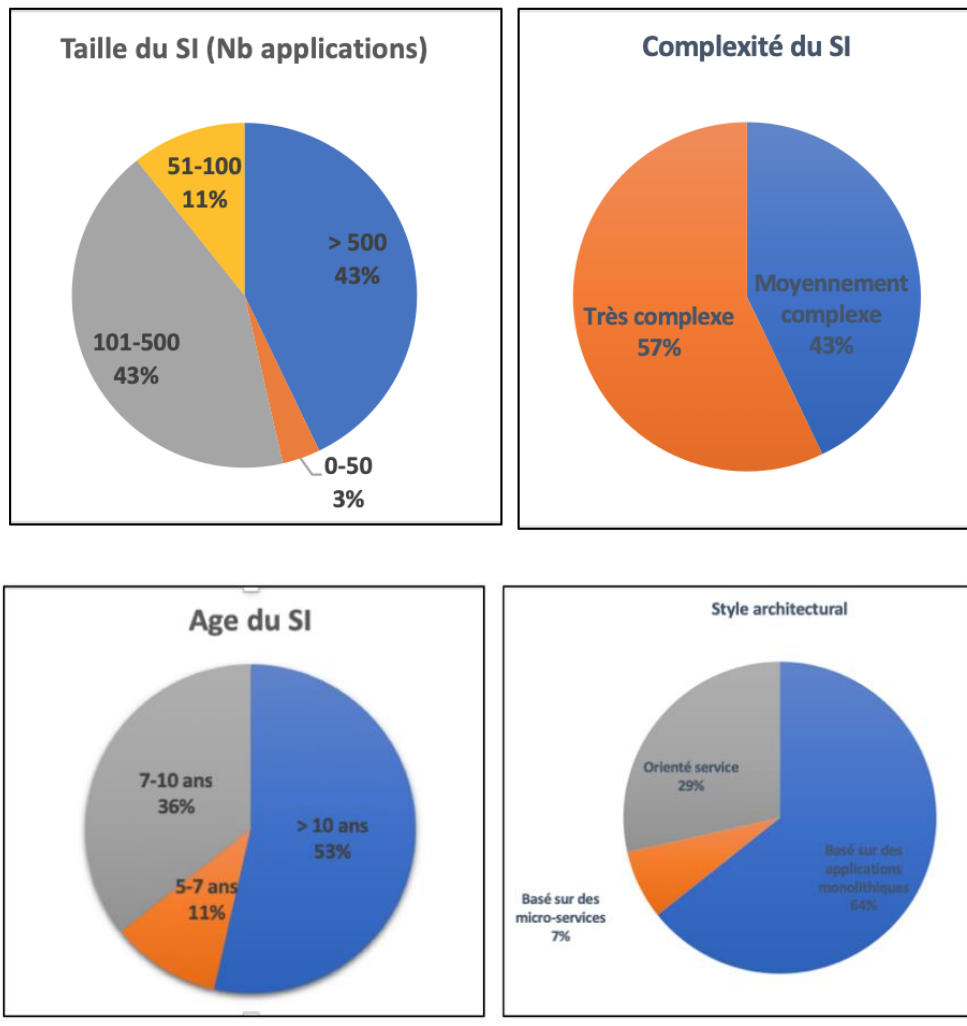
☐ Simple ☐ Peu complexe ☐ Moyennement complexe ☐ Très complexe

7. Age de votre SI (estimation de l'âge moyen des composants de votre SI) \*

☐ > 10 ans ☐ 7-10 ans ☐ 5-7 ans ☐ 2-4 ans ☐ < 2 ans

8. Le style architectural de votre SI est majoritairement : \*

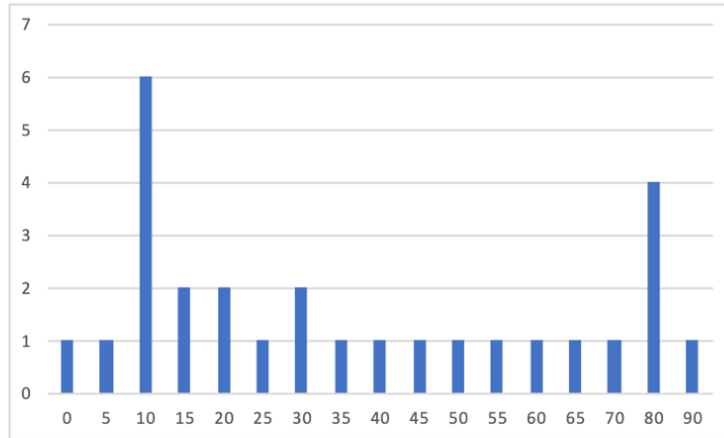
☐ basé sur des applications monolithiques ☐ orienté service ☐ basé sur des micro-services



Un SI de taille importante et très complexe, plutôt ancien et d'architecture classique

## 9. Niveau d'adoption de l'agile (part des projets de votre entité faits en agile en 2018) \*

0 %  100 %



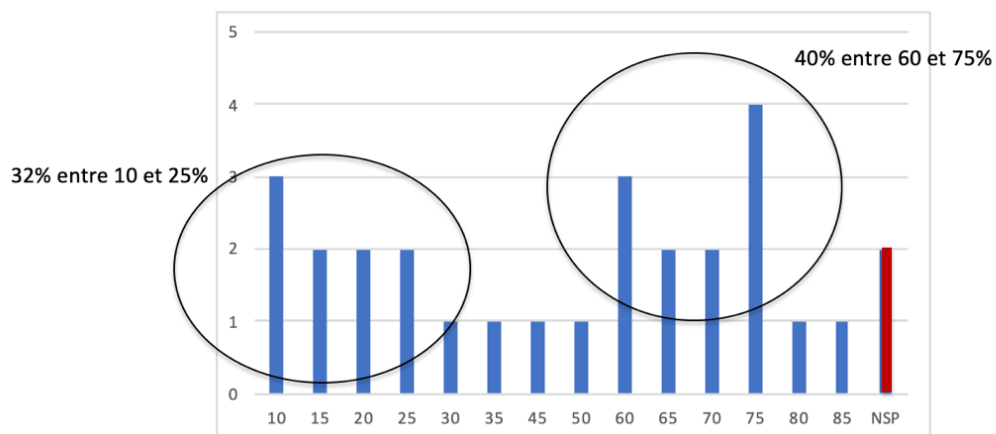
0 %  100 %

Un niveau d'adoption de l'agile encore faible, avec quelques pionniers ...

10. A quel degré avez-vous adopté la culture agile au niveau des pratiques de management ?  
 (L'agile préconise la délégation, l'autonomie et la responsabilisation des équipes)

*Vous pouvez ajouter un commentaire (optionnel)*

Contrôle  Délégation



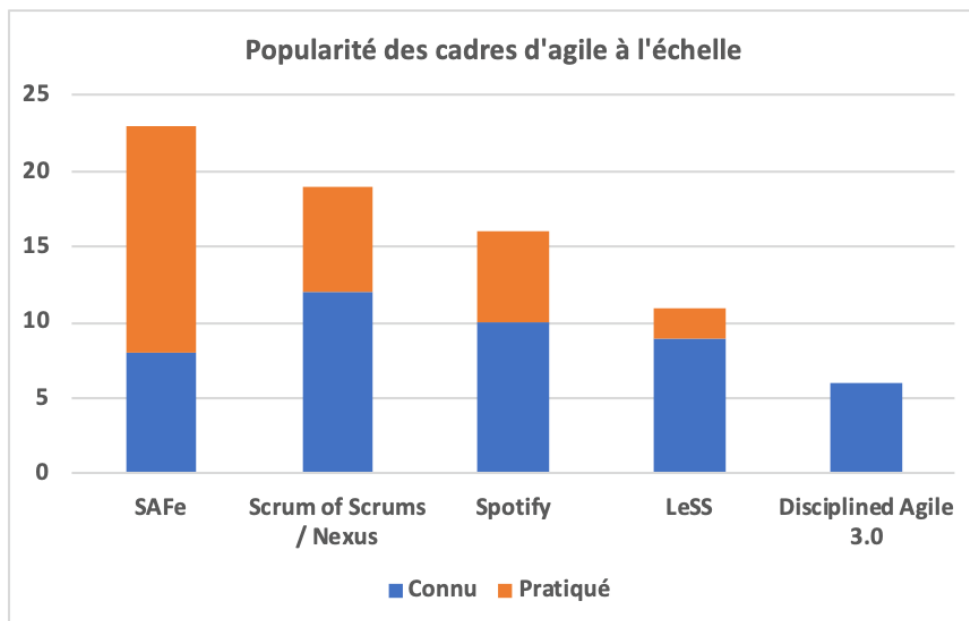
Contrôle  Délégation

Des résultats contrastés sur la culture agile au niveau des pratiques de management

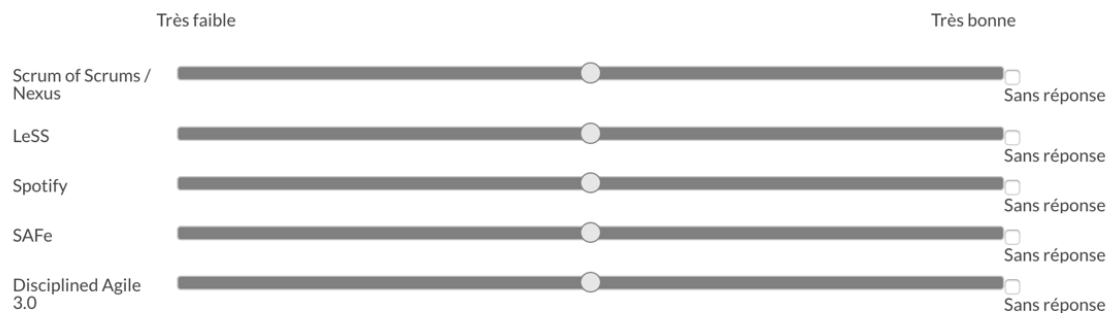


## 11. Quel(s) cadre(s) d'agile à l'échelle connaissez-vous / pratiquez-vous ? \*

	Connaissez-vous ?	Pratiquez-vous ?	Ni l'un, ni l'autre
Scrum of Scrums / Nexus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
LeSS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Spotify	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SAFe	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Disciplined Agile 3.0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Autre <input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



## 12. Pour les cadres que vous connaissez/pratiquez, comment évaluez-vous la capacité de chaque cadre à donner sa bonne place à l'architecture ? \*



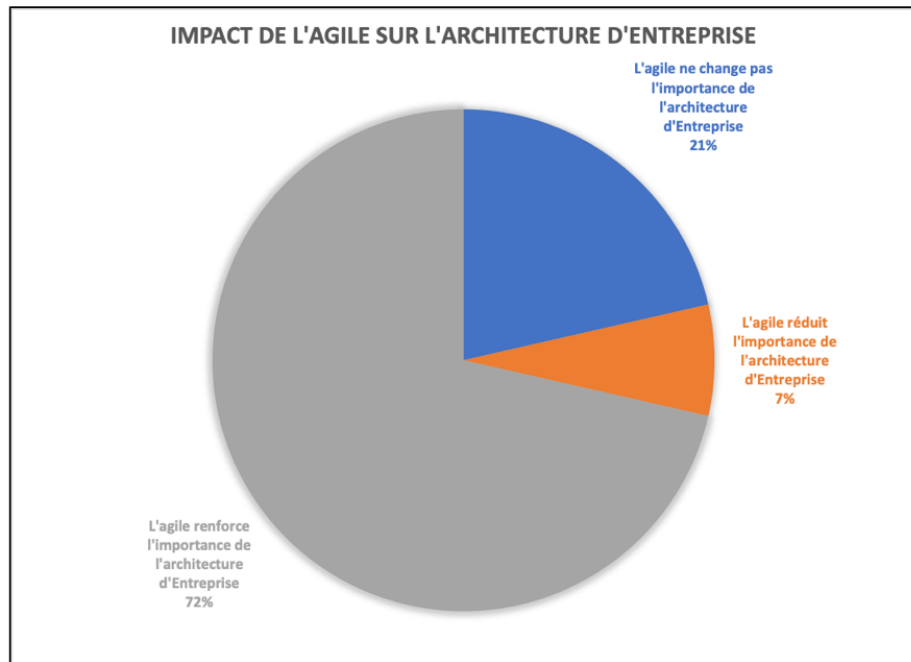
## Capacité à donner sa bonne place à l'architecture

	Très faible ← Moyenne → Très bonne	
<u>Scrum of Scrums</u> / Nexus	(31)	Nb de réponses 6
<u>LeSS</u>	(35)	4
<u>Spotify</u>	(50)	6
<u>SAFe</u>	(68)	8
<u>Disciplined Agile 3.0</u>	(45)	3

SAFe plébiscité pour l'architecture ...

13. Selon vous, quel est l'impact de l'agile sur les activités d'architecture d'entreprise ? \*

- ☐ L'agile réduit l'importance de l'architecture d'Entreprise
- ☐ L'agile renforce l'importance de l'architecture d'Entreprise
- ☐ L'agile ne change pas l'importance de l'architecture d'Entreprise



L'agile renforce l'importance de l'architecture pour la grande majorité des répondants

## Impact de l'agile sur l'architecture d'entreprise : Commentaires

La place de l'architecture d'entreprise est modifiée avec l'agile dans le sens où les moteurs de la transformation : Evolutivité et Besoin de Transversalité sont clairement alimentés par les démarches agiles.

ca devient un asset de tout le monde

L'architecture revêt plus d'importance que les produits sont davantage gérés en tout autonomie, avec un risque de négliger la transversalité et la vision long terme

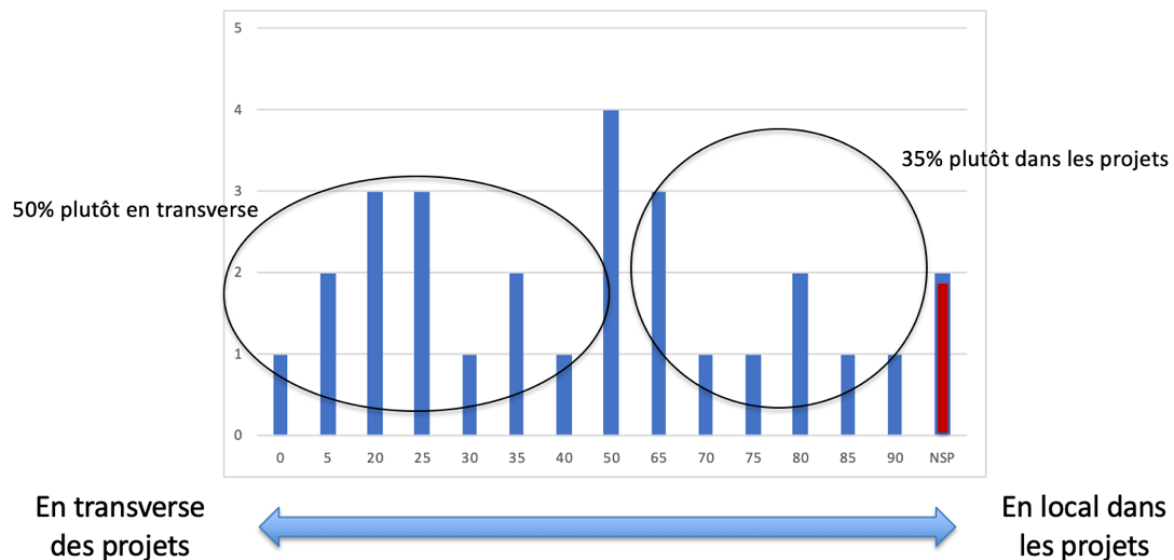
L'agile modifie la façon d'opérer des architectes d'entreprise. Ils doivent être davantage intégrés dans les équipes projets et anticiper les « enablers » d'architecture selon la roadmap des projets. L'architecture d'entreprise devient davantage itérative.

Mais il implique les experts directement dans les équipes de développement et ne fige pas l'architecture dès le début des projets

14. Selon vous, où les décisions d'architecture sont-elles prises (ou doivent-elles être prises) dans les projets agiles ? \*



## Où les décisions d'architecture sont-elles prises ?



## Où décider : Commentaires

Les décisions d'architecture ne sont pas toutes avec le même impact (composant, système, organisation, entreprise). Si on parle du projet de fabrication d'un (seul) produit il s'agit forcément du projet qui décide.

2 axes sont à prendre en compte la stratégie et le design donc les décisions sont réparties en fonction du niveau

On peut considérer des décisions à deux niveaux selon l'envergure des projets et selon l'état du SI existant au sein de l'entreprise.

d'abord dans le projet en se basant sur l'architecture intentionnelle puis en transverse pour les décisions du niveau entreprise

Tout est dans l'équilibre entre les deux : architectures intentionnelle et architecture émergente

un bon début serait d'arrêter de parler de projet ;-).

Elles devraient être prise en transverse avec une part locale

15. Selon vous, où les décisions d'architecture sont-elles prises (ou doivent-elles être prises) dans les projets agiles ? \*

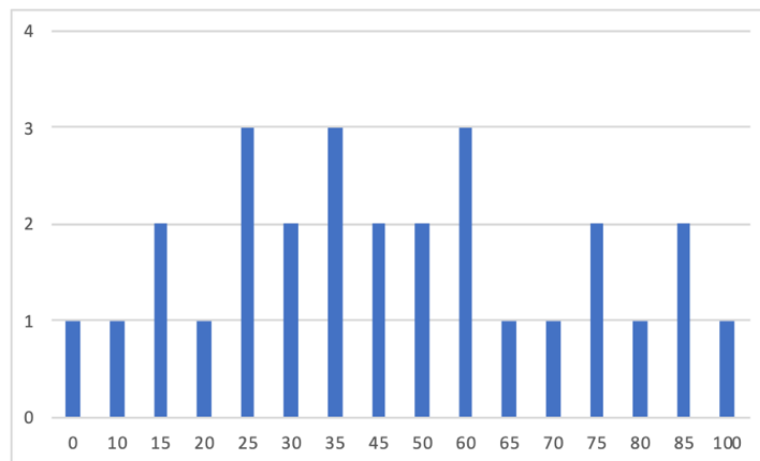
en amont des projets



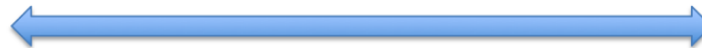
au fil de l'eau  
pendant le projet

☐ Sans réponse

## Quand les décisions d'architecture sont-elles prises ?



En amont des  
projets



Au fil de l'eau  
pendant le projet

Aucune tendance forte ne se dégage ...

### Quand décider : Commentaires

Les décisions d'architecture ne sont pas toutes avec le même impact (composant, système, organisation, entreprise). Si on parle du projet de fabrication d'un (seul) produit il s'agit forcément du projet qui décide.

Ces décisions doivent, pour la plupart, redescendre au niveau des équipes

A l'échelle, il est nécessaire de prendre les décisions le plus tard possible et de laisser des options ouvertes pour optimiser les choix économiques

16. \*

Quels livrables historiques d'architecture doit-on continuer à produire ? à l'identique, différemment ? Quels livrables avez-vous arrêté ou pensez-vous arrêter de produire ?

	Continuer à ne pas produire	Commencer à produire	Continuer à produire comme avant	Continuer à produire mais différemment	Arrêter de produire	Je ne sais pas
Schéma directeur pluri-annuel	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Cartographie fonctionnelle, POS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Cartographie applicative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Catalogue de services	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Dossier de cadrage projet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Principes et règles d'architecture	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Patterns d'architecture	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Modèle de données	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Business Model	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Cartographie de processus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Normes & Standards	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Dossier d'architecture fonctionnel	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Dossier d'architecture technique	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Backlog d'architecture	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Log des décisions d'architecture	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Autre	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

### Impact sur les principaux livrables d'architecture

	Continuer à produire mais différemment	Continuer à produire comme avant	Commencer à produire	Arrêter de produire
Schéma directeur pluri-annuel	9	1	1	1
Cartographie fonctionnelle, POS	11	1	1	0
Cartographie applicative	8	3	0	1
Catalogue de services	5	3	5	0
Dossier de cadrage projet	10	2	0	0
Principes et règles d'architecture	7	5	0	0
Patterns d'architecture	8	3	0	0
Modèle de données	11	1	0	0
Business Model	7	0	2	0
Cartographie de processus	7	1	3	2
Normes & Standards	6	4	2	1
Dossier d'architecture fonctionnel	8	2	0	1
Dossier d'architecture technique	7	2	1	1
Backlog d'architecture	3	1	8	0
Log des décisions d'architecture	2	1	9	0
Suivi des dettes d'architecture	0	0	1	0

Les livrables restent les mêmes mais sont produits différemment ...  
 ... et deux livrables apparaissent : backlog et log des décisions d'architecture

17. Indiquez votre niveau d'accord avec chacune des assertions sur l'évolution du rôle d'architecte ci-dessous : \*

	Pas du tout d'accord	Plutôt pas d'accord	Plutôt d'accord	Tout à fait d'accord
Les architectes vont déléguer davantage les travaux d'architecture aux équipes d'exécution	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Les architectes vont devoir intégrer les équipes d'exécution pour prendre en charge les travaux d'architecture	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Il faut animer des référents/correspondants sur l'architecture dans les équipes agiles	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La phase de cadrage doit être réduite	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Les concepts de principes, normes et standards d'architecture deviennent caduques	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Il faut des principes d'architecture moins nombreux et de plus haut niveau	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Les standards d'architecture doivent émerger des équipes d'exécution et être partagés	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Les architectes doivent gagner en expertise technique	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La veille et le benchmark deviennent plus importants que l'analyse des besoins métiers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Il est nécessaire de sanctuariser des budgets sur les socles SI	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
L'alignement stratégique devient une activité secondaire	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	Tout à fait d'accord	Plutôt d'accord	Plutôt pas d'accord	Pas du tout d'accord
<b>Les architectes vont déléguer davantage les travaux d'architecture aux équipes d'exécution</b>	2	8	2	1
<b>Les architectes vont devoir intégrer les équipes d'exécution pour prendre en charge les travaux d'architecture</b>	2	10	1	0
<b>Il faut animer des référents/correspondants sur l'architecture dans les équipes agiles</b>	12	1	0	0
<b>La phase de cadrage doit être réduite</b>	4	5	4	0
<b>Les concepts de principes, normes et standards d'architecture deviennent caduques</b>	0	2	5	6
<b>Il faut des principes d'architecture moins nombreux et de plus haut niveau</b>	4	8	1	0
<b>Les standards d'architecture doivent émerger des équipes d'exécution et être partagés</b>	2	8	3	0
<b>Les architectes doivent gagner en expertise technique</b>	4	5	4	0
<b>La veille et le benchmark deviennent plus importants que l'analyse des besoins métiers</b>	1	1	10	1
<b>Il est nécessaire de sanctuariser des budgets sur les socles SI</b>	7	6	0	0
<b>L'alignement stratégique devient une activité secondaire</b>	0	1	7	5

## Bibliographie et référence

---

### Introduction

Article de McKinsey d'août 2016 " How Enterprise Architects can help ensure success with digital transformations ", (<https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/how-enterprise-architects-can-help-ensure-success-with-digital-transformations>.),

### Panorama des cadres d'agile@scale

#### Historique de l'agile

Manifeste agile : <https://agilemanifesto.org/>

Lean Management : "The Toyota Way" , J. Liker, 2004

"The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses", Eric Ries, 2011

Design Thinking : "Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation", T. Brown, 2009

#### SCRUM

M. Beedle et K. Schwaber « Agile Software Development with SCRUM »

« Agile can scale : Inventing and Reinventing SCRUM in five companies » J. Sutherland

The Scrum Guide ([scrum.org](https://www.scrum.org))

#### Nexus

The Nexus Guide

#### LeSS

« Scaling Lean and Agile Development »

« Large-Scale Scrum : More with LeSS »

LeSS Website : <https://less.works/>



## **Spotify**

Spotify Engineering culture blog and videos – H. Kniberg

<https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/>

<https://labs.spotify.com/2014/09/20/spotify-engineering-culture-part-2/>

## **SAFe®**

<http://www.scaledagileframework.com>.

SAFe® 4.6 Introduction : Overview of the Scaled Agile Framework® for Lean Enterprises , D. Leffingwell

## **DAD**

« Disciplined Agile Delivery , a Practitioner's Guide to Agile Software Delivery in the Enterprise », S. Ambler, M. Lines, IBM Press, 2012.

<https://www.disciplinedagiledelivery.com/>

## **Autres approches**

### **Openspace agility**

<http://openspaceagility.com/>

### **FAST**

The FAST Guide

<http://www.fast-agile.com/>

<http://modernagile.org/>

### **RAGE**

<https://www.cprime.com/rage/>

### **ScALeD**

<http://scaledprinciples.org/>

### **XSCALE**

<https://xscaalliance.org/> et <https://xsca.wiki/>

*"Continuous Architecture : Sustainable Architecture in an Agile and Cloud-Centric World"*, Murat Erder, Pierre Pureur, Elsevier 2016

*"Building Evolutionary Architectures : Support Constant Change"*, Neal Ford, Rebecca Parsons & Patrick Kua, O'Reilly, 2017

*"Agile Architecture in the Digital Age"*, Open Group, July 2018

A White Paper by:

Hervé Barbazange, Jean-Marc Bunouf, Jean-Pierre Le, Jérôme Régnier, Société Générale

Peter Beijer, Carl Kinson, Frédéric Lé, DXC Technology

Antoine Lonjon, MEGA International

*"The Digital Practitioner Body of Knowledge™"*, Open Group

Pragmatic Architecture

IEEE Software Magazine

*"Collaborative Enterprise Architecture : Enriching EA with Lean, Agile and Enterprise 2.0 practices"*, Stefan Bente, Uwe Bombosch, Shailandra Langade, Elsevier 2012

<https://fr.slideshare.net/ClubUrbaEA/en-ea-rapport-slideshare>

*Architecture d'Entreprise et transformation numérique – Daniel Breton, Nicolas Chevalier, 2014*

René Mandel - De la stratégie business aux systèmes d'information : l'entreprise et son écosystème-  
Dunod- 2006