

Premières rencontres BUSINESS AGILITY - 22 mai 2019

Agilité & Architecture Système

Deux alliés pour un développement de produit efficient

Benjamin THOMAS, PMP

INTRODUCTION



In scope / Out of scope

Agilité

&

Architecture Système

Développement hardware

PDCA

Waterfall

TOC

Product innovation

LEAN

DevOps

DMAIC

SAFe

Agile Marketing

SIX-SIGMA

SW frameworks

Les enjeux du développement de (nouveaux) produits

« Better , Faster , Cheaper , Happier ! »

Se traduit dans nos projets par :

→ « Centrage client »

- maximisation de la valeur perçue

→ « Anticipation, Planification »

- rendre déterministe un processus divergent (culture projet)

→ « Réactivité »

- réagir vite et bien face aux changements

→ « Amélioration permanente »

- gagner sans cesse en efficience, innover

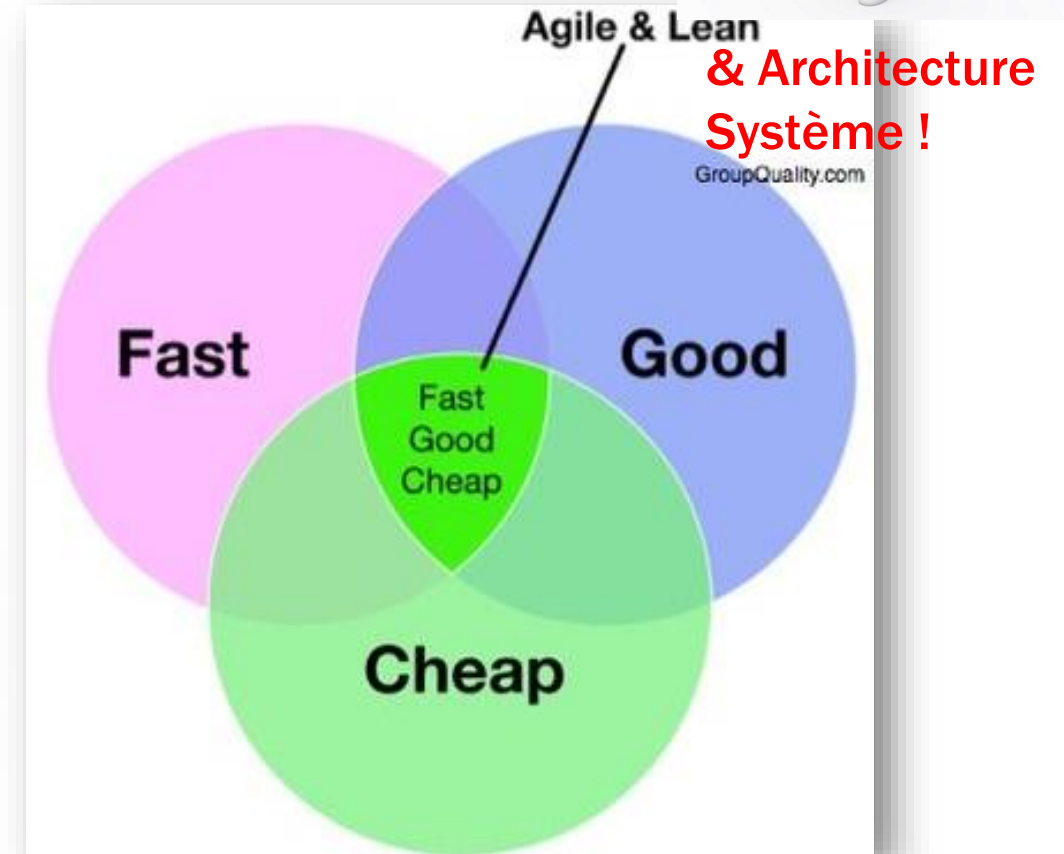


Fast, Good, and Cheap :

Y'a-t'il une « solution réelle » à cette équation ?

...ou bien seulement une solution imaginaire ?

- Voyons comment l'Architecture Système peut aider à mieux définir l'objectif et à « optimiser la trajectoire »
- Voyons comment les valeurs et principes de l'Agilité peuvent apporter **productivité, réactivité, créativité** ; ingrédients essentiels pour la réussite des projets de tous horizons.



CAS DES PROJETS DE DEVELOPPEMENT HARDWARE

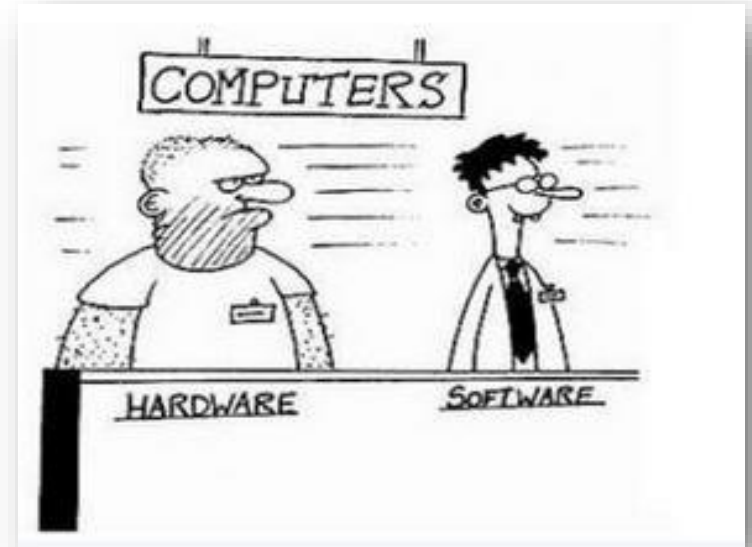


HARDWARE vs SOFTWARE

Certaines contraintes sont inhérentes au processus de conception-développement-industrialisation des objets matériels destinés à être « fabriqués » en série.

MATERIEL

- ❑ Coût de modification qui augmente drastiquement au fur et à mesure du projet (ex. coût d'un retrofit terrain !)
- ❑ Impossibilité d'ajouter des fonctions non prévues au stade de la conception
- ❑ Processus de validation-qualification long et mobilisant des moyens onéreux
- ❑ Longs délais incompressibles de certaines étapes (approvisionnement, fabrication ,...)



LOGICIEL

- ❑ Coût de modification relativement stable au cours du projet (malléabilité)
- ❑ Possibilité d'évolution par agrégation de nouvelles fonctions, pas nécessairement prévues au départ
- ❑ Possibilité d'optimiser et automatiser le processus de test et validation (quelques dizaines de secondes !)

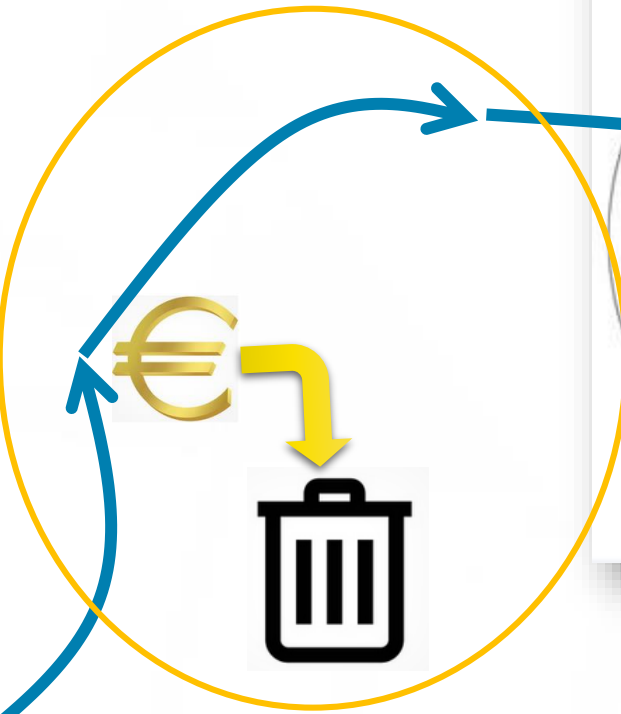
Cas des projets de conception-développement hardware

Plan driven vs. Change driven

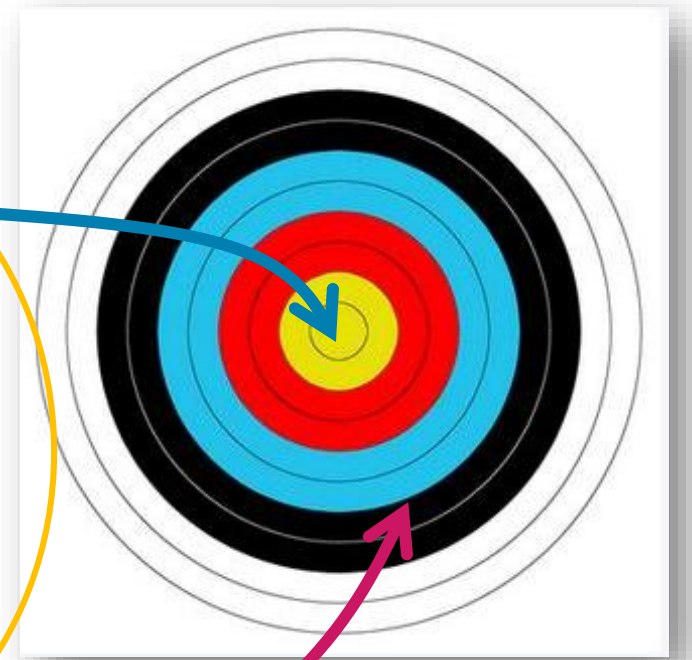
Sur les projets de développement matériel, les changements de cap passés la phase de conception préliminaire (choix d'architecture et de solutions), sont souvent « douloureux » !

=> Cause fréquente de dépassement de budget & délais sur les projets HW

Comment éviter cet écueil, ou au moins en minimiser le risque ?



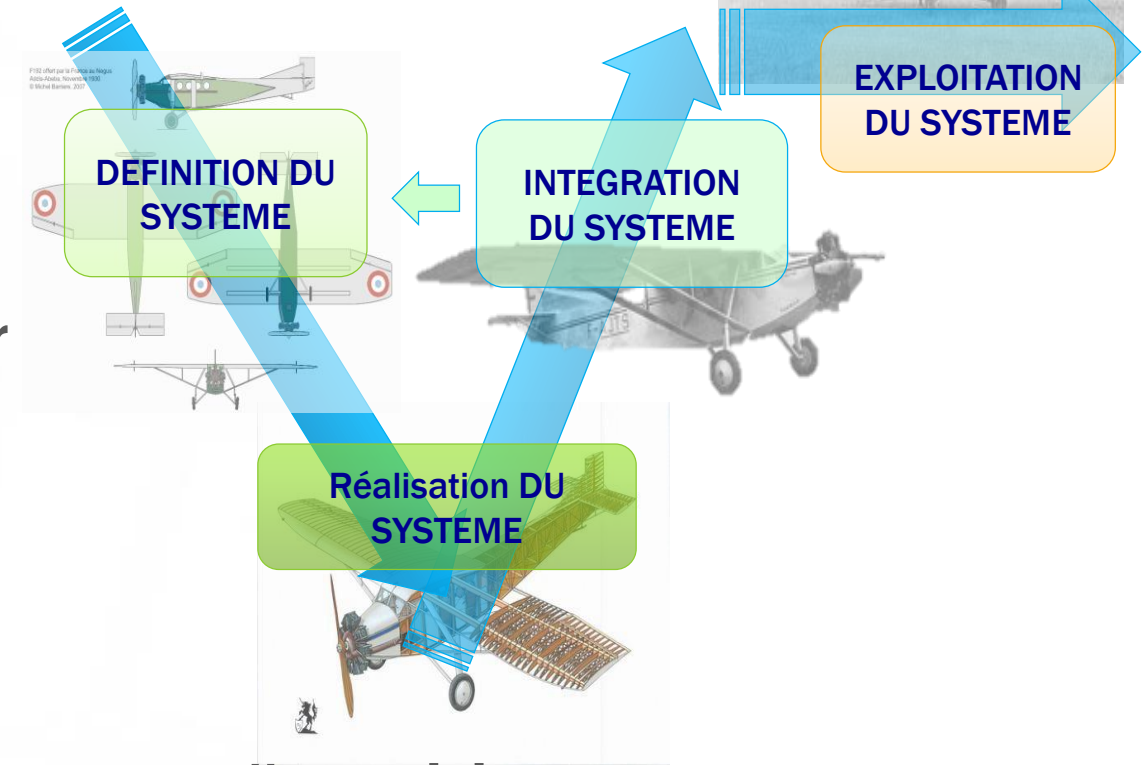
Approche prédictive



Le cycle en V : un cadre sécurisant

Les délais incompressibles de certaines étapes (Mises en plans, fabrications des pièces, fabrication d'outillages, approvisionnement) et le coût élevé induit par les modifications du produit en cours de projet, poussent à privilégier une approche prédictive pour les projets HW.
= « bon vieux cycle en V »

- Nécessite de disposer au démarrage d'une **vision complète et suffisamment précise du produit fini**,
- Nécessite d'établir un **scenario projet**



La logique de développement

Objectif de la LdD :

- Etablir un plan de dérisquage afin de « tuer les risques » le plus tôt possible,
- Faire apparaître les délais intrinsèques au processus de conception-développement HW
- Planifier le séquençement des activités en tenant compte des livrables intermédiaires (prototypes de validation, livrables documentaires...)

La méthode WBS est généralement utilisée pour recenser les livrables et délimiter les *work packages*, qui seront ensuite planifiés.

Pour construire le Logique de Développement, les principes et les outils de l'architecture système peuvent aider grandement.

Démarrer un projet de conception HW

Éléments généralement disponibles au départ

- ❑ « Product Requirements »
 - Sous forme de tableau d'exigences fonctionnelles et non fonctionnelles, plus ou moins rigoureux et exhaustif, sujet à interprétation
- ❑ Date cible de livraison des premiers équipements de production série
- ❑ Dates de livrables intermédiaires (Démonstrateur, Prototypes de validation, ...)
- ❑ Budget d'étude (NRC)
- ❑ Cout cible de fabrication (RC)

Ce dont on a également besoin pour démarrer la conception :

- ❑ Disposer d'une **vision globale partagée du produit final** :
 - ❑ Ce que les utilisateurs attendent du système ? Pour faire quoi ?
 - ❑ Ce que le système doit accomplir ?
 - ❑ De quoi le système est constitué ?
- ➔ L'architecture système a pour objectif de répondre à ces questions.



L'architecture système

- quelques notions -

Plusieurs types de système

Les systèmes



Les systèmes complexes



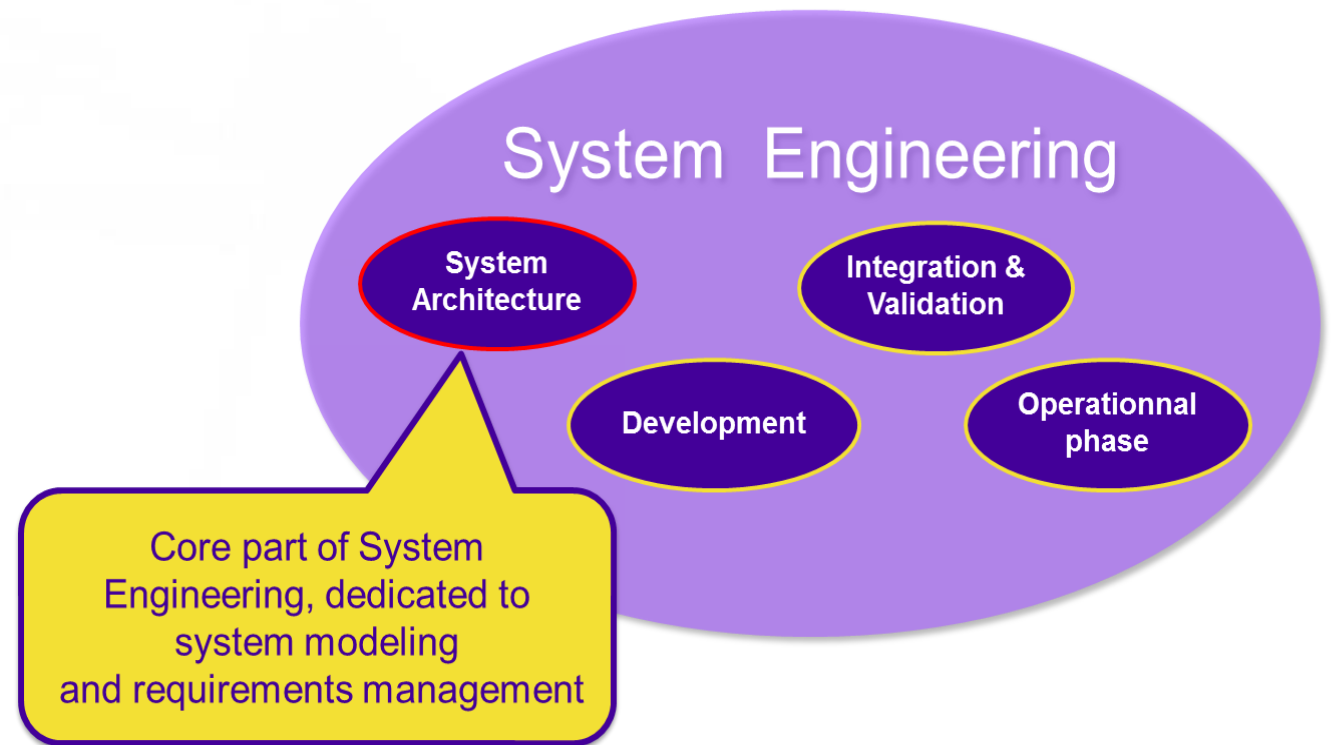
Les systèmes de systèmes ...

L'architecture système : une définition



L'architecture système est la discipline qui synthétise les méthodes et les outils permettant la modélisation d'un système (sous sa triple dimension **opérationnelle**, **fonctionnelle**, **organique**) dans le but de piloter efficacement sa mise en place.

System Architecture VS System Engineering

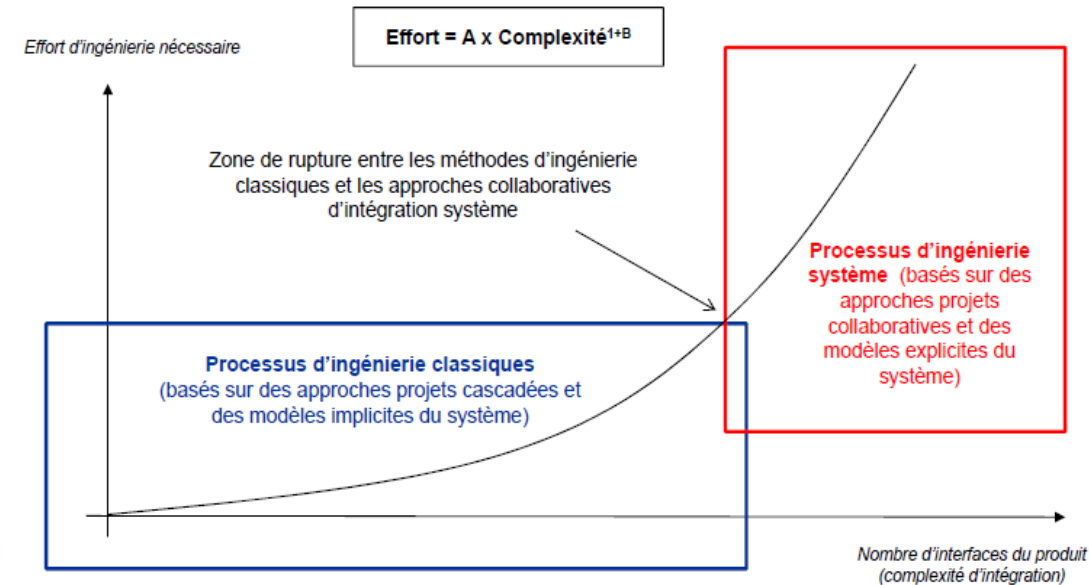


L'architecture système

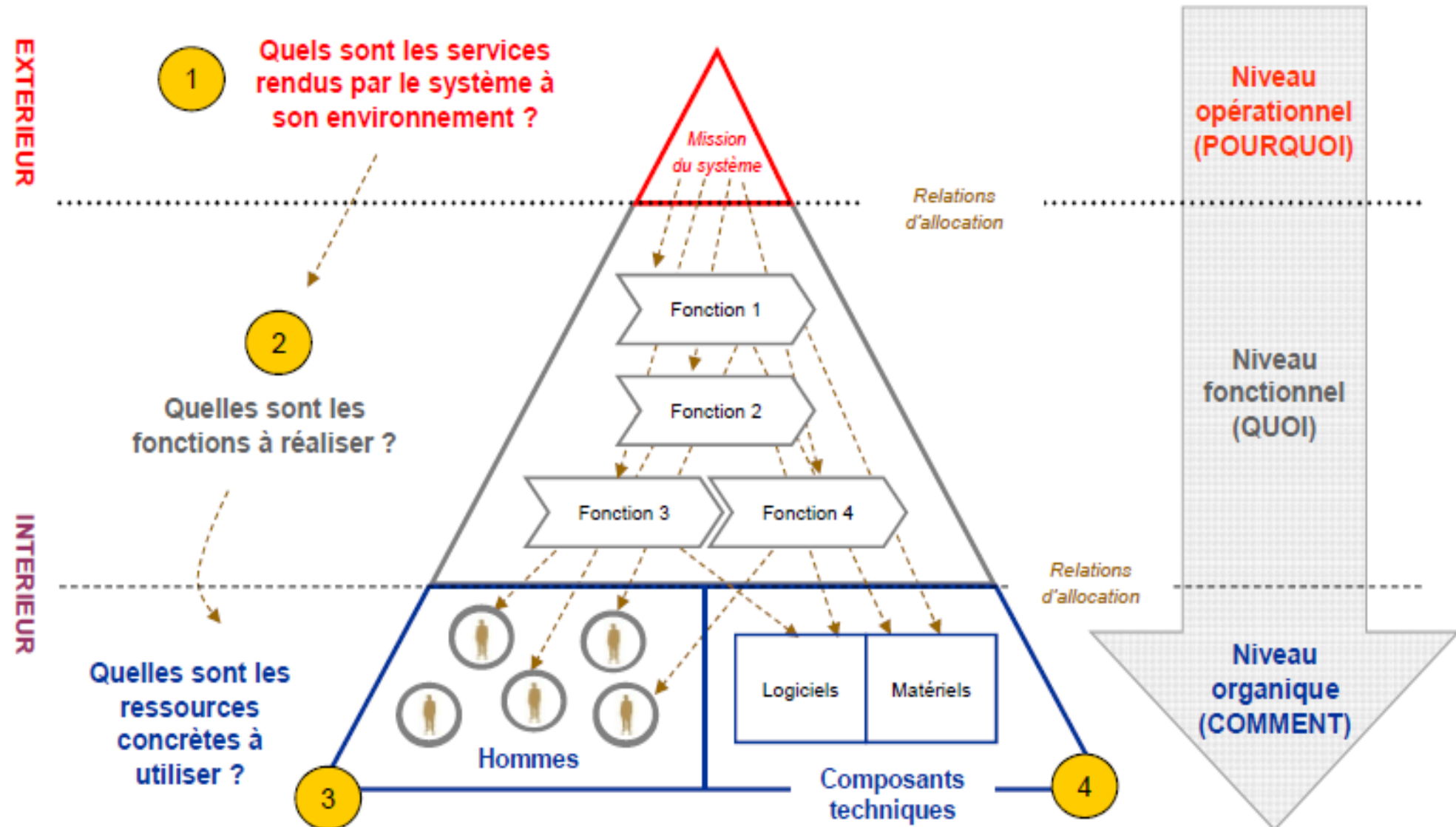
- Approche **collaborative** et interdisciplinaire, fondée sur l'analyse et l'**expérience**, qui englobe les activités pour concevoir, développer, faire évoluer et vérifier un système
- Approche **fortement itérative**
- Apporte une solution **optimisée** à des **besoins identifiés**

L'objectif ultime de l'architecture système est double :

- >> faire **coopérer les composants** du système afin d'en assurer la mission,
- >> faire **collaborer les parties prenantes** du projet afin de piloter correctement la conception et mise en place du système



Les 3 niveaux de représentation d'un système



Principe et apport de l'architecture système

Le principe fondamental de l'AS est de construire et entretenir une vision globale du système sous 3 angles :

- Architecture opérationnelle
- Architecture fonctionnelle
- Architecture organique

Lorsqu'un système (produit) est ainsi **modélisé** sous ces 3 vues architecturales, et qu'elles sont bien comprises par les équipes de développement pluridisciplinaires, alors les risques de silotage et de perte de vue de l'objectif sont fortement réduits.

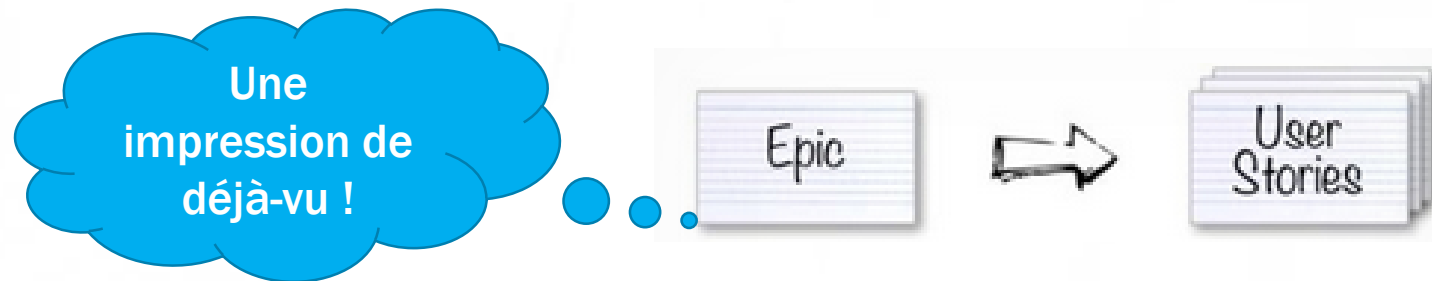
La vue opérationnelle du système

= Comprendre l'ensemble des interactions entre un système et son environnement , et les attendus (ou besoins) vis-à-vis du système.

Généralement exprimé sous la forme :



« le système externe / l'utilisateur doit pouvoir [expression du besoin] , dans le but de [bénéfice] »



Les objectifs de l'architecture opérationnelle sont de **garantir l'identification des besoins**, en tenant compte de la valeur perçue (« valeur client »).

(exemple) Vue opérationnelle

Besoins utilisateur

*Décomposition
opérationnelle*

Garantir la bonne
hygiène dentaire

Nettoyer les dents

Améliorer le nettoyage des
dents

Maintenir en condition
opérationnelle

Charger le
dentifrice

Brosser
les dents

Analyser les
brossages de dents

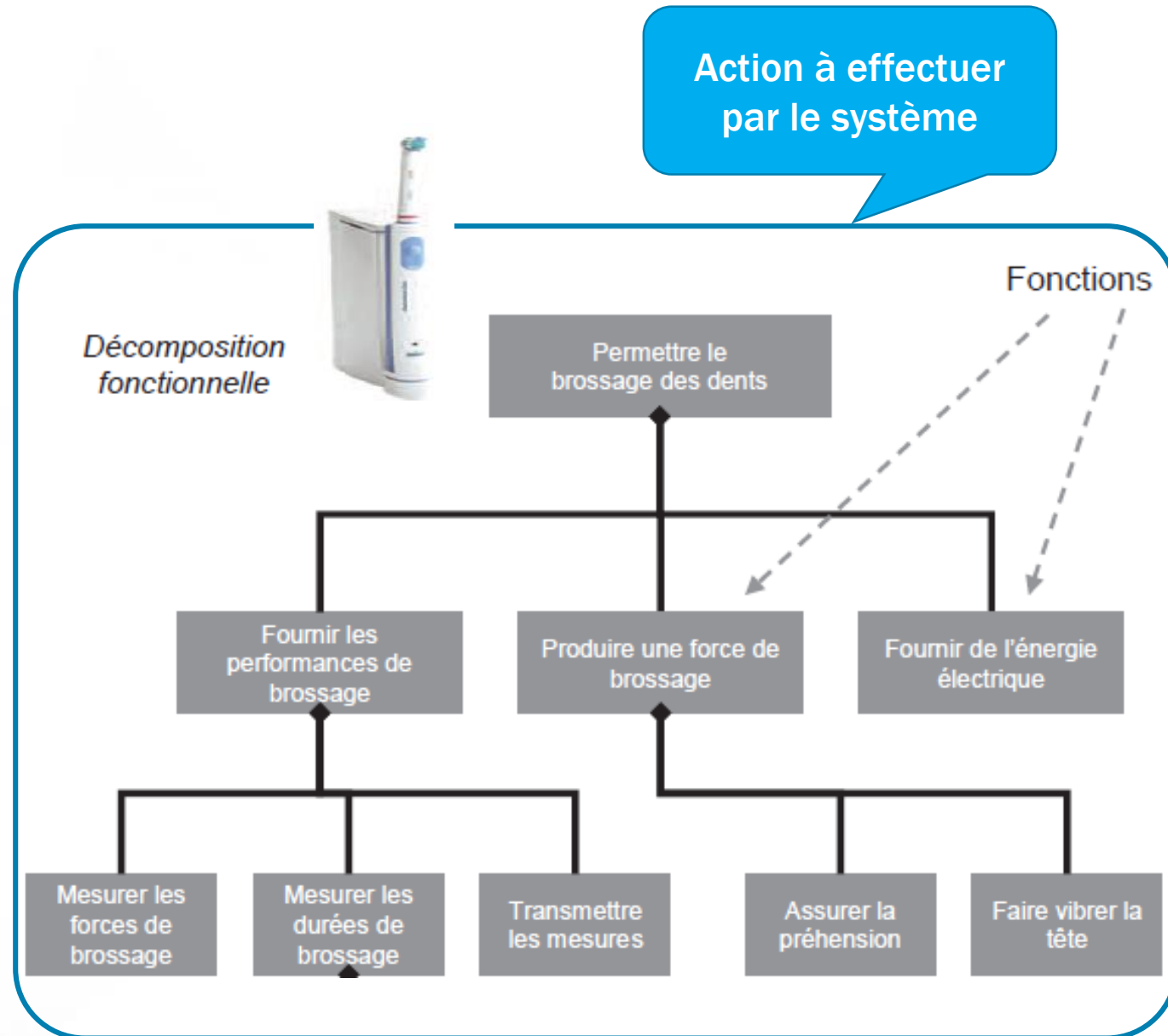
Faire des
recommandations de
brossage



La vue fonctionnelle du système

= Comprendre l'ensemble des comportements fonctionnels d'un système.

Les objectifs de l'architecture fonctionnelle sont de **garantir la couverture des exigences fonctionnelles, sans plonger trop vite dans le choix d'une solution technique.**

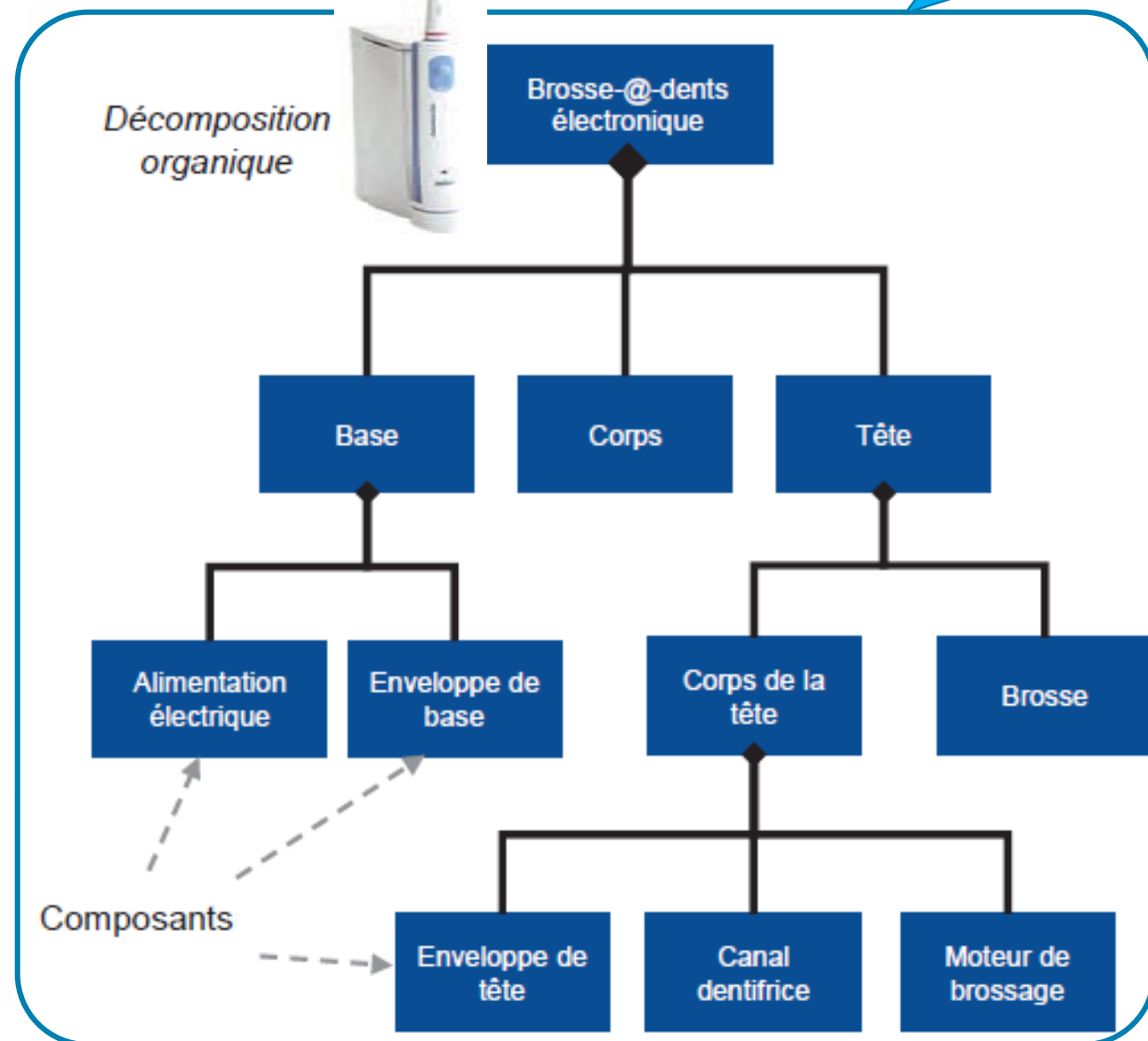


La vue organique du système

De quoi est
constitué le système

= Identifier les constituants physiques
(composants) d'un système et leurs interactions.

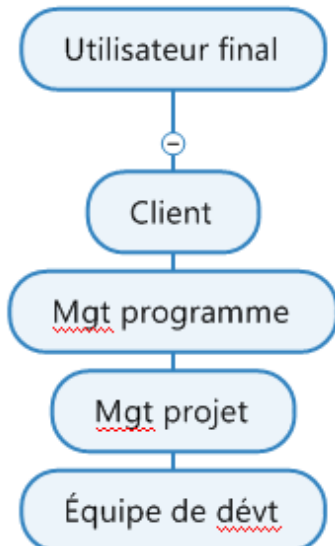
Les objectifs de l'architecture organique sont de
**garantir la cohérence et la complétude de la
vision concrète du système** vis-à-vis des
exigences opérationnelles & fonctionnelles.



Risque de désalignement au cours d'un projet

Une cause fréquente d'échec des projets ou de gaspillage de temps et d'argent est le **désalignement des parties prenantes** :

Client et utilisateurs,
management intermédiaire,...
= désalign vertical



Equipes de différentes disciplines, Entités support
(Achats, Qualité, ...) = désalign horizontal



En fournissant des représentations synthétiques et précises, facilitant la compréhension, l'AS limite les risques de désalignement des parties prenantes.

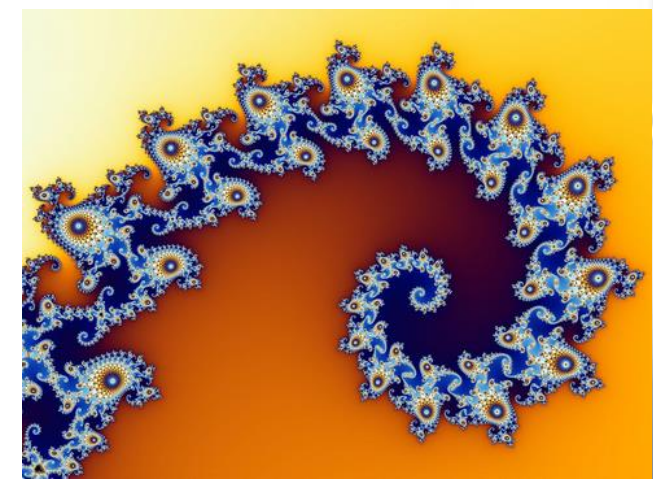


Les principes clés de l'architecture système : collaboration et communication

- L'AS est élaborée sous forme de **workshops** animés par l'architecte système, avec les experts concernés, les représentants du client,...
 - Les livrables sous le plus souvent des synoptiques de vues architecturales avec différents niveaux de détails
- Les vues architecturales sont d'excellents **vecteurs de communication**
 - Pour partager une vision commune du système entre les équipes de développement,
 - Pour communiquer vers les parties prenantes du projet, y compris le client et les utilisateurs
 - Capitalisables tout au long du projet (y compris IVVQ), et très utiles pour utilisation future (excellent point d'entrée pour replonger dans la conception ...)



En synthèse, sur l'Architecture Système



- La démarche architecture système est très utile pour :
 - **Aligner les protagonistes d'un projet** de conception-développement sur une **vision commune** du système (produit) à concevoir
 - Identifier les fonctions à **forte valeur utilisateur**
 - Identifier le niveau de **risque** associé aux différentes fonctions et constituants
 - Identifier les **interactions** entre composants et interactions avec les individus (parties prenantes)
 - Construire un **optimum global** (qui n'est pas une somme d'optimums locaux)
 - Organiser la **logique de développement** (« first things first »)
 - Poser un cadre de travail qui **facilitera l'autonomie des équipes**, notamment pour les prises de décisions raisonnées (vision globale partagée)

L'agilité : un *mindset* avant-tout

AGILE MANIFESTO, extended version !



Combien de mots faut-il changer dans le manifeste agile, pour le rendre universel à tous développements ?

3

Il suffit de remplacer à 3 endroits le mot « **software** », par « **product** », « **solution** », « **system** », ou « **service** » (au choix !)

Pour rendre le *Agile Manifesto* universel aux développements autres que logiciels.

L'adaptation au
changement
plus que le suivi d'un plan

La collaboration
avec les clients
plus que la négociation
contractuelle

Des **produits**
opérationnels
plus qu'une documentation
exhaustive

Les individus et
leurs interactions
plus que processus et outils



3. **Livrez fréquemment** un **produit** opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.

Les 4 valeurs de l'agilité

7. Un **produit** opérationnel est la principale mesure d'avancement projet.

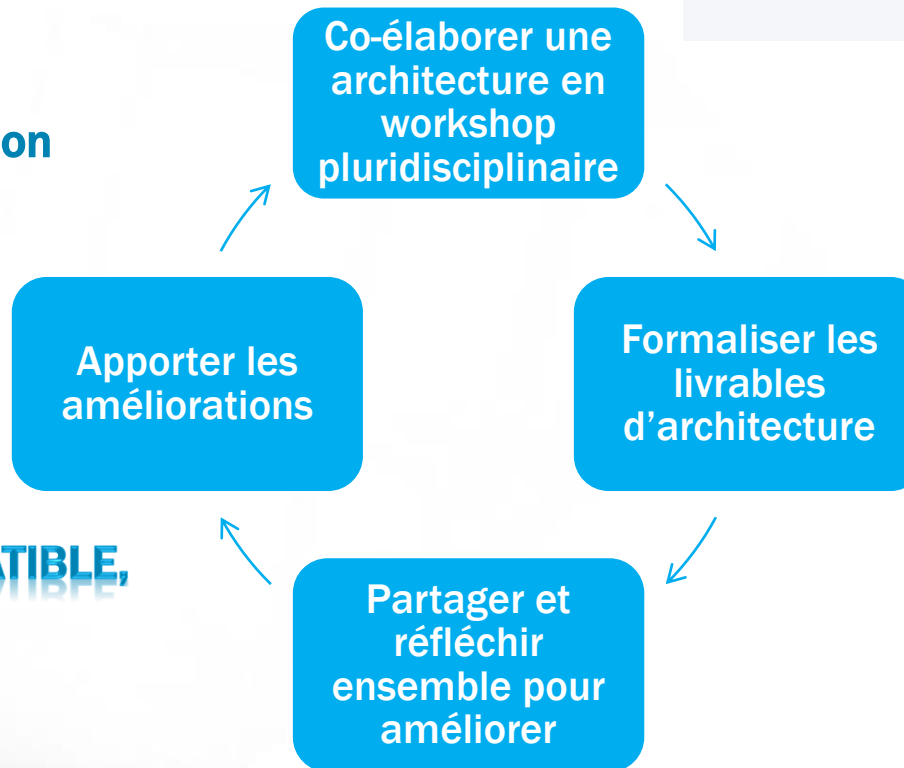
HEART OF AGILE

« Heart Of Agile », créé par Alistair Cockburn,
co-fondateur du mouvement Agile.

« Il s'agit de revenir aux valeurs et fondamentaux de
l'Agilité : Collaborer, Délivrer, Réfléchir, S'Améliorer. »



Et voici à quoi ressemble une Itération
d'architecture système ;



« **HEART OF AGILE** » COMPATIBLE,
ISN'T IT ? 😊

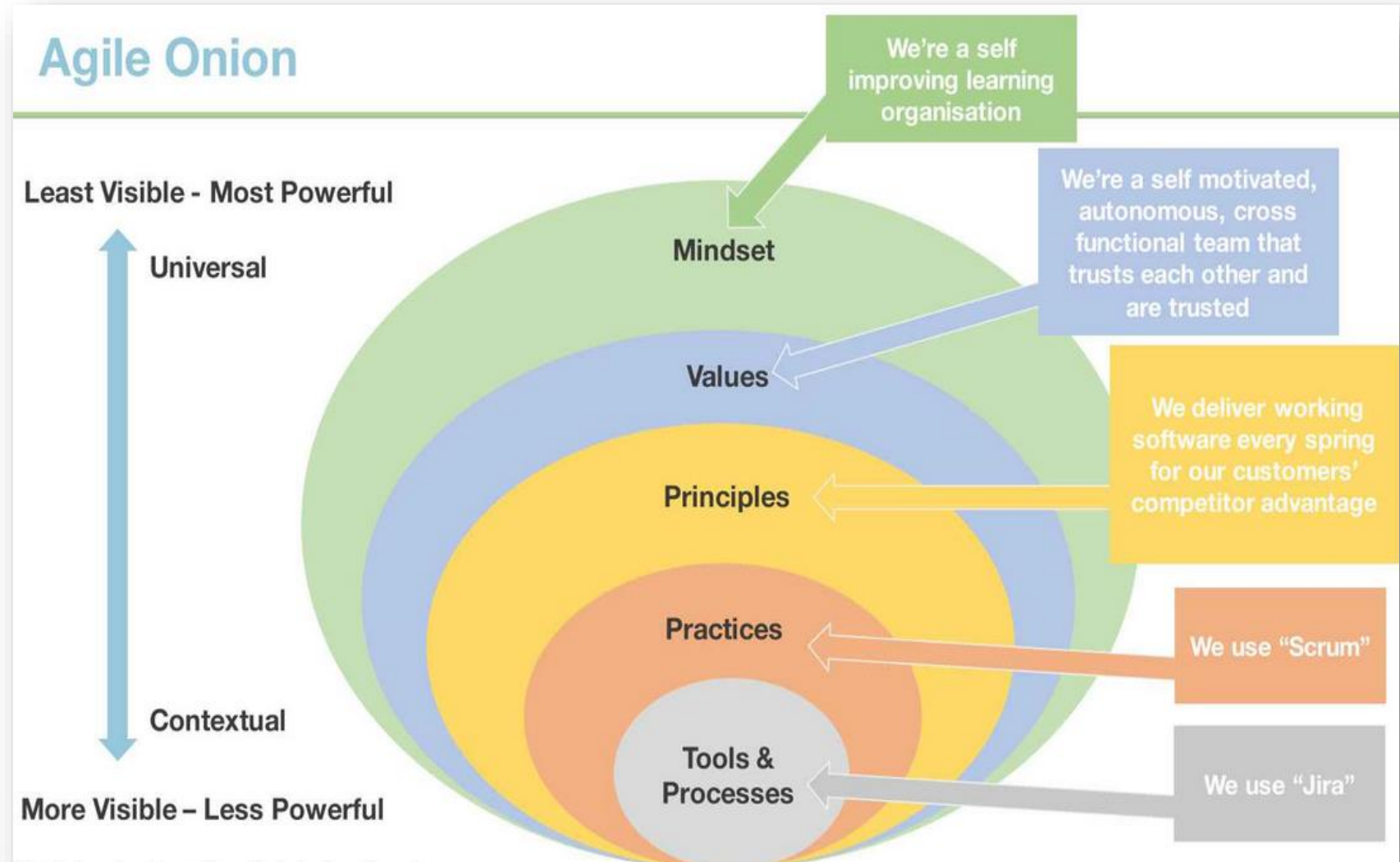
L'agilité : transposable au développement hardware ?

Le « *mindset* », les **valeurs** et **principes** agiles sont tout à fait applicables au hardware, avec les mêmes bénéfices que pour les activités software.



Et les pratiques et outils ?

- Certains également, sont transposables au hardware.



Nombreuses valeurs partagées entre agilité et architecture système

Agile is a behaviour

4 VALUES BEHAVIOURS

- Individuals and interactions
- Collaboration
- Respond to change
- Respect for people

12 PRINCIPLES BEHAVIOURS

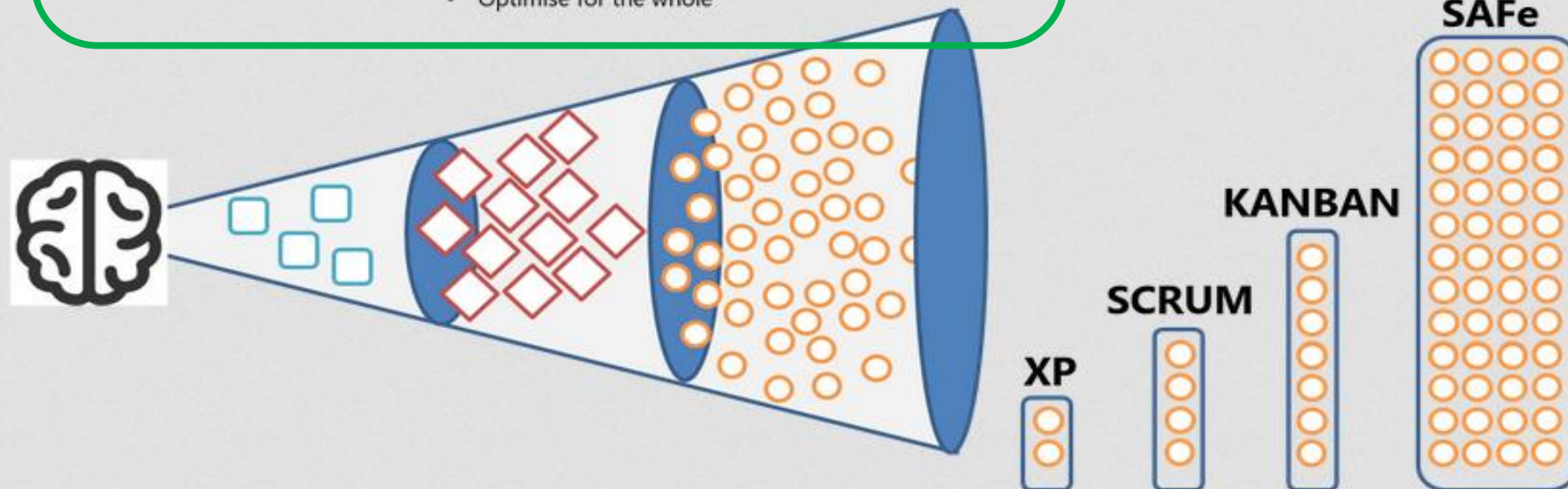
- Transparency
- Inspection
- Adaption
- Empiricism
- Trust
- Eliminate waste
- Deliver for fast feedback
- Build in quality
- Build organisational knowledge
- Optimise for the whole

MANY PRACTICES BEHAVIOURS

- Value-centred
- User-centred
- Optimised for fast feedback

NUMEROUS FRAMEWORKS BEHAVIOURS

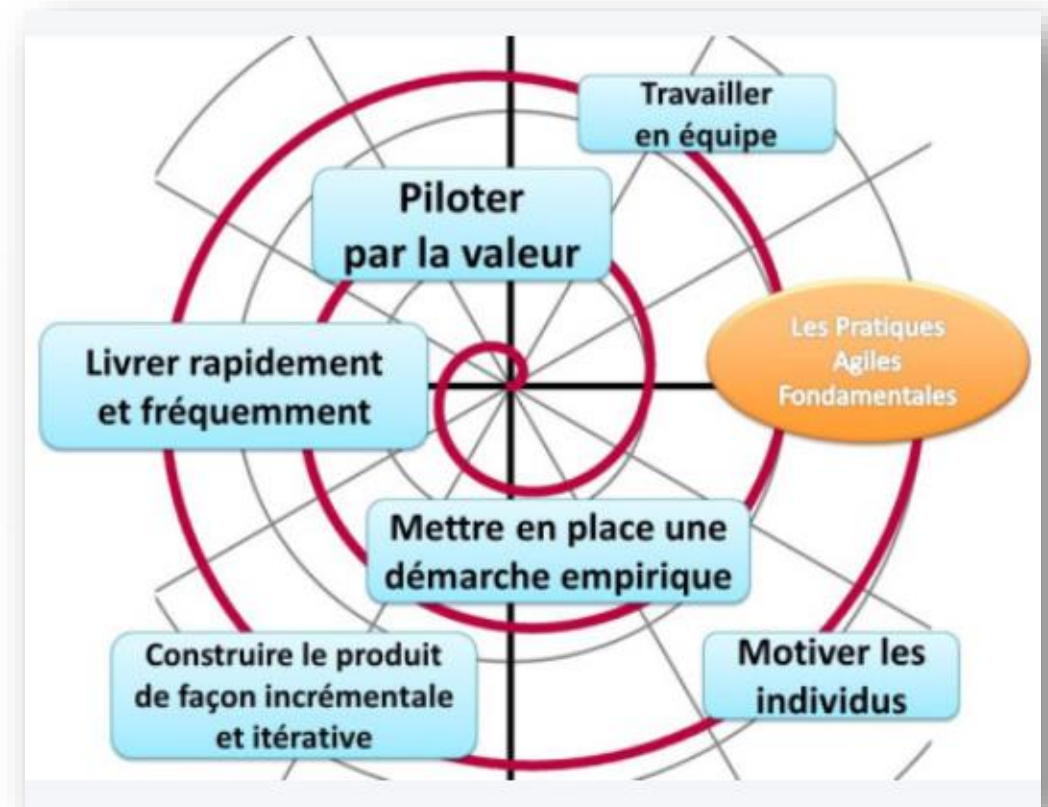
- Using a defined framework
- Disciplined in its execution
- Utilised for continuous improvement
- Utilised for continuous feedback loops



La majorité des valeurs, principes, et pratiques de l'agilité sont communes avec celles de l'architecture système

Principes Agiles : vecteurs de performance collective

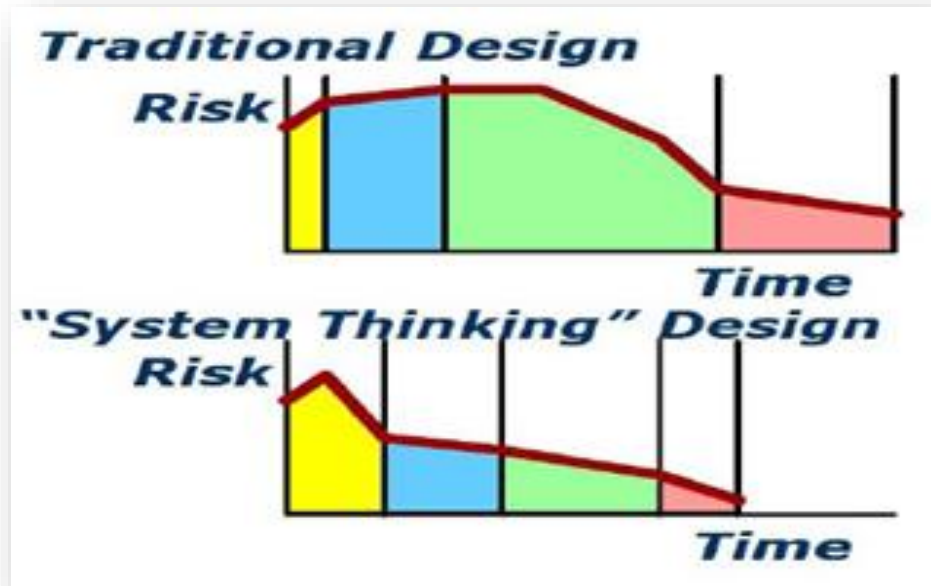
- ❑ Focalisation sur la valeur perçue
- ❑ Réactivité face aux changements de scope
- ❑ Capacité à proposer et intégrer plus rapidement des nouvelles fonctions (feature driven)
- ❑ Des équipes plus autonomes, plus motivées, plus polyvalentes, plus créatives
- ❑ Une culture du livrable qui entretient la création de valeur



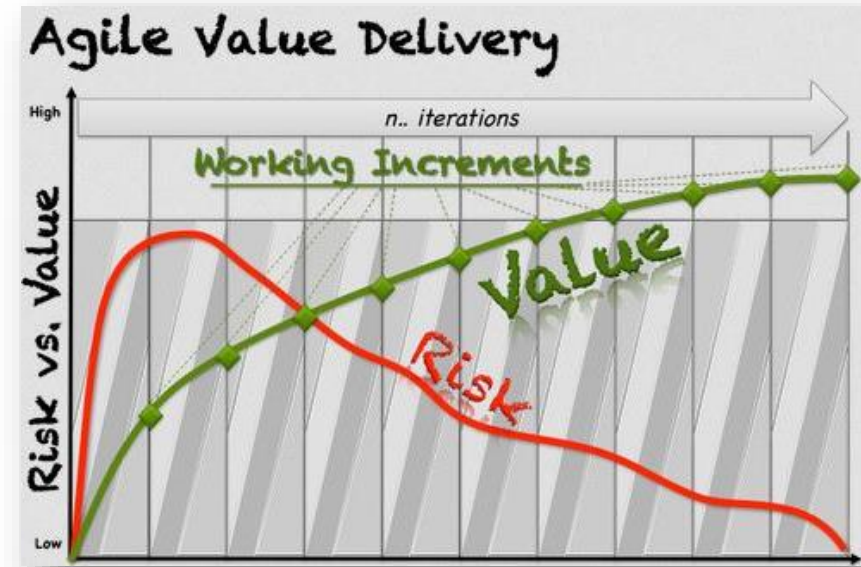
- ❑ Des bénéfices également pour les projets de développement matériel, utilisant une démarche architecture système.



Un principe directeur de l'AS et de l'Agilité : Le focus sur la réduction des risques au plus tôt



La démarche d'architecture système est utilisée pour mettre la priorité sur la réduction des risques au plus tôt dans le projet.



De même, le principe « *test fast, fail fast, learn fast* » de l'agilité, permet de valider les concepts au plus tôt en les confrontant aux utilisateurs.

The background of the slide is an abstract architectural composition. It features a series of white, three-dimensional rectangular blocks and lines that create a sense of depth and perspective. These elements are arranged in a way that suggests a modern building's structure or a complex geometric design. The overall color palette is light, with various shades of white and light gray, and a prominent blue color used for the text.

En conclusion

AS et Agilité : quelques différences ...

Architecture système

Top-down

Model based

Globale

Cycle en V

Prédictif

Structuré



Agilité

Bottom-up

Test based

Incrémentale

Livraisons
continues

Essai-Erreur

Auto-organisé

... et de nombreuses similitudes

Collaboratif

Itératif

Adaptatif

Convergence vers la réponse au
besoin réel client

Maximiser la valeur
ajoutée du processus

Approche
coût/bénéfice

Evolutif

Racines communes avec le
Lean Engineering

Utilise abondamment le
management visuel

Vise à atténuer les risques au
plus tôt

Repose sur l'intelligence
collective

Importance accordée aux
usages

Scalable

Guidé par la valeur
perçue client

Vise à fournir une
solution au juste besoin

Pluridisciplinaire

Repose sur un mindset,
des principes, et pratiques

Intimement lié à
l'organisation équipe

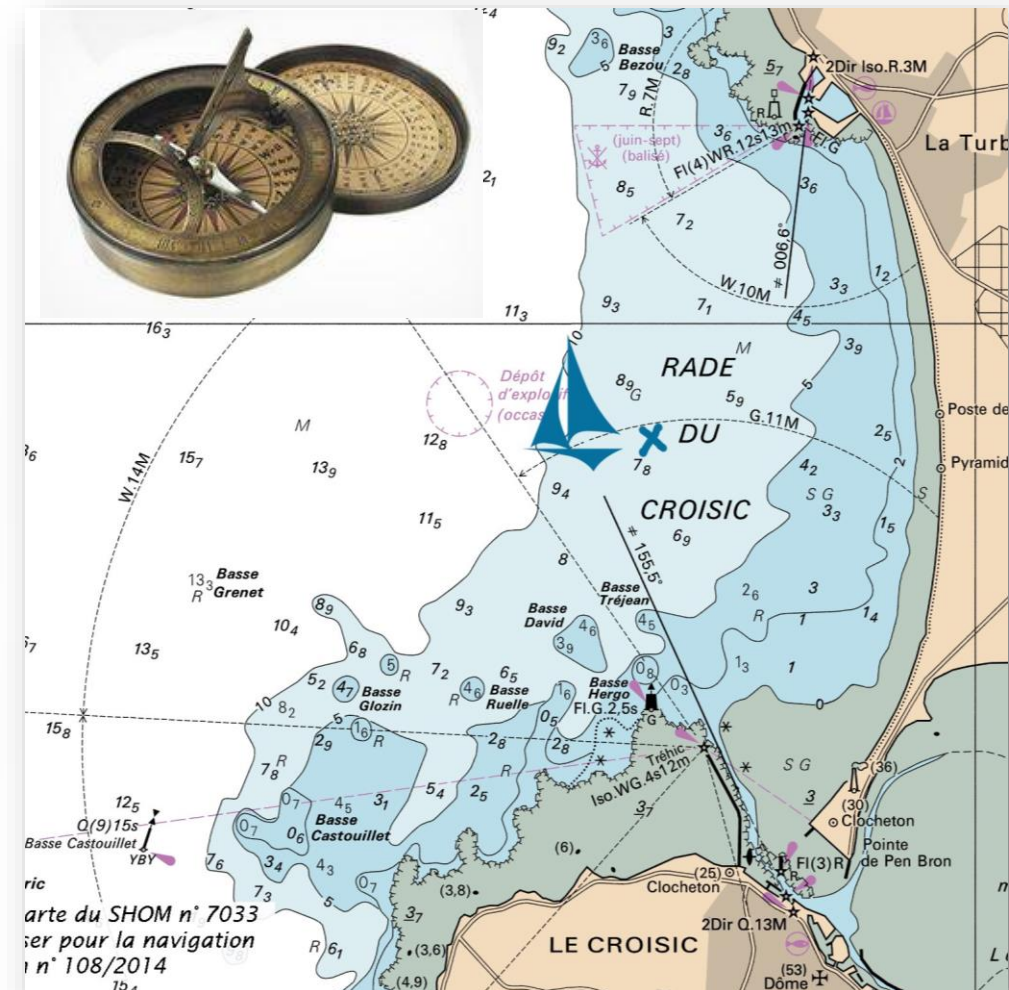
L'architecture système : « la carte et la boussole »

Dans un projet de conception-développement intégrant une part hardware, il est indispensable de « modéliser le tout », planifier les étapes du projet, et jalonner avec des points de passage (toll gates) qui permettront de mesurer la maturité du produit au cours du développement.

L'architecture système répond à ce **besoin de vision globale du produit**, en amenant les réponses :

- ❑ Quel est le service attendu par les utilisateurs ? (POURQUOI)
- ❑ Quelles sont les fonctionnalités du produit ? (QUOI)
- ❑ De quoi est constitué le produit ? (COMMENT)

Dicton marin : « Il n'est pas de vent favorable pour qui ne connaît pas son port d'arrivée. »



L 'agilité : « Du vent dans les voiles ! »

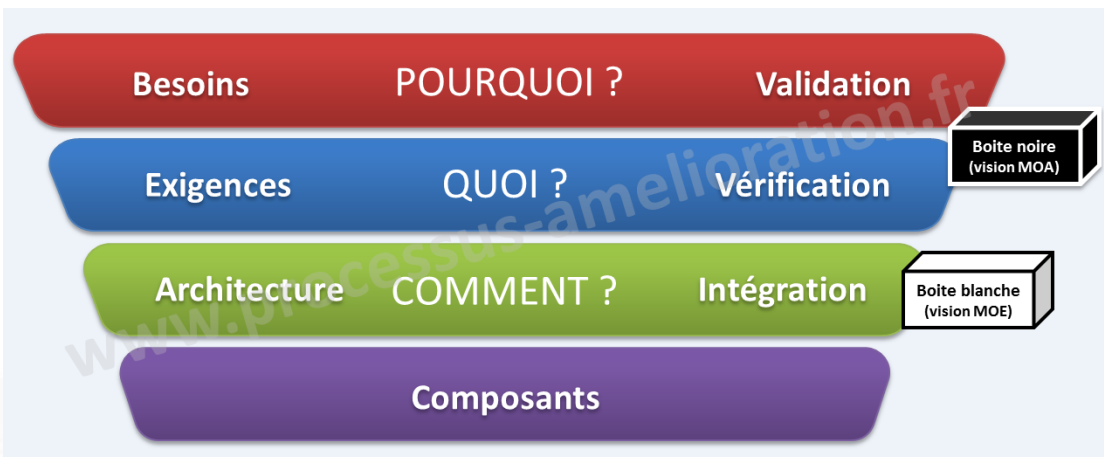
L'agilité apporte de nombreux bénéfices pour les projets de développement de produits, pas uniquement logiciels :

- ❑ Focalisation sur la valeur perçue
- ❑ Maîtrise des risques
- ❑ Réactivité face aux changements de scope
- ❑ Capacité à proposer plus rapidement des nouvelles fonctions (feature driven)
- ❑ Des équipes plus autonomes, plus motivées, plus créatives
- ❑ Une culture du livrable qui favorise la productivité

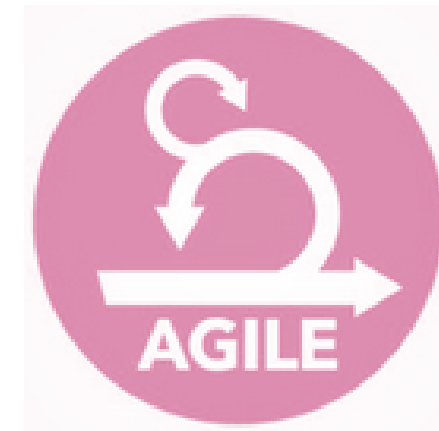


La combinaison gagnante ?

Architecture Système



Agilité

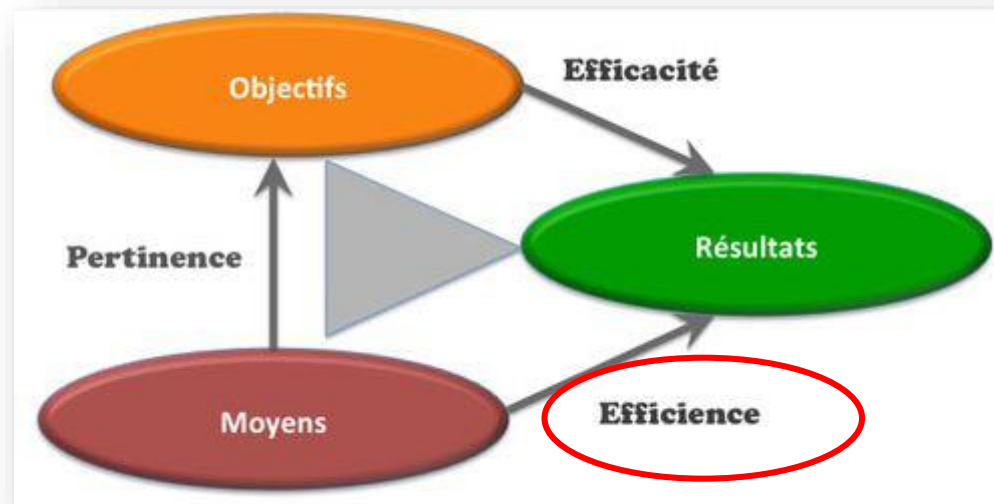


Combiner la démarche Architecture Système avec les principes et pratiques de l'agilité ;

= une « association de bienfaiteurs » pour les projets de développement de produits HW-SW



THANK
YOU



Sources

- radtac.com
- adventurewithagile.com
- freeagile.org
- MyAgilePArtnr.fr
- heartofagile.com
- cesames.net