

Guide sur l'application des approches AGILES dans le domaine automobile



EQUIPE EDITORIALE

Ce guide a été élaboré et validé par un groupe de référents de plusieurs sociétés :

			A&B	C
Babo BABAKWANZA	ALTEN	AGILE Practice Manager	X	
Svetlana DICHEVA	ALTEN Labs	Directeur programme R&D	X	X
Léonardo NASCIMENTO	BOSCH	Directeur d'Ingénierie Bosch Automotive Service Solutions	X	
Frédéric FRADET	CAPJAYA	Consultant ASPICE et MBSE		X
Jérôme CHAROUSSET	CONTINENTAL	Responsable Transformation Agile	X	
Emilie LAURENT	CONTINENTAL	Responsable Transformation Agile	X	
Daniel HUMBERT-DROZ	FAURECIA	Responsable Qualité et Sécurité	X	
Dominique BERTHIER	MICHELIN	Garant Qualité Mobilité Connectée	X	X
Alexis GAUTIER	PLASTIC OMNIUM	Mechatronics Senior Expert	X	
Djelloul REGUIEG	PLASTIC OMNIUM	Embedded Software Architect	X	
Florian TETAUD	PLASTIC OMNIUM	Mechatronics Laboratory Means & Facilities Leader	X	
Frédéric D'ARRENTIERES	GROUP RENAULT	Expert Leader Management Projet	X	
Sylvain LEJEALLE	STELLANTIS	Product Safety Audits and Processes Fellow		X
Bruno MAJOT	TECRIS	Ex. PDG TECRIS	X	
Amélie PONCHET-DURUPT	UTC	Enseignant chercheur	X	
Matthieu BRICOGNE	UTC	Enseignant chercheur	X	
Xavier CHEVALIER	VALEO	R&D Efficiency Manager		X
Laurent GARROUSTE	VEONEER	Safety Expert		X
Philippe QUERE	WOVEN by Toyota	Safety Expert		X



Version initiale publiée le 04/12/2023 (DC-17-01)

Version 2 publiée le 29/07/2026 (DC-17-02)

Publication internet : www.sia.fr (rubrique Publications)

Référence : DC-17-02

Date : 29/07/2026

Page 2

AVANT-PROPOS

Depuis quelques années les Approches AGILES se déploient dans le domaine automobile. Bien qu'initialement utilisées dans le cadre du développement logiciel où elles sont apparues, ces Approches semblent pouvoir apporter de nombreux bénéfices au monde automobile.

Elles semblent être adaptées à notre nouveau monde VUCA (Volatility Uncertainty Complexity Ambiguity) permettant aux différents niveaux d'organisation de l'Entreprise d'améliorer leurs capacités à s'adapter aux changements, à ne pas perdre de vue le besoin et la satisfaction du client final et à encourager l'autonomie et la responsabilisation des équipes.

Cependant, l'arrivée de ces approches n'est pas sans poser de nouveaux défis et questions dans un monde automobile où la réponse à l'attendu du client ne se limite pas à des applications logicielles et où le développement produit répond à ce jour à de multiples référentiels : cycle en V, ISO9001, IATF, ISO26262, cybersécurité... Elle n'est pas sans remise en cause de nos habitudes, voire de nos réflexes, ancrés depuis de nombreuses années pour certaines générations.

Plusieurs questions sont donc posées : nos processus doivent-ils évoluer pour mieux prendre en compte ces approches ? Faut-il revoir l'organisation globale de l'entreprise ou s'agit-il d'une évolution encore plus profonde de la culture d'entreprise et de l'état d'esprit de ses acteurs ?

Ce document n'a pas vocation à répondre à l'ensemble de ces questions mais la Communauté Fiabilité Qualité Sécurité de la SIA a souhaité permettre à un certain nombre d'acteurs d'échanger sur leurs pratiques et expérimentations, notamment sur son périmètre d'expertise.

Bien que la volonté ait initialement été d'animer un débat constructif entre les convaincus des approches AGILES et les plus réticents, les travaux se sont rapidement orientés vers la formalisation d'éléments de définition, permettant d'éviter les mauvaises interprétations et les faux débats, mais également d'identifier les questions fondamentales qui se posent, puis de partager les bonnes pratiques, existantes ou nouvelles.

PARTIE A. INTRODUCTION ET RÉFÉRENTIELS	8
1 STRUCTURE DU GUIDE.....	8
1.1 versions	8
1.2 Pictogrammes	8
1.3 Notations et acronymes	9
2 INTRODUCTION	11
3 ENJEUX	11
4 DEFINITIONS	12
4.1 Vocabulaire lié aux Approches	12
4.2 Vocabulaire lié aux Processus, méthodes et outils	13
5 FRAMEWORKS.....	15
5.1 SCRUM	15
5.1.1 Historique	15
5.1.2 Description du FLUX D'Activité SCRUM:	16
5.1.3 Points clés du framework :	17
5.2 Kanban	18
5.2.1 Historique	18
5.2.2 Description du flux d'activités kanban	18
5.2.3 Points clés du framework	19
5.3 Les frameworkS d'Agilité à l'échelle.....	20
5.3.1 Scrum of Scrums	21
5.3.2 Le framework LeSS.....	22
5.3.3 Le framework SAFe	22
5.3.4 points clés de comparaison entre LeSS et SAFe.....	23
5.4 Approche Hybride	24
5.4.1 différents modes d'hybridation	24
6 IDENTIFICATION DES POINTS DE VIGILANCE VIS-A-VIS DE L'APPROCHE AGILE :	25
6.1 caractérisation des contextes d'application.....	25
6.2 Identification des points d'attention.....	27
6.3 Gestion des risques liés aux évolutions d'Approche ou à la transformation de l'organisation.....	28
6.3.1 nécessité d'un pilotage du changement	28
6.3.2 Type de transformation et stratégie d'entreprise :	30
7 RECOMMANDATIONS ET BONNES PRATIQUES PAR THEMATIQUES	31
PARTIE B. AGILE ET QUALITE.....	32
1 IDENTIFICATION DES PROBLEMATIQUES.....	32
2 AGILE ET BESOIN DU CLIENT	32
2.1 Rappel des principes et des exigences	32
2.2 Avantages et difficultés.....	32
2.3 Conditions de succes	33
2.3.1 Adapter le processus de Revue et de Priorisation des exigences	33
2.3.2 Demonstrations rapides en clientèle	35
2.3.3 Gestion des évolutions d'exigences	35
3 AGILE ET PLANIFICATION	37
3.1 Rappel des principes et des exigences	37
3.2 Avantages et difficultés.....	37
3.3 conditions de SUCCÈS.....	37
3.3.1 Planification basée sur le flux de valeur.....	37
3.3.2 Planification Adaptée à l'horizon de temps	38
3.3.3 Planification à l'échelle dans le domaine automobile.....	38
3.3.4 Maitrise du Processus de Planification	40
3.3.5 Adaptation de la planification des livrables et de leurs contenus	40
4 AGILE ET CONCEPTION	41
4.1 Rappel des principes et des exigences	41
4.1.1 Principes Agiles et pratiques des frameworks Agiles relatif à la conception :	41

4.1.2 La phase de conception est également soumise à de nombreux standards :	42
4.1.3 La conception est soumise aux contraintes de l'industrie automobile	42
4.2 Avantages et difficultés	43
4.2.1 Avantages :	43
4.2.2 Difficultés :	43
4.3 conditions de succès	44
5 AGILE ET MAITRISE DES RISQUES	48
5.1 Rappel des principes et des exigences	48
5.2 Avantages et difficultés	50
5.3 conditions de succès sur la Gestion des risques projets	51
5.3.1 Choix de l'approche	51
5.3.2 Rituel de gestion des risques	51
5.3.3 Modes d'Interactions et comportement au sein de l'équipe de développement	52
5.3.4 disponibilité des experts	52
5.3.5 Capitalisation	52
5.4 conditions de succès sur la Gestion des risques techniques	52
5.4.1 Gestion des configurations en environnement VUCA	52
5.4.2 maîtrise des risques techniques en mode de conception itératif	53
5.4.3 Formalisation des livrables qualité en mode de conception itératif	54
5.4.4 Capacité à gérer les risques liés aux démonstrations (plus) rapides sur le marché	54
6 AGILE ET VERIFICATION & VALIDATION	55
6.1 Rappel des principes et des exigences	55
6.2 Problématique	57
6.3 Conditions de succès	58
6.3.1 Gestion de la configuration pour les Vérification et Validation	58
6.3.2 Vérifier et valider rapidement	58
6.3.2.1 Valider virtuellement	59
6.3.2.2 Etendre le process de CI	60
6.3.2.3 Optimisation des coûts	62
6.3.3 Couvrir l'ensemble des exigences par une stratégie V&V adaptée	64
6.3.3.1 Scalable Definition of Done	64
6.3.3.2 Scenarios de validation	65
6.3.4 Conserver la capacité de Test & Learn	66
6.3.5 Validation des évolutions logicielles en clientèle	66
7 AGILE ET RELATION CLIENT-FOURNISSEUR	68
7.1 Rappel des principes et des exigences	68
7.2 Avantages et difficultés	68
7.3 Modèles Contractuels client - Fournisseur	71
8 AGILE ET COMPETENCES	73
8.1 Rappel des principes et des exigences	73
8.2 Avantages et difficultés	73
8.3 Conditions de succès	74
8.3.1 Compétences élargies	74
8.3.2 Montée en compétences adaptée et coaching	75
8.3.3 Implication des services Ressources humaines	76
8.3.4 Implication du management et top management	77
9 SYNTHESE ET POINTS OUVERTS	78
PARTIE C. AGILE ET SECURITE FONCTIONNELLE	80
1 IDENTIFICATION DES PROBLEMATIQUES	80
1.1 Compléments sur les enjeux et besoins	80
1.1.1 Rappel sur les normes de la sécurité automobile	80
1.1.2 Pourquoi déployer des approches agiles dans ces domaines ?	82
1.1.3 Que propose cette partie C du guide ?	83
1.2 les Problématiques identifiées	84
2 SAFETY CULTURE	85
2.1 Rappel des principes et des exigences	85
2.2 Avantages et difficultés	85
2.3 Conditions de succès	85
2.3.1 Culture Safety & AGILE	85

2.3.2 Orientation Processus, contrôle et traçabilité	86
2.3.3 Compétences requises.....	86
3 CYCLE DE DEVELOPPEMENT	87
3.1 Rappel des principes et des exigences	87
3.1.1 Exigences ISO26262	87
3.1.2 Exigences SOTIF-ISO21448.....	89
3.2 Avantages et difficultés.....	91
3.3 Conditions de succès	92
3.3.1 Préliminaires requis et phases de développement itératives.....	92
3.3.1.1 Principe :	92
3.3.1.2 Cas d'application ISO26262 :	93
3.3.1.3 Cas d'application SOTIF :	96
3.3.1.3.1 Phase Pre-Game :	96
3.3.1.3.2 Phase Game	97
3.3.1.3.1 Cas particulier du cycle de vie en Opération.....	99
3.3.2 Définition de la priorité des exigences lors des itérations	100
3.3.2.1 Principe.....	100
3.3.3 conditions de lancement et de validation d'une itération et d'une release	101
3.3.3.1 Conditions pour lancer une itération :	101
3.3.3.2 Conditions pour valider la fin d'une itération :	101
3.3.3.3 Conditions pour valider une release (interne ou externe) :	102
3.3.4 Organisation et synchronisation des équipes	103
3.3.4.1 Organisation statique des équipes	103
3.3.4.2 Organisation dynamique des équipes	104
Outils associés au processus de développement.....	108
4 GESTION DES EXIGENCES	109
4.1 Rappel des principes et des exigences	109
4.1.1 Caractéristiques des exigences Safety	109
4.1.2 Gestion des exigences safety	110
4.2 Avantages et difficultés.....	111
4.3 Conditions de succès	111
4.3.1 Conditions de succès spécifiques au mode agile	111
4.3.1.1 Intérêt et limite à la notion de priorisation d'exigences et de MVP :	111
4.3.1.2 Gestion et traçabilité des exigences.....	112
4.3.1.3 Rigueur de la maintenance documentaire	113
3.3.2 Quelques point d'attention à ne pas oublier	114
5 SYNTHESE ET POINTS OUVERTS	116

LISTE DES ANNEXES

Annexe 1 : Liste des figures et des tableaux.....	118
Annexe 2 : Références bibliographiques	120
Annexe 3 : Grille d'identification des points de vigilance liés aux approches Agiles ou Classiques.....	122

PARTIE A. INTRODUCTION ET RÉFÉRENTIELS

1 STRUCTURE DU GUIDE

Ce guide est prévu pour être construit de manière itérative.

Cette première version donne un cadre de définition des Approches AGILES et un état de l'existant que nous avons voulu conserver synthétique et non exhaustif. Le but de ce cadre étant de faciliter un alignement de la communauté sur des notions considérées comme fondamentales.

Le guide aborde ensuite les problématiques posées par l'arrivée des approches AGILES dans le domaine Automobile (voir [liste des problématiques](#)) en proposant des recommandations sans caractère systématique.

Des versions suivantes devraient donc compléter la réflexion avec les autres thématiques de la Communauté d'Experts – CE FQS - et notamment celle de la Sécurité, voire élargir le champ de réflexion à d'autres domaines.

1.1 VERSIONS

Date de publication	Thématiques abordées
04/12/2023	Définitions, État de l'Art
04/12/2023	AGILE & Qualité
29/07/2026	AGILE & Safety

1.2 PICTOGRAMMES

Des icônes jalonnent ce guide afin d'attirer l'attention du lecteur sur certains passages. La signification de ces icônes est indiquée ci-après.

Pictogramme	Message d'avertissement
	Ce pictogramme indique au lecteur qu'il s'agit d'un INCONTOURNABLE, d'une INFORMATION INDISPENSABLE.
	Ce pictogramme indique au lecteur qu'il s'agit d'un PIEGE A EVITER.
	Ce pictogramme indique au lecteur qu'il s'agit d'une BONNE PRATIQUE, d'une BONNE IDEE.

1.3 NOTATIONS ET ACRONYMES

<u>ADAS</u>	Advanced Driver Assistance Systems
<u>AMDEC</u>	Analyse de Modes de Défaillances de leurs Effets et de leur Criticité
<u>ART</u>	Agile Release Train
<u>ASIL</u>	Automotive Safety Integrity Level
<u>ASPICE</u>	Automotive Software Performance Improvement and Capability dEtermination
<u>CE FQS</u>	Communauté d'Experts Fiabilité Qualité Sécurité (sûreté de fonctionnement) de la SIA.
<u>CI</u>	Continuous Integration
<u>CD</u>	Coutinuous Delivery
<u>DIL</u>	Driver In the Loop
<u>DoD</u>	Definition Of Done
<u>GEPP</u>	Gestion des Emplois et Parcours Professionnels
<u>HIL</u>	Hardware in the Lopp
<u>IATF</u>	International Automotive Task Force
<u>INTACS</u>	International Assessor Certification Scheme
<u>LeSS</u>	Large Scale Scrum
<u>MBD</u>	Model Based Design
<u>MIL</u>	Model In the Loop
<u>MMF</u>	Minimum Marketable Feature
<u>MVP</u>	Minimum Viable Product
<u>OTA</u>	Over The Air (peut-être appliqué à un Firmware FOTA ou à une Configuration COTA)
<u>PDCA</u>	Plan-Do-Check-Adjust
<u>PI</u>	Planning Interval
<u>PMI</u>	Project Management Institute
<u>RTE</u>	Release Train Engineer
<u>SAFe</u>	Scaled Agile Framework
<u>SyIL</u>	System In the Loop
<u>UNECE</u>	United Nations Economic Commission for Europe
<u>VIL</u>	Vehicle In the Loop
<u>VUCA</u>	Volatility Uncertainty Complexity Ambiguity

V&V	Vérification & Validation
WIP	Work In Progress
WSJF	Weighted Shortest Job First

2 INTRODUCTION

L'objectif du document est donc de mettre à disposition de la communauté des acteurs de la filière automobile :

- Les éléments de base sur ce que sont les approches AGILES afin d'éviter les mauvaises interprétations voire les faux débats,
- Les questions fondamentales qui se posent ainsi que les bonnes pratiques existantes ou que nos échanges ont fait émerger.

Parmi ces acteurs, nous pensons particulièrement :

- À ceux des métiers de la Fiabilité, de la Qualité et de la Sécurité qui découvrent les approches Agiles,
- Aux Agilistes provenant du domaine purement logiciel et ne connaissant pas encore le contexte automobile et ses contraintes particulières telles que le développement matériel, la production en grand volume et les contraintes associées de Fiabilité, Qualité et Sécurité.

3 ENJEUX

Pourquoi le monde Automobile s'intéresse-t-il de si près à des approches nées dans le domaine du génie logiciel et du manufacturing (Berthou, 2022) ?

Si les approches AGILES intéressent autant ce n'est pas uniquement parce que le monde Automobile a entamé une forte mutation en intégrant toujours plus de software aux traditionnels systèmes Hardware mais également parce que ces approches semblent apporter des éléments de réponse à de nombreux nouveaux besoins internes et externes à ce domaine.

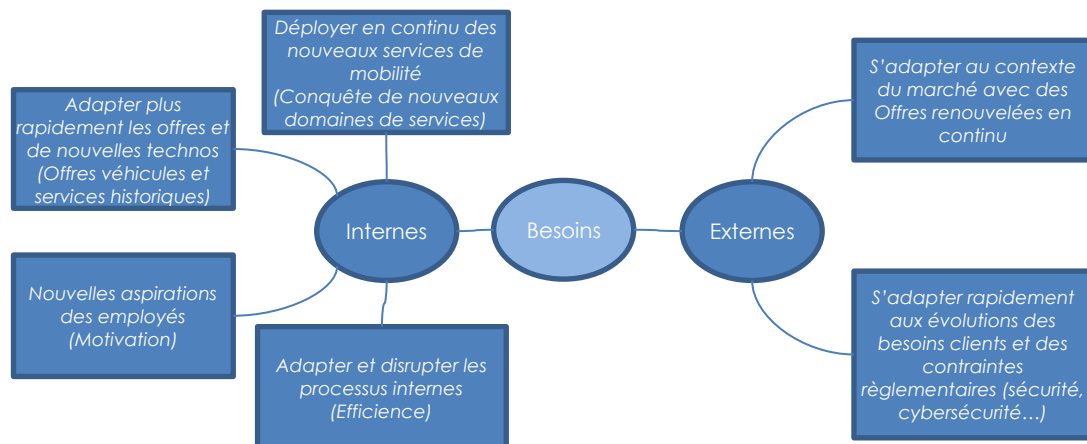


Figure 1- Besoins Internes et Externes du monde Automobile.

4 DEFINITIONS

Approche/Processus/Méthode/Outil :

Approche : façon de penser, d'appréhender un contexte de développement, d'opération de Produit et/ou Service,

Processus : enchaînement d'étapes, organisées temporellement qui visent à atteindre un but,

Méthode : ensemble de règles et pratiques qui visent à organiser une ou plusieurs étapes du processus,

Outil : élément effectivement manipulable pour réaliser une tâche circonscrite.

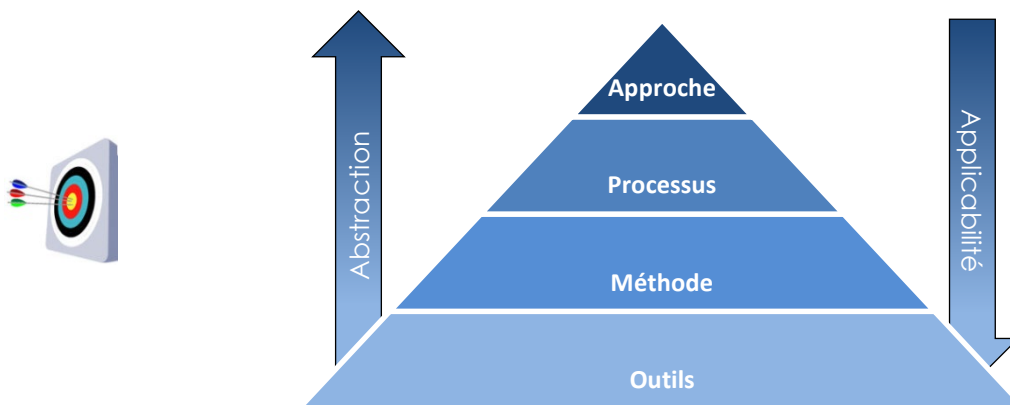


Figure 2 - Canevas de classification des techniques et méthodes [Guérineau et al., 2018]

Ces 4 niveaux ne présument pas des choix d'organisation et de management faits dans l'entreprise et ils ont vocation à évaluer la portée des concepts et techniques utilisés lors du développement de Produits et Services. Ils auront nécessairement un impact sur les choix d'organisation, de management et la définition des rôles et responsabilités.

4.1 VOCABULAIRE LIE AUX APPROCHES

Approche Prédictive :

Approche pour laquelle le périmètre projet, sa planification et son coût peuvent être déterminés dès les premières phases du projet. Toute évolution de ce cadrage est soigneusement gérée. Les enjeux et objectifs du projet sont en général clairement définis pour son lancement. Un exemple d'approche prédictive couramment utilisé dans le domaine automobile est l'approche Waterfall ([Project Management Institute](#)).

Approche Adaptative :

Approche pour laquelle le périmètre projet est évolutif, l'organisation des activités itératives et les livrables peuvent être obtenus de manière incrémentale. Ce type d'approche permet de s'adapter aux évolutions d'enjeux et objectifs du projet ou en cas d'incertitude élevée quant aux solutions à mettre en œuvre.

Approche Hybride

Approche basée sur une combinaison d'éléments des approches Prédictives et Adaptatives.

AGILE : Qui a de la facilité et de la rapidité dans l'exécution de ses mouvements. Prompt dans les opérations intellectuelles [\[Définition Robert\]](#).

AGILE Manifesto : Le Manifeste pour le développement agile de logiciels ([The Manifesto for Agile Software Development](#)) est un texte rédigé en février 2001 par dix-sept experts du développement d'applications informatiques. Ces experts estimaient que le traditionnel cycle de développement en cascade ne correspondait plus aux contraintes et aux exigences des organisations en évolution rapide. Il est constitué de 4 valeurs et de 12 principes qui définissent une Approche.

Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire. Ces expériences nous ont amenés à valoriser :



Les individus et leurs interactions *plus que les processus et les outils*

Des logiciels opérationnels *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*

Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers.

Approche AGILE : Ensemble des valeurs et principes du Manifeste Agile qui se déclinent en différents « Framework » regroupant Processus, Méthode (Règles et Pratiques) et Outils de développement, avec les adaptations nécessaires de l'organisation de l'entreprise et de son management. Initialement utilisée dans le domaine du logiciel, leur domaine de déploiement tend à s'étendre et à se généraliser sur l'ensemble des métiers de la conception et de l'opération des Produits et Services.

Agilité à l'échelle : Approche visant à appliquer les approches AGILE au-delà d'une équipe. Dans cette perspective, l'approche peut décrire le déploiement à de multiples équipes jusqu'à une organisation complète.

4.2 VOCABULAIRE LIE AUX PROCESSUS, METHODES ET OUTILS

Artéfact : Dans son acception générale, Artefact désigne tout objet, processus ou effet produit par l'être humain. En Agile, les artefacts sont des informations qu'une équipe et les parties prenantes utilisent pour détailler le produit en cours de développement, les actions à mettre en place pour y parvenir ainsi que les mesures prises durant le projet [\[définition Altassian\]](#).

Backlog : liste priorisée des fonctionnalités qu'un produit/service doit contenir. Elle peut être raffinée en liste de tâches afin que les membres d'une équipe puissent développer la solution lors du sprint.

Baseline / Base de référence : une spécification ou un produit logiciel qui a été officiellement revu et accepté, qui ensuite sert de base pour des développements futurs, et qui ne peut être modifié que selon un processus formel de contrôle des modifications (ISO24765/IEEE610).

Configuration : ensemble de caractéristiques fonctionnelles et physiques corrélées d'un produit ou d'un service défini par les exigences relatives à la conception, à la réalisation, à la vérification, à l'utilisation et au soutien d'un produit ou d'un service (ISO 10007 : 2003).

Definition of Done : ensemble de règles définies, partagées et approuvées permettant de considérer un élément (exemple : User Story, tâche, livrable...) comme réalisé.

DevOps : est un mouvement en ingénierie informatique et une pratique technique visant à l'unification du développement logiciel (**Dev**) et de l'administration des infrastructures informatiques (**Opérations**), notamment l'administration système.

Apparu autour de 2007 en Belgique avec Patrick Debois, le mouvement DevOps se caractérise principalement par la promotion de l'automatisation et du suivi (monitoring) de toutes les étapes de la création d'un logiciel, depuis le développement, l'intégration, les tests, la livraison jusqu'au déploiement, l'exploitation et la maintenance des infrastructures. [\[définition wikipédia\]](#).

Le DevOps peut ainsi être décliné par métier ou application comme par exemple en cybersécurité avec le DevSecOps.

Enabler : éléments du backlog qui permet de prendre en charge les activités nécessaires pour étendre l'Architectural Runway pour fournir les futures fonctionnalités métier. Ces activités incluent le développement de l'exploration, de l'infrastructure, de la conformité et de l'architecture.

EPIC : un epic est un large corpus de travail qui peut être décomposé en plusieurs user stories plus petites, chacune livrable sur une période donnée. Il correspond souvent à un sujet de grande envergure, avec une portée et un impact significatifs nécessitant l'alignement des stakeholders.

Sprint : période courte délimitée dans le temps, durant laquelle les différents membres d'une équipe vont travailler sur un objectif détaillé et atteignable basé sur une sélection d'éléments de Backlog raffinée en tâches (sprint backlog).

User Story : une user story est utilisée pour capturer la description d'une fonctionnalité du point de vue de l'utilisateur final. C'est la plus petite unité de travail dans un framework Agile, conçue pour articuler la valeur apportée au client.

Validation : confirmation par des preuves objectives que les exigences pour une utilisation spécifique ou une application prévue ont été satisfaites (ISO 9000).

Vélocité : quantité moyenne de travail qu'une équipe Scrum réalise pendant un sprint ; elle est généralement mesurée en "points de complexité" (story points) en sommant les estimations de tous les éléments du backlog qui ont pu être livrés au cours du sprint. Il faut quelques itérations pour que l'équipe converge vers une Vélocité moyenne sur laquelle elle pourra se baser pour planifier les sprints à venir.

Vérification : confirmation par des preuves objectives que les exigences spécifiées ont été satisfaites (ISO 9000).

5 FRAMEWORKS

Nous avons choisi de rappeler brièvement les principaux Frameworks couramment déployés dans le monde Automobile que ce soit pour de l'agilité au niveau d'une petite équipe ou pour l'application à plus grande échelle.

5.1 SCRUM



Scrum est un framework ou cadre de développement de logiciel répondant aux préconisations du manifeste AGILE. Il décrit à la fois les éléments de processus, de méthode et certains outils.

5.1.1 HISTORIQUE

C'est en 1986 que la métaphore de la mêlée, SCRUM en Anglais, est pour la première fois mentionnée dans un article de Hirotaka Tekeuchi et Ikujiro Nonaka publié par Harvard Business Review [1]. Les auteurs s'interrogent sur la capacité du modèle de développement Produit traditionnel à adresser les nouveaux challenges de vitesse et flexibilité :

« The traditional sequential or "relay race" approach to product development—exemplified by the National Aeronautics and Space Administration's phased program planning (PPP) system—may conflict with the goals of maximum speed and flexibility. Instead, a holistic or "rugby" approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today's competitive requirements. »

Sur la base de différents retours d'expérience dans des domaines variés (Canon, NEC, Honda, Fuji-Xerox...), ils proposent un processus de développement caractérisé par : une part d'instabilité, des équipes auto-organisées, des phases de développement se chevauchant, des apprentissages multiples (multi-niveaux (individu/groupe/organisation) et multi-compétences), un contrôle « subtil », et un transfert des apprentissages au niveau organisationnel.

En 1995, Ken Schwaber présente à la conférence de l'[OOPSLA](http://www.oopsla.org) les fondements de ce qui deviendra la méthode Scrum (Schwaber, 1995). En 2002, il publie avec Mike Beedle « *Agile Software Development With Scrum* » et en 2010, avec Jeff Sutherland, les principes de la méthode dans le « *Guide Scrum* » (version en ligne www.scrum.org).

5.1.2 DESCRIPTION DU FLUX D'ACTIVITE SCRUM:

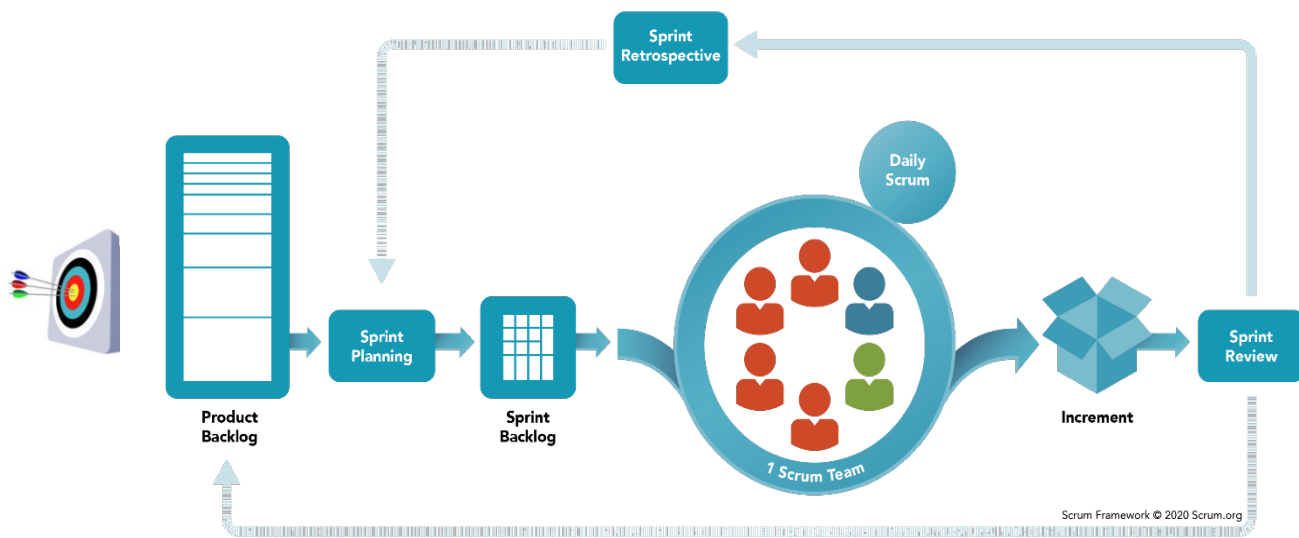


Figure 3 – Flux d'activité selon Scrum Framework @ Scrum.org

Le product backlog correspond à une liste hiérarchisée par la valeur gérée par le product owner.

Le sprint backlog est issu de la cérémonie de sprint planning. À cette occasion, le Product Owner présente l'ensemble des fonctionnalités souhaitées pour le sprint. Les développeurs raffinent ces fonctionnalités en tâches afin de s'assurer de la bonne compréhension et pouvoir mieux décider de ce qui peut être fait durant le sprint à venir. Ils constituent ainsi le sprint backlog. L'objectif du sprint est défini par l'équipe entière : Product Owner et Equipe de développement.

L'équipe scrum se réunit quotidiennement lors du « daily scrum » sur un temps court pour communiquer de façon transparente sur les tâches en cours, celles qui vont être difficiles à atteindre et les tâches à venir.

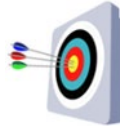
L'incrément est l'état du produit après un sprint. Son état est démontré lors de la revue de sprint et permet de terminer les tâches selon des critères d'acceptation définis en amont.

La rétrospective offre à l'équipe un temps de prise de recul sur ses pratiques pour définir ensemble des actions d'amélioration à intégrer dans le sprint suivant, ces actions sont intégrées au sprint planning.

Concernant les rôles :

- Le Product Owner a une vision claire. Il est responsable de gérer le product backlog, les livraisons et de la gestion des parties prenantes (utilisateurs, clients, direction de l'organisation...). Le Product Owner est responsable de s'assurer que l'équipe de développement apporte une valeur ajoutée au client.
- Le scrum master est un leader-serviteur. Il aide à l'adoption de l'approche Agile, est responsable qu'elle fonctionne correctement, il aide le Product Owner à définir la valeur, les développeurs à fournir de la valeur et l'équipe à s'améliorer.
- Les membres de l'équipe de développement correspondent aux personnes qui réalisent le travail.

5.1.3 POINTS CLES DU FRAMEWORK :



Valeurs	Courage, focus, engagement, respect, ouverture
Piliers	Transparence, inspection, adaptation
Rôles	Scrum master, product owner, développeurs
Rituels	Sprint planning, daily scrum, sprint review, sprint retrospective
Artefact	Product backlog, sprint backlog, increment
Management visuel	Scrumboard réinitialisé à chaque itération
Métrique	Vélocité de l'équipe
Développement	Itération fixe et courte (sprint de quelques semaines)
Livraison	En fin de sprint
Philosophie de gestion du changement	Les changements au cours d'un sprint doivent être autant que possible limités

Tableau 1 - Point clé du Framework SCRUM

5.2 KANBAN

Apparue au milieu des années 2000, la méthode Kanban est moins prescriptive que certaines approches agiles et moins perturbatrice, car il s'agit de l'approche originale « "start where-you-are" ». Les équipes de projet peuvent commencer à appliquer la méthode Kanban avec une relative facilité et progresser vers d'autres approches agiles si elles le jugent nécessaire ou approprié.

5.2.1 HISTORIQUE

L'origine de la méthode Kanban remonte aux années 1950 où Toyota utilisa un système de carte « étiquette » ou Kanban (カンバン) pour optimiser ses niveaux de stocks de matériau ou produit intermédiaires sur ses chaînes de production. Ainsi, lorsqu'un conteneur ou un stock intermédiaire était vide, un Kanban indiquant la référence du matériau, la quantité attendue, le client, le fournisseur... était remonté en amont dans la chaîne d'approvisionnement permettant ainsi d'ajuster au mieux ces stocks tampons sans risquer un arrêt de production. Ces Kanbans ont ensuite évolué vers des solutions électroniques basées sur des codes-barres ou RFID.

Les équipes de développement logiciel ont repris ce principe pour décrire les tâches à réaliser sous forme de carte Kanban et ainsi suivre la progression du travail de façon visuelle. On dit souvent que Kanban est orientée flux continu. C'est donc un processus et une méthode de développement simplifiée qui se base sur un outil « le tableau de bord Kanban ».

5.2.2 DESCRIPTION DU FLUX D'ACTIVITES KANBAN

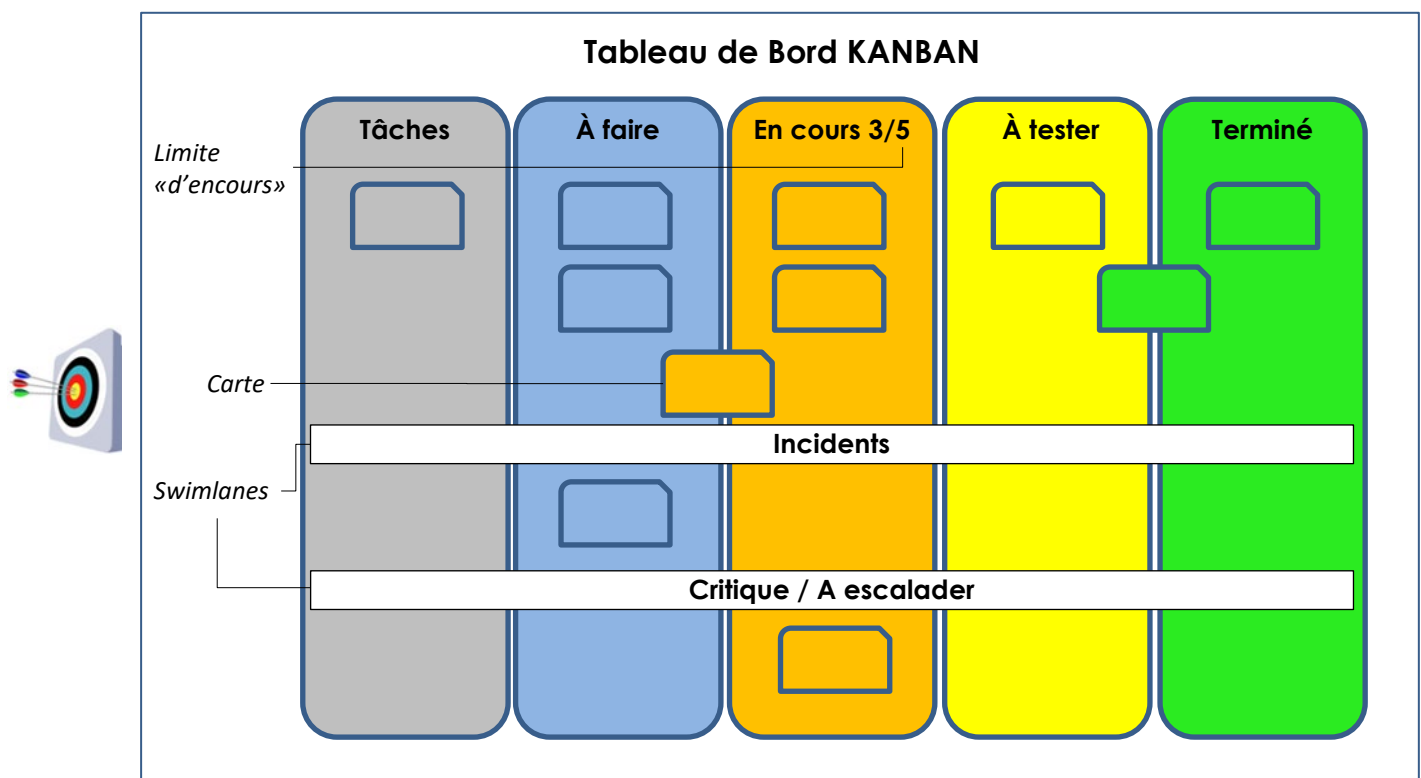


Figure 4 – Flux d'activité KANBAN représenté selon un Tableau de bord KANBAN

La méthode Kanban peut séduire par sa simplicité de mise en œuvre au travers d'un flux continu de développement (comparativement à Scrum orienté Itération) géré à l'aide du tableau de bord Kanban.

Six pratiques fondamentales :

- Visualiser le flux de travail, avec si besoin des sections pour segmenter le travail à réaliser (notion de «Swimlanes » Figure 4),
- Limiter le travail en cours (limite d'encours ou Work In Progress WIP),
- Gérer le flux en identifiant et en supprimant les goulots d'étranglement,
- Rendre les normes et règles du processus claires et explicites telles que par exemple la gestion du tableau de bord Kanban,
- Mettre en œuvre des boucles de rétroactions,
- Améliorer de manière collaborative.

Un point souvent discuté est la manière d'établir la limite d'encours ou WIP. Cette limite permet à l'équipe de se concentrer sur les tâches lancées et de les finaliser avant d'en lancer de nouvelles. Elle permet également de communiquer aux autres parties prenantes que la capacité de travail de l'équipe de développement est limitée.

5.2.3 POINTS CLES DU FRAMEWORK



Valeurs	Transparence, respect, compréhension, leadership, flux, focalisation sur le client, accord-entente, équilibre, collaboration
Piliers	Principes liés à l'amélioration continue (Kaizen) : • Commencez par ce que vous faites, • Acceptez de poursuivre des changements graduels et évolutifs, • Encouragez les initiatives à tous les niveaux, Principes liés à la prestation de services : • Concentrez-vous sur les besoins et les attentes du client, • Gérer le travail et non les travailleurs, • Évaluez régulièrement le réseau de services Pratique liée à la préservation du flux : • Respect de la limite d'en cours ou WIP
Rôles	Pas de rôle imposé
Rituels	Pas de rituels imposés
Artefact	Tableau de bord, Ticket/Carte
Management visuel	Kanban board continu
Métrique	Délai de traitement d'une demande, temps de cycle
Développement	En flux continu tiré
Livraison	En continu
Philosophie de gestion du changement	Possible à n'importe quel moment

Tableau 2 - Point clé du Framework KANBAN

5.3 LES FRAMEWORKS D'AGILITE A L'ECHELLE



Lorsque les développements nécessitent de faire travailler ensemble plusieurs équipes, on parle d'**AGILE à l'échelle** et l'application des frameworks de base Scrum ou Kanban n'est pas suffisante. Le choix d'un framework devra être fait en portant une attention particulière à l'adaptation au contexte de l'entreprise et au périmètre de déploiement visé.

Parmi les frameworks les plus répandus dans le domaine Automobile nous pouvons citer : **Scrum of Scrums**, **LeSS** pour Large Scale Scrum et **SAFe** pour Scale Agile Framework. Le rapport State of Agile 2022 en dresse la répartition d'usage suivante :

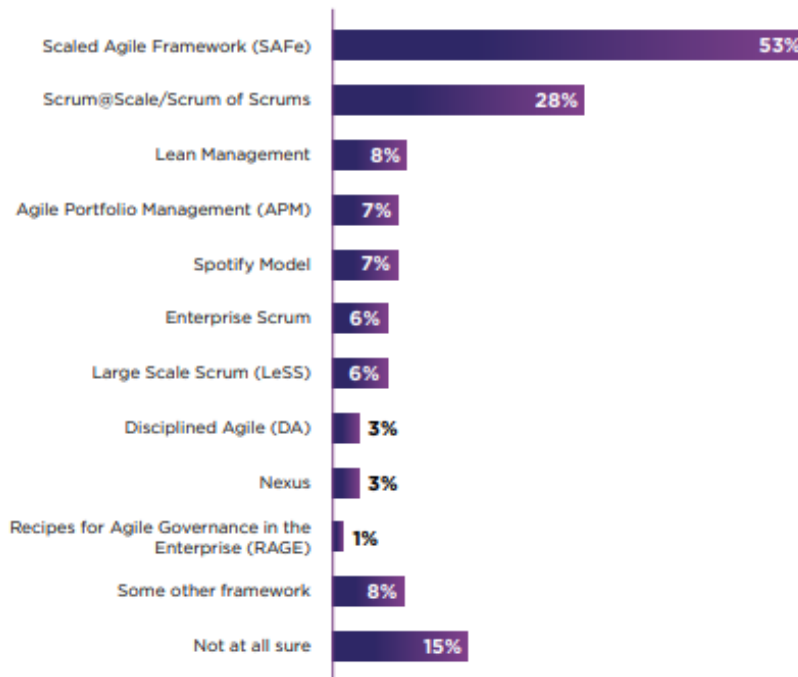


Figure 5 - Répartition de l'usage des Frameworks par entreprise selon StateOfAgile 2022

Certaines entreprises développent également leur propre Framework en s'inspirant de ceux existants et en les adaptant à leur propre contexte et leurs besoins particuliers.

5.3.1 SCRUM OF SCRUMS

La construction de ce Framework date de 1996 où Jeff Sutherland et Ken Schwaber ont eu tous deux besoins de synchroniser le travail de plusieurs équipes (Sutherland, 2001).

Les auteurs s'appuient sur plusieurs études dont celle de Richard Hackman et Neil Vidmar (Hackman & Vidmar, 1970) qui dès 1970 démontrent que la qualité des interactions dans un groupe est liée à quelques critères clés tels que l'homogénéité des membres du groupe ou le nombre de membres du groupe. Pour ce dernier critère, 4,6 personnes constituent une « équipe de taille optimale ».

Afin de maintenir une efficacité globale élevée, Sutherland et Schwaber proposent de créer une équipe « Scrum of Scrums » composée avec des délégués de chaque petite équipe qui coordonnera la livraison finale au client au travers d'une solution entièrement intégrée à la fin de chaque sprint.

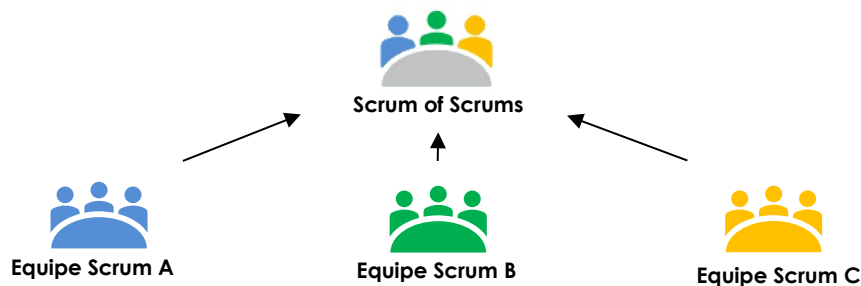


Figure 6 - Organisation du Scrum of Scrums (Altlassia)

Ces premières expériences sont aujourd'hui consolidées et décrites sur le site <https://www.scruminc.com/> et un guide est accessible en ligne <https://www.scrumatscale.com/scrum-at-scale-guide-online/>.

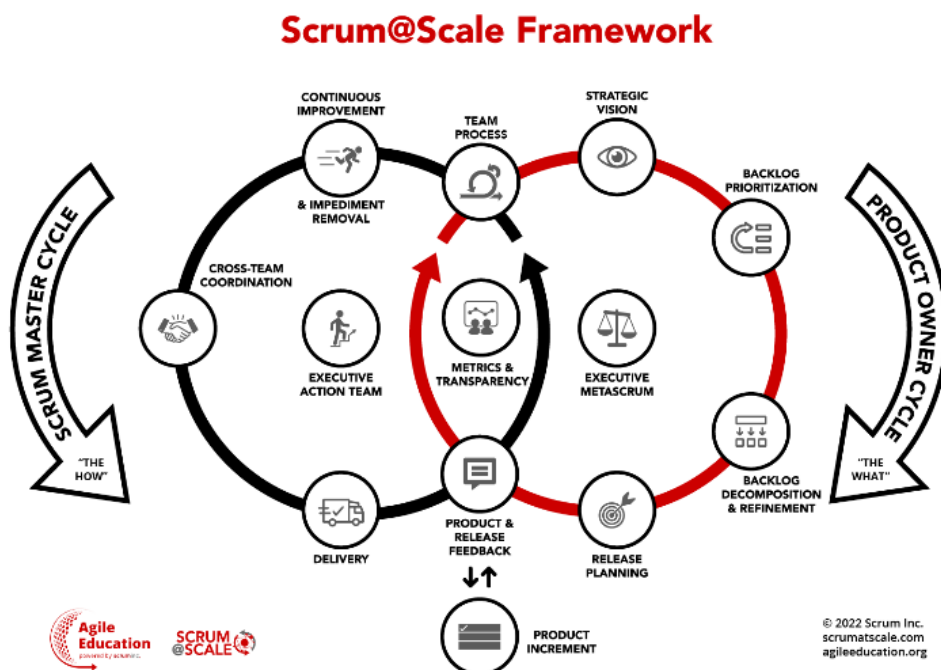


Figure 7 - Framework Scrum@Scale © 2022 Scrum Inc.

Elles ont également donné naissance à de nombreux autres Frameworks de passage à l'échelle dont SFA et LeSS les plus déployés dans le domaine automobile.

5.3.2 LE FRAMEWORK LESS

LeSS a été développé par Bas Vodde et Craig Larman, formalisé en 2013. Sa description détaillée est disponible sur le site <https://less.works/>.

LeSS propose un cadre léger très proche de Scrum avec quelques règles supplémentaires permettant l'organisation du travail entre plusieurs équipes et leur alignement : Sprint Planning 1 et 2, Overall Product Backlog Refinement, Sprint Overall Restrospective...

LeSS séduit souvent par sa relative simplicité de mise en œuvre.

5.3.3 LE FRAMEWORK SAFe

SAFe a été développé par Dean Leffingwell dans la même période. Sa description détaillée est disponible sur le site <https://www.scaledagileframework.com/>.

SAFe propose des principes, des pratiques et des compétences pour développer l'Agilité d'Entreprise, définie comme « la capacité de rivaliser avec la concurrence et de prospérer à l'ère du numérique en faisant rapidement face aux évolutions du marché et en saisissant les nouvelles opportunités grâce à des solutions métier innovantes et numériques » (© Scaled Agile, Inc). Le référentiel se décline en différentes configurations, adressant les différents niveaux de l'entreprise :

- *Essential SAFe* aborde l'agilité d'équipe et les pratiques permettant à un regroupement de plusieurs équipes (Agile Release Train ART) de collaborer efficacement au développement d'un même Produit, dans une logique de livraison en continu de type DevOps,
- *Large Solution SAFe* aborde le développement de solutions très complexes, nécessitant la collaboration de multiples Agile Release Trains et/ou fournisseurs, en conciliant la logique de livraison en continu avec les problématiques spécifiques d'ingénierie des grands systèmes soumis à des contraintes réglementaires et/ou de longue durée de vie,
- *Portfolio SAFe* aborde la définition et le pilotage d'un Portefeuille Produits, en alignement avec les orientations stratégiques de l'entreprise, à travers des pratiques de "Lean Portfolio Management" centrées sur l'optimisation en continu des chaines de valeur, l'ajustement régulier des moyens alloués (budget), et des mécanismes de gouvernance assurant la rapidité des prises de décision.

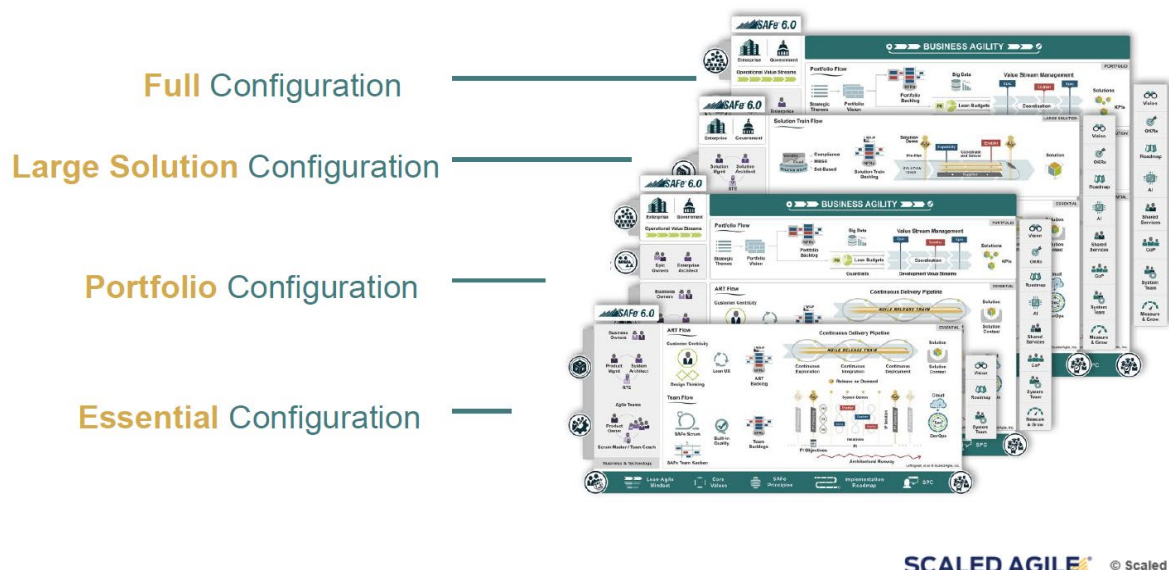


Figure 8 - Déclinaison des configurations de SAFe aux différents niveaux de l'Entreprise.

5.3.4 POINTS CLES DE COMPARAISON ENTRE LeSS ET SAFe

Des rituels supplémentaires sont mis en avant pour permettre le passage à l'échelle :

- Rituel d'Alignement : permet aux équipes et parties prenantes de partager une mission et une vision communes : ce sont les Sprint Planning ou Planning Interval (PI),
- Rituel de Synchronisation : mis en avant par SAFe, il permet aux équipes de se synchroniser sur le périmètre de l'itération, d'identifier et de gérer les dépendances, d'élaborer le planning détaillé et de gérer les risques de l'itération.

	LeSS	SAFe
Périmètre	Pertinent dès plus d'une équipe	Pertinent de quelques équipes à de nombreuses équipes
Responsabilité Valeur		Business Owner
Responsabilité du contenu Produit	Product Owner	Product Management / Product Owner
Alignement des équipes	Sprint Planning 1 et 2	Planning Interval (PI)
Architecte	Non formalisé	System Architect
Synchro des équipes	Non formalisé	ART Sync Event
Livraison	Livraison potentielle d'un Incrément	
Fonctionnement des équipes	Équipe Scrum	Équipe Scrum et/ou Kanban

Tableau 3 - Comparaison des Frameworks LeSS et SAFe

5.4 APPROCHE HYBRIDE

Une approche Hybride résulte de la combinaison de cycles adaptatifs (agiles) et prédictifs (en V). Les développements des systèmes automobiles sont habituellement cadencés autour de « logiques » de développement et de « gates ». Ces derniers jalonnent et organisent les grandes étapes du développement, depuis l'intention produit en amont jusqu'à la réalisation d'un objet validé dans un système industriel capable en aval. Cela permet à tout l'écosystème (clients/fournisseurs des domaines hardware/software) du projet de se synchroniser compte tenu des nombreux processus se déroulant en « simultané » (Ingénierie concourant), parfois à grande échelle ou liant différents portefeuilles de développements produit.

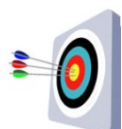
L'hybridation permet donc de bénéficier des méthodes et outils Agiles en les intégrant dans les processus, soit dans certaines phases, soit sur l'ensemble du cycle. On peut aussi combiner des domaines qui suivent une approche classique (exemple un constructeur) avec un autre qui se structure en agile (ex un fournisseur) sous réserve de bien travailler les points de synchronisation.

Ainsi la combinaison peut être temporelle, les approches alternant dans le temps en fonction du niveau d'incertitude ; ou structurelle, certains niveaux du système ou de management étant pilotés en mode adaptatif et d'autres en mode prédictif selon leur maturité propre ou en fonction du type de pilotage décidé.

Le mode Hybride peut également être une réponse transitoire permettant de gérer les risques de changement pour une organisation souhaitant évoluer d'une organisation classique vers une organisation AGILE (voir Partie A.6.3).

5.4.1 DIFFERENTS MODES D'HYBRIDATION

Les modes d'hybridation sont multiples. Le PMI [Project Management Institute](https://www.pmi.org/) décrit plusieurs exemples dans son PMBOK Guide V7 (PMI, 2022) en combinant Approche Agile et Approche Prédictive classique (ex. Waterfall) :



- option 1 - hybridation top-down : gouvernance Waterfall et équipes délivrant en AGILE,
- option 2 – hybridation temporelle : option 1 puis Waterfall par la gouvernance quand le niveau d'incertitude semble adapté à une Approche Prédictive,
- option 3 – hybridation au niveau des équipes : option 1 avec au niveau des équipes certaines délivrant en AGILE et d'autres en Approche prédictive.

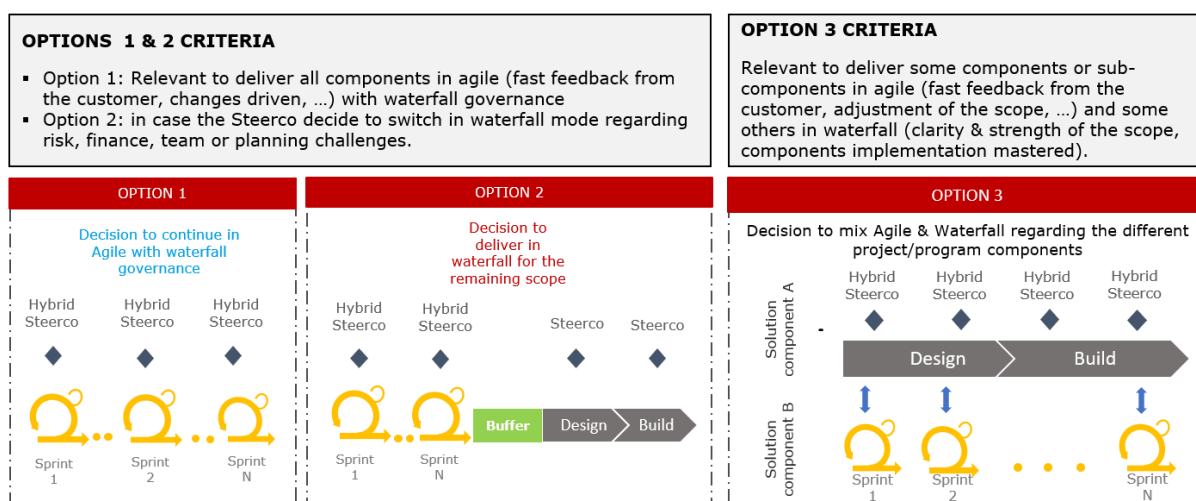


Figure 9 - Différents modes d'hybridation possible entre Agile et Waterfall @PMI 2022.

6 IDENTIFICATION DES POINTS DE VIGILANCE VIS-A-VIS DE L'APPROCHE AGILE :

6.1 CARACTERISATION DES CONTEXTES D'APPLICATION

Plusieurs outils existent et permettent de caractériser des contextes d'application que ce soit dans l'objectif général de résolution de problèmes et de prise de décision, que la mise en place d'Approches de Management Projet ou de choix d'organisation :

- [Le Cynefin Framework :](#)

Le modèle Cynefin, développé par Dave Snowden chez IBM en 1999 est assez général. Il permet de caractériser le niveau de complexité d'un problème et propose des Démarches et des typologies de Pratiques adaptées à une bonne prise de décision.

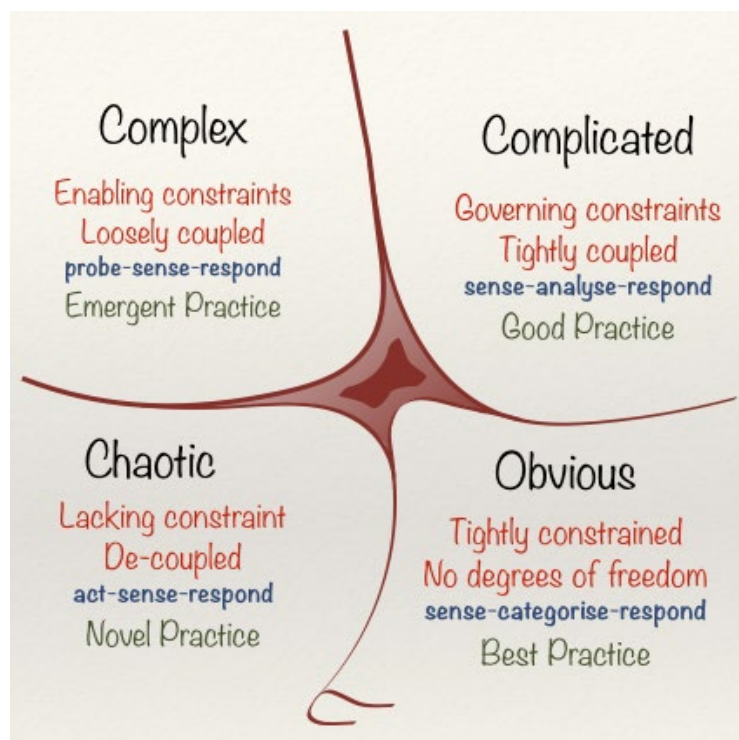


Figure 10 - Domaines du Framework Cynefin

- [Matrice de Stacey :](#)

Issue des travaux de Ralph Stacey (1948-2021) sur le management des organisations, l'outil vise à identifier les facteurs de complexité avec :

- Un niveau d' « Agreement » souvent associé à la notion d'incertitude sur la définition du besoin en management projet (axe vertical),
- Un niveau d' « Uncertainty » souvent reliée à la notion de l'incertitude sur le Comment (axe horizontal).

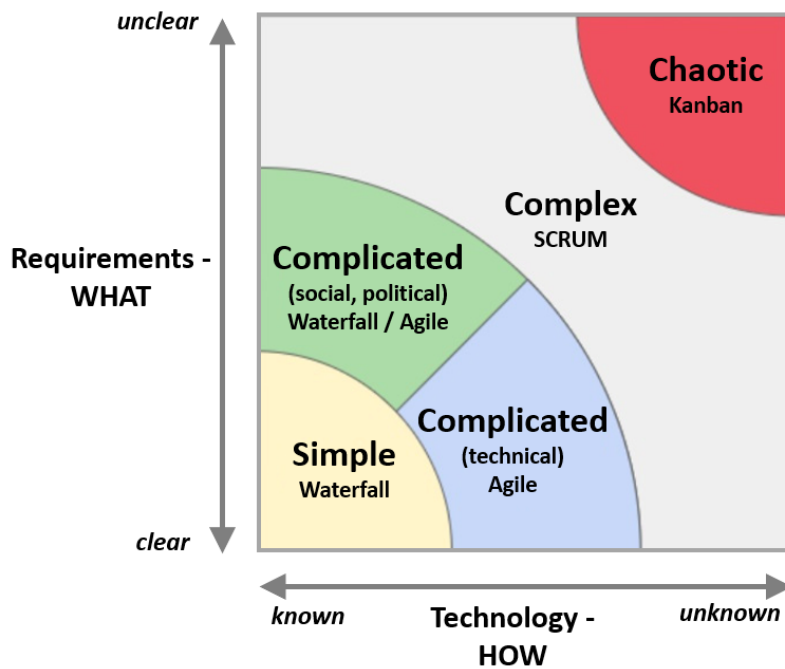


Figure 11- Matrice de Stacey adaptée aux managements projets logiciels

Cynefin et Stacey sont très centrés sur le niveau de maîtrise des chaînes causales dans les contextes d'application visés.

- **Des Matrices multi-critères** : de plus nombreux critères de caractérisation peuvent être utilisés. Celui proposé en **Figure 12 / Annexe 3** considère :
 - Des critères d'incertitude sur la Nature et les Enjeux Produit,
 - Des critères d'incertitude sur les dépendances organisationnelles et contraintes d'environnements.



Par méconnaissance et/ou manque d'information, les Approches Agiles peuvent être perçues comme un outil permettant de résoudre plus rapidement des situations de crises, au même titre que les Task Forces. Or ce sont bien des outils différents, avec des modalités d'utilisation spécifiques :

- **Task Force** : dans les situations de crise, ressources et budget non limités, actions hors procédure, outils de résolution de problèmes,
- **Approches Agiles** : pour les applications tout au cours du développement, avec ressources et budget fixés, accélération de la livraison de valeurs par optimisation du flux et focalisation sur le juste besoin, dans le respect des procédures (Built In Quality).

6.2 IDENTIFICATION DES POINTS D'ATTENTION



Afin de mieux appréhender les éléments considérés comme plus ou moins favorables aux approches agile ou classique, 12 caractéristiques sont proposées sur la figure ci-après. Elles sont regroupées en 2 catégories, intitulées "enjeux produit" et "enjeux organisationnels". Pour chaque caractéristique, le risque associé à la mise en œuvre d'une approche classique ou d'une approche agile est analysé. Une case grisée signifie que la situation semble peu probable. En revanche, les cases bleues permettent, pour chaque situation, de caractériser l'environnement du projet (exemple de positionnement via les "x"). Avec le code couleur jaune pour l'approche classique et bleu pour l'approche agile, les pouces levés permettent de préciser les bénéfices, alors que les pouces baissés les risques.

Bénéfices et risques différenciants associés aux approches agiles ou classiques						
		Faible			Fort	
Caractérisation du périmètre produit ou du projet		--	-	0	+	++
ENJEUX PRODUIT	1	Instabilité des exigences et des demandes produit	👍 x			👍
	2	Capacité de modifier à faible coût le Produit et à livrer partiellement	👎	x		
	3	Taux de nouveauté vs. standardisation	👍			x 👍
	4	Enjeu de rupture planning				x 👍
	5	Quantité d'innovations peu matures à intégrer				x 👍
ENJEUX ORGANISATIONNELS	6	Autonomie de compétences de l'équipe pour délivrer le produit ou le projet	👎	x		👍
	7	Capacité à avoir des lots techniques indépendants sur le projet	👎		x	
	8	Capacité de «design change» sans impact contractuel	👎			
	9	Autonomie de décision sur le terrain et subsidiarité	👎			x 👍
	10	Possibilité d'optimiser le niveau d'exigence documentaire	👎		x	
	11	Capacité du processus et des moyens de validation à valider en boucle courte les changements	👎			x 👍
	12	Maturité digitale de l'organisation	👎		x	

x caractérisation de l'environnement du projet

Bénéfice / risque projet classique

Bénéfice / risque projet agile

👍👍

👍👎

Figure 12- Grille d'identification des points de vigilance liés aux approches Agile ou Classique.

Afin de faciliter l'interprétation de ce tableau, un exemple "enjeu produit" et un exemple "enjeu organisationnel" sont proposés ci-après :

- Enjeu 3 "Taux de nouveauté vs. Standardisation" : selon la nature du produit à développer, un taux plus ou moins important de nouveautés est envisagé. Si ce taux est faible (fort carry-over dans une logique de "rafraichissement", nouvelle génération d'un produit existant avec peu d'innovations, etc.), une approche classique sera considérée comme bénéfique. A contrario, si le taux de nouveautés est important, une approche agile sera considérée comme positive, et l'approche classique sera jugée risquée.

- Enjeu 12 "Maturité digitale de mon organisation" : si cette notion de maturité numérique ne semble pas spécifiquement apporter de bénéfice particulier à un contexte classique ou agile, une faible maturité est considérée comme un véritable risque pour la mise en place d'une approche agile. En effet, des changements fréquents pouvant intervenir, y compris tardivement, la gestion des impacts et le fait de maîtriser les activités nécessitant une mise à jour partielle semble essentielle. Cette gestion des impacts, avec une granularité fine, paraît être une des clés pour permettre les modifications induites pour une approche agile.

6.3 GESTION DES RISQUES LIES AUX EVOLUTIONS D'APPROCHE OU A LA TRANSFORMATION DE L'ORGANISATION

6.3.1 NECESSITE D'UN PILOTAGE DU CHANGEMENT

Les évolutions d'Approche de gestion projet ou de transformation du mode de pilotage d'une Organisation nécessitent une gestion du changement structurée. Plusieurs options sont possibles :

- Des méthodes traditionnelles de gestion du changement : modèle ADKAR, 8 étapes de Kotter, courbe de changement de Kubler-Ross...
- Des roadmaps de transformation telle que celle proposée par [SAFe](#),
- La méthode ShuHaRi issue des arts martiaux japonais qui décrit 3 étapes d'apprentissage :
 - o Shu: protéger, obéir (守) : apprendre et appliquer les fondamentaux,
 - o Ha: se détacher, digresser (破) : respecter les valeurs, trouver les exceptions, optimiser, trouver de nouvelles approches,
 - o Ri : quitter, se séparer (離) : transcender, optimiser les approches voire sortir du cadre théorique initial pour trouver ce qui est optimal au niveau de l'organisation.

L'organisation [DACH30](#) propose une transformation basée sur ces principes.

Au-delà des changements dans les pratiques d'ingénierie et de gestion projet, les approches Agiles peuvent entrer en friction avec l'état d'esprit, les règles tacites et la culture d'entreprise. Par exemple une entreprise qui promeut le « faire bien du premier coup » devra clarifier ce que cela signifie dans un contexte nouveau de livraisons itératives en boucle courte, sous peine de provoquer des incompréhensions, des conflits, des frustrations et peut-être pour finir un rejet de l'approche Agile.

À titre d'exemple, la réalisation d'un système prototype nécessite usuellement 6 semaines minimum. Dans le cadre d'un sprint Agile, il est demandé de réaliser un système prototype en 2 semaines. La première réaction est de refuser l'action puisqu'usuellement 6 semaines minimum sont nécessaires. Cependant le niveau d'exigence et fonctionnalité pour chaque itération en Approche Agile est plus faible que lors d'un déroulement de projet classique. De fait le délai pour obtenir un nouveau système prototype peut-être raccourci. Le besoin de recul et remise en cause est donc nécessaire afin de ne pas abandonner ou faire échouer le projet.

L'exemple de ce délai de prototypage peut bien entendu être transposé aux outillages ou à des procédés.



Il est capital de **sensibiliser** et **d'accompagner la direction et les managers** de proximité dans cette évolution culturelle, qui peut prendre beaucoup de temps. Certaines entreprises se dotent d'accompagnateurs du changement (parfois appelés 'responsables de transformation') et de **coachs pour faciliter cette phase**. Les services support devront eux-mêmes prendre en compte ces évolutions et intégrer ces nouvelles Approches pour être capables d'accompagner ce changement.

Dans l'étude « The impact of agility: How to shape your organization to compete » – Mai 2021 par Wouter Aghina, Christopher Handscomb, Olli Salo, and Shail Thaker McKinsey & Company, les résultats d'une enquête de 1000 répondants ont été étudiés. Dans le top 4 des éléments clés, on retrouve le fait que le top management prenne part à la transformation agile. Tout d'abord, le premier élément réside dans la phase de préparation. La direction et les managers doivent s'engager en consacrant suffisamment de temps pour se former et si besoin se faire accompagner pour maîtriser les concepts de l'agilité, comprendre les changements associés et se sentir capable de mener les actions de changements. Dans la phase de transformation, l'élément clé repose sur le fait que le top management s'investisse pour piloter l'organisation pour atteindre les objectifs de création de valeur liés à l'agilité et ne pas déléguer ce sujet en bottom-up.

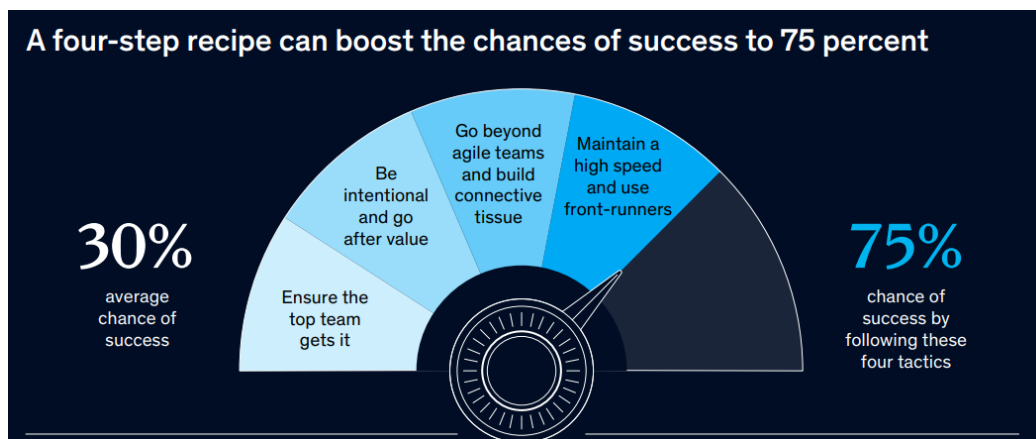


Figure 13 - Les 4 leviers principaux pour booster les chances de réussite de votre transformation AGILE selon l'étude McKinsey - The Impact of agility www.mckinsey.com.

6.3.2 TYPE DE TRANSFORMATION ET STRATEGIE D'ENTREPRISE :

Suivant la stratégie et les ressources de l'entreprise, la transformation sera soit globale, soit limitée à un certain périmètre : entité, projet ou portefeuille de projet de développement.

L'intégration et l'application des approches Agiles sont plus larges que le périmètre de l'équipe projet. L'entreprise doit donc prendre en compte l'impact de ces approches sur son organisation et fonctionnement, à titre d'exemples non exhaustifs :



- La décomposition des étapes projet,
- Le mode de pilotage projet,
- Les méthodes et fréquence de livraison des livrables,
- La définition du contenu des livrables,
- La constitution de l'équipe projet,
- La gestion des heures de ressources humaines,
- Le budget,
- La définition des contrats,
- La définition des jalons et des revues qualité,
- Le recrutement.

La transformation devra être progressive et itérative afin d'assurer une application des approches Agiles sans perturber la productivité de l'entreprise. De nouveaux métiers et compétences seront nécessaires, cela implique donc un suivi régulier et soutenu qui ne peut être obtenu sans un plan de formation et des plans de carrière (voir Partie B.8).

7 RECOMMANDATIONS ET BONNES PRATIQUES PAR THEMATIQUES

La suite du guide a pour objectif de partager des recommandations et bonnes pratiques par thématiques :

- Partie B : Agile et Qualité,
- Partie C : Agile et Sécurité (à venir en 2024).

Pour chaque thématique, les problématiques rencontrées lors de la mise en œuvre d'Approche AGILE sont traitées par :

- Un rappel des principes et des exigences à respecter,
- Une synthèse des avantages, difficultés et propositions de conditions de succès,
- Une description plus détaillée des Conditions de succès et de leurs implémentations.

PARTIE B. AGILE ET QUALITE

1 IDENTIFICATION DES PROBLEMATIQUES



Figure 14 - Thèmes des problématiques "Agile et Qualité"

2 AGILE ET BESOIN DU CLIENT

2.1 RAPPEL DES PRINCIPES ET DES EXIGENCES

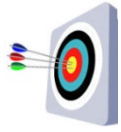
L'importance à accorder au besoin réel du client est une valeur cardinale des approches AGILES. Elle est de plus clairement établie dans le premier principe du Manifeste Agile :

- "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."
- Il en est de même pour les référentiels Qualité ISO9001 : 2015 ou IATF 16949 : 2016 au travers :
- Du leadership et de l'engagement de la direction d'une entité certifiée sur son "Orientation client" (§5.1.2),
- De la détermination et de la revue des exigences relatives aux produits et services avant de s'engager à fournir des produits et services au client (§8.2.2 et §8.2.3).

2.2 AVANTAGES ET DIFFICULTES

Le déploiement d'une approche AGILE implique également selon les principes 2 et 3 du Manifeste :

- « Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage »,
- « Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale ».



Ce dernier principe est également souvent associé à la notion de MVP (*Minimum Viable Product*) et de MMF (*Minimum Marketable Feature* - voir (Scaled Agile, Lean UX, 2023)), et permet de mieux prioriser les fonctions à livrer au plus tôt au client selon le niveau de valorisation. L'exposition des MMFs et/ou d'un MVP auprès du client permettra également de rapidement boucler sur le niveau de satisfaction de ce dernier et d'adapter la solution. Ce sont des avantages clés vis-à-vis de « l'Orientation Client » requise, en particulier dans un contexte de forte incertitude sur le besoin client, comparativement à une approche classique du cycle en V.

La gestion des exigences relatives aux produits et services selon les normes qualités implique quant à elle la détermination et la revue des exigences relatives aux produits et services afin que l'engagement à fournir des produits et services au client puisse in fine être concrétisé.

En synthèse, le développement par évolutions d'exigences successives devient donc le mode de fonctionnement « nominal » comparativement aux approches traditionnelles plus prédictives où il est considéré comme un mode dégradé. La gestion et revue des exigences doit être maintenue dans ce nouveau cadre.



Pour mener à bien de tels développements en approche AGILE et dans ce cadre normatif, nous identifions donc les conditions de succès suivantes :

- Prioriser les exigences pour se focaliser sur la valeur apportée au Client,
- Rendre les pratiques de Revues d'exigences compatibles aux nouveaux modes de développement de solution : itérations courtes, construction de solution incrémentale,
- Intégrer le rôle du Product Owner dans les revues d'exigences,
- Tirer pleinement parti du processus d'exposition rapide du Produit auprès du client afin de vérifier l'adéquation entre la compréhension acquise des exigences, leur priorisation, leur valorisation et le besoin réel du client,
- Lorsque nécessaire, optimiser le processus de gestion des évolutions puisque celui-ci devient de plus en plus fréquent.

La capacité à accueillir le changement est également très liée aux choix d'architecture, de technologies (aussi bien produit que procédé) et à leur capacité à accepter ces évolutions (voir 0).

Les points clés liés aux évolutions nécessaires dans la gestion de la relation Client-Fournisseur sont analysés en Partie B.7.

2.3 CONDITIONS DE SUCCES

2.3.1 ADAPTER LE PROCESSUS DE REVUE ET DE PRIORISATION DES EXIGENCES

Avant de s'engager à fournir un Produit, la Revue des exigences telle que requise par l'ISO9001 doit couvrir plusieurs types d'exigences :

- Des exigences fonctionnelles,
- Des exigences non fonctionnelles liées à des contraintes que devra respecter le Produit (contraintes d'environnement, conditions d'usage, profil de mission...),

- Des exigences non fonctionnelles liées à des contraintes légales et réglementaires, normatives.

Bien qu'il puisse exister des incertitudes sur l'ensemble de ces domaines d'exigences, la raison pour privilégier une Approche Agile est souvent liée aux incertitudes portant sur les exigences fonctionnelles, et à leur valorisation possible auprès du client. Les exigences non fonctionnelles sont souvent moins flexibles voire non négociables lorsqu'il s'agit d'exigences légales ou réglementaires. Ce sont donc ces différentes contraintes qui définiront les niveaux de flexibilité autorisés et à prendre en compte lors de ces Revues d'Exigences.

Il faudra également mettre en œuvre cette Revue dans le cadre du Contrat établi entre Client et Fournisseur (voir Partie B.7) en prenant en compte les modes d'élaboration des exigences : stabilisées en début de contrat ou élaborées en plusieurs phases de maturation.

Dans les cas de faibles niveaux d'incertitudes, une adaptation possible de la Revue pourra donc être de :

- Revoir les exigences stabilisées au plus tôt,
- Revoir les exigences non stabilisées et définir un plan d'action lorsque nécessaire,
- Prioriser l'ensemble des actions de développement et de gestion des risques avec un focalisation valorisation client.

Le plan d'action suite pourra contenir entre autres :

- Une enquête client complémentaire,
- Une démonstration rapide sur le marché auprès d'une clientèle ciblée d'une simulation, d'un prototype avec un certain niveau de fidélité,
- La mise à jour de la revue initiale liée aux exigences non stabilisées lors des Sprint Planning, PI planning à venir...



Pour les cas de plus fortes incertitudes, il faudra aller vers une revue « en continu » tout au long du développement, où à chaque itération ce qui est stabilisé et non-stabilisé sera revu et priorisé. Pour chaque exigence non-stabilisée, une analyse des risques permettra d'identifier les activités d'ingénierie (architecture, conception, implémentation) qui peuvent être menées en l'état, i.e. avant stabilisation, et/ou les points qui doivent être absolument clarifiés au préalable.

Dans les deux cas, les Backlog refinement meeting, Sprint Planning et PI planning sont des moments privilégiés pour contribuer à cet exercice et permettent aux différentes parties prenantes de clarifier leurs exigences.

Le Product Owner sera clé pour l'exercice puisqu'il est le principal représentant du client au sein de l'équipe et est chargé de maximiser la valeur fournie par l'équipe en s'assurant que la backlog reste alignée avec les besoins des clients et des parties prenantes. Mais il ne pourra à lui seul assurer la représentativité de l'ensemble des parties prenantes (usage, sécurité, maintenabilité...).

Un écueil fréquemment rencontré est la difficulté de faire vivre en parallèle et de maintenir en cohérence :

- Une structure de backlog par essence très dynamique et volatile,
- Une structure documentaire formelle & et à évolution très contrôlée (car souvent liée à des aspects contractuels et/ou réglementaires) définissant les spécifications applicables du produit.





Pour faciliter cette gestion, on cherchera à réduire le plus possible les redondances de contenu entre les deux structures, et à maintenir la synchronisation via des liens/renvois entre les outils (par exemple chaque élément de backlog renvoie vers la liste des exigences détaillées associées à cette fonctionnalité dans l'outil de gestion des spécifications), ou à adopter une **chaîne intégrée d'outils d'ingénierie** qui assure cette synchronisation par construction.

L'approche AGILE permettra donc globalement de gagner en efficacité de communication en renforçant les échanges directs, en évitant les communications en cascade habituelles et les aléas associés : mauvaise compréhension, interprétation... et les redondances d'informations si possible au moyen de chaîne d'outils permettant une continuité numérique.

2.3.2 DEMONSTRATIONS RAPIDES EN CLIENTELE

Les démonstrations rapides en clientèle peuvent permettre :

- D'améliorer la compréhension des besoins du client :
 - o Le parcours client constitué par l'ensemble des cas d'usage sur l'ensemble du cycle de vie Produit,
 - o La confirmation des exigences à priori stabilisées mais qui pour certaines évolueront après confrontation du client à la solution proposée,
 - o La confirmation ou évolution des exigences non stabilisées,
 - o Ainsi que le niveau de valorisation et de priorité à retenir sur l'ensemble de ces exigences.

Il sera utile de définir les objectifs précis de l'exposition et les méthodes d'évaluation associées, ainsi que de pouvoir recueillir de façon ouverte les satisfactions et insatisfactions éprouvées par les utilisateurs,

- D'améliorer l'évaluation de risques techniques (voir 0 et Partie B.6),
- D'assurer un bouclage avec les différentes parties prenantes lors des démonstrations.

Ces démonstrations doivent rester maîtrisées d'un point de vue du respect des exigences légales et réglementaires, de la sécurité des utilisateurs et de la satisfaction client. Cette maîtrise passera par :

- La vérification du respect des exigences légales et réglementaires selon la phase de développement,
- Une identification des risques résiduels spécifiques à ces démonstrations et si besoin la mise en place des mesures de maîtrise de ces risques (voir 0 et Partie B.6),

2.3.3 GESTION DES EVOLUTIONS D'EXIGENCES

Afin de pouvoir accueillir pleinement les évolutions d'exigences du client, il est nécessaire de pouvoir en évaluer rapidement l'impact. Ceci nécessite quelques prérequis :

- Disposer d'une documentation précise de la configuration ou des configurations existantes afin de pouvoir rapidement analyser l'impact d'une évolution,
- Disposer d'un processus de gestion des évolutions efficace et efficient car il sera sollicité plus souvent en approche Agile,
- Être capable de prioriser les demandes d'évolutions par rapport aux développements en cours. Le pilotage de la Backlog le permettra plus facilement qu'en mode projet traditionnel.

Si l'on souhaite également pour certaines de ces évolutions les propager sur les produits déjà sur le marché, il sera en plus nécessaire :

- De connaître les configurations de l'ensemble des Produits mis sur le marché,
- De disposer des moyens économiques & techniques de mise à jour du parc Produits, et en particulier pour le logiciel, de produit et process compatibles OTA (Over The Air – Firmware/Configuration Over The Air FOTA ou COTA).



L'optimisation du processus de gestion des évolutions pourra prendre en compte les points suivants :

- Différencier les domaines de l'organisation stabilisés en Agile qui délivrent des évolutions produits itératives et fréquentes sur un périmètre donné, par exemple la mise à niveau de logiciels ADAS (Advanced Driver Assistance Systems ou Systèmes d'aide à la conduite) en clientèle, des domaines en charge de développer des sous-ensembles techniques concourant à un nouveau véhicule, par exemple une carrosserie, une nouvelle plateforme, ou une nouvelle ligne d'assemblage. La synchronisation des plannings et des logiques de développement est à bien maîtriser pour gérer la cohérence des évolutions entre ces différents domaines qui partagent des "assets techniques" communs,
- Réaliser l'analyse d'impact sur le bon périmètre :
 - o Pour les demandes d'évolutions provenant du client ou du business, l'analyse sera top-down permettant de couvrir l'ensemble d'un système, et des équipes AGILES,
 - o Pour les demandes d'évolutions locales (optimisation, amélioration continue...), le risque sera de ne pas détecter un impact sur d'autres lots du système, sur l'équipe « voisine », ou à un niveau Architecture. Il sera nécessaire de mettre en place des mécanismes « d'élargissement » de cette analyse d'impact d'une équipe à un ensemble d'équipes. La maîtrise de l'Architecte (exemple : System Architect dans SAFe) permet de prendre en compte cette problématique.
- Clairement évaluer le niveau d'impact afin de déclencher les actions au juste nécessaire : architecture, définition produit, définition processus de production, d'opération...
- En fonction du niveau d'impact, clairement identifier pour quelle release la livraison du client sera possible : release en cours « n », release n+1, release n+2... (voir Partie B.3.3.3).

3 AGILE ET PLANIFICATION

3.1 RAPPEL DES PRINCIPES ET DES EXIGENCES

Le manifeste AGILE préconise la capacité à répondre au changement plutôt que de suivre un plan.

Par ailleurs, les normes ISO9001 et IATF requièrent :

- Une planification et maîtrise opérationnelles des processus permettant de satisfaire les exigences clients (§8.1),
- Une planification de la conception et du développement (§8.3.2) au travers d'outils tels que l'APQP par exemple (§8.3.2.1-a).

3.2 AVANTAGES ET DIFFICULTES

Dans un environnement VUCA et face à un certain niveau d'incertitude, il est donc nécessaire de définir le bon compromis entre niveau de détail de la planification et capacité à accueillir le changement et à s'adapter rapidement.

Les conditions de succès identifiées sont les suivantes :



- Focaliser la planification sur la création de valeur,
- Définir le niveau de planification adapté,
- Définir les mécanismes de synchronisation permettant une planification à l'échelle.

3.3 CONDITIONS DE SUCCÈS

3.3.1 PLANIFICATION BASEE SUR LE FLUX DE VALEUR

Comme décrit dans le §2.2, les principes du Manifeste Agile et la logique MVP conduisent à prioriser et donc à macroscopiquement planifier selon le flux de valeur attendu par le client.



Mais cette priorisation souvent liée à des besoins fonctionnels ne doit pas faire oublier la prise en compte des besoins non fonctionnels (réglementaires, levée de risques, ou autres contraintes) et la planification associée. Ces besoins sont parfois moins visibles du client et de l'utilisateur final mais restent clés et souvent non négociables.

Des méthodes comme WSJF - Weighted Shortest Job First, popularisée par (Reinertsen, 2009), permettent d'intégrer l'ensemble de ces contraintes dans cette évaluation de la priorité d'un élément du backlog.

$$WSJF = \frac{\text{Coût du délais}}{\text{Durée de la tâche}} = \frac{CdD}{\text{Durée de la tâche}}$$

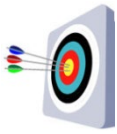
$CdD = \text{Valeur Business} + \text{Criticité temporelle} + \text{Reduction de risques et Création d'Opportunités}$

Par ailleurs cette logique de priorisation peut être étendue à un périmètre plus large que le simple projet, par exemple à un produit et ses évolutions voire une ligne de produit complète. Cela permet de mettre en place des organisations de développement qui restent plus stables dans la durée et qui favorisent l'efficacité, la montée et le maintien en compétence des équipes. On parle alors **d'Approche « Produit »** par opposition à une approche « Projet ».



3.3.2 PLANIFICATION ADAPTEE A L'HORIZON DE TEMPS

Le choix de l'Approche décrit en fin de Partie A.6 constitue une première décision quant au mode de planification à mettre en œuvre.



Ensuite, l'approche AGILE et les différents Frameworks proposent des principes de planification adaptés au contexte de forte incertitude. Il s'agit principalement de planifier aussi précisément que possible et nécessaire le court terme et de façon de plus en plus macroscopique le moyen et le long terme en s'adaptant donc au niveau d'incertitude. On parle de planification par vague « Rolling wave planification » ou de planification par horizon de temps multiple « Multiple horizon planning » [[Industrial DevOps](#)].

L'écueil d'une planification détaillée pour les horizons de temps lointains qui serait toujours fausse est ainsi évité. Et par ailleurs, l'écueil de non-planification sur le moyen long terme est évité en focalisant sur les éléments structurants pour ces horizons de temps.

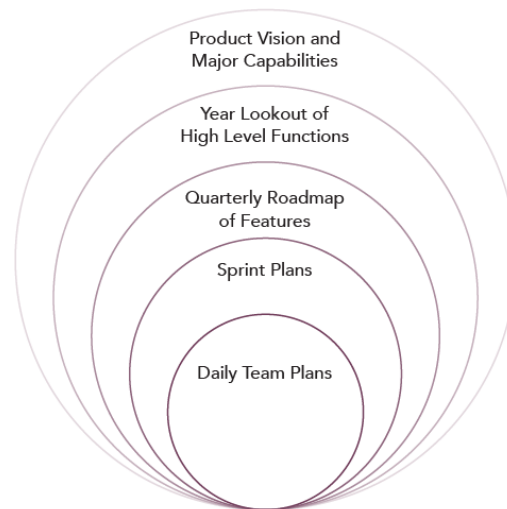


Figure 15 - "Multiple horizon planification" - Industrial DevOps

Le mode de planification, le niveau de détail de cette planification et les attendus associés doivent nécessairement être discutés entre les parties prenantes et faire l'objet d'un accord (voir §B8).

3.3.3 PLANIFICATION A L'ECHELLE DANS LE DOMAINE AUTOMOBILE

La planification doit également être adaptée au domaine de l'automobile et à ses spécificités telles que :

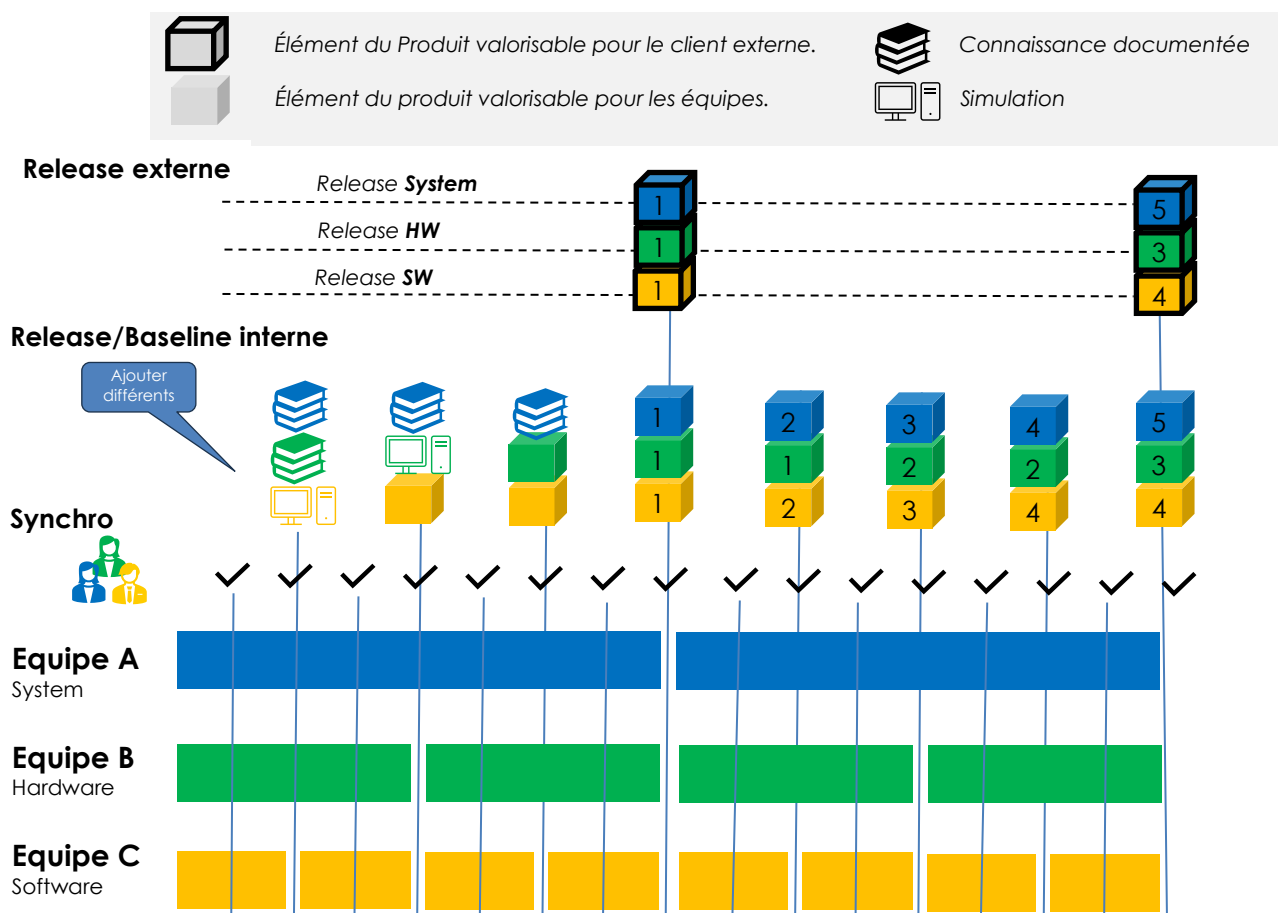
- Une forte complexité au sein de certains métiers, entre métiers (marketing, design, co-conception entre mécanique, chimie et matériaux, Hardware, Software...),
- Une complexité multi-niveaux (composants, sous-systèmes, systèmes, véhicules, industrie, opération...) avec des besoins supplémentaires de synchronisation et d'interfaçage robuste.

Même si des choix de conception permettent de plus en plus de découpler ces métiers et niveaux (voir Partie B.4.3), les synchronisations et le bon fonctionnement aux interfaces restent clés.

Le cadre de planification doit permettre la meilleure flexibilité possible et la bonne prise en compte des interdépendances entre ces domaines et niveaux.

Il pourra s'appuyer sur :

- La mise en place de mécanismes de synchronisation entre équipes et entre niveaux,
- Des Périodes de Synchronisation identiques,
- Des « Releases (ou Baselines) internes » sur des Périodes identiques, permettant à chaque équipe la livraison d'éléments de valeur au niveau de « l'équipe d'équipes » et nécessaire à l'avancement et à la cohérence globale (problématiques d'intégration, de validation, d'atténuation des risques, de convergence au jalon en mode Hybride...). SAFe parle d'itération et associe à chaque itération une boucle « PDCA : Plan-Do-Check-Adjust »,
- Des « Releases externes » ou mise en production valorisables pour le client externe (Constructeur Automobile, utilisateur final...) pouvant être Périodiques, « On Demand » ou ciblées sur un événement à une date donnée (démonstration clientèle, mise sur le marché...). Le périmètre de ces « Releases externes » peut-être variable pour s'adapter au besoin (ex : release complète de la Solution ou simple mise à jour d'un logiciel d'un sous-système) comme indiqué sur le schéma ci-dessous.



3.3.4 MAITRISE DU PROCESSUS DE PLANIFICATION

Dans les approches Agiles, la responsabilité du processus de planification n'est plus confiée à des rôles spécifiques (par exemple « chef de projet » ou « responsable de lot ») mais devient une responsabilité partagée par l'ensemble des membres de l'équipe : c'est l'équipe dans son ensemble qui va s'engager sur l'atteinte de certains objectifs à la fin d'une itération ou d'un incrément.

Selon le principe de « Sagesse des Foules » (voir livre éponyme de James Surowiecki) on s'attend à ce que les orientations, décisions et engagements pris collectivement par un groupe constitué d'expertises, d'expériences et de personnalités diverses soient en moyenne plus robustes et fiables que ceux provenant d'une personne seule. En pratique, toutefois, le changement n'est pas si simple : les équipes doivent s'approprier cette responsabilité nouvelle et acquérir les compétences qui leur permettent de l'exercer dans de bonnes conditions (voir §8.3.1). Les points essentiels à maîtriser sont :

- La priorisation (estimation du coût du délai et de ses différentes composantes business et techniques),
- L'estimation de la complexité des tâches, et des risques associés,
- L'estimation de la capacité de travail de l'équipe (généralement basée sur un historique de Vitesse (voir Partie A.5.1.3)).



Pour une équipe avec peu d'expérience, un accompagnement et un suivi de la montée en compétence sera nécessaire. La planification en boucles courtes et les mécanismes de rétrospective facilitent un ajustement rapide. L'objectif est d'éviter le sur- ou sous- engagement et d'atteindre un mode de fonctionnement pérenne, où l'équipe sait tenir avec un bon niveau de confiance des objectifs ambitieux tout en maintenant un effort soutenable dans la durée.

3.3.5 ADAPTATION DE LA PLANIFICATION DES LIVRABLES ET DE LEURS CONTENUS

Comme expliqué dans l'exemple du paragraphe précédent, la fréquence des livraisons internes et externes étant plus élevée, les produits associés acquérant progressivement l'intégralité de leurs fonctions et performances, ce changement de définition et des attendus implique donc la mise à jour en conséquence des livrables et des attendus qualité.

Il est donc nécessaire de rendre la formalisation des livrables plus itérative et progressive afin qu'elle accompagne et supporte les activités dans le respect des requis qualité voire réglementaire et afin d'éviter toute dette documentaire.

Ceci peut être assuré en intégrant ces attendus dans la définition du DoD voire avec une approche graduelle telle que celle proposée par SAFe du Scaled DoD (voir Partie B.6.3.3.1).



4 AGILE ET CONCEPTION

4.1 RAPPEL DES PRINCIPES ET DES EXIGENCES

4.1.1 PRINCIPES AGILES ET PRATIQUES DES FRAMEWORKS AGILES RELATIF A LA CONCEPTION :

Le Manifeste Agile contient deux principes dédiés à la phase de conception : D'une part "The best architectures, requirements and designs emerge from self-organising teams" (principe n°11). Il recommande de permettre aux équipes Agiles de prendre des décisions de conception de façon autonome. Et le second : "Continuous attention to technical excellence and good design enhances agility" (principe n°9). Il recommande de développer l'excellence technique de prêter attention à la conception qui permettra de recevoir des évolutions donc de faciliter l'Agile.

Le Framework SAFe d'Agilité à l'échelle adresse la conception Agile sur différents aspects pour le logiciel comme pour les autres disciplines en collectant et synthétisant des approches qui font leur preuve sur le terrain.

La méthodologie SAFe a produit fin 2021 un ensemble de consignes sur l'introduction de l'agilité dans un environnement hardware (voir <https://scaledagileframework.com/applying-safe-to-hardware-development/>). On trouve des recommandations sur les choix technologiques (préférer l'utilisation des FPGA et des SoC plutôt que des ASIC, permettre une évolution découplée des composants grâce à des interfaces stables), sur les méthodes de validation (valider chaque composant de façon autonome), et sur l'évolution de la mentalité.

SAFe met en avant l'exemple de conception des fusées SpaceX avec une nouvelle fusée livrée tous les 2 ou 3 mois.

SAFe insiste sur le "Design for Change", c'est à dire que la conception en Agile doit permettre l'évolution. Ce qui implique de trouver un juste milieu entre faire des choix de conception tôt mais qui ne limitent pas les évolutions futures donc l'Agilité du développement du produit. Ce point est très important quand les changements de conception sont difficiles comme en mécanique et électronique.

SAFe définit l'approche "Set-based" design par opposition au "point-Based" design. Le principe est de préserver les options de conception et progressivement, intégration après intégration, de converger vers une solution finale. Cela implique de considérer plusieurs solutions de conception, d'explorer, d'intégrer/tester souvent et de déployer en apprenant afin de converger vers la solution (voir <https://v5.scaledagileframework.com/set-based-design/>).

SAFe met en avant la nécessité d'une double conception d'architecture : d'une part descendante sous la responsabilité du "ART team" et ascendante venant des équipes Agiles et sous leur responsabilité. Il faut veiller à ce que les deux flux de conception se rejoignent et surtout qu'ils ne s'ignorent pas.

Par ailleurs, SAFe promeut l'approche "DevOps" (2007) qui cherche à avoir un flux continu entre développement (Dev) et production (Ops). SAFe reprend aussi le "Design thinking" (by R. Faste 1980) (voir <https://v5.scaledagileframework.com/agile-product-delivery/>), une méthode de design collaboratif centré sur le besoin client.

LeSS, le framework basé sur Scrum mais à l'échelle, en s'appuyant sur le principe Agile n°9 (voir en début de paragraphe), explique et donne des astuces et conseils pour faire de la conception dans un environnement Agile.

Dans une succession d'articles (Articles Less et [livres](#)) (voir <https://less.works/less/technical-excellence/architecture-design>) avec des retours d'expériences de Less sur la conception, ils insistent sur l'adaptation permanente de l'architecture par les personnes qui font le produit. Il suggère d'organiser des workshops de conception régulièrement où les personnes qui font le produit "discutent" de la conception, plutôt que d'avoir des architectes d'un côté qui pensent l'architecture et que d'autres réalisent. LeSS met en avant la première valeur Agile, on privilégie les interactions afin de développer l'architecture collectivement afin que toutes les personnes concernées se l'approprient. Ils parlent "d'infection collective" de la conception.

Au-delà des méthodes comme Scrum, Kanban, SAFe, Less, les approches peuvent être adaptées au contexte spécifique du domaine : Hardware, Véhicule...

4.1.2 LA PHASE DE CONCEPTION EST EGALEMENT SOUMISE A DE NOMBREUX STANDARDS :

- La norme IATF 16949 est un standard de gestion de la qualité de l'industrie automobile. (Voir chapitre 8.3 « Conception et développement de produits et services » [IATF 16949](#)).
- La norme de cybersécurité ISO 21434 décrit plusieurs processus nécessaires pour garantir le niveau de cybersécurité du véhicule.
- La norme ISO 26262 introduit des contraintes de safety sous plusieurs formes. On trouve des contraintes sur le matériel à utiliser (ASIL C ou ASIL D), des contraintes logicielles (nécessité de redondances) et des contraintes dans la gestion des interférences (voir partie C).
- ASPICE : Les constructeurs exigent que les développements satisfassent les exigences des niveaux 2 ou 3 de ASPICE qui est une méthode d'évaluation basée sur la norme ISO/CEI 15504. Dédiaée initialement au développement système et logiciel dans l'automobile, ASPICE devrait s'étendre à d'autres domaines métiers et plus généralement se généraliser au développement de produits dans sa prochaine version (voir [ASPICE](#)).
- Le document Agile ASPICE "1.3 : Introduction de l'Agilité" (voir [SPICE PAM V1.3](#), accès limité) publié par INTACS (International Assessor Certification Scheme).



Un des défis majeurs de l'introduction de l'Agilité dans la phase de conception consiste à introduire des développements itératifs et à donner de l'autonomie aux équipes, malgré les normes, les standards portant sur cette phase ainsi que l'ensemble des réglementations auxquelles l'industrie Automobile est soumise.

4.1.3 LA CONCEPTION EST SOUMISE AUX CONTRAINTES DE L'INDUSTRIE AUTOMOBILE

La plupart des produits sont "physiques" : mécanique, électronique, mécatronique et il est difficile de changer rapidement leur conception pour s'adapter.

L'introduction des méthodes de design itératif est une autre difficulté majeure. Historiquement, le design des composants électroniques est figé dans les premières phases du projet. Les phases d'implémentation et de validation apparaissent dans un second temps. Les approches Agiles ont pour but d'introduire un design itératif voire continu, ce qui implique une validation continue. À nouveau, une évolution des technologies utilisées est nécessaire pour permettre un design continu et une validation continue. Par exemple, l'utilisation de FPGA sera préférée aux ASIC pour leur modularité. Les bénéfices de chaque nouvelle technologie doivent être mesurés par rapport au coût supplémentaire qu'elles introduisent.

Les produits automobiles doivent être "fabricables", testables et homologables ce qui pousse à une conception "stable", quand l'Agile prône une conception qui évolue pour s'adapter au marché, à l'environnement d'affaires en mouvement rapide.

Le produit doit être maintenable d'un point de vue logiciel, y compris en production tout en respectant les homologations. Par exemple, comment faire évoluer en série un logiciel sans remettre en question l'homologation sur les émissions de CO₂, la classe de consommation énergétique du véhicule, son empreinte carbone lors de sa fabrication et lors de son usage.

Dans l'approche Lean proposée par Toyota, on cherche à stabiliser le produit avant la mise en série quand les approches récentes prônent une conception en permanente évolution (voir « Du Toyotisme au Teslisme » (Valentin, 2020 (2nd édition))). On peut voir une application de DevOps du monde logiciel appliqué à l'industrie automobile.

4.2 AVANTAGES ET DIFFICULTES

4.2.1 AVANTAGES :

- Architecture conçue pour être résiliente aux changements,
- Boucle courte de design, intégration, feedback qui permet de faire converger progressivement la conception,
- Conception pensée pour et par le test,
- Ordre de conception priorisé en fonction de la valeur attendue par le client,
- Conception proche voir conjointe avec le client (OEM, utilisateur final),
- Apprentissage progressif quand le contexte ou la technologie est mal connue ou innovante,
- Approche multi solutions permettant de "pivoter" en fonction des retours (retour interne et externe),
- Conception plus collaborative qui évite les malentendus et les difficultés de maintenance,
- Conception progressivement fixée au fur et à mesure des boucles itérations/résultats/changements,
- Itération dédiée à l'exploration pour apprendre/découvrir et ensuite décider,
- Conception et déploiement/Production sont connectés,
- Niveau de qualité de la conception plus maîtrisé (par exemple : à un moment fixé hebdomadaire, les membres des équipes échangent avec les experts),
- Réduction des risques de mauvaise conception grâce aux boucles d'itération courtes.

4.2.2 DIFFICULTES :

- Les experts qui ne sont pas des membres des équipes Agiles doivent intervenir régulièrement pour réduire les risques,
- La conception étant plus collaborative, les membres des équipes Agiles doivent être plus multi compétences,
- Les équipes étant cross-fonctionnelles, les membres doivent être compétents en profondeur sur leur spécialité et suffisamment sur les autres sujets pour permettre la collaboration,
- Les équipes étant "autonomes" et 'auto organisées", elles doivent être capables de prendre à leur charge la conception qui auparavant était faite par d'autres,
- Architecte et développeur doivent s'entendre.
- Introduire les approches Agiles dans les développements matériels présente des défis similaires à l'introduction des approches Agiles dans les développements logiciels. Un changement de mentalité, des processus et des habitudes est nécessaire. Les approches Agile permettent de renforcer la levée de risques avec par exemple le lancement de test au plus tôt.
- Les dépendances en développement matériel prennent des aspects variés :

- Le logiciel bas-niveau est dépendant des choix de contrôleurs de domaines, des actionneurs, des capteurs et des réseaux,
- L'implémentation de certaines contraintes de safety ou de cybersécurité ont un impact sur le choix du matériel. Par exemple, la norme ISO 26262 nécessite l'utilisation de solutions impliquant plus fortement le matériel en ASIL C ou D,
- Les normes et règlements évoluent constamment et cela nécessite d'avoir des spécialistes pour s'assurer que les équipes Agiles adressent ces évolutions. Les équipes Agiles dépendent donc fortement de ces spécialistes,
- Difficultés pour évaluer la totale faisabilité d'un backlog (voir Partie B.3.3.4),
- Maîtrise des nouvelles technologies (exemple : infrastructure Cloud, écosystème technologique, collecte dynamique des données des véhicules),
- Difficulté de déléguer les estimations aux équipes Agiles quand estimer est nouveau pour elles.

4.3 CONDITIONS DE SUCCES

- Un travail de contextualisation de certains processus clés de développement ou d'industrialisation en usine est recommandé en parallèle de l'implémentation des approches Agiles. Ce travail d'adaptation des standards facilitera la mise en cohérence avec les éléments issus des approches Agiles tout en préservant des éléments fondamentaux du système de conception ou de production en grande série,
- Développer les champs de valeurs mais en considérant les aspects économique, performance, qualité, délai,
- Développer des conditions de priorisation : on va maintenir constant l'effort et jouer sur les priorités donc on va ajuster le contenu livré et ne pas toucher à la qualité,
- Créer des équipes de spécialistes et construire plutôt des équipes cross fonctionnelles afin qu'elles puissent prendre en charge la conception avec autonomie,
- Acceptation conjointe par les OEM et fournisseurs qu'une place plus importante soit laissée à l'exploration au détriment du contenu livrable,
- Étendre les compétences des personnes si elles sont au départ très spécialisées. Chercher à établir des personnes avec une compétence en "T" (voir https://en.wikipedia.org/wiki/T-shaped_skills et Partie B.8.3.1),
- Au niveau équipe, établir une "architectural runway (descendant/montant - Partie B.4.1.1) lorsque l'on passe à l'agilité à l'échelle, donc une structure d'équipe d'équipes (une équipe qui organise entre elles plusieurs équipes Agiles). Ainsi plusieurs équipes Agiles doivent collaborer pour contribuer à une conception commune cohérente du produit. Dans ce cas, une coordination entre architecte(s) et équipes est nécessaire pour assurer la cohérence des sous éléments produits développés et leurs intégrations. L'architecture "runway" est le support d'intégration entre les équipes. Tandis que pour une équipe seule, l'architecture ne dépend que de l'équipe elle-même et l'intégration est sous son entier contrôle,
- Établir un état d'esprit et des pratiques Agiles qui vont permettre une conception "Agile" en formant les personnes,
- Documenter les hypothèses prises lors de proposition de conception,
- Assumer la variabilité et préserver les options de conception,
- Définir les critères pour accepter/rejeter les solutions, pour savoir "pivoter",



- Être prêt à pivoter, c'est à dire à abandonner ou changer de conception,
- Expérimenter (par exemple Pop-corn méthode, voir <https://agilesensei.com/popcornflow/>).
- Célébrer et considérer l'apprentissage acquis quand l'option testée n'est pas la bonne,
- La conception doit permettre l'intégration continue de valeur dans le produit : si le design n'est pas flexible, l'intégration ne sera pas possible (Intégration continue => Design continu). Par exemple : Bootloader, flashage, mise à jour à distance (OTA),
- Mettre en place des livraisons régulières de "working product", ce qui peut être la livraison d'un ou d'éléments permettant d'apprendre, de conforter un choix de conception,
- Investir dans la simulation et la construction de mules : définir quand passer dans le physique ? (Voir "La simulation [numérique](#) au service des projets") :
 - o Par exemple : la livraison d'une simulation avec la conclusion associée peut être résultant d'une itération Agile.
 - o Par conséquent, pour des disciplines comme l'électronique, la mécanique, une simulation ou une impression 3D ou une carte électronique modulaire "wrapped", sont des livrables permettant d'avancer sur la construction de la valeur produit.
 - o On comprend donc qu'il est nécessaire d'investir dans ces moyens qui vont permettre d'avancer vite en conception et des moyens de test automatique.
- Investir dans des moyens rapides de génération du hardware :
 - o Fast prototyping : peut-être du code logiciel généré, de l'électronique,
 - o 3D printing (même pour des travaux finis, pour des petites séries, la principale difficulté est la vitesse d'impression. Les propriétés mécaniques et matériaux sont suffisamment développées pour permettre de la production en petite série. La technologie du 3D printing avance de jour en jour donc le seuil de rentabilité des moules par rapport à l'impression va se décaler de plus en plus vers le 3D printing.
- Construire une architecture de façon modulaire à tous les niveaux :
 - o La capacité à livrer en itération rapide des éléments de valeurs pour le logiciel comme pour l'électronique, la mécanique, passe par une architecture modulaire permettant cet assemblage par brique de valeur. Par exemple : une base de châssis de voiture, sur lequel on peut greffer des éléments comme suspension, transmission, direction, etc...
- Structurer les équipes pour faciliter la conception en Agile :
 - o La structuration des équipes impacte le design du système.
 - o Lorsque l'on construit des équipes Agiles, la manière dont on les constitue (quelles compétences, quelles disciplines, quelle mission/périmètre on leur affecte ?) va fortement influencer la conception du système. Rappel sur la Loi de Conway : "Toute organisation qui conçoit un système (défini au sens large) produira une conception dont la structure est une copie de la structure de communication de l'organisation." (Voir https://fr.wikipedia.org/wiki/Loi_de_Conway).



- Les dépendances entre les équipes vont aussi influencer l'Agilité qui va se répercuter sur la conception. Des équipes très indépendantes et autonomes vont plus facilement avoir une forte cohésion interne, mais il faudra veiller à la relation entre elles. On distingue l'esprit d'équipe et l'esprit d'équipe d'équipes, qui vont s'influencer l'une par rapport à l'autre. Selon l'équilibre établi, cela va favoriser la conception au sein de l'équipe ou entre les équipes. L'idéal serait de trouver un juste milieu entre la conception au sein des équipes et la conception entre les équipes.
- On parle de topologie d'équipe et il n'y a pas une bonne topologie mais des solutions avec chacune leurs avantages et inconvénients.

On peut retrouver plusieurs catégories de topologies :

- Structure d'équipe orientée "feature"/ fonctionnalité, c'est à dire que chaque équipe focalise sur la création de la valeur associée à une ou plusieurs "features"/fonctionnalités,
- Structure d'équipe orientée Architecture, c'est à dire que chaque équipe se focalise sur un élément d'architecture. Par exemple : Basic SW, communication, Operating système, etc.,
- Structure d'équipe orientée disciplines, c'est à dire par exemple des équipes mécatroniques, des équipes logiciels, etc.,
- Structure mixant les trois premières topologies.

Possibilité de topologie d'équipe d'équipes (Guide non exhaustif d'aide à choisir)				
(Équipe Agile appliquant Scrum, Kanban ou un framework hybride)				
	Equipe Fonctionnalité(s) (dans SAFe Stream-Aligned Team)	Structure d'équipe selon l'architecture (Dans SAFe Complicated Subsystem Teams ou platform team)	Structure par discipline (Dans SAFe Complicated Subsystem Teams)	Hybride (ex: in SAFe a Stream-Aligned teams connect to a Complicated Subsystem Teams)
Description	Equipes dédiées fonctionnalités Elles « livrent » de la fonctionnalité produit à intégrer	Equipes distribuées selon les éléments de l'architecture. Exemple : BSW, Application, OS, ...	Equipes distribuées par discipline. Exemple : équipe SW, équipe électronique, équipe système, ...	Est une combinaison des autres. Exemple : 2 équipes fonctionnalités et une mécatronique
Distribution géographique des équipes	Chaque équipe est Colocalisée	Chaque équipe Component/Block team est Colocalisée	Chaque équipe est Colocalisée	Chaque équipe est Colocalisée ou mélangée
Gestion des variants du produit	Pas facile nativement	Facile par construction à partir des éléments produits	Pas facile nativement	-
Gestion des dépendances	Les équipes sont très indépendantes. L'intégrateur construit le produit livrable (Working product)	Une ou plusieurs équipes utilisent les résultats d'une ou d'autres équipes pour construire le produit livrable. (Working product)	Au niveau de chaque discipline moins de dépendances mais plus vis à vis des autres disciplines. Exemple intégration SW/HW	Peut-être plus ou moins forte selon comment le backlog est distribué aux équipes

Inconvénients	Risque de travail redondant entre les équipes (par exemple conception en double d'éléments). Risque sur l'architecture d'ensemble	Difficultés d'intégration. Difficultés de construire un produit avec toute sa valeur. Perte de vue de la valeur client	Les difficultés de collaboration entre les silos perdurent. Difficultés de trouver une cadence commune. Conception par disciplines et non globale produit	Le compromis peut finalement additionner les inconvénients
Avantages	Les équipes sont orientées valeur, donc plus centrées sur le client	Optimisation de l'architecture, du taux de réutilisation	Facile à mettre en œuvre par rapport à des équipes cross disciplinaires	Le compromis peut combiner les avantages



Le lecteur pourra se référer au livre "Team Topologies " de Matthew Skelton et Manuel Pais, pour approfondir le sujet (Matthew Skelton, 2019).

Il faut constituer la structure d'équipe Agile en fonction du flux de valeur ou des livrables communs à fournir au-delà de l'organisation standard préexistante. Selon la topologie qui sera choisie, on facilitera plus la collaboration, plus la réutilisation, ou plus la vitesse de livraison, etc...

Chaque équipe peut être dans une zone géographique différente mais on veillera au maximum à colocaliser les membres d'une même équipe afin de faciliter le principe numéro 11 de l'Agile.

Dans tous les cas, on doit prévoir des intégrations régulières entre les équipes qui permettront de conforter le design et la progression de la livraison de valeur.

Il est fortement recommandé, comme le fait SAFe, de mettre toutes les équipes sur la même cadence d'intégration/validation/livraison.

En résumé, la structure des équipes et de l'équipe d'équipes va influencer dans un sens ou un autre la conception de la solution. On prendra garde à bien analyser les possibilités et il n'y a pas de choix idéal mais différents compromis.

On gardera aussi à l'esprit le principe 11 de l'Agile "The best architectures, requirements, and designs emerge from self-organizing teams.", c'est à dire que nous obtiendrons une meilleure conception en nous appuyant sur une équipe auto-organisée, responsable qui va chercher la meilleure conception plutôt que de lui imposer une conception non concertée (l'intelligence collective l'emporte sur l'intelligence individuelle).

5 AGILE ET MAITRISE DES RISQUES

5.1 RAPPEL DES PRINCIPES ET DES EXIGENCES

Bien que le Manifeste Agile n'adresse pas directement la notion de Maitrise des Risques, cette notion est fortement liée aux quatre valeurs proposées.

Une lecture rapide du Manifeste peut conduire à se concentrer sur les valeurs décrites à gauche du Manifeste (voir Figure 18 - Extrait du Manifeste AGILE.Figure 18) en oubliant celles de droite alors que le Manifeste dit très explicitement que « Nous reconnaissons la valeur des seconds éléments mais privilégions les premiers ».



Les individus et leurs interactions	...	<i>plus que les processus et les outils</i>
Des logiciels opérationnels	...	<i>plus qu'une documentation exhaustive</i>
La collaboration avec les clients	...	<i>plus que la négociation contractuelle</i>
L'adaptation au changement	...	<i>plus que le suivi d'un plan</i>

Figure 17 - Extrait du Manifeste AGILE.

Par exemple, le manifeste ne préconise pas de ne pas documenter sa conception, ou de ne pas définir un plan, ou encore de travailler sans le support de processus. Il met l'accent sur le fait que respectivement un Produit fonctionnel, la capacité à répondre au changement ou les individus et leurs interactions sont des éléments qui ont plus de valeur comparativement aux seconds éléments. Les bases des pratiques de maitrise des risques ne sont donc pas absentes des approches Agiles et il ne semble pas y avoir d'incompatibilité.

Par ailleurs les approches Agiles s'appuient sur des processus beaucoup plus itératifs comparativement aux approches traditionnelles plus linéaires et séquentielles. Cette particularité très adaptée à un développement dans un contexte VUCA peut paraître incompatible avec les processus de maitrise des risques habituellement mis en œuvre grâce à une forte planification et anticipation des risques.

Les requis ISO9001 et IATF mettent d'ailleurs un fort accent sur cette maitrise des risques et notamment sur leur identification et prise en compte au travers de la planification des actions nécessaires :

- Au niveau de l'organisation, de ses objectifs et du contexte dans lequel elle évolue (ISO 9001 - §6.1),
 - o Avec des exigences supplémentaires et systématiques de l'IATF sur :
 - Les Analyses de risques (§6.1.2.1),
 - La mise en place d'actions préventives visant à éviter une nouvelle récurrence de causes liées à des non-conformités (§6.1.2.2),
 - Ainsi que la mise en place de plan d'urgence prenant en compte les risques et impacts pour les clients (§6.1.2.3),
- Au niveau opérationnel (ISO 9001 - §8.1),
- Au niveau des processus de conception et de développement (ISO 9001 - §8.3),
 - o Par une approche pluridisciplinaire (IATF - §7.1.3.1),
 - o Par la réalisation et la revue des analyses de risques de la conception et de fabrication (AMDEC produit et procédé) (IATF - §8.3.2.1),
 - o Pour le développement de logiciel embarqué (IATF §8.3.2.3),
 - o Pour la sélection des prestataires externes (IATF - §8.4.1.2),
 - o Pour la gestion des évolutions (ISO 9001 - §8.5.6 et IATF §8.5.6.1).

Le processus général de maîtrise de risque est basé sur les étapes suivantes :

- Connaître son Processus / Produit,
- Identifier les risques,
- Analyser et Évaluer les risques,
- Traiter les risques jugés non acceptables,
- Suivre les risques

Les connaissances acquises doivent être capitalisées afin d'éviter :

- De nouvelles occurrences du risque sur le même périmètre,
- Que le risque ne se reproduise sur des Processus/Produits similaires.

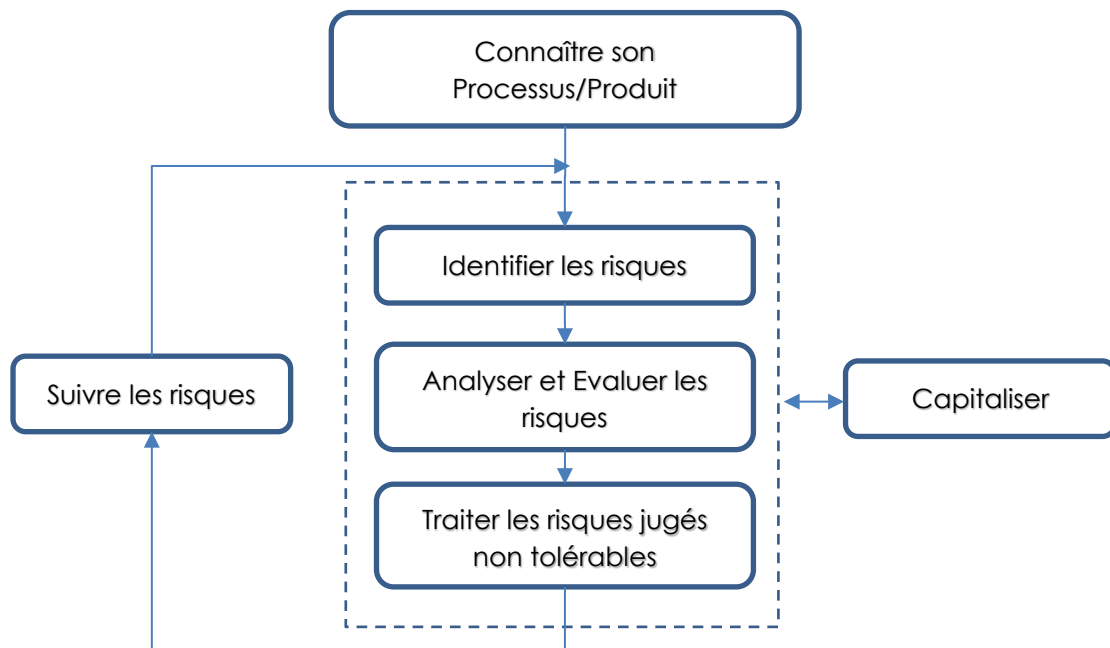
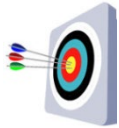


Figure 18 - Processus général de Maîtrise des risques.

5.2 AVANTAGES ET DIFFICULTES

Une approche Agile présente un certain nombre d'avantages quant à la maîtrise des risques :



1. Une meilleure identification des besoins clients par une forte proximité entre Business et Concepteur et une exposition rapide du Produit auprès de l'utilisateur final,
2. Une meilleure identification de nouveaux risques Produit par une exposition rapide sur le marché,
3. Une meilleure capacité de réaction aux changements par la capacité à réaliser des boucles courtes (par exemple grâce à une forte automatisation),
4. Une meilleure communication entre personnes en favorisant les interactions directes en face à face.

Néanmoins, ceci n'est pas sans remettre en cause certains usages, traditions ou habitudes de nos processus de maîtrise des risques du fait de :

1. L'absence de spécification stabilisée en début de cycle de conception,
2. Une planification qualité qui pourrait être insuffisante,
3. Une fréquence de changement beaucoup plus élevée et des cycles de développement beaucoup plus courts,
4. Une volonté d'exposer en clientèle beaucoup plus rapidement que dans des cycles de développement traditionnels.

Pour mener à bien la gestion des risques en approche Agile nous identifions les conditions de succès suivantes :

Au niveau gestion de projet :

1. Choisir l'approche projet adaptée au contexte,
2. Tirer pleinement profit de la proximité renforcée avec le client qu'apporte l'approche Agile pour limiter les risques de non-réponse au vrai besoin du client (voir §2.3),
3. Tirer pleinement profit d'interactions facilitées entre personnes,
4. Maintenir les contributions des experts nécessaires au sein des équipes Agiles,
5. Maintenir une capitalisation aux seins des équipes et aux seins de l'organisation.



Au niveau ingénierie :

1. Gérer les très fréquentes évolutions de la configuration du système,
2. Maintenir une planification qualité et une maîtrise des risques adaptée en mode de conception itératif,
3. Formalisation sans accumulation de dette des livrables définis par la planification qualité.

5.3 CONDITIONS DE SUCCES SUR LA GESTION DES RISQUES PROJETS

5.3.1 CHOIX DE L'APPROCHE

Le choix de l'approche projet la plus adaptée au contexte de développement telle que décrit en Partie A.6 est en lui-même une première mesure apportée à la gestion des risques projets :

- en zone d'incertitude technique ou sur la définition du besoin : l'approche Agile permet une réponse adaptée avec exposition rapide, itération et adaptation de la solution au besoin,
- en zone de plus forte maturité technique et de bonne connaissance du besoin une approche classique associée aux outils habituels de planification et de maîtrise des risques peut rester le meilleur choix.

5.3.2 RITUEL DE GESTION DES RISQUES

Si une approche Agile est choisie, les choix de priorisation d'activité pourront se faire en intégrant la notion de maîtrise de risque comme décrit en Partie B.3.3.1.

SAFe va plus loin sur le formalisme et propose d'intégrer des activités d'identification des risques et de revue des risques de façon récurrente pendant les phases de planification (événement PI Planning) et d'exécution (événement ART Sync) au travers du ROAM Board et de 4 catégories de Risques :

- o Les risques « Acceptables » : aucune action nécessaire, le risque en question est accepté en l'état,
- o Les risques « Non Acceptables » avec des statuts :
 - « Owned » si le plan d'action reste à construire sous la responsabilité d'une personne identifiée,
 - « Mitigated » si le plan d'action est défini,
 - « Resolved » si le risque peut maintenant être considéré comme acceptable.



Figure 19 - ROAM board proposé par SAFe (@Scale Agile, Inc)

Si besoin un animateur pourra aider les équipes à ancrer la pratique, ce rôle pouvant être pris en charge par un animateur de gestion de risque traditionnel ou par le coach Agile / Scrum Master ou le Release Train Engineer (RTE) dans un cadre SAFe.

5.3.3 MODES D'INTERACTIONS ET COMPORTEMENT AU SEIN DE L'EQUIPE DE DEVELOPPEMENT

La préférence donnée aux interactions directes entre personnes, les rituels d'équipes permettent d'améliorer la qualité des échanges entre personnes, et de les rendre plus continus.

Le fonctionnement en boucle courte permet également de multiplier les points de rendez-vous, de démonstration et de partage des résultats sans attendre le prochain jalon.

Pour cela les valeurs de transparence et de confiance sont indispensables au sein de l'équipe.



Il est également intéressant de maintenir l'équipe de développement en interactions avec des personnes « externes » afin de bénéficier d'un œil « neuf » sur certaines questions et d'éviter les effets de biais comportementaux qui pourraient apparaître au sein de l'équipe. Les rituels de démonstration peuvent être un bon moment pour provoquer ces échanges avec la participation du ou des représentants des clients, des responsables hiérarchique et/ou fonctionnel, d'experts, etc...

5.3.4 DISPONIBILITE DES EXPERTS

La complexité des systèmes développés, l'intervention de multiples métiers conduisent de plus en plus fréquemment au besoin d'avis d'expert lors des étapes l'identification et de maîtrise des risques.

La norme IATF requiert d'ailleurs une approche pluridisciplinaire pour ces activités (§7.1.3.1, §8.3.2.1).

Ces expertises sont rarement des ressources dédiées et disponibles en continu au sein d'une équipe Agile. Il convient donc de définir comment ces experts peuvent intervenir dans cet environnement.



Les rituels Agiles (démonstration, rétrospective...) peuvent être des moments privilégiés pour provoquer ces échanges. Si besoin un rituel complémentaire et dédié peut être ajouté au processus itératif.

5.3.5 CAPITALISATION

En fin d'itération, les rituels de rétrospective permettent de capitaliser aussi bien sur le Produit que sur le processus de développement.

Il peut être ensuite important de partager certains points clés en dehors de l'équipe lorsque nécessaire. Soit parce qu'ils impactent d'autres équipes sur le même développement en cours, soit pour une capitalisation sur un métier donné pour autre développement en cours ou à venir.



La mise en place d'une nouvelle organisation Agile dans l'entreprise doit prendre garde à ne pas faire disparaître des moyens de capitalisation préexistants à la transformation sans y apporter de nouvelle réponse adaptée.

Par exemple, une équipe « Process et Qualité » pourra être mise en place et travailler à l'amélioration des processus avec une backlog alimentée par les problèmes identifiés lors des rétrospectives.

5.4 CONDITIONS DE SUCCES SUR LA GESTION DES RISQUES TECHNIQUES

5.4.1 GESTION DES CONFIGURATIONS EN ENVIRONNEMENT VUCA

Dans un environnement où les itérations de développement et les mises à jour Produit/Service sur le marché sont plus fréquentes, il est impératif d'assurer une traçabilité précise de toute évolution sur ce Produit/Service ou des éléments contribuant à sa conception, à sa production et à son exploitation afin d'assurer la maîtrise des évolutions.

Ceci peut être facilité par la mise en place d'outils digitaux de définition, de gestion et de partage de la configuration entre les fonctions de conception et validation, production et exploitation.

5.4.2 MAITRISE DES RISQUES TECHNIQUES EN MODE DE CONCEPTION ITERATIF

5.4.2.1 Évolution des méthodes d'Analyse de Risques

L'évolution du processus de conception d'un mode continu à un mode itératif amène à réfléchir au besoin d'adapter les méthodes habituelles d'analyse de risque.



Une analyse de risque de type AMDEC généralement très séquentielle pourra évoluer en **rituel périodique** intégré à un Sprint où les risques seront identifiés et évalués afin de générer si besoin des tâches à intégrer à une backlog.

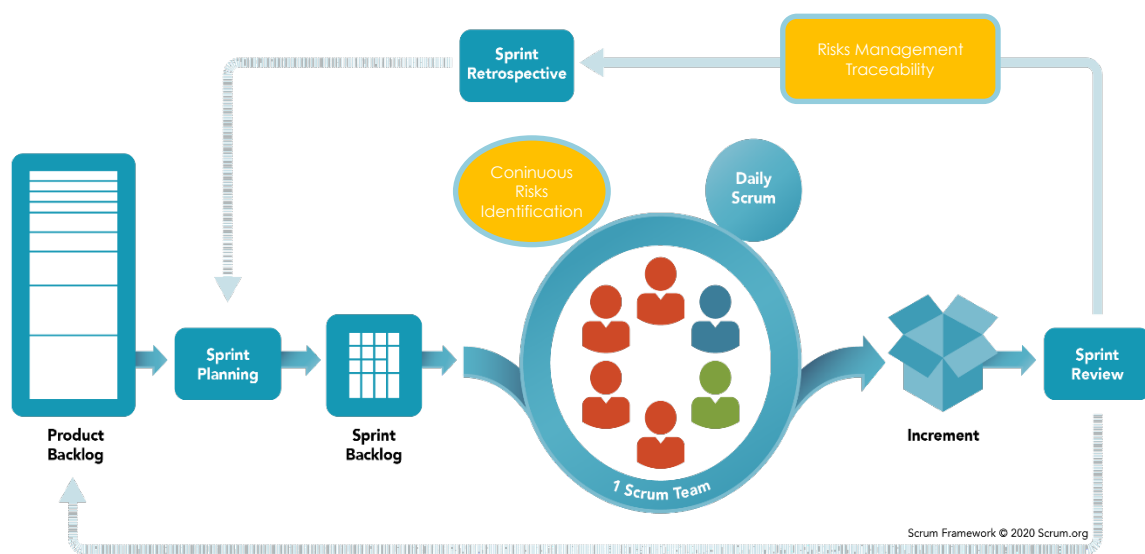


Figure 20 - Possible complément des activités SCRUM avec les rituels d'identification, de gestion et traçabilité des risques

Les différentes itérations de conception peuvent également être considérées comme autant d'évolutions successives dont l'impact doit être maîtrisé. Par exemple, l'ajout tardif dans une backlog d'une nouvelle exigence à implémenter afin « d'accueillir le changement » doit avoir fait l'objet d'une analyse d'impact et si nécessaire d'action permettant la maîtrise des risques jugés non acceptables. Il pourra donc être nécessaire d'optimiser le processus de gestion des évolutions afin qu'il s'intègre dans le nouveau processus de développement itératif.

Enfin, la maîtrise des risques ne pourra se faire sans augmenter :

- La capacité à disposer d'architectures résilientes et la préservation d'options (voir Partie B.4.1.1),
- La capacité à capter des signaux faibles : suivi d'un système en test ou d'une solution exposée en clientèle, alerting des équipes support...,
- La capacité à réagir en cas de problème : définition au préalable des mesures de secours, capacité à activer ces mesures.



5.4.2.2 Critères de priorisation de la gestion des risques techniques

Les critères traditionnels de priorisation des risques s'appuient en général sur :

- Une évaluation du niveau d'impact du risque : Sévérité ou Gravité,
- Une évaluation de son occurrence,
- Et pour certaine méthode, une évaluation de la capacité à détecter l'occurrence du risque (ex : AMDEC).



En mode de conception itérative, il peut être intéressant de compléter ces critères par une notion de « **bon timing** ». Certains risques identifiés devront être traités dans les premiers sprints de conception (adhérence forte à l'architecture, nombre de dépendances élevé, faible maturité technique...) d'autres pourront être traités lors de sprint plus tardifs (découplage élevé, forte maturité technique...).

5.4.2.3 Évolution des moyens et outils

L'utilisation d'outils digitaux peut aider à ce travail par un meilleur partage d'information voire par l'automatisation de certaines étapes d'analyse (exemple : logiciel ANSYS medini analyse – Journée SIA ISO26262 2022).

La **continuité des outils digitaux** peut également permettre d'aller plus loin par exemple en identifiant voire en exécutant automatiquement les tests de non-régression qui seraient nécessaires en automatisant des activités d'intégration continue.

5.4.3 FORMALISATION DES LIVRABLES QUALITE EN MODE DE CONCEPTION ITERATIF

Afin de ne pas accumuler de dette de formalisation des livrables qualité, dont ceux liés à la maîtrise des risques, les attendus devront être identifiés et pris en compte au bon moment dans la planification (tâches de la Backlog, conditions du Done...).

Ils devront couvrir les différents domaines et métiers, et également être formalisés de façon adaptée au contexte (voir Scalable DoD Partie B.6.3.3.1).

5.4.4 CAPACITE A GERER LES RISQUES LIES AUX DEMONSTRATIONS (PLUS) RAPIDES SUR LE MARCHE

Le besoin de démonstrations en clientèle devra se faire dans le respect des contraintes réglementaires et sécuritaires et avec les mesures de maîtrises des risques spécifiques à ces démonstrations comme décrit en Partie B.2.3.2.

6 AGILE ET VERIFICATION & VALIDATION

6.1 RAPPEL DES PRINCIPES ET DES EXIGENCES

Suivant la norme ISO 9000 : 2015, la vérification est le processus d'évaluation d'un système, d'un produit ou d'un composant afin de déterminer s'il satisfait aux exigences pour atteindre un niveau spécifique de son développement. Elle vise à confirmer par des preuves objectives comme des examens, des inspections, des tests ou des analyses que les exigences spécifiées ont été implémentées comme attendus en vertu des critères préalablement définis (performance, fiabilité,...).

La validation est un processus d'affirmation qu'un système, produit ou composant est en adéquation avec les attentes prévues et les besoins de toutes les parties prenantes. La validation est également la confirmation, par une preuve objective, que les exigences sont satisfaites dans le cas d'une utilisation spécifique ou une application prévue, réelle ou simulée de l'ensemble. La validation peut impliquer des tests, des démonstrations, des essais sur terrain ou d'autres moyens pour établir la conformité.

La vérification et la validation (V&V) sont deux processus distincts, mais complémentaires. Ils peuvent être soumis à une ou plusieurs revues pour déterminer la pertinence, l'adéquation ou l'efficacité du produit, système, composant mais aussi service à atteindre des objectifs définis.

Dans des modèles de gestion de projet, tels que le cycle en V, la vérification et la validation sont généralement effectuées dans la seconde partie du cycle de développement en prenant en compte :

- Les différents niveaux de validation du logiciel :

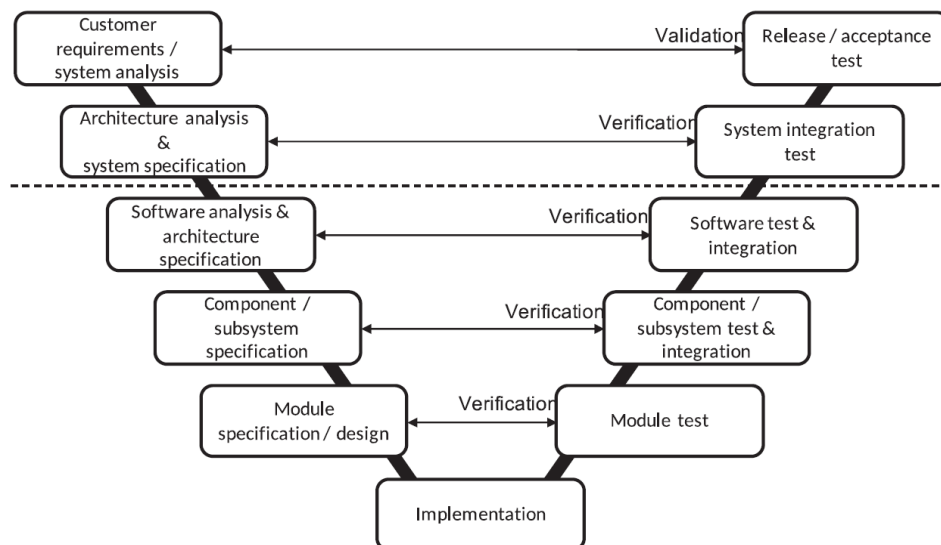


Figure 21 - Cycle en V dans le développement de logiciel automobile proposé par [Pfeffer, 2019]

- Et également les différents niveaux de vérification et de validation liés au matériel :
 - Hardware,
 - Intégration Hardware et Software,
 - Intégration Système,
 - Intégration et validation Véhicule,
 - Acceptance finale.

Pour ces étapes de vérification et validation, le niveau d'intégration pris en compte (« In the Loop ») est souvent utilisé. On parle de test :

- HIL – Hardware In the Loop: étape de validation des éléments Logiciel en temps réel dans un environnement contrôlé (Hardware et environnement réel ou simulé),
- SyIL – System in the Loop: étape de validation des interactions entre sous-systèmes,
- DIL – Driver in the Loop avec l'implication du conducteur pour permettre la validation des interactions Système-Conducteur,
- VIL – Vehicle In the Loop : étape essentielle pour valider l'interopérabilité de tous les sous-systèmes au niveau d'un véhicule.

Certaines exigences devront être validées par des tests finaux sur piste ou route.

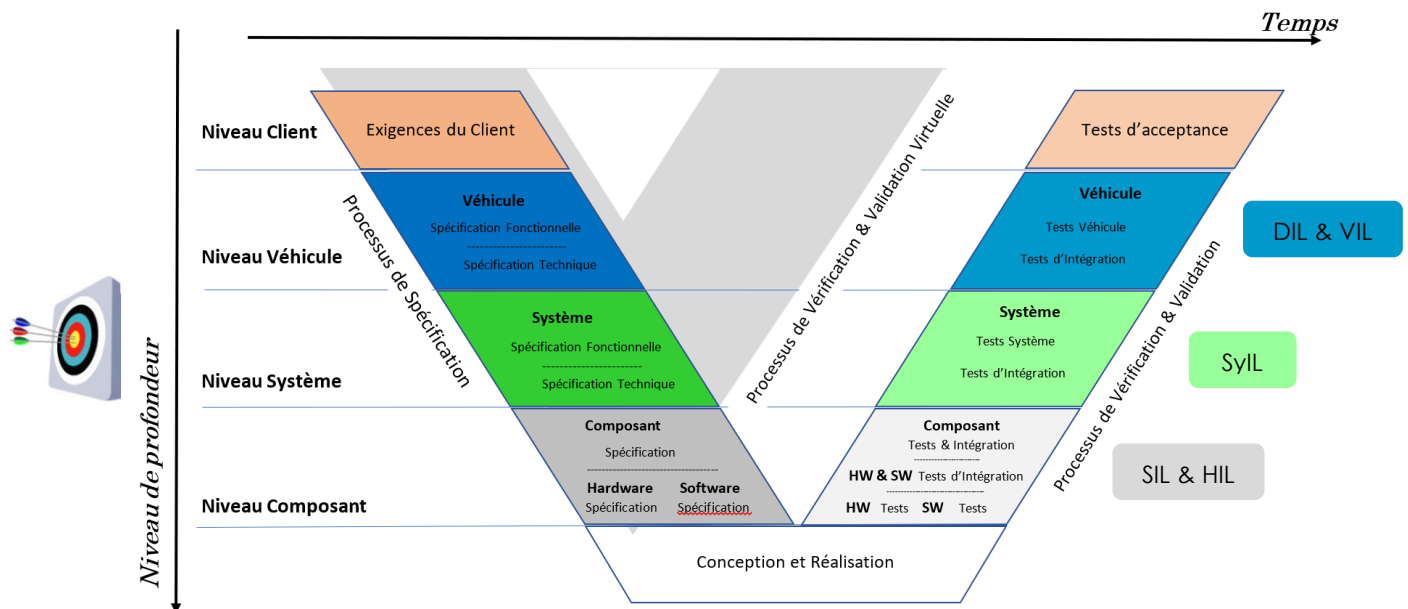


Figure 22 - Modèle de développement en V

Dans les approches Agiles, le développement est un processus itératif, incrémental et continu et la validation est basée sur les principes suivants :

- Définition of Done,
- Continuous Integration and Delivery - CI/CD,
- Quick Test and Learn.

Définition of Done : Le processus de vérification et de validation pour chaque incrément dépend fortement de la « Definition of Done (DoD) » utilisée pour définir les critères qui doivent être satisfaits pour considérer un élément complètement terminé. Cette définition implique la livraison du produit en étapes intermédiaires au client (interne) tout au long du processus de développement avec une vérification et validation (V&V) complétant chaque étape. Chaque incrément représente un engagement envers le client. Le produit final fait l'objet d'une affirmation pour garantir que tous les incréments fonctionnent ensemble. La DoD peut également inclure un produit, prêt à être livré au client final, tout en veillant à ce que les étapes de V&V soient toujours maintenues. La DoD s'applique généralement à une équipe Agile individuelle.

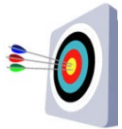
Dans une équipe Agile individuelle, les développeurs sont responsables d'assurer la qualité en établissant d'une manière claire et précise la DoD pour chaque incrément, et ils doivent se conformer à cette définition. Le Product Owner (PO) participe également à la définition des critères. Le PO a l'autorité sur le Product backlog et doit être capable de comprendre tous les critères de validation et les normes liées à la vérification. L'équipe Agile est chargée de fournir toutes les preuves nécessaires pour garantir la conformité réglementaire et fonctionnelle du produit. Une personne au sein de l'équipe Agile peut être spécifiquement responsable uniquement du suivi du processus de vérification et de validation, complétant ainsi le rôle du Scrum Master.

Dans les grands groupes, il est tout à fait envisageable que des ressources et des équipes spécifiques soient dédiées à la tâche de vérification.

Continuous Integration and Delivery - CI/CD : La livraison du logiciel est un exemple de processus incrémental et continu. Des paradigmes tels que l'Intégration et la Livraison Continues (Continuous Integration and Delivery - CI/CD) supposent que la liste des exigences n'a jamais une version finale ou figée. La CI englobe un développement continu avec les tests d'intégration correspondants, tandis que le CD comprend les recettes et la mise en production. L'équipe Agile a la responsabilité d'assurer des incréments de valeur atteignables, facilitant ainsi la gestion des projets liés à des produits complexes.

Quick Test and Learn : En Agile, l'objectif est de construire et d'apprendre le plus rapidement possible. La modélisation et la simulation dans un environnement virtuel, ainsi que le prototypage rapide contribuent à accélérer la phrase d'apprentissage.

6.2 PROBLÉMATIQUE



L'évolution d'un processus de conception et de validation très linéaire, à un processus itératif et incrémental, pose les problématiques suivantes vis-à-vis des activités de V&V :

- Assurer une bonne gestion de la configuration entre les activités de conception et celles de V&V,
- Assurer une capacité à Vérifier, Valider et Livrer rapidement,
- Assurer une bonne couverture des exigences par le « Définition of Done ».

De plus l'extension des pratiques Agiles au domaine Automobile apporte les problématiques complémentaires suivantes :

- Étendre la gestion de la configuration logicielle à celle de système incluant du matériel,
- Étendre les pratiques de validation itérative et incrémentale au domaine matériel,
- Conserver une bonne couverture de validation des exigences dans le cas de systèmes complexes et évolutifs,
- Optimiser les coûts de validation : limiter les redondances entre fournisseur et constructeur, assurer une bonne complémentarité entre validation sous-systèmes et systèmes...,
- Assurer une capacité à faire évoluer le produit en clientèle.

Nous identifions donc les conditions de succès suivantes :

- Gérer de façon précise la configuration du système entre conception et validation,
- Vérifier et Valider rapidement des systèmes complexes et évolutifs selon les besoins de release pour les clients utilisateurs,
- Définir et déployer des stratégies de V&V permettant de couvrir l'ensemble des exigences attendues en optimisant les coûts,



- Conserver une capacité à tester et à apprendre rapidement aussi bien sur des questions techniques et sur des aspects business,
- Définir les stratégies de maîtrise des logiciels en clientèle lors des évolutions à distance.

6.3 CONDITIONS DE SUCCES

6.3.1 GESTION DE LA CONFIGURATION POUR LES VERIFICATION ET VALIDATION

Les activités de V&V doivent être réalisées sur des configurations clairement définies et maîtrisées.

Par exemple, il est indispensable de garantir le suivi de chaque version d'un logiciel embarqué dans un système électronique de contrôle moteur qui doit être correctement documentée et archivée afin de bien distinguer les versions précédentes et actuelles. La mise en place d'un système de gestion des branches de développement permet de travailler simultanément sur plusieurs versions de logiciel. Néanmoins, les approches modernes d'intégration continue incitent plutôt à limiter ce nombre de branches et à mettre en place des mécanismes d'architecture permettant d'activer les fonctionnalités en cours de développement pour certains groupes d'utilisateurs.



Un autre exemple peut être pris sur les phases d'intégration de différentes parties d'un système. Lors des tests d'intégration continue, les fonctionnalités sont testées et intégrées dès qu'elles sont prêtes, ce qui permet de minimiser les risques d'erreurs et de conflits lors des releases.

Mais la fréquence élevée des intégrations peut conduire à une désynchronisation entre les phases de développement et de test. Si les tests valident une version inférieure du produit alors que le développement a itéré sur une nouvelle version dont les changements sont significatifs, la version précédente devient obsolète.

La configuration testée et validée doit être la même que celle livrée lors du Sprint. La synchronisation des équipes de développement, de vérification et de test de validation est donc obligatoire.

Globalement, la qualité et la traçabilité des éléments de configuration de l'ensemble du périmètre visé peuvent être suivies et mesurées par des indicateurs de conformité pour garantir leur maîtrise au cours du temps. Des outils digitaux paraissent adaptés pour répondre à ce besoin et il faudra veiller à assurer une continuité numérique sur l'ensemble du périmètre de gestion de configuration (système et cycle de vie).

Il est également fortement recommandé de clairement identifier des rôles et missions associés à cette activité.



6.3.2 VERIFIER ET VALIDER RAPIDEMENT

Afin de livrer plus rapidement sur le marché, il est impératif de pouvoir concevoir mais également réaliser les activités de V&V de façon rapide et continue. À cette fin, nous retiendrons les points clés suivants :

- Limiter l'effort nécessaire en V&V en limitant les dépendances et couplages lors de la conception (voir Partie B.4.3 sur les choix d'architectures adaptées) :
 - o Couplage entre module logiciel lors de la définition de l'architecture globale du logiciel,
 - o Couplage entre matériel et logiciel ou entre sous-systèmes matériels lors de la définition de l'architecture globale d'un système,
- Valider virtuellement par simulation,
- Étendre le processus de CI.



Pour garantir la rapidité de la validation, il est avantageux de placer autant de tests que possible vers l'amont et de renforcer des approches telles que le TDD (Test-Driven Development) et le BDD (Behaviour-Driven Development). Ces méthodes ont la particularité de compléter voire de remplacer la formalisation des spécifications.

Le TDD ou Développement Piloté par les Tests est une pratique itérative et incrémentale dans le développement logiciel Agile. Elle repose sur des tests automatisés rigoureux qui font office d'exigences formelles prenant systématiquement en compte les besoins du client. Les avantages du TDD incluent la possibilité de tester rapidement dans une démarche CI de nouvelles fonctionnalités et de centrer l'objectif des développeurs davantage sur la quantité de tests réalisée avec un niveau de confiance élevé plutôt que sur la quantité de code produit à la fin de la journée.

La méthode BDD (Behaviour-Driven Development) se concentre sur le comportement global attendu du système/produit plutôt que sur les détails de ses implémentations logicielles unitaires. Le BDD génère des spécifications comportementales sous la forme de scénarios concrets pour servir de tests fonctionnels dans une démarche de CI/CD (voir aussi 6.3.3.2). Dans certains cas, ces scénarios de test tiendront aussi lieu d'exigences formelles dans la Backlog en assurant que les conditions de V&V sont respectées grâce aux critères d'acceptation.

Cependant, ces approches sont actuellement peu rependues dans le secteur automobile et surtout déployées dans le domaine du logiciel.

6.3.2.1 Valider virtuellement



La conception et la V&V de maquettes numériques permettent de simuler une remontée du V au plus tôt avec des intérêts multiples :

- Concevoir par modélisation :
 - o La faisabilité des différentes options de conception du produit,
 - o Définir des modèles répondant au besoin,
 - o Optimiser ces modèles et ainsi d'anticiper les meilleures configurations,
- Vérifier la cohérence de l'ensemble des différents niveaux de spécifications,
 - o Valider les modèles des composants individuels ainsi que leur intégration,
 - o Vérifier si les fonctionnalités se comportent comme prévu et identifier tout écart par rapport aux exigences,
 - o Vérifier les performances attendues, identifier des problèmes potentiels et faire des ajustements en fonction des résultats,
 - o Tester les modèles dans des conditions extrêmes, difficilement reproductibles dans des situations réelles,
 - o Automatiser les tests pour assurer une grande couverture de scénarios,
- Vérifier l'adéquation de la spécification avec les besoins de l'utilisateur final,

et ainsi économiser de temps et réduire les coûts grâce à cette optimisation de la conception et l'anticipation de test physiques.

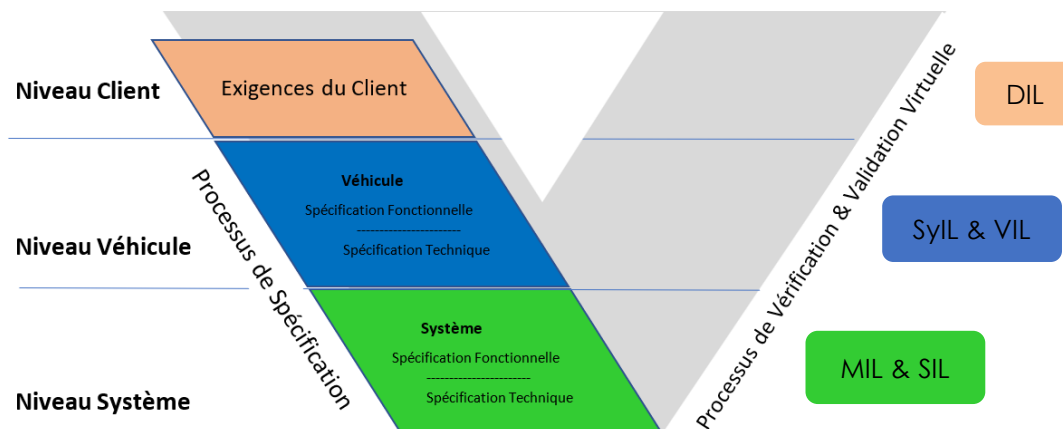


Figure 23 – Étapes de Validation Virtuelle "au plus tôt"

Ici également, plusieurs niveaux de test peuvent être réalisés basés sur la modélisation MBD (Model-Based Design). On parle de tests MIL (Model In the Loop), Système-SyIL, Conducteur-DIL comme décrit en Partie B.6.1.

Des solutions de Prototypage rapide peuvent également permettre d'aller plus loin que de la spécification et validation virtuelle en se rapprochant progressivement d'une solution représentative de la solution finale jusqu'à des tests Vehicle in the Loop - VIL.

6.3.2.2 Étendre le process de CI



Les outils numériques permettent également (W BECAMEL L DI VALENTIN - Renault Group & Stellantis, 2023):

- De vérifier des non-régressions fonctionnelles et de performances,
- D'augmenter la couverture des tests en réduisant le nombre d'essais physiques,
- De recalculer/améliorer les modèles numériques de conception.

L'intégration continue (CI) qui précède la livraison continue (CD) a pour objectif de supprimer les opérations manuelles et chronophages en automatisant les différentes étapes du développement. Dans le domaine du logiciel embarqué automobile, la génération automatique de cas de test est largement utilisée pour la vérification et la validation, permettant ainsi d'optimiser les processus et de garantir la qualité du produit. Des outils de type JIRA, offrent une **plateforme centralisée** pour gérer les activités liées à la vérification et à la validation, ce qui permet de rationaliser et d'automatiser les processus. Ils permettent également de gérer les cas de test de manière structurée et définir les critères de réussite, suivre l'état d'avancement et les rapports de résultats. Il est tout à fait possible de faciliter la gestion des anomalies détectées lors des activités et suivre leurs résolutions, et ainsi que de fluidifier la transition entre les différentes étapes de V&V. Le CI consiste à automatiser le processus de compilation afin de détecter rapidement les erreurs (des outils comme Jenkins ou GitLab sont également envisageables).



Le CD vise à automatiser le déploiement du produit dans son environnement cible. La livraison continue et automatique favorise un cycle de feedback rapide et s'adapte aux besoins changeants en améliorant d'une manière continue la qualité. La livraison continue exige aussi une gestion de configuration rigoureuse.

La mise en place d'une approche CI/CD (Continuous Integration/Continuous Delivery) dans le domaine de logiciel est déjà reconnue comme un investissement important en termes de coûts de licences et de formation du personnel.

Dans le cas de système matériel et logiciel, il est nécessaire d'étendre cette approche CI/CD jusqu'à l'automatisation de test sur banc HIL voire sur les niveaux suivants d'intégration par exemple Système. Les bancs d'Hardware-In-the-Loop (HIL) sont souvent utilisés pour simuler l'environnement de fonctionnement des divers composants. Cela permet de réduire les risques associés aux tests sur le terrain et d'itérer beaucoup plus fréquemment les plans de validation. Mais la mise en place et la maintenance des bancs HIL peuvent représenter des investissements et coûts d'exploitation significatifs qu'il sera nécessaire d'anticiper.

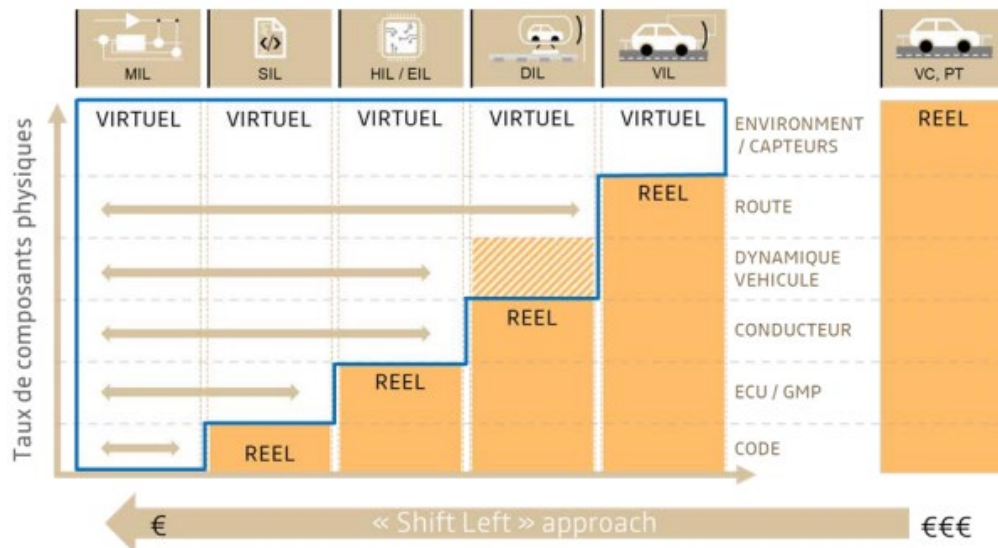


Figure 24 - Complémentarité Numérique / Physique - Becamel & Di Valentin - SIA 2023

L'intégration de tests véhicules plus fréquents est un aspect clé de la validation en hardware. Les tests sur terrain, en utilisant des véhicules réels, permettent de valider le fonctionnement des composants matériels dans des conditions réelles d'utilisation. Ces tests peuvent aussi être coûteux et nécessitent une planification et une exécution minutieuse en termes de sécurité.

Les tests réels peuvent alors permettre de recalibrer les modèles numériques et d'améliorer leur précision :

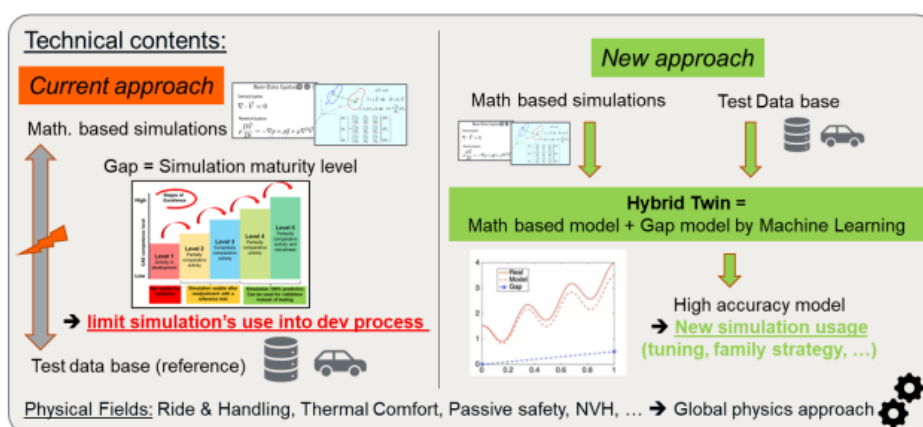
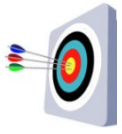


Figure 25 - Enrichissement des modèles numériques par les données de test - Becamel & Di Valentin - SIA 2023

6.3.2.3 Optimisation des coûts



La collaboration plus étroite en mode Agile sur l'ensemble de la chaîne de validation doit permettre de réduire les redondances et les incohérences coûteuses, et de détecter au plus tôt les problèmes grâce aux quick-feedback dans une démarche d'intégration plus progressive et continue.

SAFe illustre l'utilisation de ces différents outils de validation dans l'enchaînement d'itérations allant de la mise en œuvre de modèle numérique, de prototype et enfin l'utilisation de la solution finale.

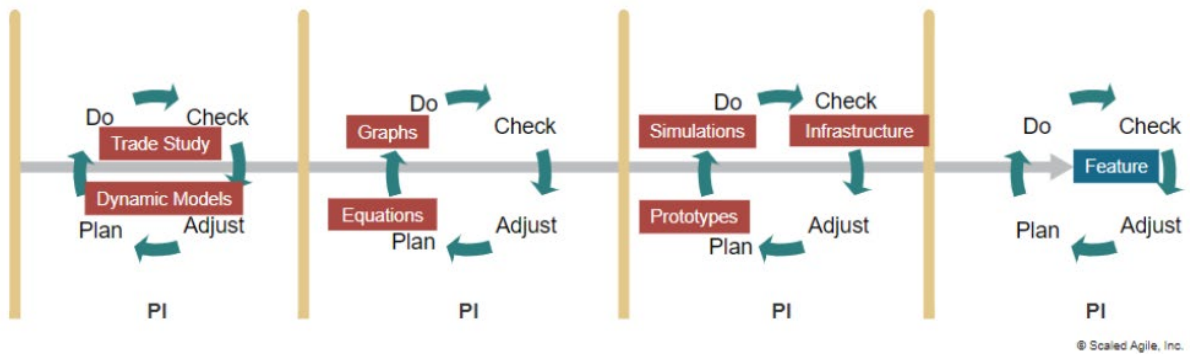


Figure 26 - Build incrementally with fast feedback - SAFe - Scale Agile Inc.



Un exemple d'évolution des processus de conception et de validation vers des cycles plus itératifs et progressifs peut être pris dans le domaine de la définition des Interfaces Homme Machine. Auparavant intégré dans le cycle de développement logiciel très linéaire (analyse, implémentation, validation, livraison), ce domaine est aujourd'hui traité spécifiquement, de façon beaucoup plus itérative et incrémentale avec des outils adaptés (outil de prototypage avec des niveaux de fidélité bas/moyen/élevé) assurant un bouclage de V&V plus fréquent avec les utilisateurs et une meilleure robustesse avec des nouvelles compétences telles que l'ergonomie, le design graphique, la prise en compte de la charge mentale de l'utilisateur...

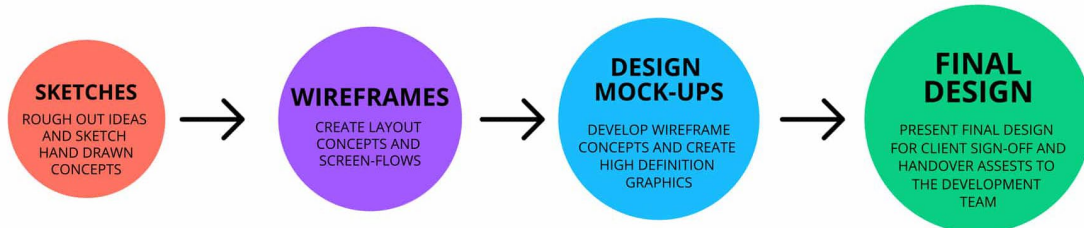


Figure 27 - Processus de conception IHM @Veethree Group

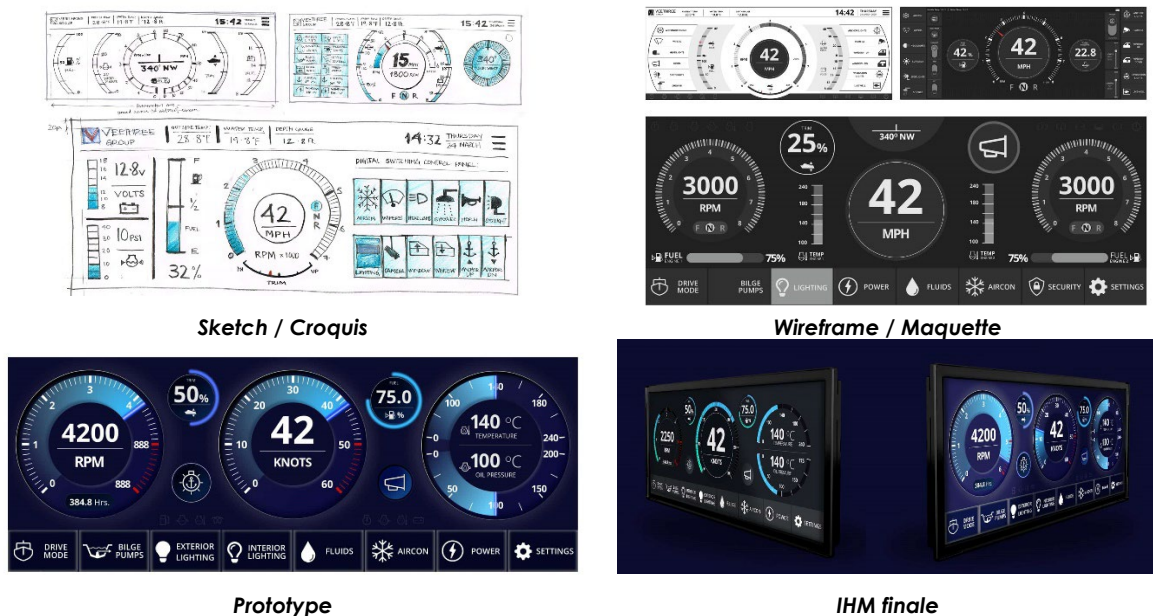


Figure 28 - Résultat IHM à chaque étape du processus de conception @Veethree Group

Le framework SAFe insiste également sur la nécessité de forcer le plus fréquemment possible une intégration « end-to-end » afin de détecter les problèmes au plus tôt :

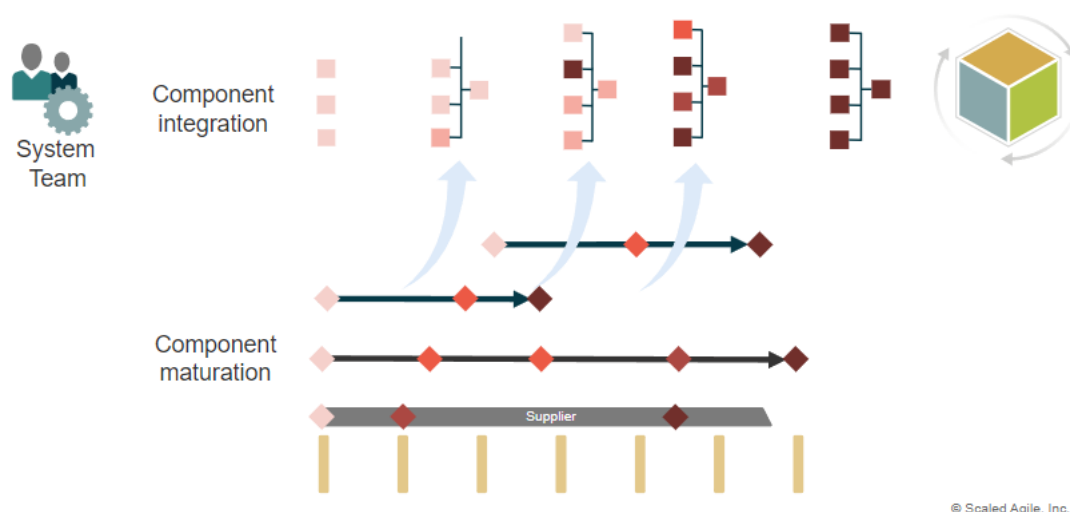


Figure 29 - Fréquente intégration "end-to-end" - SAFe - Scale Agile Inc.

6.3.3 COUVRIR L'ENSEMBLE DES EXIGENCES PAR UNE STRATEGIE V&V ADAPTEE

6.3.3.1 Scalable Definition of Done



Selon SAFe, la notion de « Scaled Definition of Done (Scaled DoD) » est une extension de la DoD qui est adaptée pour répondre aux besoins d'un développement Agile à grande échelle en assurant que le travail soit réalisé au bon moment, par les bonnes équipes, avec des exigences adaptées et donc en optimisant les coûts.

Team Increment	System Increment	Solution Increment	Release
- Stories satisfy acceptance criteria - Acceptance tests passed (automated where practical) - Unit and component tests coded, passed, and included in the build verification test - Cumulative unit tests passed - Assets are under version control - Engineering standards followed - NFRs met - No must-fix defects - Stories accepted by Product Owner	- Stories completed by all teams in the ART and integrated - Completed features meet acceptance criteria - NFRs met - No must-fix defects - Verification and validation of key scenarios - Included in build definition and deployment process - Increment demonstrated, feedback achieved - Accepted by Product Management	- Capabilities completed by all trains and meet acceptance criteria - Deployed/installed in the staging environment - NFRs met - Solution end-to-end integration, verification, and validation done - No must-fix defects - Included in build definition and deployment process - Documentation updated - Solution demonstrated; feedback achieved - Accepted by Solution Management	- All capabilities done and meet acceptance criteria - End-to-end integration and solution V&V done - Regression testing done - NFRs met - No must-fix defects - Release documentation complete - All standards met - Approved by Solution Management - Satisfies release governance requirements

© Scaled Agile, Inc.

Figure 30 - Exemple de définition de « Scaled DoD » selon SAFe

Ce concept renforce la pratique habituelle Automobile de vérifier la cohérence et la robustesse des différents plans de V&V à chaque niveau d'intégration lors de la remontée du cycle en V en assurant une optimisation des ressources et de la couverture de la validation globale, ceci dans un mode de conception itératif et incrémental.



Le CI/CD incrémental dédié à un système fonctionnel exige une Scaled Definition of Done afin de garantir que le travail demandé est affecté au moment approprié. Chaque équipe, entité ou entreprise doit adopter sa propre définition. Bien que cette dernière puisse différer d'une équipe à l'autre, toutes doivent partager le même noyau d'éléments.



Plus largement, et face à certaines dérives des approches Qualité classique où les fonctions Opérationnelles sont parfois déresponsabilisées des objectifs Qualité au profit de fonctions spécifiques, l'approche Built-In Quality de SAFe insiste sur une plus forte responsabilisation des équipes opérationnelles avec la prise en compte de Pratiques standards génériques et spécifiques aux 5 domaines clés : Business, Software, Hardware, Systèmes IT et Systèmes Cyber-physique.

6.3.3.2 Scenarios de validation



Les scenarios sont des cas de test conçus pour évaluer le comportement du système dans des situations réalistes et représentatives. Pour garantir la sécurité et la fiabilité du système, la maîtrise de conditions et des scénarios de test est essentielle. Les scénarios de test doivent être soigneusement élaborés pour couvrir un large éventail de conditions de conduite y compris des situations critiques et proches d'accident.

Un enjeu important de la qualité de V&V est liée à l'augmentation de la complexité des nouveaux systèmes de conduite. Qu'elle soit due à des causes internes aux systèmes (augmentation du nombre de fonctionnalités, augmentation des interactions internes au système...) ou à des causes externes (interactions entre systèmes embarqués, entre véhicules voire avec l'infrastructure, multiplicité des situations de vies rencontrées) cette évolution entraine une augmentation exponentielle des cas de tests à prendre en compte.

Par exemple, la validation basée sur des scénarios (Scenario-based testing) est actuellement en développement pour évaluer les fonctions ADAS et/ou les différents niveaux d'autonomie des véhicules. L'objectif est de s'assurer que les systèmes réagissent conformément aux attentes face aux divers événements qui peuvent survenir lors de la conduite.

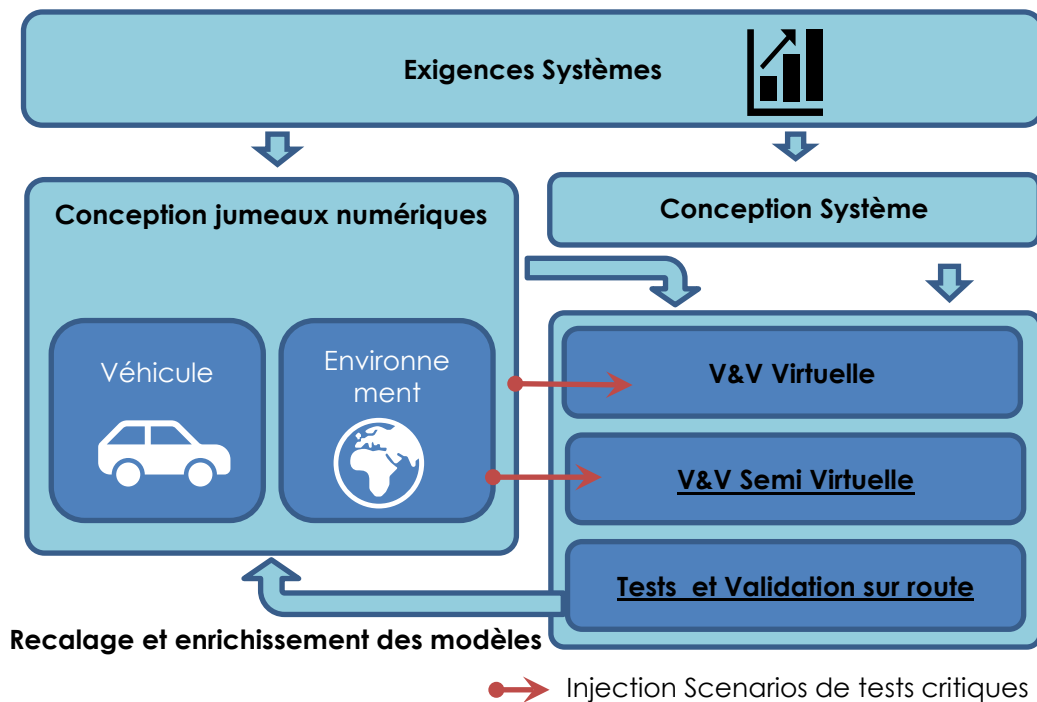


Figure 31 - Interactions entre MBD et Validation du modèle final.



L'élaboration de modèles numériques (jumeaux numériques) permet de Vérifier et Valider virtuellement et de progressivement mieux identifier les cas de tests critiques (voir Stratégie d'exploration des scénarios de test dans (Koné, 2021)). L'analyse des résultats de V&V aux différents niveaux d'intégration permet quant à elle d'améliorer la robustesse des modèles. Globalement, on augmente donc le taux de couverture des exigences et des situations de vie et la qualité de validation.

Les activités de validation semblent donc évoluer vers un processus plus itératif pour lesquels les Approches Agiles semblent être bien adaptées.

6.3.4 CONSERVER LA CAPACITE DE TEST & LEARN



La capacité de Test & Learn dans le domaine automobile est à la fois :

- Limitée par les contraintes de fiabilité et de sécurité qui s'imposent souvent aux solutions matérielles :
 - o La fiabilité est longue à démontrer,
 - o La sécurité nécessite un certain niveau de maîtrise des risques pour les utilisateurs d'un système en cours de validation,
- Limitée par les contraintes réglementaires à respecter lors de test sur le marché,
- Favorisée par la possibilité de tester au moyen de maquette numérique ou prototypage rapide comme décrit en Partie B.6.3.2.1,
- Favorisée par le développement des systèmes connectés permettant un suivi de l'usage, de la performance des solutions, la détection d'écart par rapport au prévu et lorsque nécessaire la mise à jour en cours de test.

Il est donc fortement recommandé que les aspects limitants soient anticipés, traités au plus tôt (voir les modes de planification B-3.3.3) et découplés au mieux du reste de la conception qui pourra alors évoluer à des fréquences plus élevées.

6.3.5 VALIDATION DES EVOLUTIONS LOGICIELLES EN CLIENTELE

La capacité à faire évoluer les logiciels en clientèle de façon rapide et simple pour l'utilisateur, donc à distance et sous le contrôle du constructeur automobile, est un levier de flexibilité qui est en train de se généraliser. Il génère cependant des contraintes à prendre en compte :

- La maîtrise des configurations en clientèle :
 - o En complément de la gestion de configuration initiale, il conviendra d'assurer la collecte et la transmission des données nécessaires à la traçabilité des évolutions de la configuration véhicule dans le temps pour anticiper des mises à jour adaptées et les maîtriser,
- Le respect des contraintes réglementaires telle que le R156 UNECE sur les mises à jour du logiciel et les systèmes de gestion de ces mises à jour,
- La maîtrise de ces évolutions :
 - o Pour les évolutions avec impact sur la sécurité du véhicule : le document SIA DC-10-01 élaboré par Renault Group, STELLANTIS et l'UTAC précise que pour toute évolution de la configuration, une évaluation de l'impact selon ISO26262 part2 - §6.4.3.3 doit être réalisée. Pour toute évolution avec impact non couverte par le document package, une extension de l'approbation par type est nécessaire.



De façon plus général il conviendra :

- D'élaborer une stratégie de validation représentative de configurations qui couvrent les principales variations dans le parc automobile. Ces configurations communes sont choisies par leurs paramètres pertinents présents dans l'ensemble du parc. En assurant la validation de ces configurations clés, les résultats peuvent être transposés aux autres configurations similaires,
- De maîtriser les risques de validation et de mise à jour par des moyens tels que :
 - o Le test des configurations clés dans un environnement contrôlé avant de déployer sur l'ensemble du parc automobile,
 - o La réalisation d'essais tout d'abord sur une flotte restreinte,
 - o Des mises à jour progressives et suivies afin d'assurer la capacité à réagir et à corriger le déploiement en cas d'anomalies,
 - o L'identification de configuration critique par des méthodes IA-machine learning,
- De communiquer auprès des clients et utilisateurs finaux les informations nécessaires à la maîtrise des risques et des contraintes réglementaires.

7 AGILE ET RELATION CLIENT-FOURNISSEUR

7.1 RAPPEL DES PRINCIPES ET DES EXIGENCES

La filière Automobile se caractérise par un maillage complexe de fournisseurs, d'équipementiers et de sous-traitants sur lesquels s'appuient les constructeurs. La conception d'un véhicule (ou plus généralement d'une solution de mobilité) met donc en œuvre un nombre considérable de relations Client-fournisseur, cascades sur plusieurs niveaux.

La collaboration avec le Client figure explicitement comme une valeur cardinale du Manifeste Agile ("Customer collaboration over contract negotiation"). Plus spécifiquement, quatre principes de ce même Manifeste viennent préciser les modes de collaboration Client-Fournisseur encouragés par la démarche Agile :

- "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."
- "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage."
- "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."
- "Working software is the primary measure of progress."

On y voit apparaître les notions de livraisons fréquentes en boucles courtes, de flexibilité face à une expression changeante du besoin, et de pilotage/suivi des développements par une évaluation directe de la maturité réelle du produit.

De leur côté les normes ISO et IATF mettent en avant la responsabilité pleine et entière du Client (« Vous pouvez externaliser l'activité, mais vous ne pouvez pas externaliser le risque » - Michael Gallagher), qui doit assurer la maîtrise des processus, produits et services fournis par des prestataires externes (ISO 9001 §8.4). Elles requièrent notamment :

- Des critères pour l'évaluation, la sélection, la surveillance des performances et la réévaluation des prestataires externes,
- Une vérification ou autres activités nécessaires pour s'assurer que les processus, produits et services fournis par des prestataires externes satisfont aux exigences,
- La communication aux prestataires externes de l'ensemble des exigences applicables.

Des exigences similaires se retrouvent dans la section §3.4 « Acquisition Process Group » d'Automotive SPICE.

7.2 AVANTAGES ET DIFFICULTES

L'approche Agile remet en cause les usages, traditions ou habitudes dans la relation Client-Fournisseur, en promouvant notamment :

1. Une définition "faible" du produit au moment de la contractualisation. Le contrat borde la façon dont le Client et Fournisseur vont travailler ensemble, définit le cadre général (périmètre technique et fonctionnel, exigences réglementaires applicables, ...) mais ne cherche pas à figer une définition détaillée du produit.

2. Une implication plus forte du client dans la phase de développement. Les équipes du Client et du Fournisseur vont élaborer conjointement une définition détaillée du produit, par itérations & ajustements successifs, en tenant compte des nouvelles données qui émergent au cours du développement (évolutions marché & compétition, veille techno, études de faisabilité technique, retours sur l'utilisabilité et les performances du produit dans ses versions intermédiaires, ...).
3. Une co-gouvernance, avec prise de risques en commun. Les environnements VUCA induisent un haut niveau de risque ambiant qu'il conviendra de reconnaître et de partager équitablement entre le Client et le Fournisseur. Les choix de priorités et les arbitrages technico-économiques (p.ex. augmenter de X€ le coût du produit pour intégrer une réserve mémoire plus importante et favoriser l'évolutivité future) devront donc être faits de façon concertée.
4. Des horizons de prévision plus courts. On travaille généralement sur des feuilles de route de 3 à 9 mois. Au-delà on s'accorde sur les grands jalons (liés notamment à l'industrialisation, aux qualifications & homologations) mais on se laisse volontairement une grande marge de manœuvre en évitant de figer un planning détaillé.

Pour mener à bien une telle transformation, touchant à la fois les pratiques d'ingénierie et la culture, nous identifions les conditions de succès suivantes :

Au niveau ingénierie et gestion de projet

1. Le Client doit planifier et mettre en place les moyens, notamment humains, nécessaires à une réelle collaboration technique tout au long du développement
 - Maturation & raffinement itératif du besoin (voir Partie B.2)
 - Dimensionnement, définition des marges de conception (voir Partie B.4)
 - Évaluations objectives de l'état du produit ; être en mesure de donner des "feedbacks" rapides à chaque incrément fonctionnel, tout en assurant la validation complète à terme (voir Partie B.6)
 - Alignement des priorités, co-pilotage (voir 0), gestion commune des risques (voir 0)
2. Pour la bonne efficacité de cette collaboration technique, le Client et le Fournisseur devront choisir des outils communs ou au minimum compatible entre eux, mettre en place des « passerelles » permettant le partage et l'échange routinier d'éléments techniques divers (ex. spécifications techniques & demandes d'évolution, rapports de tests, enregistrement de défauts, ...) et des livrables.
3. Le Référentiel Agile choisi doit être maîtrisé par les deux parties, les personnels des deux parties doivent être formés en conséquence (voir Partie B.8)
4. La cadence de développement (durée des itérations, fréquence et calendrier des livraisons) doit être définie en commun. Dans le cas particulier où le Client intègre des modules ou sous-systèmes issus de plusieurs Fournisseurs, il devra veiller à ce que l'ensemble des Fournisseurs s'alignent sur une cadence « maitresse », rendant ainsi possible une approche incrémentale en boucles courtes au niveau du système complet.
5. Le Client doit être associé aux principales cérémonies Agiles du Fournisseur (ex. PI Planning et/ou System Demo de SAFe) et s'y impliquer activement. De même il devra bénéficier d'un accès transparent à certains artefacts, notamment les feuilles de route, le backlog et les priorités.



6. Le Client et le Fournisseur doivent s'aligner sur un « Definition of Done » commun, qui précise le niveau minimum de maturité, de qualification et de documentation associé à chaque type de livrable, conditionnant ainsi l'usage qui pourra en être fait par le Client (p.ex. le roulage en véhicule est-il possible avec chaque version intermédiaire, ou juste avec certaines versions ayant bénéficié d'une qualification plus poussée ?).



Au niveau commercial et contractuel

7. Limiter l'utilisation du modèle "Prix Fixe / Périmètre Fixe / Durée fixe" qui, appliqué en parallèle d'une démarche de développement Agile conduit trop souvent à une inflation déraisonnable du nombre d'avenants ; le traitement de ces avenants engendre inévitablement des frictions qui risquent de détourner le Client et le Fournisseur de leur objectif commun. On trouvera dans la section §0 des recommandations sur les types de contrat susceptibles de réduire ces frictions.
8. Aligner autant que possible les processus commerciaux (passage de commande, facturation) sur la cadence des développements Agiles. On pourra notamment s'appuyer sur les « Planning Intervals » (PI) de SAFe, d'une durée habituellement comprise entre 8 et 12 semaines. Ainsi, par exemple, le Fournisseur déclencherait une facturation à la fin de chaque PI, en fonction de l'atteinte des engagements pris pour ce PI.
9. Identifier et isoler les éléments économiques du contrat liés à une prise de risque en commun ; par exemple un dimensionnement CPU ou mémoire intégrant une marge de sécurité pour d'éventuelles évolutions futures aura un impact sur le coût de l'équipement – le choix de cette marge devra être fait en commun par les deux parties.

Le modèle Agile ASPICETTM (Intacs, 2022), conçu comme une extension de ASPICETTM adaptée aux pratiques Agiles, propose un cadre d'évaluation spécifique pour le processus de collaboration Client- Fournisseur, axé autour de 6 pratiques de base :

AGL.2 Partner Collaboration Management – Practice Summary

- BP1 Establish a collaboration model.** Establish and maintain an agreement on how partners collaborate, how and how often they align and synchronize, and which information is shared among the partners.
- BP2 Share information.** Establish communication to share all agreed information between partners.
- BP3 Establish a technical approach.** A technical approach is jointly developed, agreed upon and kept up to date on a regular basis as documented in the collaboration agreement.
- BP4 Review joint technical content.** Perform reviews on jointly developed technical content with regard to Acceptance Criteria and Definition of Done on regular basis as agreed in the collaboration model.
- BP5 Inspect joint progress.** Measure, visualize, and inspect joint progress continuously with regard to quality, effort, cost and schedule against plan. Identify technical and business risks, impediments, and deviations from desired status collaboratively.
- BP6 Act on risks, impediments, and deviations.** Mitigate technical and business risks, impediments, and deviations from desired status collaboratively.

Source : <https://knuevenermackert.com/en/agile-spice-ebook-download/> Page 25

On retrouve l'accent particulier mis sur le partage d'informations, la collaboration technique étroite, l'alignement fréquent, la co-gouvernance et le partage des risques.

7.3 MODELES CONTRACTUELS CLIENT - FOURNISSEUR

- **Temps & Matériels (Time & Material)**

Le fournisseur est rémunéré en fonction des moyens et ressources mis en œuvre, avec éventuellement des exigences de productivité et de performance (vélocité). Le contrat prend fin à l'initiative du client, lorsque la valeur des éléments restants dans le backlog produit ne justifie plus de poursuivre les développements. Ce modèle exige un niveau de confiance élevée entre les deux parties. Des variantes avec bonus/malus ou plafond permettent de limiter l'exposition du Client.

- **Prix Fixe par Incrément ou par Point / Unité d'Œuvre**

Dans ce type de contrat, la rémunération est associée aux livraisons d'incrément fonctionnels par le Fournisseur. Cette rémunération peut être constante d'un incrément à l'autre, ou modulée par l'ampleur de chaque incrément livré (quantifiée en « Story points » ou en « Function Points »)

Chaque jalon de revue est l'occasion d'inspecter l'incrément, d'en évaluer la qualité et les performances. Si des écarts sont observés, les deux parties conviennent des adaptations nécessaires.

- **Modèles hybrides à pertes & profits partagés**

Un exemple de modèle hybride consiste pour le Fournisseur à se rémunérer par une combinaison de T&M (à un tarif promotionnel) et de Prix fixe par Point (à un tarif promotionnel également).

Imaginons un incrément estimé à 1000 points – le Fournisseur pense pouvoir le réaliser en 8000 heures et souhaite en tirer une rémunération de 800k€, soit un taux horaire de 100€. Dans ce modèle hybride, il pourra proposer (à titre d'illustration) un T&M à 30€ de l'heure combiné à un prix fixe de 560€ par point livré. Le tableau ci-dessous montre l'évolution de la rémunération en fonction du nombre de points réellement livrés et de l'effort réel associé :

	Nbr de Points Réalisés	Effort Passé (heures)	Rémunération (€)	Taux horaire effectif (€/h)
Variation du nbre de points réalisés à effort constant	900	8000	744000	93
	1000	8000	800000	100
	1100	8000	856000	107
Variation de l'effort passé à nbre de points constant	1000	7200	776000	108
	1000	8000	800000	100
	1000	8800	824000	94

Il y a donc bien une incitation réelle pour le Fournisseur à maximiser sa vélocité, en revanche un dépassement d'effort lié à un aléa projet sera partiellement pris en charge par le Client (par opposition à un contrat Prix fixe par Incrément ou par Point).

- **SAFe Managed Investment Contracts** (Scaled Agile, 2021)

Avant que les deux parties puissent s'engager dans un contrat d'envergure pour le développement d'un système complexe comportant de nombreuses inconnues, une phase préparatoire est nécessaire.

Durant cette phase de pré-engagement (Pre-commitment), le Client et le Fournisseur travaillent ensemble pour préciser la mission et les objectifs, confirmer la faisabilité et dimensionner le développement. Selon le contexte, le Client pourra mener cette phase de pré-engagement soit avec un unique Fournisseur préalablement sélectionné, soit en parallèle avec plusieurs Fournisseurs potentiels – le ou les Fournisseurs seront éventuellement rémunérés spécifiquement pour cette phase.

En sortie de cette phase on disposera d'une base d'engagement claire pour le développement proprement-dit :

- Définition de la vision et de la roadmap produit
- Définition du périmètre MVP (Minimum Viable Product) et d'éventuelles fonctionnalités supplémentaires
- Backlog produit priorisé pour les premiers « Planning Intervals » (PI)
- Définition précise des responsabilités d'exécution
- Cadrage économique, engagement de financement des PI (nombre de PI engagés), niveaux de financement initiaux et autres conditions contractuelles

Ensuite, pendant le déroulement du contrat de développement, le Client et le Fournisseur préparent, planifient et exécutent ensemble chaque PI, puis évaluent ensemble le résultat atteint.

À chaque jalon, une démonstration de la solution est organisée et la solution est évaluée. Les métriques convenues sont compilées et analysées, et des décisions sont prises pour le prochain PI. La progression de la solution et les améliorations du programme sont évaluées lors de l'événement « Inspect and Adapt » (I&A). À ce stade, le Client peut décider d'augmenter, de diminuer ou de maintenir les niveaux de financement, ou même de commencer à mettre fin à l'initiative selon qu'une valeur suffisante a ou n'a pas été atteinte.

Le recours au cycle « Lean StartUp »¹ permet de réduire le risque en optimisant la livraison des fonctionnalités prioritaires à la fin de chaque PI. Cela permet également une meilleure maîtrise des dépenses requises pour les fonctionnalités supplémentaires en fonction des résultats économiques.

- **Contrats Multi-phases**

Avec sa séparation en deux phases (pré-engagement et engagement) le « SAFe Managed Investment Contracts » est une illustration du principe consistant à découper un contrat de grande envergure, notamment par sa durée, en une succession de contrats plus courts, de façon à limiter les risques et à pouvoir adapter la relation contractuelle aux circonstances, en tenant compte notamment du niveau de confiance qui peut évoluer entre les deux parties, ainsi que des profils de risques qui fluctuent avec le temps.

Chacune des phases définies pourra s'exécuter selon un modèle T&M, un modèle à Prix fixe ou un modèle hybride.

¹ Voir (Scaled Agile, 2021) Figure 5

8 AGILE ET COMPETENCES

8.1 RAPPEL DES PRINCIPES ET DES EXIGENCES

Les normes ISO 9001 et IATF établissent des exigences concernant :

- La gestion des compétences (ISO §7.2), qui vise à garantir, par des formations ou des expériences appropriées, l'aptitude individuelle de chaque collaborateur à remplir sa mission.
- La gestion des connaissances organisationnelles (ISO §7.1.6), qui consiste à déterminer, acquérir, tenir à jour et mettre à disposition au sein de l'organisation l'ensemble des connaissances & savoirs nécessaires pour la maîtrise des processus et l'obtention de la conformité des produits et des services.

Conformément à ces exigences, lorsqu'elle s'engage dans une transformation Agile une organisation devra identifier le besoin de connaissances organisationnelles supplémentaires, déterminer comment elle peut les acquérir ou y accéder, puis assurer la mise à disposition de ces nouvelles connaissances & la montée en compétences des collaborateurs. Cette démarche, qui implique l'ensemble des salariés, les RH et le Management, pourra s'inscrire dans le contexte plus global de la Gestion des Emplois et Parcours Professionnels (GEPP) telle que définie dans les articles L. 2242-20 et suivants du Code du travail français.

L'ensemble de ces connaissances peuvent être décomposées en 3 parties :

- Le « Mindset Agile » qui définit le mode de fonctionnement et les différences avec une gestion de projet classique que ce soit au niveau de la fréquence des réunions, le type de livrable et leur contenu ou les séquences.
- Les différentes approches Agiles citées en partie A.
- Les rôles spécifiques dans une équipe Agile dont la composition varie en fonction du framework choisi tels que par exemple : Scrum Master, Product Owner, Architecture Owner.

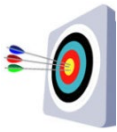
8.2 AVANTAGES ET DIFFICULTES

Comme mentionné dans les chapitres précédents les approches Agiles, les avantages évidents sont de livrer plus rapidement des livrables en se concentrant sur la juste valeur et avec une fréquence de suivi plus élevée.

A ces avantages se rajoutent pour les membres de l'équipe Agile :

- Une meilleure interaction : par les réunions d'équipes plus fréquentes et des périmètres élargis, les membres d'équipe entrent plus en interaction entre eux mais également avec les équipes supports.
- Une plus grande autonomie : le fonctionnement Agile ne peut se faire sans une autonomie accordée à l'équipe projet qui permet à ses membres de mieux gérer ses actions et avec moins de contraintes.
- Une meilleure satisfaction : à une meilleure autonomie, une meilleure interaction s'ajoute le fait de voir le projet se dérouler plus rapidement avec moins d'attente ou blocages, et l'ensemble sans sacrifier à la qualité du produit.

Cependant l'ensemble des avantages cités ne pourra être obtenu sans prendre en compte les points d'attention suivants qui seront traités et développés dans le chapitre 8.3 :



- Des compétences élargies,
- Une montée en compétence adaptée et la présence de coaching,
- L'implication des services de ressources humaines,
- L'implication du management et du top management par la stratégie de l'entreprise,
- L'acceptation du changement (voir Partie A.6.3.1 sur le pilotage du changement).

8.3 CONDITIONS DE SUCCES

8.3.1 COMPETENCES ELARGIES

En complément des formations spécifiques aux approches Agiles, les collaborateurs appelés à travailler dans une équipe Agile devront posséder ou acquérir un socle minimum de compétences pour assurer le fonctionnement autonome de l'équipe :

- Estimation de complexité et d'effort :
 - o Identification et gestion des risques,
 - o Identification et maîtrise des dépendances,
 - o Suivi d'avancement & reporting,
 - o Capitalisation et retour d'expérience.
- Les compétences "soft skills" suivantes : communication, autonomie, transparence, négociation, résolution de conflits, prise de décision, dynamique d'équipe. L'ouverture d'esprit et l'engagement sont primordiaux.

A ces compétences, s'ajoute le besoin d'avoir des profils « T-shaped » / multi-casquettes au sens où ce type de profil est expert dans son domaine (profil « I-shaped ») mais dispose également d'un bon niveau de connaissances dans les domaines transverses (profil « Dash-shaped »). Il maîtrise ainsi les livrables, les processus, les méthodes et outils liés à son domaine d'expertise avec des compétences et connaissances élargies nécessaires au bon déroulement du projet.

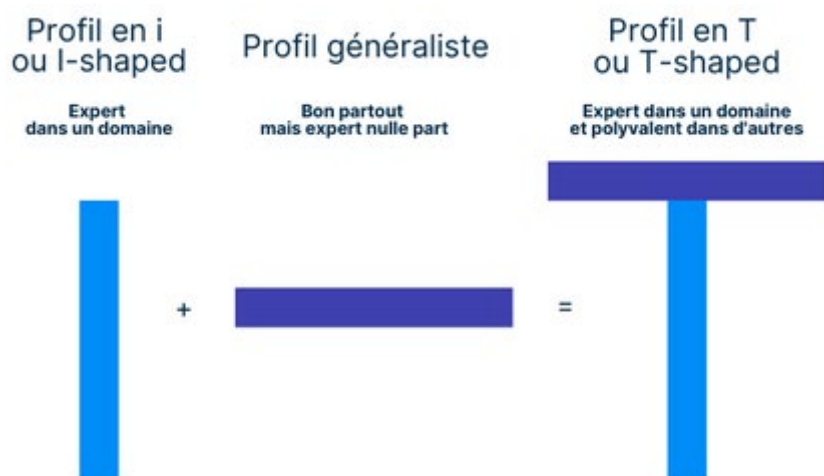


Figure 32 - Représentation visuelle du « T-shaped », du site www.taleez.com

A titre d'exemples :

- La mécatronique est par définition l'intersection entre 3 domaines : la mécanique, l'électronique et le logiciel. Il est rare d'avoir une ressource qui sache à la fois dimensionner les éléments mécaniques, optimiser un circuit imprimé par rapport à la CEM (Compatibilité électromagnétique) et également effectuer une modification de code. Cependant on souhaite fortement que la ressource qui est plus orientée mécanique ait conscience des besoins et problématiques de l'électronique et du logiciel par rapport au diagnostic d'une défaillance détectée sur véhicule.
- Le responsable du développement d'un composant doit certes maîtriser la conception de son produit mais aussi connaître et comprendre les besoins et contraintes de l'industrialisation, de la qualité et des achats.
- Le besoin d'avoir des ingénieurs logiciels qui sont capables de faire une grande partie des spécifications, un peu de conception, du codage, et une partie des tests.

Le besoin de ce type de profil n'est pas anodin pour les raisons suivantes :

- En proportion souvent faible dans les entreprises
- Vient presque à l'encontre des normes qualité et des souhaits de certains salariés où il est demandé d'avoir une description de poste précise avec un périmètre précis et figé afin de déterminer les responsabilités de chacun. A l'inverse, d'autres salariés seront demandeurs de ce type d'évolution afin de diversifier leurs actions et être impliqués dans presque l'entièreté d'un produit plutôt que d'être restreints à un périmètre bien plus petit.

Il sera donc nécessaire de créer ou augmenter la quantité de ces profils « T-shaped » dans l'entreprise soit par recrutement, soit par évolution avec des membres d'équipe projet ouverts aux domaines transverses et des descriptions de poste ad hoc. Une telle évolution n'est pas possible sans plan de carrière et support des services RH et du management.

8.3.2 MONTEE EN COMPETENCES ADAPTEE ET COACHING

Comme bon nombre de domaines, la montée en compétences et organisation des équipes projet ne peut se faire uniquement par des formations. C'est un travail qui doit être progressif et adapté. On peut favoriser la montée en compétence par coaching et mentoring, en intégrant des gens déjà formés (embauche ou prestation de services) à des postes clés, comme Release Train Engineer (RTE) ou Scrum Master.

En fonction de la taille et de la stratégie de l'entreprise, les formations et le « coaching » peuvent être réalisés soit en interne, soit en externe.



A l'heure actuelle, il n'existe pas de formation Agile dédiée aux métiers de l'automobile. Les formations disponibles sont soit qualifiantes soit certifiantes et proposées par des organismes et instituts. Les certifications et qualifications sont généralement décomposées par rôle dans l'équipe Agile et niveaux d'autonomie. Dans tous les cas ce type de formation a un programme figé et n'offre pas de programme adapté spécifiquement à l'entreprise concernée.

Compte tenu des offres de formations certifiantes, il est légitime de se demander si la certification est nécessaire pour être formateur en interne ou externe. Les certifications sont un gage de crédibilité quant aux connaissances du formateur dans les approches Agiles, cependant elles ne valident ou ne créditent pas l'expérience. Les certifications sont donc un plus, mais la condition nécessaire et suffisante pour une formation efficace est l'expérience du formateur aussi bien par rapport aux approches Agiles que dans le secteur automobile.

Si la formation et le « coaching » sont externalisés, une étape d'adaptation et d'assimilation de la culture de l'entreprise et du contexte métier & produit sera nécessaire au prestataire afin d'assurer que les méthodes, outils et conseils soit appropriés et cohérents. Cette assimilation de la culture de l'entreprise peut prendre plusieurs mois et doit donc être prise en compte dans le planning du plan de déploiement et réalisée bien en amont des formations.

8.3.3 IMPLICATION DES SERVICES RESSOURCES HUMAINES

Compte tenu des conditions de succès précédentes, l'implication des services RH est obligatoire. Par exemple des plans de carrière favorisant l'émergence de profils T-shaped doivent être élaborés et proposés aux collaborateurs, les descriptions de poste doivent être alignées avec la réalité d'une équipe Agile.

D'autre part la montée en compétences doit elle aussi être validée par les services RH tout comme dans une matrice classique de compétences dans l'ISO IATF. En se basant sur cette dernière, une proposition de décomposition pourrait être la suivante (cf. Figure 34) :

- Niveau 1 / Sensibilisation : les objectifs sont d'adhérer et acquérir un « Mindset » des approches Agiles, comprendre les différences avec les approches classiques, comprendre les spécificités des rôles dans une équipe Agile.

Elle s'adresse donc à un très large public :

- o toutes les personnes impliquées dans un projet utilisant une approche Agile,
- o les preneurs de décisions d'une entreprise afin de comprendre et anticiper les impacts de l'utilisation des approches Agiles.
- Niveau 2 / apprentissage : l'objectif est de connaître et appliquer les approches Agiles sous la supervision d'un formateur ou d'une personne suffisamment expérimentée (de niveau 4),
- Niveau 3 / Autonomie : l'objectif est de réaliser les actions correctement et en accord avec le « Mindset » des approches Agiles afin d'assurer la qualité des livrables.
- Niveau 4 / Formateur/coach : les objectifs sont d'assurer que les procédures et actions sont alignés avec l'approche Agile choisie, d'optimiser la méthode et les outils, d'assurer la formation et le niveau de compétences, d'assurer un support rapide et efficace (que ce soit en interne ou en prestation de service). Typiquement pour les personnes voulant renforcer leur position d'expertise en approches Agiles et/ou former d'autres personnes en interne, assurer un support transversal, proposer des services de consultation.

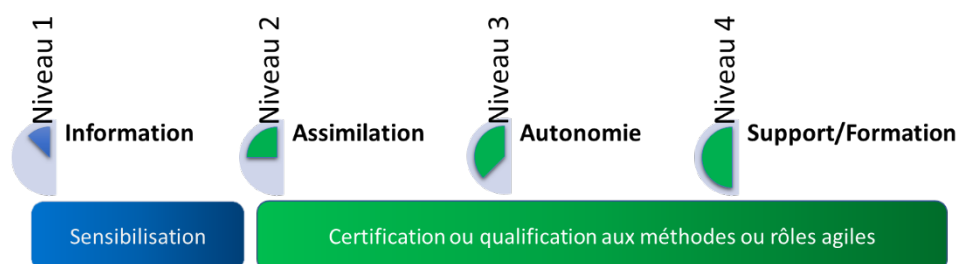


Figure 33 - Matrice des compétences

Les termes de certification ou qualification ne se rapportent pas nécessairement aux diplômes/certificats des organismes et instituts de formation, mais peuvent être des confirmations des compétences internes à l'entreprise.

8.3.4 IMPLICATION DU MANAGEMENT ET TOP MANAGEMENT

Une équipe Agile se doit d'être autonome et disponible pour le projet.

Pour cela le Top Management et Management doivent être impliqués dans la transformation Agile afin de garantir ces 2 composantes :

- Autonomie de terrain : les membres de l'équipe doivent être autonomes afin de réaliser sans avoir besoin d'un support ou d'une expertise externe qui viendrait perturber le bon déroulement du projet Agile. Ce degré de latitude devrait être offert par le management direct, cependant ce n'est pas toujours le cas par des organisations trop en silo.
- Autonomie des processus et organisation : l'ensemble de l'équipe doit être également autonome au sens que si elle est soumise aux mêmes contraintes de prises de décision que lors d'un fonctionnement classique le gain de l'approche Agile sera en dessous des attendus. À titre d'exemple, les processus achats sont généralement conçus avec différents de niveaux de signature afin d'assurer le suivi du budget et la gestion de finances adéquates. Cependant, il est fréquent que le processus de la commande soit supérieur à celui de la livraison du produit. Dans le cadre d'un fonctionnement Agile le processus doit donc être simplifié et raccourci au minimum. Il est en est de même pour bon nombre de processus, d'où le besoin de l'implication du top management afin d'autoriser ces processus.
- Gestion des ressources et disponibilité : un fonctionnement en Agile ayant une fréquence de suivi plus élevée, il demande une allocation /disponibilité supérieure de ses membres afin d'éviter de « papillonner » entre différents projets. Il sera donc préférable que les ressources soient multi-rôle/T-Shape et avec une disponibilité/allocation importante.
- Le fonctionnement en Agile demande des membres Agiles qui soient dédiés à une équipe et non partagés entre différentes équipes qui génèreraient des interventions ponctuelles. L'implication d'experts (qui ne peuvent être dédiés à 100%) est également nécessaire et un équilibre entre membres Agiles et experts est à trouver :
 - o Si l'équipe doit faire appel à trop d'experts, les méthodes d'avancement seront les approches de projet classique. De plus les experts ne pourront offrir une disponibilité importante par leur position multi-projets et leur nombre dans l'entreprise. De ce fait la performance sera en dessous des attendus d'une approche Agile voire inférieure à celle d'une approche classique.
 - o Si l'équipe n'a pas accès à des experts et doit se contenter des profils Agiles « T-shaped », les actions et décisions demandant un avis pointu et expérimenté ne pourront être réalisées avec le bon niveau de qualité. Le résultat en sera soit un blocage soit une prise de risque non maîtrisée.



- L'équilibre entre le nombre de membre Agiles et celui d'experts dépend des ressources et compétences disponibles, ainsi que des domaines d'activités. Cependant une proportion de 80% de membres Agiles et 20 % d'experts pourrait être une bonne base.

9 SYNTHÈSE ET POINTS OUVERTS

En synthèse, les échanges ont permis d'identifier les principaux apports des Approches Agiles vis-à-vis des requis qualité du domaine Automobile :

- Pas d'incompatibilité entre les Approches Agiles et les objectifs qualité de notre domaine,
- Retour d'expérience maintenant notable sans constat de dégradation de la qualité livrée. Ces approches facilitent le déploiement de nouvelles technologies (architecture à fort découplage matériel logiciel, OTA, giga-casting...) mais doivent s'adosser aux particularités du domaine automobile,
- Approche flexible et déployable au juste besoin (exemple : boîte à outil très complète SAFe),
- Changement de mode de contractualisation initié entre Constructeur et Équipementier plus adapté au contexte VUCA,

Vis-à-vis des difficultés de mise en œuvre, nous avons vu que, généralement, des adaptations de pratiques permettent de déployer ces Approches tout en restant conformes aux principes et exigences qualités de notre domaine. Nous avons noté les points ouverts ou pistes d'améliorations suivantes :

- Porter une sensibilisation de la culture Agile vers l'ensemble des fonctions transverses de l'entreprise (qualité, fiabilité, industrialisation, achat...) et également vers les auditeurs de certification,
- Associer au choix d'une Approche projet une analyse du contexte d'application et l'identification des risques associés (voir Partie A.6.2),
- Même si nous n'avons pas noté d'incompatibilité fondamentale entre les Approches Agiles et les référentiels qualité du domaine Automobile, des évolutions du cadre normatifs pourraient permettre de mieux intégrer et prendre en compte ces nouvelles approches sur les points suivants :
 - o Gestion d'un certain niveau d'incertitude sur la définition du besoin en début de processus de conception,
 - o Meilleure prise en compte de processus itératif,
 - o Meilleure prise en compte des nouveaux modes de contractualisation clients – fournisseurs,
 - o Meilleure intégration des nouveaux modes de fonctionnement dans les CoreTools IATF et en particulier dans l'APQP.

Le travail réalisé pour définir un cadre d'évaluation ASPICE dans un contexte Agile par le groupe de travail Intacs.com (voir (Intacs, 2022)) donnent quelques pistes sur les points clés à prendre en compte.

- Prendre en compte les évolutions d'exigences du Client final, de perception et d'acceptation des problèmes sur des objets très technologiques. Les approches Agiles devraient permettre de suivre ces évolutions et de s'adapter plus rapidement que ce que nos anciens processus ne le permettaient,
- Consolider et déployer plus largement de nouveaux modes de contractualisation Client – Fournisseur pour permettre une réelle connexion des Approches Agiles côté Constructeur et Équipementiers, notamment sur les phases amont de sélection fournisseur (sourcing) qui restent souvent rigides et lentes,

- Les procédures d'Homologation véhicule commencent à prendre en compte les problématiques de mise à jour sur le marché qui seront plus fréquentes en mode Agile. C'est par exemple le cas pour le logiciel couvert par le Règlement R156 UNECE – Software Update et Software Update Management System. Pour les systèmes de sécurité, le DC-10-01 - SIA – **Management of Functional Safety in the context of Complex Electronic Vehicle Control Systems homologations for passenger cars and light trucks vehicles** en reste à la nécessité de demande d'extension d'approbation en cas d'évolution avec impact sur les performances fonctionnelles ou techniques.
- Les Approches Agiles nécessitent un investissement fort sur la formation du personnel, ce qui implique un temps long et peut poser un problème dans des environnements très fluctuant avec des turnovers importants ou plan d'embauche massif,
- Le profil « T-shape » suppose une augmentation des domaines de compétences, une évolution des descriptions de postes et des parcours professionnels adaptés. Certains salariés peuvent être réticents à ce changement et devront être accompagnés dans cette évolution.
- L'offre de formation des approches Agiles dans l'enseignement reste à consolider pour une réelle prise en compte de ces nouvelles Approches dans les cursus avant l'entrée dans la vie professionnelle,
- Le profil « T Shape » et le besoin de cross fonctionnalité peuvent être en friction avec des organisations très verticalisées où chaque manager oriente ses collaborateurs vers plus d'expertise dans le domaine dont il est responsable sans valoriser la pluridisciplinarité. Il faudra donc intégrer dans l'organisation une démarche plus forte dans ce sens, soit en renforçant des objectifs de pluridisciplinarité et/ou en faisant évoluer la structure hiérarchique pour casser ces silos,
- Avec une très forte focalisation valeur produit, le risque de perte de capacité à capitaliser en discipline est réel. Il sera nécessaire de faire évoluer les canaux de capitalisation par exemple via la mise en place de communautés de pratiques au sens de SAFe (similaire aux Guildes de Spotify (Henrik Kniberg, 2012)).

PARTIE C. AGILE ET SECURITE FONCTIONNELLE

1 IDENTIFICATION DES PROBLEMATIQUES

1.1 COMPLEMENTS SUR LES ENJEUX ET BESOINS

1.1.1 RAPPEL SUR LES NORMES DE LA SECURITE AUTOMOBILE

La conception de système sûr pour le domaine automobile nécessite aujourd'hui la prise en compte de plusieurs normes permettant de couvrir notamment :

- Le domaine de la sécurité fonctionnelle :
 - o ISO 26262 – Functional Safety (FuSa): dangers causés par des défaillances des systèmes Electriques et Electroniques,
 - o ISO 21448 - SOTIF : dangers causés par des insuffisances de spécification, des limitations de performances, des Interfaces Homme Machine non adapté, des mauvais usages,
- Le domaine de la cybersécurité,
- Des techniques particulières comme celles de l'intelligence artificielle.

Le tableau ci-dessous décrit les différentes causes possibles de dangers et les normes associées selon l'ISO21448.

Source of hazard	Cause of hazardous events	Within scope of
System	E/E system faults	ISO 26262 series
	Functional insufficiencies	ISO 21448 (SOTIF)
	Incorrect and inadequate Human-Machine Interface (HMI) design (inappropriate user situational awareness, e.g. user confusion, user overload, user inattentiveness)	ISO 21448 (SOTIF) European Statement of Principles on human-machine interface [1]
	Functional insufficiencies of artificial intelligence-based algorithms	ISO 21448 (SOTIF)
	System technologies EXAMPLE: Eye damage from the beam of a lidar.	Specific standards EXAMPLE: IEC 60825
External factor	Reasonably foreseeable misuse by the user or by other road participants	ISO 21448 (SOTIF) The ISO 26262 series
	Attack exploiting vehicle security vulnerabilities	ISO/SAE 21434
	Impact from active infrastructure and/or vehicle to vehicle communication, and external systems	ISO 21448 (SOTIF) ISO 20077; ISO 26262 series; IEC 61508 series
	Impact from vehicle surroundings (e.g. other users, passive infrastructure, weather, electromagnetic interference)	ISO 21448 (SOTIF) The ISO 26262 series ISO 7637-2, ISO 7637-3 ISO 11452-2, ISO 11452-4, ISO 10605 and other relevant standards

Tableau 4 - Vue d'ensemble de sujets Safety adressés par différentes normes.

Ces normes définissent des cadres de développement différents mais globalement assez contraints, et donc, à priori assez éloignés de ceux proposés par des Approches Agiles.

Celui de l'ISO 26262 est basé sur :

- Des exigences associées à un cycle de développement séquentiel où les tâches s'enchaînent les unes après les autres,
- Des exigences associées à des méthodes de conception, de vérification et validation, d'audit, d'évaluation.

Celui de la norme ISO21448 est un peu plus itératif et vise à couvrir progressivement :

- l'identification et l'analyse des risques permettant la conception d'un système sûr,
- la vérification et la validation de ce système lorsqu'il est exposé à des cas d'usages de plus en plus riches, complets et donc de plus en plus proches de situations de vies que le système rencontrera en condition d'usage réelle,
- l'identification et la résolution de problème SOTIF qui apparaîtrait en opération.

La norme ISO21434 associée à la réglementation R155 sur la cybersécurité quant à elle établit une structuration assez proche de celle de l'ISO26262. Alors que cette dernière adresse les étapes permettant la mise sur le marché d'un véhicule complètement validé, la norme couvrant le domaine de la cybersécurité doit faire de même et également, comme l'ISO21448, permettre de traiter l'apparition de nouveaux risques, de nouvelles menaces après sa mise sur le marché. Pour cela, elle propose un cycle continu de surveillance, d'évaluation du risque et si besoin d'ajustement de la conception initiale.

Ces ajustements sont dans la plupart des cas rendus possibles, ou grandement facilités, par les fonctions de mise à jour logiciel à distance encadré également par une nouvelle réglementation UNECE R156 et la norme ISO 24089 associée.

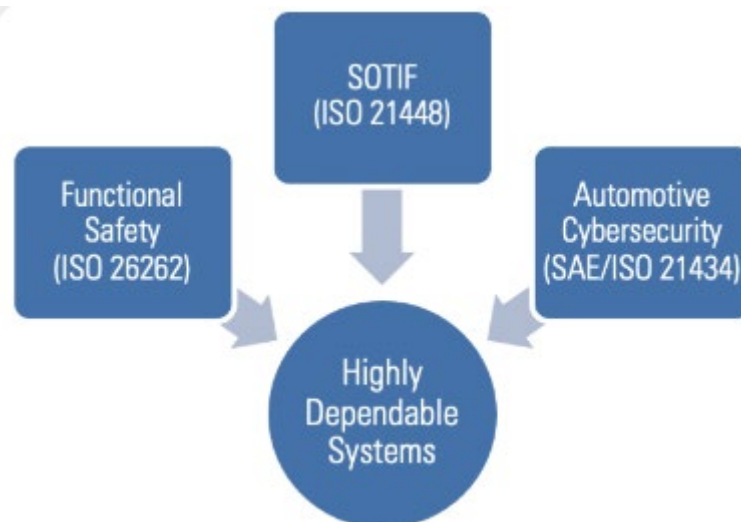


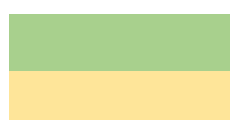
Figure 34 - Contribution des principales normes automobiles à l'obtention de Système Hautement Sûr.

1.1.2 POURQUOI DEPLOYER DES APPROCHES AGILES DANS CES DOMAINES ?

Le bénéfice des approches AGILES est déjà acquis sur un certain domaine comme discuté dans les parties A et B du guide :

- Boucle de conception en amont de la mise sur le marché permettant une prise de risque plus importante,
- Domaine d'incertitude/d'inconnu élevé donc adapté aux approches AGILE,
- Capacité à déployer et à corriger rapidement sur le domaine logiciel.

Phase projet	Recherche avancée	Projet pré-dev	Projet Marché
Technology Readiness Level	Faible TRL 1-5	Moyenne TRL 6-7	Elevée TRL 8-9
Système	Nouvelle architecture	Gestion d'évolution avec analyse d'impact	Gestion d'évolution avec analyse d'impact
Matériel	Nouvelle technologie	Nouvelle conception	Gestion d'évolution avec analyse d'impact
Logiciel	Nouvelle technologie	Nouvelle conception	Fonctions principales liées à la mobilité : Gestion d'évolution avec analyse d'impact
			Fonctionnalités additionnelles : fonctions non principales et non liées à la mobilité : Gestion d'évolution avec analyse d'impact.



Domaines déployés

Domaines non systématiquement déployés et approche préconisée

Tableau 5 - Domaine de déploiement des Approches AGILES

Mais son extension à des domaines complexes, à des hauts niveaux de maturité TRL et qui plus est sur des périmètres impactant la sécurité est encore loin d'être systématique :

- Temps de cycle de développement de bout en bout plus long,
- Impact d'une défaillance en usage plus élevé et pas forcément acceptable car avec des effets immédiats sur la sécurité, les risques financiers, l'image de l'entreprise,
- Difficulté à réagir rapidement pour couvrir le risque.

Alors pourquoi vouloir aller plus loin et élargir à ces nouveaux domaines de déploiement ?

- S'adapter aux évolutions des besoins et accélérer les mises sur le marché
 - Lorsque l'AGILE est déployée uniquement sur le logiciel et permet une accélération des développements et des livraisons, le matériel et le système deviennent les facteurs limitants (le corolaire est que déployer sur tous les domaines permet de tirer pleinement profit de l'approche),
 - Développer et tester à un fort niveau d'intégration plus rapidement permet de détecter plus tôt certains bugs, certaines limites de spécification, de compréhension du besoin et d'éviter les effets tunnels,
- S'adapter à des domaines de développement à plus forts niveaux d'incertitudes :
 - prendre en compte l'identification de nouveaux scénarios SOTIF en particulier pour les ADAS,
 - prendre en compte l'identification de nouvelles menaces,
- S'adapter plus rapidement aux évolutions réglementaires.

Mais alors quelles sont les limites ?

- la livraison de la Sécurité Fonctionnelle associée à des lots fonctionnels ne peut pas être découpée en MVP pour l'utilisateur final,
- les choix d'architectures initiaux en particulier liés au safety concept doivent être le plus résiliants possibles par rapport aux changements,
- la disponibilité des ressources de l'organisation (ex : safety engineer...) pour lever les risques au bon moment sur les choix d'architecture dimensionnants,
- les processus de développement sur les métiers liés au matériel doivent évoluer pour adresser cette logique de développement incrémental,
- l'organisation de l'entreprise traditionnelle peut limiter les bénéfices de l'AGILE si elle ne s'adapte pas à tous les niveaux,
- les modes de fonctionnement AGILE entre OEM et fournisseur doivent être partagés (principes de livraisons incrémentales, principes de tests d'intégration continues...).

1.1.3 QUE PROPOSE CETTE PARTIE C DU GUIDE ?

Nous nous focaliserons sur la mise en œuvre de développement ISO26262 et SOTIF ISO21448 en approche Agile afin de définir les avantages, les difficultés et les facteurs clés de succès.

Le périmètre de la cybersécurité ne sera pas traité ici. Néanmoins la structure de la norme ISO21434 étant proche de celle de l'ISO26262, les propositions faites ici pourront aider au traitement de la cybersécurité en Approche Agile.

1.2 LES PROBLEMATIQUES IDENTIFIEES

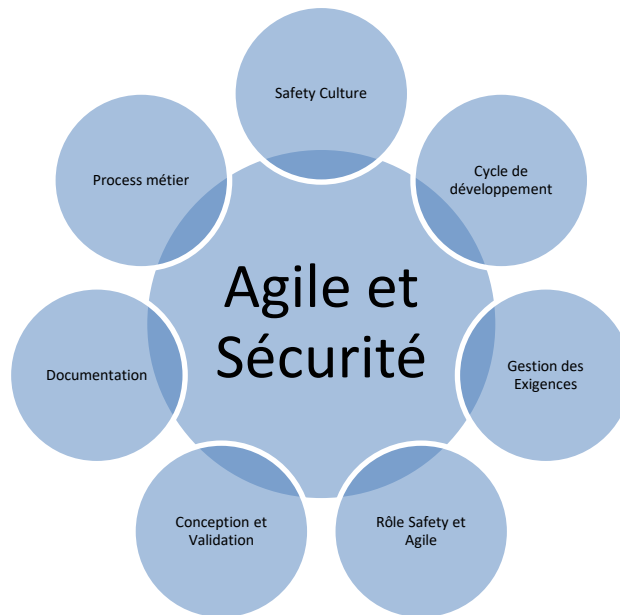


Figure 35 - Thèmes des problématiques "Agile et Sécurité"

La version 02 du guide couvre les 4 premiers thèmes :

- Safety Culture,
- Cycle de développement,
- Gestion des exigences,
- Rôle Safety et Agile (intégré dans le même paragraphe que le Cycle de développement).

Une prochaine version traitera les autres thématiques listées sur la Figure ci-dessus.

2 SAFETY CULTURE

2.1 RAPPEL DES PRINCIPES ET DES EXIGENCES

La norme ISO26262 impose le développement d'une Culture de la Sécurité Fonctionnelle dans l'entité « The organization shall create, foster, and sustain a safety culture that supports and encourages the effective achievement of functional safety – ISO26262-2018 – part 2 - §5.4.2.1 ».

La norme donne en annexe B les principaux points que doit adresser cette Culture Safety.

Les approches AGILES mettent en avant également une culture spécifique, un état d'esprit AGILE via notamment l'AGILE Manifesto comme évoqué dans la partie A.

2.2 AVANTAGES ET DIFFICULTES

Faire travailler ensemble des personnes issues de ces deux « mondes », s'ils n'ont pas eu la possibilité dans leurs parcours respectifs de connaître et comprendre ces deux cultures, peut conduire à des difficultés majeures.

Il conviendra donc :

- De permettre à chacun de comprendre les fondements et la culture de l'autre,
- D'identifier les éléments de culture communs et ceux représentant des risques.

Points communs aux cultures de Sécurité Fonctionnelle et AGILE :

- Travail en groupe multi-compétences et avec respect de la diversité,
- Orientation amélioration continue,
- Attitude proactive pour anticiper, détecter et traiter les risques,
- Alignement charge, compétences et capacité des ressources.

Points de vigilance :

- Capacité à comprendre la culture globale du domaine alliant Sécurité Fonctionnelle et AGILE,
- Orientation processus, contrôle et traçabilité,
- Gestion des exigences (voir §4),
- Définition des responsabilités associées à la Sécurité Fonctionnelle (voir §3.3.4.1),
- Prise en compte de la Sécurité Fonctionnelle dans les mécanismes de priorisation (voir §3.3.2) et de décision,
- Compétences en Sécurité Fonctionnelle requises dans les équipes AGILE et qualification au poste.

2.3 CONDITIONS DE SUCCES

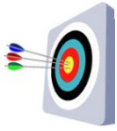
2.3.1 CULTURE SAFETY & AGILE

Une action simple consiste à mettre en place des formations sur les deux domaines s'adressant à des groupes de personnes issues des deux Cultures. Compte tenu de la difficulté à faire passer des messages culturels à distance, il est fortement recommandé d'organiser ces formations en présentiel, de plus cela permettra de faciliter des échanges directs et le partage d'expériences concrètes issues de ces deux mondes. L'idéal étant que le formateur ait lui-même une expérience avérée sur les domaines de la Sécurité Fonctionnelle et des Approches AGILE.

2.3.2 ORIENTATION PROCESSUS, CONTROLE ET TRAÇABILITE

La conformité à l'ISO26262 ne peut pas se satisfaire de processus qui seraient « implicites », non formalisés, non contrôlés et ne s'appuieraient pas globalement sur un Système de Management des processus.

Adopter une approche AGILE n'exclut pas de formaliser et de documenter les processus d'application associés.



La documentation et la traçabilité restent une bonne pratique indispensable à l'application de l'ISO26262, par exemple via la production de livrables nécessaires aux « Safety Cases ».

Néanmoins, on veillera à adopter une approche au juste nécessaire notamment en limitant le nombre d'itérations et de revues des documents, en accord avec le Manifeste Agile.

2.3.3 COMPETENCES REQUISES

Compétences spécifiques sur les métiers :

- Ingénierie système incluant la spécification fonctionnelle,
- Sécurité Fonctionnelle ISO26262 et ISO21448,
- Développement logiciel embarqué,
- Conception Hardware,

Sur les processus support :

- Gestion de projet en Approche Agile,
- Méthodes d'analyse de risques,
- Gestion de la Configuration,
- Gestion des Exigences.

3 CYCLE DE DEVELOPPEMENT

3.1 RAPPEL DES PRINCIPES ET DES EXIGENCES

3.1.1 EXIGENCES ISO26262

Le processus de développement proposé par l'ISO26262 se caractérise notamment par :

- La description d'une suite d'activités,
- La définition de rôles particuliers associés à la Sécurité Fonctionnelle,
- Des contraintes sur la mise en œuvre de méthodes, de moyens et outils de développement.

Chaque activité de développement est décrite avec :

- Des objectifs,
- Des prérequis,
- Des entrées,
- Des exigences et recommandations associées à l'activité,
- Des sorties ou work products.

Dans la plupart des cas, les sorties d'une activité sont toutes ou en partie reprises comme prérequis de l'activité suivante rendant l'ensemble du cycle très séquentiel.

Par exemple pour les étapes initiales de développement :

Après avoir défini tous les objectifs de sécurité (HARA), les exigences de sécurité émanent d'une part du « Functional Safety Concept », afin de :

- a) spécifier le comportement fonctionnel ou fonctionnel dégradé de l'élément conformément à ses objectifs de sécurité ;
- b) spécifier les contraintes concernant la détection et le contrôle approprié des défauts pertinents conformément à ses objectifs de sécurité ;
- c) spécifier les stratégies ou mesures au niveau de l'élément pour atteindre la tolérance aux défauts requise ou atténuer de manière adéquate les effets des défauts pertinents par l'élément lui-même, par le conducteur ou par des mesures externes ;
- d) attribuer les exigences de Sécurité Fonctionnelle à la conception architecturale du système ou à des mesures externes ;
- e) vérifier le concept de Sécurité Fonctionnelle et de spécifier les critères de validation de la sécurité.

Puis, d'autre part, selon le chapitre 4 de la norme ISO 26262, clause 6, les exigences de sécurité émanent du « Technical Safety Concept », afin de spécifier la mise en œuvre technique des exigences de Sécurité Fonctionnelle à leur niveau hiérarchique respectif, en tenant compte à la fois de la définition de l'élément et de la conception architecturale du système, et en abordant la détection des défaillances latentes, l'évitement des pannes, l'intégrité de la sécurité et les aspects d'exploitation et de service.

Ces exigences sont ensuite allouées aux domaines électronique et logiciel permettant de lancer la conception dans ces domaines.

L'ensemble est donc très séquentiel comme le montre la figure 2 – Structure of Safety Requirements de la partie 8 de l'ISO 26262.

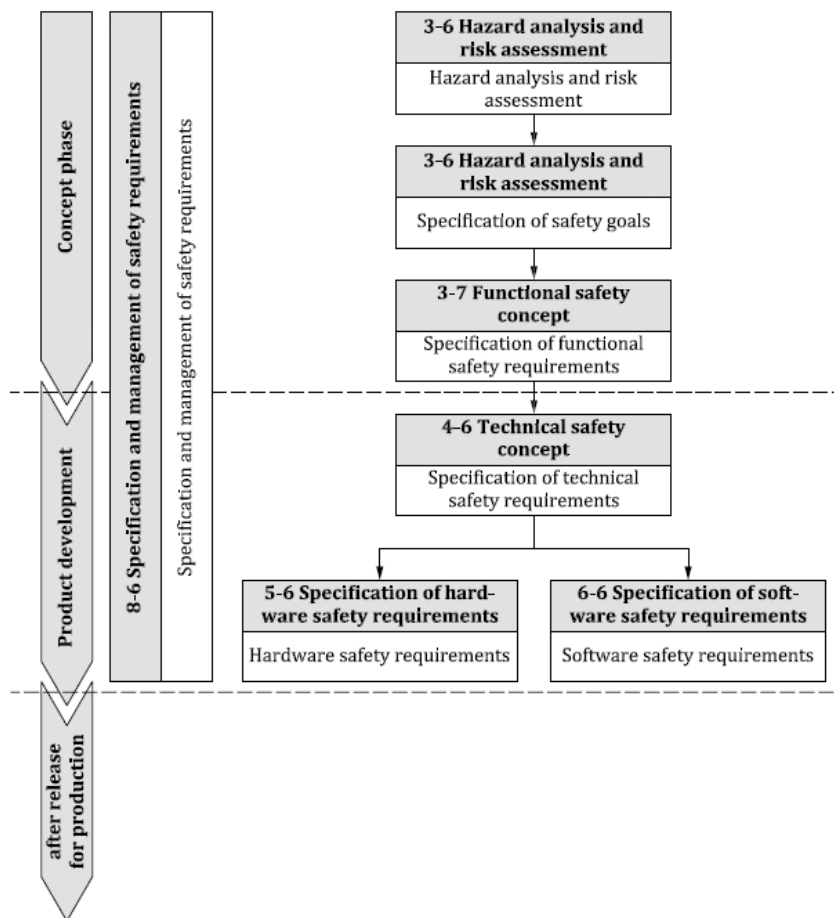


Figure 36 - Structuration des Exigences selon ISO26262 (Figure 2 part 8)

Le processus de développement fait également appel à des rôles et responsabilités particuliers tels que :

- Safety Manager (ISO26262-Part.1-§3.140) : personne ou organisation chargée de superviser et d'assurer l'exécution des activités nécessaires à la réalisation de la Sécurité Fonctionnelle,
- Safety Engineer (principalement ISO26262-Part. 3 à 6) : concepteurs en charge des activités impactés par la Sécurité Fonctionnelle,
- Safety Auditors & Assessors avec pour ces derniers des exigences d'indépendance vis-à-vis des activités (ISO26262-Part. 2- §6.4.11 & 6.4.12).

Les moyens et outils logiciels mis en œuvre doivent répondre à des requis d'évaluation et de qualification comme décrit dans la part 8 chapitre 11 de la norme « Confidence of Software tools ».

Et enfin, des contraintes sur les processus de développements (V&V incluses) sont décrites dans chacune des parties 3 à 6 sur le processus de conception.

3.1.2 EXIGENCES SOTIF-ISO21448

SOTIF (Safety Of The Intended Functionality) est le complément de l'ISO26262 (Sécurité Fonctionnelle en cas de défaillance) et est nécessaire pour garantir la sécurité du système en l'absence de défaillance.

En effet, en raison de l'augmentation des fonctionnalités avancées (ADAS) incluses dans les véhicules :

- Assistance au maintien dans la voie, assistance au changement de voie, reconnaissance des panneaux de signalisation...
- « Chauffeur » d'embouteillage, « chauffeur » d'autoroute, régulateur de vitesse adaptatif, ...
- Direction évasive, évitement d'obstacles, freinage d'urgence automatique, ...

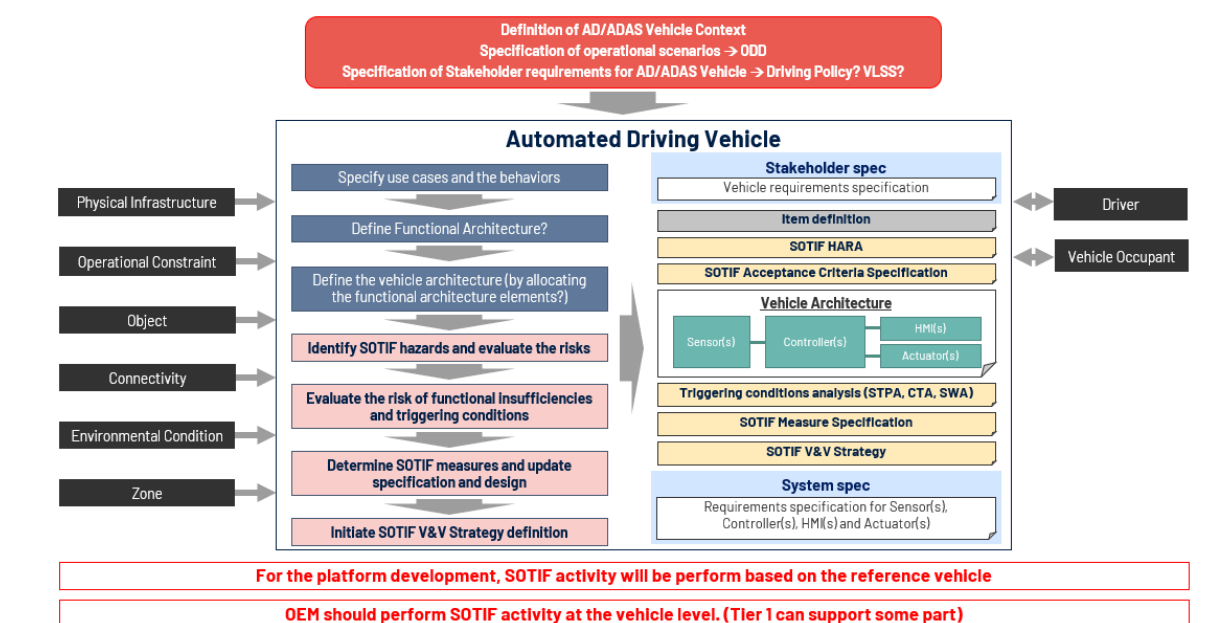


Figure 37 - Rappel des activités principales SOTIF

La norme SOTIF (ISO 21448) a été conçue pour s'intéresser aux dangers qui peuvent être causés par des problèmes de performance, des lacunes dans la spécification du système ou des mésusages et donc pouvant survenir même si le système est sans défaillance du point de vue de l'ISO26262.

La SOTIF a souvent un domaine d'incertitude plus élevé quant à la quantification des risques comparativement à l'ISO26262 comme le montre la figure ci-dessous :

- En ISO26262 : les cibles de défaillances aléatoires du matériel sont quantifiées et démontrées, les cibles FTTI sont définies et démontrées, ceci pour chaque Objectif de Sécurité,
- En SOTIF : un niveau de risque acceptable est défini basé sur par exemple le principe GAME (Globalement Au Moins Équivalent) ou GAMAB (Globalement Au Moins Aussi Bon) par comparaison au risque existant sur le marché établi pas des statistiques d'accidentologie. Le système sera mis à l'épreuve par une exploration progressive des conditions opérationnelles de vie réelle.

SOTIF & Functional Safety

Perform a HARA ... and then what?

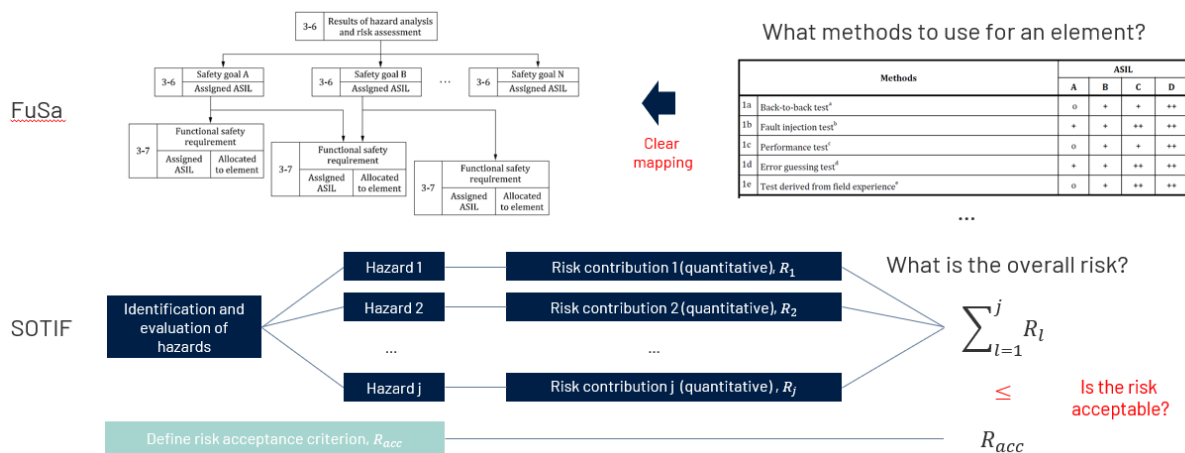


Figure 38 - Niveau d'incertitude des développements ISO26262 et SOTIF

Pour définir le niveau de risque résiduel, la SOTIF requiert plusieurs phases d'itération telles que décrites sur la figure ci-dessous :

- Itérations sur la phase de développement :
 - o Analyse pour satisfaire les clauses 5, 6, 7, 8,
 - o V&V pour satisfaire les clauses 8, 9, 10, 11,
- Itérations sur la phase opérationnelle pour satisfaire à la clause 13.
 - o Capitalisation sur les produits similaires déjà en série pour améliorer le développement en cours,
 - o Veille sur le produit développé une fois lancé en série,
 - o Amélioration continue de produits aussi bien en cours de développement que déjà développés.

L'approche Agile facilite naturellement l'amélioration continue du produit en développement et en vie série.

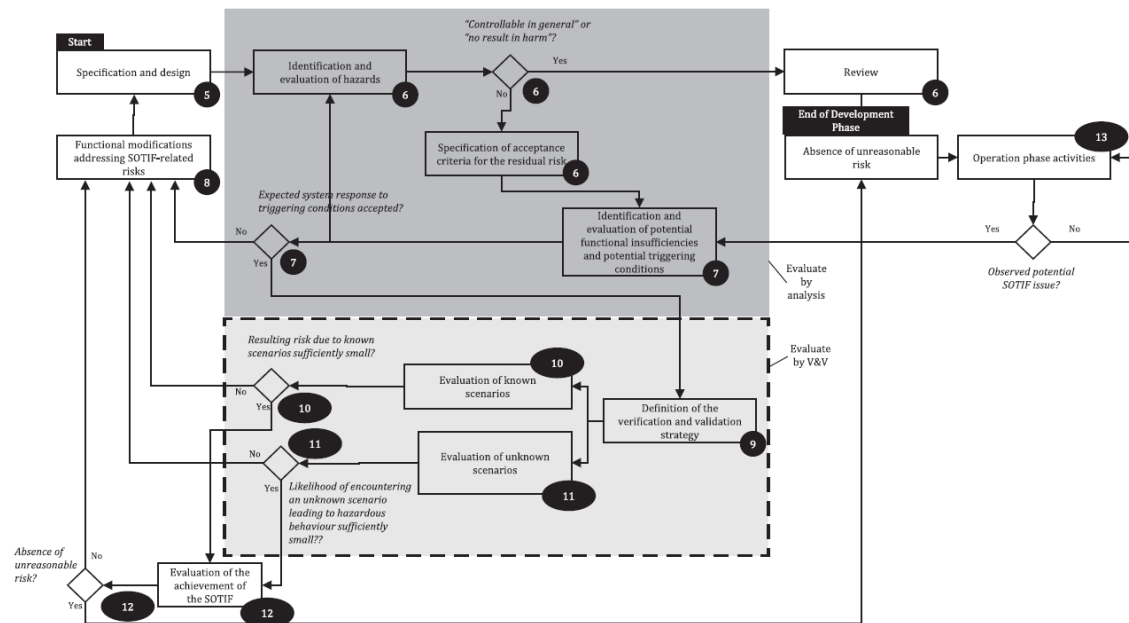


Figure 39 - Enchaînement des activités décrites par ISO21448

Ces itérations visent à améliorer progressivement le niveau de maîtrise des risques de la solution autant par l'exploration du domaine de fonctionnement et donc l'identification de risques, que par la définition et validation de solutions permettant la maîtrise de ces risques.

Les exigences de Safety Management de l'ISO26262 (ISO26262:2018 Parties 2, 7 et 8) dont entre autres celles associées aux rôles spécifiques liés à la Sécurité Fonctionnelle peuvent être étendues à la SOTIF.

Les outils doivent également répondre aux mêmes exigences d'évaluation et qualification que l'ISO26262.

3.2 AVANTAGES ET DIFFICULTES

La confrontation des exigences normatives à celles des approches Agiles fait apparaître les points de tension suivants qu'il faudra éclairer :

- comment développer de façon itérative dans le cadre de l'ISO26262, ou du SOTIF et quelles peuvent être les logiques de définition des priorités des tâches dans les phases itératives. Et comme nous l'avons évoqué, les réponses seront plus difficiles à apporter pour le périmètre ISO26262 de par sa nature plus séquentielle comparativement au SOTIF,
- Comment être conforme aux normes ISO26262 ou SOTIF en appliquant des approches Agiles,
- Quelle peut être l'organisation des équipes dans le cas d'un développement en approche AGILE d'un produit avec des contraintes de sécurité :
 - o comment intégrer les rôles spécifiques aux normes de la Sécurité Fonctionnelle dans des organisations d'équipes AGILES ?
 - o ou comment adapter des organisations ou équipes de Sécurité Fonctionnelle existantes à une approche AGILE ?
- Comment les outils supportant les approches AGILES satisfont les exigences de Sécurité Fonctionnelle; ou quelle capacité ont les outils habituels de Sécurité Fonctionnelle à répondre à la problématique des itérations Agiles ?

3.3 CONDITIONS DE SUCCES

3.3.1 PRELIMINAIRES REQUIS ET PHASES DE DEVELOPPEMENT ITERATIVES

3.3.1.1 Principe :

Avant de démarrer des cycles de développement itératifs sur des systèmes complexes portant de la sécurité, il convient de définir et partager les connaissances acquises et leurs niveaux de robustesse/incertitude associés que ce soit au sujet de la compréhension du besoin, de sa valorisation, de son mode de réalisation ou de son mode d'opération.

Pour cela le développement peut être organisé en deux séquences (Leite, 2023) :

- Une phase "Pre-Game",
- Une phase "Game" constituée par une série d'itération.

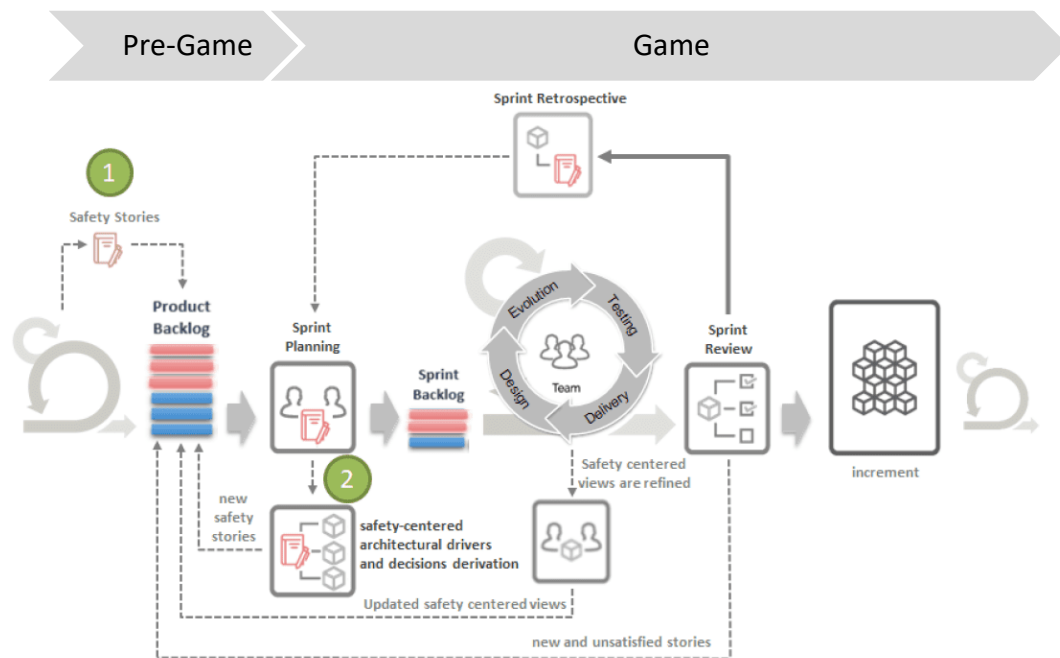


Figure 40 - Schéma itératif en Sécurité Fonctionnelle selon SCA3DA

La phase Pre-Game doit permettre de définir un premier backlog du système en développement (sur le schéma, les « functional stories » sont décrites en bleu et les « safety stories » en rouge), qui pourra si nécessaire être mis à jour au cours de la phase Game. Elle permet aussi de définir le périmètre, prioriser et planifier les itérations à réaliser en phase Game.

Il est donc nécessaire en fin de phase Pre-Game de disposer d'un niveau d'Analyse safety suffisamment robuste. Sur la base des éléments connus du backlog système et de sa première hypothèse d'architecture, la phase Pre-Game devra couvrir les livrables de l'ISO26262 et de l'ISO21448 suivant :

- Une première version du Safety Plan avec ses éléments structurants,
- L'Item Definition du système et si applicable l'analyse d'impact,
- La HARA dont les Safety Goals qui pourront être associés à des activités de Sécurité Fonctionnelle dans le cadre d'EPIC,
- Les premières hypothèses de Concept de Sécurité (Fonctionnel et/ou Technique).
- Les scénarios et conditions de déclenchement liés à la SOTIF si non inclus dans la HARA,
- Les contre-mesures du système et de ses éléments liés aux risques SOTIF.

La phase **Game** a ensuite pour but le déroulement des itérations sur la base du backlog et des données d'entrée établies en phase Pre-Game. Cette phase devra notamment couvrir les livrables de l'ISO26262 et de l'ISO21448 suivant :

- Les Analyses Safety déductives et inductives, notamment les arbres de défaillances pour les systèmes complexes, incluant l'analyse des modes communs.
- Les Safety Concepts incluant les Safety Requirements jusqu'à leur allocation aux niveaux hardware et software.

Les activités de cette phase Game peuvent conduire à modifier les données d'entrée établies en phase Pre-Game et donc le backlog du système, conduisant à la mise à jour de l'analyse d'impact.

Chaque itération est testée/validée avant livraison. Les éléments développés via les itérations sont intégrés et testés jusqu'au niveau du système développé, d'abord grâce à des simulations au cours des premières itérations, puis progressivement de simulation et modèle vers des moyens d'intégration physiques de plus en plus proche du système fini.

Pour les tests de validation nécessitant un temps long comparativement aux durées d'itération de la phase Game (tests d'endurance, tests environnementaux, tests liés à une industrialisation...), les itérations permettront de cadencer le lancement et le suivi des tests et d'assurer la validation complète de la livraison associée. L'exécution pourra se faire au travers de plusieurs itérations.

3.3.1.2 Cas d'application ISO26262 :

En phase **Pre-Game**, Il est possible de décliner tous les objectifs de sécurité en exigences de manière itérative, en définissant une « safety story » à chaque objectif, et d'attribuer les exigences de sécurité dérivées aux sous-systèmes ou éléments concernés.

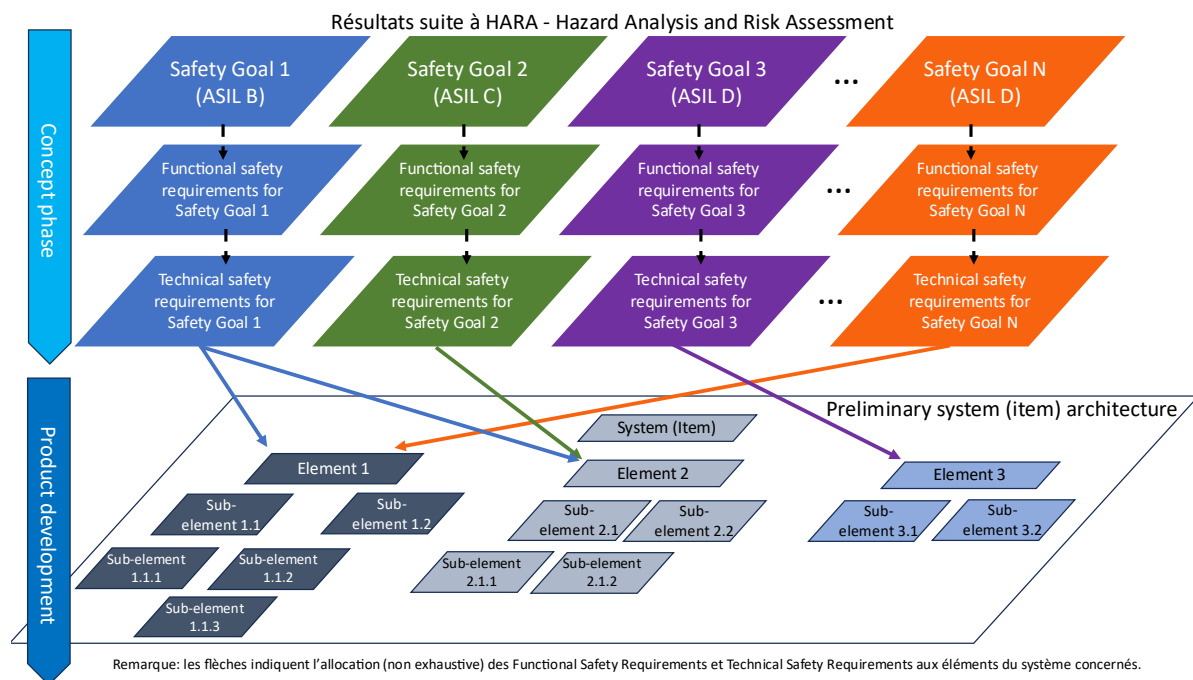
Pour ce faire, il est nécessaire de commencer par les objectifs de sécurité dont les niveaux d'intégrité de sécurité (ASIL) attribués sont les plus élevés (par ordre décroissant de niveau, D → A). Cela signifie que le concept de Sécurité Fonctionnelle pourrait être réalisé par le biais d'itérations couvrant fonctionnel et objectifs de sécurité associés.

Les prérequis nécessaires sont l'identification de toutes les fonctions du système à l'étude (item definition) et après que la HARA est effectuée sur cette base fonctionnelle, les Objectifs de Sécurité dédiés à toutes les fonctions identifiées.

Durant la phase **Pre-Game**, cette liste d'objectifs de sécurité est supposée la plus complète possible afin de récolter les niveaux d'intégrité de sécurité (ASIL) les plus forts pour appliquer une décomposition ASIL des éléments de l'item.

Remarque : particulièrement en phase de **Pre-Game** durant laquelle il peut encore y avoir des hypothèses de périmètre fonctionnel et technique encore non confirmées, la vision incrémentale, peut s'appuyer sur des principes d'analyses SEooC (Safety Element out of Context) si certains prérequis ou certaines données d'entrée sont manquants.

Pour illustrer les précautions nécessaires avant d'entamer les itérations dédiées aux safety stories, voici un exemple de décomposition et d'allocation des exigences de Sécurité Fonctionnelle sur un système abstrait en s'appuyant sur son architecture :

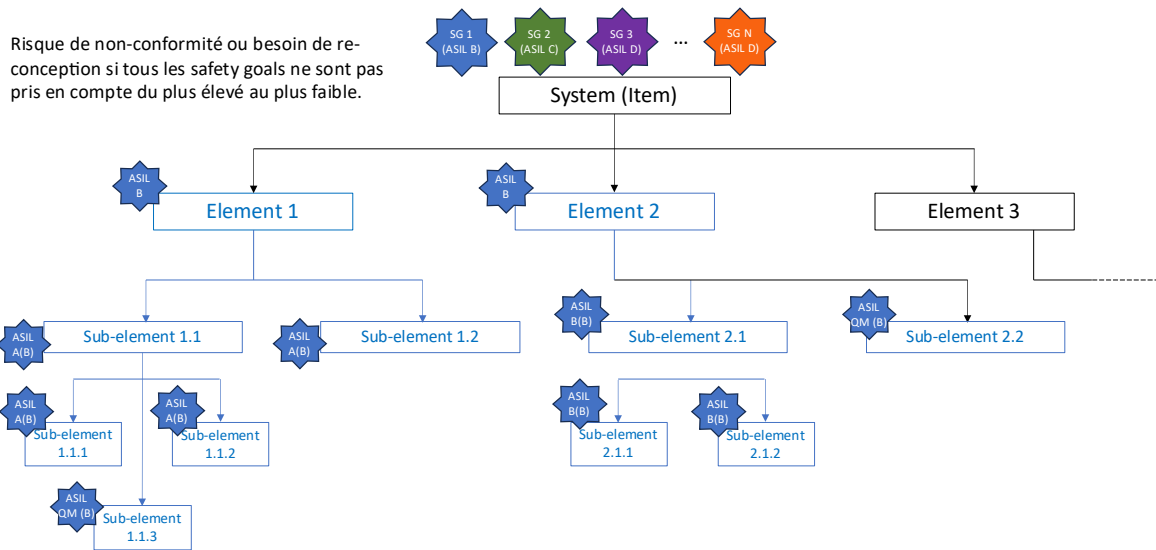


La décomposition et l'allocation aux éléments du système (item) réalisée par Objectif de Sécurité est représentée sans vue architecturale du système en admettant que les sprints dédiés aux safety stories 1, 2, 3 et 4 aient été tous accomplis dans l'ordre chronologique (1 → 4).

Or deux objectifs de sécurité (SG 1 et SG N) sont alloués au même élément (élément 1). Pour une meilleure efficacité, il est préférable de s'assurer de l'exhaustivité de la liste des objectifs de sécurité pour toutes les fonctions portées par le système.

Si ces activités sont réalisées sans prendre en compte l'ordre décroissant des niveaux ASIL (D → A), le résultat des premières décompositions et allocations des exigences aux éléments du système (item) sera caduque du fait des niveaux ASIL insuffisants si un objectif de sécurité ultérieur avec un niveau ASIL plus important est détecté au travers de la finalisation de l'HARA qui n'a pas été exhaustive jusqu'à lors.

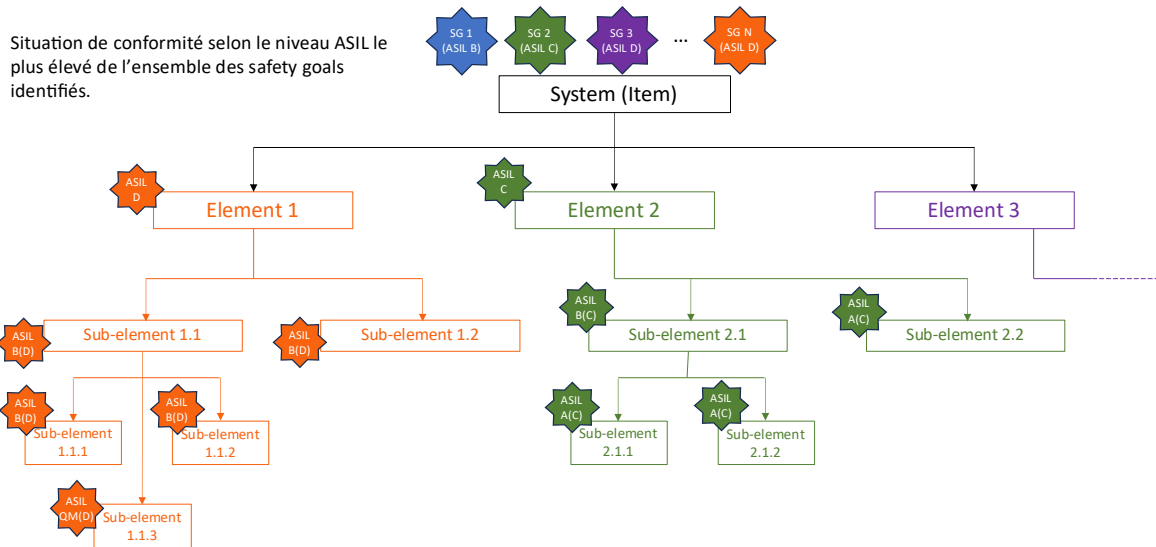
Dans la figure ci-dessous, on peut voir le résultat de cette analyse dans le cas chronologique (en considérant uniquement l'objectif de sécurité 1) :



Notes:

- En bleu figure la décomposition ASIL selon le Safety Goal 1 (ASIL B) seulement.
- Exemple de décomposition basée sur système abstrait.

Dans la figure suivante, voici ce que donnerait le résultat de cette analyse en prenant en compte l'ordre décroissant de tous les objectifs de sécurité :



Remarque :

Il est important de considérer les plus hauts niveaux ASIL alloués aux éléments du système en phase Pre-Game car le niveau ASIL déterminera par les tables liées aux exigences de l'ISO 26262 en phase de développement produit (mode game), les méthodes à suivre en termes d'implémentation, de vérification et de validation et ce, aux niveaux système, matériel (électronique voire mécanique) et logiciel.

3.3.1.3 Cas d'application SOTIF :

Pour chacune de ces itérations, il est possible d'appliquer l'approche AGILE, avec les mêmes précautions que pour la Sécurité Fonctionnelle et donc de définir des « SOTIF EPIC » puis « SOTIF stories » en Pre-Game pour les impacts de la SOTIF sur la définition du système et la spécification du design (architecture de l'item) et donc de s'assurer que les activités SOTIF sont intégrées au backlog.

La SOTIF story est applicable à tout le système (capteurs, unités logiques, actionneurs) :

- Élément de détection comme : caméra, lidar, radar...
- Post-traitement des données comme : fusion de données de détection, classification de données....
- Traitement d'algorithmes complexes comme : logiques de prise de décision, gestion des priorités, transfert du contrôle au conducteur...
- Actionnement comme : la performance pour réaliser les commandes de contrôle dans les délais nécessaires sous des contraintes spécifiques...

Tous ces sous-systèmes ou éléments du système sont concernés par les SOTIF epics/stories à leurs niveaux respectifs.

C.1.3.3.1.3.1 Phase Pre-Game :

En phase pregame, le plus important est de réaliser les étapes 5 et 6 de la première itération.

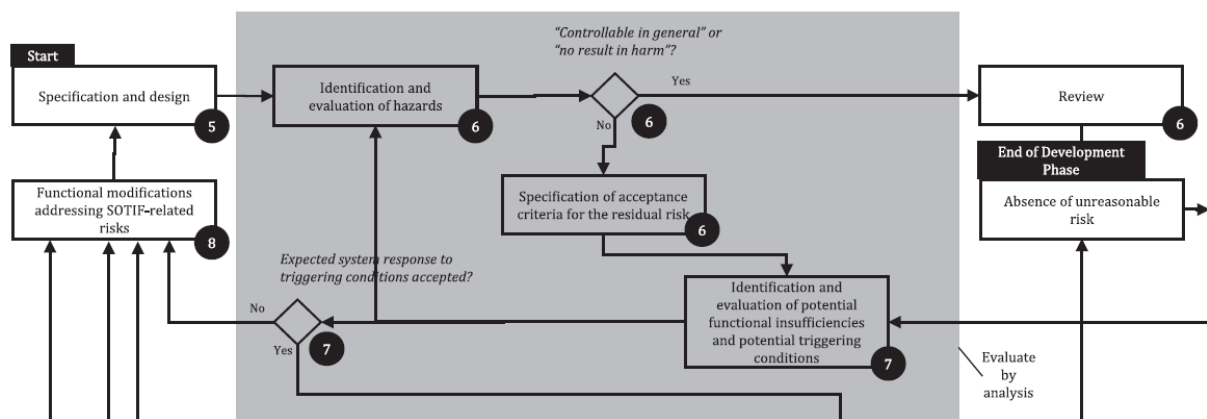


Figure 41 - Enchaînement des activités décrites par l'ISO21448 (partie 1/2)

Pour la clause 5 « Spécification et conception », les principaux aspects à prendre en compte sont les éléments suivants :

- une première version d'un Safety Plan avec les éléments structurants,
- la description de la fonctionnalité prévue et les fonctionnalités des sous-systèmes et composants de support, notamment avec l'ODD (domaine de conception opérationnelle) ;
- la conception du système concerné et de ses éléments mettant en œuvre la fonctionnalité prévue ;
 - les objectifs de performance des capteurs, contrôleurs, actionneurs ou autres entrées et composants installés permettant la fonctionnalité prévue ;
- les dépendances, interactions ou interfaces de la fonctionnalité prévue avec :
 - le pilote ;

- l'interface du pilote (par exemple, IHM) et la manière dont l'interface est utilisée pour atténuer les abus connus et raisonnablement prévisibles ;
- l'opérateur distant/de back-office (usagers de la route, conditions environnementales, l'infrastructure routière, ...)
- les usages impropres raisonnablement prévisibles (directs et indirects) ;
- les limitations de performances potentielles, les conditions de déclenchement identifiées et les contre-mesures du système et de ses éléments ;
- les architectures du système et du véhicule mettant en œuvre la fonctionnalité prévue ;
- le concept d'avertissement et de dégradation ;
- les points structurants des procédures de collecte et de surveillance des données pendant et après le développement de la fonctionnalité prévue ; et
- le mécanisme, la conception et les exigences qui prennent en charge les capacités d'atténuation des risques pendant l'exploitation.

Cette clause 5 peut être associée à une « SOTIF story » réitérée en vertu des critères d'acceptation des risques résiduels en mode Pre-Game pour la clause 6 et en mode Game en vertu des résultats V&V faisant ressortir des insuffisances fonctionnelles liées à des limites de performances.

Pour la clause 6, « Identification et évaluation des dangers/hasards », les objectifs sont de définir :

- le danger au niveau du véhicule,
- l'évaluation des risques liés aux comportements dangereux,
- les critères d'acceptation,

Cette clause est moins exposée à des réitérations car tous les dangers sont supposés être identifiés durant cette analyse. Cependant des tests de vérification et validation effectués en simulation ou sur route peuvent détecter des limites de performances et/ou des dangers jusqu'alors non-envisagés avant cette activité.

Pour la clause 7 « Identification et évaluation des limitations de performance et des événements déclencheurs potentiels », cela consiste à obtenir :

- Les insuffisances potentielles identifiées des spécifications, insuffisances de performances et conditions de déclenchement (y compris une mauvaise utilisation directe raisonnablement prévisible) ;
- L'évaluation de la réponse du système aux conditions de déclenchement identifiées pour leur acceptabilité par rapport au SOTIF.

Les clauses 6 et 7 peuvent aussi faire l'objet de « SOTIF stories » en Pre-Game mais aussi en Game s'il est nécessaire de retoucher l'architecture du système pour atteindre certains critères d'acceptation en traitant certaines insuffisances fonctionnelles.

Enfin, la clause 8 « Modifications fonctionnelles pour répondre aux risques liés à la SOTIF » peut aussi faire l'objet d'une « SOTIF story » en Pre-Game afin de réaliser la spécification SOTIF, prérequis pour enrichir la spécification et la conception du système. Elle sera certainement réitérée en mode Game à la suite des résultats de tests V&V.

Toutes les exigences qui découlent de ces phases de spécification (dont les critères d'acceptation) et d'analyse des risques sont prises en compte pour le mode Game.

C.1.3.3.1.3.2 Phase Game

Les clauses 9, 10, 11 et 12 figurant dans la partie V&V, seront traitées en mode **Game** car l'implémentation des exigences du système et leur vérification sont nécessaires (phase de développement produit) pour s'assurer que les spécifications SOTIF issues de la clause 8 ont

été prises en compte dans les modifications du design, c'est-à-dire après une itération de la clause 5.

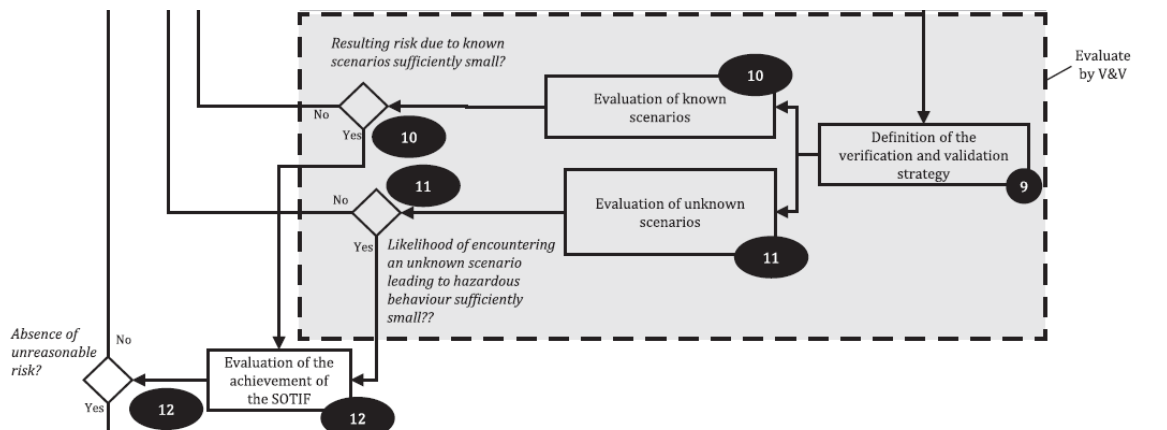


Figure 42 - Enchaînement des activités décrites par l'ISO21448 (partie 2/2)

Chaque clause fera également l'objet d'une « SOTIF story » qui pourra être réitérée via la définition d'itération d'exploration définis en Approche AGILE avec un objectif d'apprentissage.

La déclinaison de ces activités en sprints sera très utile pour :

- l'évaluation des scénarios potentiellement dangereux connus,
- l'exploration et évaluation de scénarios inconnus.

La figure ci-dessous représentant le modèle de zone de scénario SOTIF :

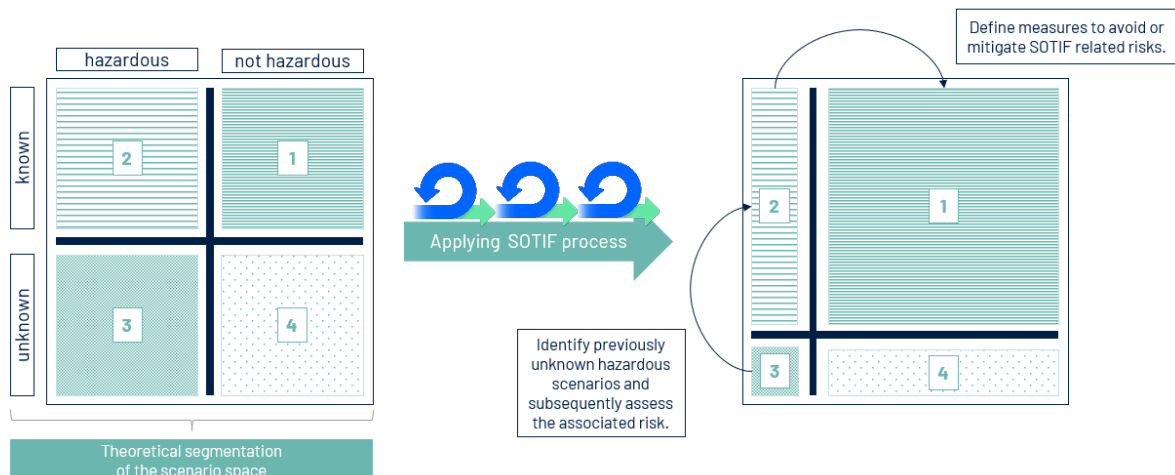


Figure 43- Modèle des zones des scénarios SOTIF

Comme le montre cette figure, l'objectif de la SOTIF est de réduire les aires 2, 3 et 4 au profit de l'aire 1.

Les scénarios à spécifier ou à jouer soit en simulation et réaliser avec des tests sur route font l'objet de définition de sprints dédiés :

- critères de performance (insuffisances fonctionnelles),
- critères d'acceptation de risque résiduel.

La satisfaction de ces critères permet de valider que les objectifs SOTIF sont atteints. Cela est mesurable par la diminution significative des aires 2, 3 et 4 correspondant à un nombre de scénarios définis initialement et le nombre résiduels de ces scénarios avec des risques non encore atténués (aire 2), voire non explorés pour les « scénarios inconnus » (aires 3 et 4, non directement quantifiable).

$$\sum \text{scenarios} = \sum \text{scenarios}_{(\text{aire } 1)} + \sum \text{scenarios}_{(\text{aire } 2)} + \sum \text{scenarios}_{(\text{aire } 3)} + \sum \text{scenarios}_{(\text{aire } 4)}$$

Soit $\tau_{\text{scenario coverage}}$ le taux de couverture des scénarios permettant d'obtenir un taux de risque résiduels acceptable :

$$\tau(\text{scenario coverage}) = \frac{\sum \text{scenarios}(\text{aire } 1)}{\sum \text{scenarios}}$$

Un critère de couverture des cas étant défini, on peut comparer cette valeur un taux de risques résiduels acceptables (par les critères d'acceptation) $\tau_{\text{SOTIF_fullfillment}}$.

Si $\tau_{\text{scenario_coverage}} \geq \tau_{\text{SOTIF_fullfillment}}$ alors les objectifs SOTIF sont atteints,

Sinon une exploration de scénarios supplémentaires doit être effectuée et des sprints dédiés doivent être définis pour ce faire.

En approche AGILE, cette métrique peut être utilisée au cours du développement pour évaluer l'état d'avancement du développement :

$$\tau(\text{scenario coverage}) = \frac{\sum \text{scenarios_identifiés_traités}(\text{aire } 1)}{\sum \text{scenarios_identifiés}(\text{aire } 2 + \text{aire } 1)}$$

Elle pourra être complétée avec le nombre de nouveaux scénarios hazardous identifiés pendant l'itération qui est un indicateur révélateur statistique du nombre de scénario hazardous non encore découverts.



new hazardous scenarios per iteration = *number of new scenarios identified in area 2*

C.1.3.3.1.3.1 Cas particulier du cycle de vie en Opération

Durant la phase de développement, les phases opérationnelles des générations déjà sur le marché génèrent des nouveaux scénarios impactant la phase Game (ou Pre-Game).

Durant la phase opérationnelle, de nouveaux scénarios non pris en compte peuvent être détectés et impacteront les équipes de développement chargées des évolutions sur ces versions de produit avec de nouveaux scénarios à prendre en compte. Sur ce type de développement, une phase Game semble également adaptée. Il conviendra tout de même de réaliser une analyse d'impact sur l'existant qui pourra éventuellement justifier une nouvelle phase Pre-Game.

3.3.2 DEFINITION DE LA PRIORITE DES EXIGENCES LORS DES ITERATIONS

3.3.2.1 Principe

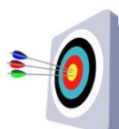
Comme décrit précédemment, le backlog système devra être classé par ordre de priorité pour définir le contenu de chaque itération. Plusieurs techniques d'aide à la décision peuvent être utilisées (The New Rational Manager – Decision analysis, CESAF...). Parmi ces techniques, le Framework CESAF propose d'identifier des Key Design Driver – KDD afin d'élaborer une matrice d'aide à la décision ou bien d'identifier les risques clés à traiter.

		Type A			Type B		
		Respecting all requests (size, mnt perf, opt perf)			Respecting mnt/opt perf requests while ensuring manufacturing yield		
Criteria	How to evaluate	low	mid	high	low	mid	high
Cost	Parts cost, assembly cost, Maintenance cost total	20%			10%		
Durability	Expected on-market failure rate	None					
Look	Design (does not include size)						
Weight	Total lens weight				100%		
Size	Total lens size		Same as competition		+10%		
Mnt perf	AF speed, AF follow perf, stabilization steps number			AF speed 2x AF follow = Stabilize +			AF speed 2x AF follow = Stabilize +
Ease of use	Evaluation (does not include weight)						
Optical perf	MTF value			MTF +10%			MTF +10%
Manufacturing	Lens assembly line yield rate (including parts supplier yield rate)	Yield rate: YY					Yield rate: XX

Figure 44 - Identification des KDD selon le Framework CESAF

Pour le périmètre de la Sécurité Fonctionnelle, il conviendrait donc d'identifier les critères clés associés à la satisfaction des exigences de Sécurité Fonctionnelle que nous nommerons KSDD Key Safety Design Driver. Ces critères peuvent être de différents types :

- Sécurité Fonctionnelle :
 - o Niveau ASIL d'un Safety Goal,
 - o Cibles FTTI – Fault Tolerant Time Interval (ISO26262-1/3.61), FHTI – Fault Handling Time Interval (ISO26262-1/3.56),
 - o Cible sur les défauts aléatoires HW,
 - o Fréquence d'apparition de scénario SOTIF...
- Disponibilité :
 - o Objectif de disponibilité...
- Résilience au changement :
 - o Niveau de modularité d'une architecture,
 - o Niveau de découplage aux interfaces (y compris sur des paramètres de calibration)...
- Autres contraintes notamment environnementales (ISO26262-part5 §10.4.4 Table 10 et §10.4.6 Table 12) :
 - o Hypothèses d'indice de Protection IP d'un calculateur,
 - o Niveau d'immunité en compatibilité électromagnétique,
- Production et Maintenance :
 - o Classification des caractéristiques,
 - o Capabilité pour la maîtrise des Caractéristiques Spéciales de Sécurité...



Les activités à réaliser dans chaque itération doivent être définies avec un niveau de priorité comme déjà décrit en Partie B.3.3 - Planification et Partie B.5.4.2 - 1.1.1 Maîtrise des risques techniques en mode de conception itératif que ce soit en ISO26262 ou SOTIF. Si une méthode de définition des priorités est utilisée (critère WSJF ou autre), il conviendra de s'assurer que la Sécurité Fonctionnelle est pondérée de manière à être correctement prise en compte dans une approche multicritères.

3.3.3 CONDITIONS DE LANCEMENT ET DE VALIDATION D'UNE ITERATION ET D'UNE RELEASE

3.3.3.1 Conditions pour lancer une itération :

Les conditions pour démarrer une itération sont les suivantes :

- "Definition of Ready" scalable au niveau des Stories:
Les exigences associées à la Story sont "prêtes" (DoR), c'est à dire qu'elles satisfont la gestion des exigences en vigueur dans le projet. Par exemple, elles ont été revues, discutées, ajustées (notamment si des **hypothèses** issues de la phase Pre-Game sont remises en cause), acceptées et prêtes pour implémentation,
Cette préparation se fait via le principe du "backlog refinement" qui consiste à préparer les éléments du backlog qui vont être pris en compte dans le ou les prochain(s) sprints.
Le découpage des "Stories" permet de les implémenter dans la durée d'une itération.
Il est recommandé de vérifier leur qualité avec le modèle "INVEST" (Independent Negotiable Valuable Estimable Small Testable) ou « 3C » (Card Conversation Confirmed).

Le processus de revue des exigences amonts défini dans l'ISO26262 (part 8 §9.2), doit donc évoluer vers un process plus continu et itératif en approche AGILE.

3.3.3.2 Conditions pour valider la fin d'une itération :

Les conditions pour valider la fin d'une itération sont les suivantes :

- les critères d'acceptation orientés valeur sont définis et vérifiés,
- les exigences process de la Sécurité Fonctionnelle sont intégrées dans les DoD « Scalable » (voir Partie B.6.3.3.1) avec par exemple des éléments parmi les suivants :
 - o Revue des analyses d'impact,
 - o Preuve du respect de règles métier (exemple : règles de conception, de codage...),
 - o Preuve de conformité aux exigences de performances, de conformité réglementaire, Sécurité Fonctionnelle, Cyber sécurité...
 - o Les verification review et confirmation mesures sont réalisées en cohérence avec le contenu de l'itération : ces revues partielles ou complètes permettront de détecter des écarts vis-à-vis des requis au plus tôt ou au moins de façon progressive,
 - o Les analyses de risques sont mises à jour.

L'ensemble de la documentation supportant l'itération est mise à disposition (fonctionnelle ou technique, rapport de test, analyses de risque, rapport issue des outils de modélisation MBSA...).



3.3.3.3 Conditions pour valider une release (interne ou externe) :

Les conditions pour valider le résultat d'une release sont les suivantes :

- La livraison est forcément testée :
 - o En s'appuyant sur les validations réalisées aux DoD de chaque itération,
 - o En prenant en compte les essais supplémentaires au plus haut niveau d'intégration,
 - o Soit par écarts liés aux évolutions avec en complément des tests de non-régression si nécessaire,
 - o Soit en totalité.
- Les critères d'acceptation (Sécurité Fonctionnelle et SOTIF) permettant de répondre à un besoin client interne ou externe sont remplis,
- Les Audits Sécurité fonctionnelles sont réalisés : le Plan de Sécurité Fonctionnelle est défini conformément aux exigences de l'ISO26262 et il est mis en œuvre,
- Les Verification Review et Confirmation Measures nécessaires sont finalisées,
- Les autorisations de roulage sont validées.

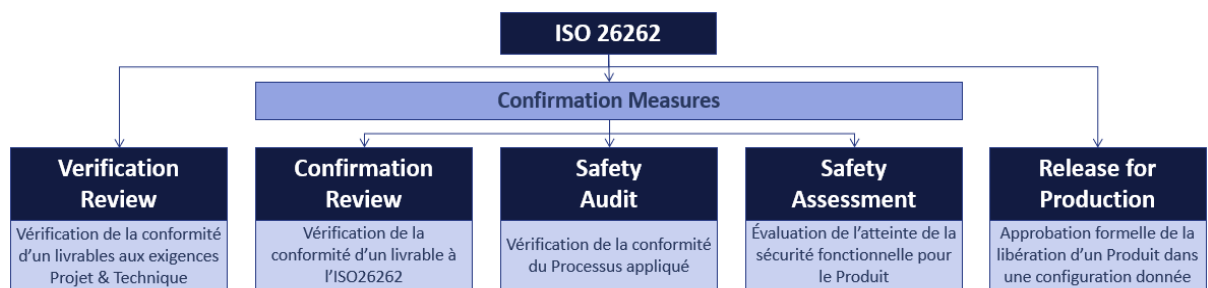
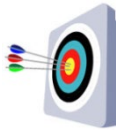


Figure 45 - Revue de vérification et de confirmation selon l'ISO26262

3.3.4 ORGANISATION ET SYNCHRONISATION DES EQUIPES

L'organisation des équipes sera clé pendant la phase itérative comme déjà discuté en Partie B.4.3.

Afin de faire les choix d'organisation et d'intégration des rôles liés à la Sécurité Fonctionnelle adaptés, les critères suivants pourront être pris en compte :

- Le niveau de complexité du Système :
 - Un haut niveau de complexité nécessitera une organisation favorisant les échanges et interactions,
- Le niveau d'indépendance des sous-systèmes les uns par rapport aux autres :
 - plus le niveau d'indépendance de certains lots sera élevé, plus les équipes associées pourront travailler en autonomie,
- Le niveau de maîtrise des risques produits :
 - un faible niveau de maîtrise des risques produits nécessitera de fortes interactions des équipes. Ces interactions seront d'autant plus une nécessité que les risques seront liés à des aspects multi-compétences,
- Le niveau d'innovation (intégration de nouveau principe/technologie, ou à l'opposé forte réutilisation de composants existants) :
 - de même que le niveau de maîtrise des risques, un fort niveau d'innovation ou un faible niveau de réutilisation nécessitera de fortes interactions des équipes. Ces interactions seront d'autant plus une nécessité que le niveau d'innovation sera lié à des aspects multi-compétences,
- Le niveau Safety :
 - les niveaux de Safety ASIL élevés nécessiteront de fréquentes interactions des rôles liés à la Sécurité Fonctionnelle avec les autres membres de l'équipe de développement.

3.3.4.1 Organisation statique des équipes

Pour rappel, plusieurs niveaux de responsabilité sont définis lors du déploiement d'approche AGILE sur des systèmes complexes :

- L'Equipe AGILE :
 - o Est responsable du livrable attendu avec deux rôles particuliers,
 - ⇒ Le Scrum Master responsable de la bonne prise en compte des règles applicables à l'équipe (règles AGILE, associées aux métiers techniques, à l'Entreprise...),
 - ⇒ Le Product Owner définit les priorités et objectifs pour l'équipe,
- L'Equipe d'Equipes :
 - o Assure la cohérence et la synchronisation des différentes équipes,
 - o Définit les livrables à usage interne ou externe y compris les livrables intermédiaires permettant la synchronisation des équipes,
- Les Services Partagés :
 - o Assure la disponibilité d'Expertises techniques, Qualité, Safety, Audit...



L'intégration des rôles liés à la Sécurité Fonctionnelle dans ces équipes pourra être adaptée au niveau de contrainte de Sécurité Fonctionnelle. Par exemple :

Niveau de contrainte Safety (ASIL, FTTI...)		
	Faible	Fort
Conception Safety	Peut-être réalisée par une contribution externe à l'équipe AGILE (Services Partagés)	Intégration dans l'Equipe AGILE et présence en continue (détection des écarts, analyse d'impact réactive, détection d'hypothèse non respectée...)
Safety Management	Peut-être réalisée par une contribution externe à l'Equipe d'Equipe (Services Partagés)	Intégration dans l'Equipe d'Equipes
	Constantes: - Comme pour le management des autres disciplines, le management Safety supporte l'organisation AGILE, - Le management est moins impliqué dans le suivi d'avancement des activités comparé à un mode traditionnel et plus dans le suivi du respect des process et de la qualité des work product livrés,	
	Inflexions & points d'attention: - la cohérence du planning de développement Agile par rapport aux requis de la Sécurité Fonctionnelle (process et fonctionnel), - le contenu des backlogs par rapport aux livrables projets attendus, - la vérification du bon niveau de Culture SAFETY des équipes, - un niveau d'implication variable en fonction du niveau de Culture SAFETY.	
Safety Assessment	Constante: - L'organisation choisie ne devra pas dégrader le niveau d'indépendance requis pour les rôles liés à la Sécurité Fonctionnelle et en particulier pour le rôle associé aux Safety Assessment.	
	Inflexion & point d'attention: - La charge associé peut être impactée par la fréquence des itérations et des changements plus élevée.	

Tableau 6 – Intégration des Fonctions Safety dans les équipes AGILES

3.3.4.2 Organisation dynamique des équipes

Les points suivants sont à prendre en compte pour l'organisation des équipes en dynamique:

Pour des Systèmes complexes avec contrainte de Sécurité Fonctionnelle, il peut être intéressant de définir des groupements d'équipes Agiles (Train pour SAFe) avec les équipes qui ont de fortes dépendances, par exemple avec des équipes fortement impactées par un concept de sécurité. Cela peut conduire à l'organisation des trains suivante :

- Groupes au niveau véhicule,
- Groupes « domaine » sur les systèmes interdépendants : ADAS, ESP, ABS, contrôle moteur,
- Groupes autonomes : infodivertissement, airbag, BMS, TPMS...

Dans le cas de domaines systèmes interdépendants, il est fortement recommandé de dédier une équipe aux activités d'intégration.

Il peut également y avoir un intérêt à créer des mécanismes de communication et synchronisation entre équipes de groupe / trains différents pour traiter des thématiques inter systèmes et fortement « réparties » dans le véhicule :

- Mode commun sur les concepts de sécurité,
- Performance bruit afin de traiter des dépendances entre isolation phonique habitacle et émission d'un organe...,
- Perfo thermique : émission d'un organe et point chaud dans l'habitacle,
- Et également : CEM, vibratoire, crash...

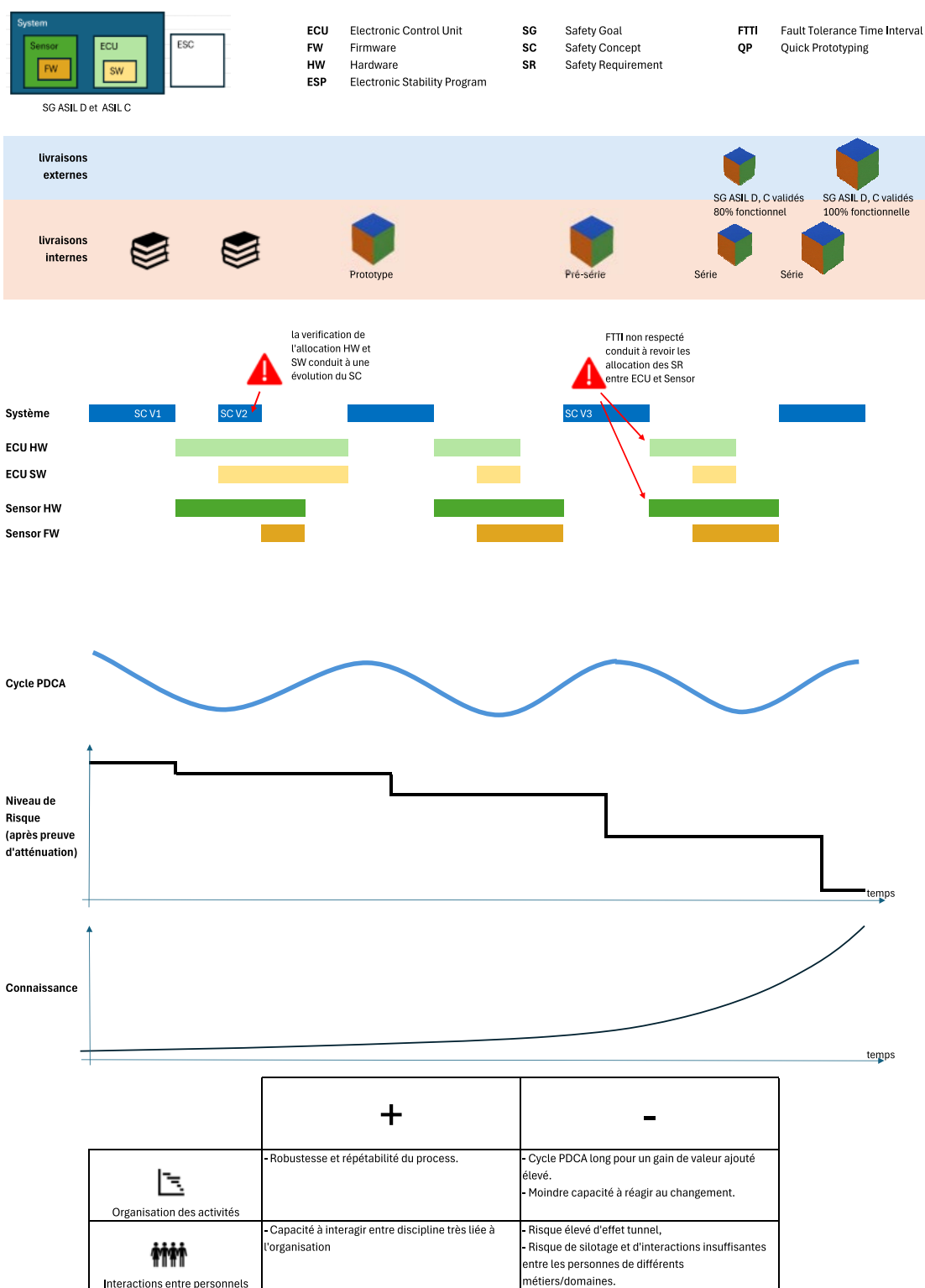
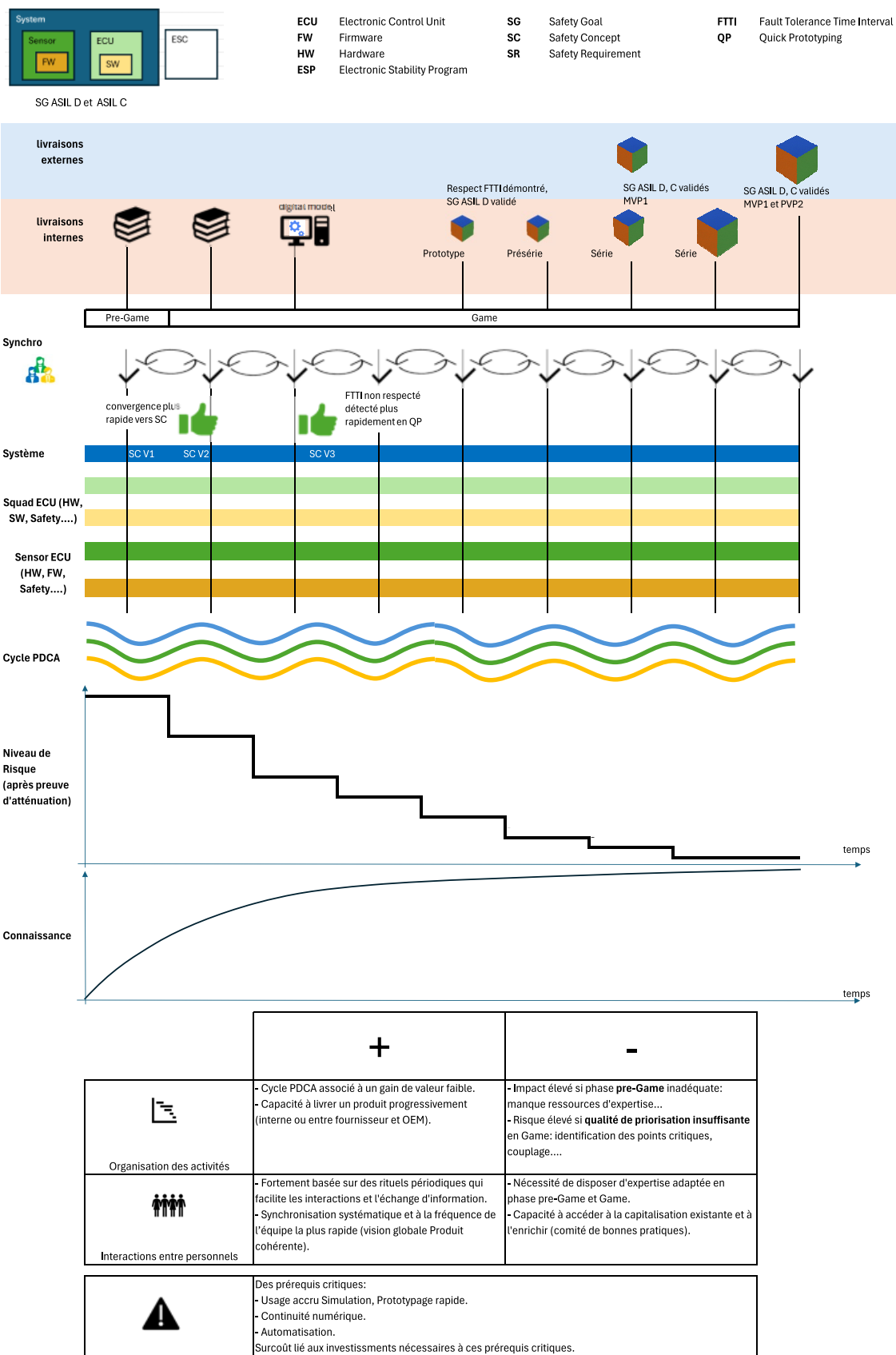


Figure 46 - Organisation dynamique des équipes en approche projet traditionnelle



* conformité démontrée à l'ensemble des exigences de l'ISO26262, y compris celles liées au process de développement et notamment des tables.

Figure 47 - Organisation dynamique des équipes en approche Agile

OUTILS ASSOCIES AU PROCESSUS DE DEVELOPPEMENT

Les moyens et outils logiciels mis en œuvre y compris ceux supportant les approches AGILE doivent répondre à des requis d'évaluation et de qualification comme le montre la figure ci-dessous.4 - part 8 chapitre 11 – Confidence in the use of Software tools.

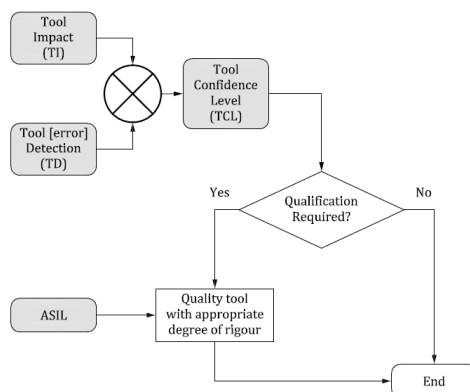


Figure 48 – Evaluation et Qualification d'un Software Tools (ISO26262 - Part 8 - §11 - Fig. 4)

Selon le résultat de l'évaluation (TCL),

		Tool error detection		
		TD1	TD2	TD3
Tool impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

Tableau 7 - Détermination du niveau TCL - Tool Confidence Level (selon ISO26262 - Part8 - Table 3)



le logiciel devra répondre à une qualification selon les process décrit par les tableaux 4 et 5 de la norme. Le tableau ci-dessous donne par exemple les conditions pour un niveau TCL3 (les méthodes '++' sont « highly recommended » et '+' « recommended » selon la définition de l'ISO26262).

Methods		ASIL			
		A	B	C	D
1a	Increased confidence from use in accordance with 11.4.7	++	++	+	+
1b	Evaluation of the tool development process in accordance with 11.4.8	++	++	+	+
1c	Validation of the software tool in accordance with 11.4.9	+	+	++	++
1d	Development in accordance with a safety standard ^a	+	+	++	++
^a No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected. EXAMPLE Development of the software tool in accordance with ISO 26262, IEC 61508, EN 50128 or RTCA DO-178C.					

Tableau 8 - Exemple de méthodes de qualification possibles pour un niveau TCL3

Et par ailleurs les outils liés à la Sécurité Fonctionnelle devront répondre aux contraintes AGILE avec les critères suivants :

- Ergonomie d'usage,
- Continuité numérique au sein d'un même outil, ou interfaces « seamless » entre outils,
- Capacité à gérer les nombreuses évolutions en Approche Agile de manière la plus automatisée possible.

4 GESTION DES EXIGENCES

4.1 RAPPEL DES PRINCIPES ET DES EXIGENCES

Dans le développement de systèmes critiques, la gestion des exigences de Sécurité Fonctionnelle constitue l'un des piliers les plus importants. Loin d'être une simple étape documentaire, elle se base sur l'ossature de l'ISO 26262 depuis la définition du safety goal (l'exigence de plus haut niveau) jusqu'à sa déclinaison et son allocation aux différents niveaux de conception système (ISO 26262, partie 4), matériel (ISO 26262, partie 5) et logiciel (ISO 26262, partie 6).

Respecter cette gestion rigoureuse des exigences n'est pas une option, elle impose la traçabilité, la cohérence et la qualité des exigences tout au long du cycle de vie. Ce socle est indispensable pour démontrer l'atteinte des objectifs de Sécurité Fonctionnelle. C'est dans ce cadre que nous rappelons ici les caractéristiques clés qu'une exigence safety doit respecter : clarté, cohérence, complétude, faisabilité, vérifiabilité parmi d'autres, afin d'assurer la robustesse de l'architecture et la fluidité du développement en agile.

4.1.1 CARACTÉRISTIQUES DES EXIGENCES SAFETY

Les exigences safety doivent répondre aux critères suivants :

a) **Caractéristiques essentielles des exigences.**

Les exigences safety doivent être :

- Sans ambiguïté : Formulation claire sans multiples interprétations ;
- Compréhensibles : Accessibles à toutes les parties prenantes ;
- Atomiques : Non divisibles exprimant un seul besoin ;
- Cohérentes : Sans contradiction avec d'autres exigences ;
- Réalistes et faisables : Techniquement et économiquement réalisables et applicables dans les contraintes de développement ;
- Vérifiables : Mesurables avec des critères d'acceptation définis ;
- Nécessaires : Indispensables au produit ou à la solution ciblés ;
- Sans contraindre la solution technique : décrivent le besoin sans imposer de solution technique spécifique ;
- Complètes : Suffisamment détaillées pour éviter des interprétations erronées ;
- Conformes : Alignées avec les normes et les réglementations applicables.

Chaque exigence doit disposer des attributs suivants :

- Une identification unique, inchangée tout au long du cycle de vie ;
- Statut d'avancement (en cours, à revoir, validée ...) ;
- ASIL attribué reflétant son niveau de criticité.

b) **Identifiables comme exigences safety** : Les exigences safety doivent être explicitement identifiées. Cela peut être réalisé en les regroupant ou en les marquant à l'aide d'un attribut spécifique ;

- c) **Héritage de l'ASIL** : Les exigences *safety* héritent de l'ASIL initial associé au *safety goal* tout au long de leur dérivation. Le *safety goal*, formulé pour prévenir ou atténuer un risque inacceptable, constitue le point de départ de ce processus. Par défaut, l'ASIL reste inchangé pour les éléments dérivés sauf si une décomposition permet de démontrer que ces exigences sont allouées à des composants suffisamment indépendants. Ce mécanisme de décomposition est essentiel pour ajuster le niveau d'intégrité requis, en cohérence avec le risque de défaillance estimé et peut permettre une réduction justifiée de l'ASIL ;
- d) **Allocation à un élément spécifique** : Chaque exigence *safety* doit être attribuée à un élément. Cela inclut des éléments au niveau système, matériel ou logiciel ou des niveaux plus granulaires tels que les composants logiciels visibles dans l'architecture et les unités logicielles (units), définies dans la conception détaillée;
- e) **Allocation à la Fabrication** : Les exigences produit et les études associées (AMDEC par exemple) permettent d'identifier les caractéristiques spéciales nécessaires à la conformité de production.

4.1.2 GESTION DES EXIGENCES SAFETY

La gestion des exigences *safety* repose sur une structuration et une organisation rigoureuse afin de garantir leur cohérence et leur maintenabilité tout au long du cycle de vie de développement.

L'ensemble des exigences *safety* dérivées d'un ou de plusieurs *safety goal(s)* doivent également respecter les principes suivants :

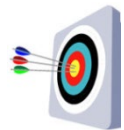
- Structuration: Les exigences sont organisées selon plusieurs niveaux successifs et regroupées par affinité (exemple par fonction). Il peut exister plusieurs niveaux hiérarchiques liés à la décomposition du Système;
- Traçabilité bidirectionnelle : des liens permettent d'identifier les exigences « parents » et « enfants » d'une exigence. Cela permet l'analyse d'impact et la couverture et de justifier les exigences dérivées (i.e. caractéristique « Nécessaire ») à partir de leurs exigences « parents » jusqu'au *Safety Goal* d'origine,
- Complétude : Les exigences *safety* d'un niveau doivent couvrir l'intégralité des exigences des niveaux précédents. Cela garantit la couverture de besoins identifiés,
- Absence de répétition : Chaque exigence doit être unique, aucune duplication ou répétition n'est tolérée dans un même niveau hiérarchique ;
- Possibilité d'évolution : les exigences doivent être versionnées et associées à une configuration (baseline). Le respect des caractéristiques du §4.1.1 facilitera la modification ou l'extension des exigences.

Lors de la conception d'un produit, différents outils sont mobilisés (par exemple des systèmes de gestion des exigences ou des gestionnaires de documents). Ces outils doivent permettre de structurer les exigences de manière hiérarchique, de les catégoriser (au moyen d'attributs) et d'en assurer la traçabilité.

Pour la gestion des évolutions, il est nécessaire d'identifier les exigences impactées pour l'ISO9001 (8.4.5.3), et d'assurer un lien bidirectionnelle entre la demande d'évolution et les exigences impactées (cf ASPICE).

Alors que la gestion des exigences en ingénierie classique se base sur une exhaustivité visée dès le début du cycle de conception puis leur traçabilité tout au long de ce processus, l'approche AGILE va privilégier :

- Une identification des exigences au fur et à mesure du développement (e.g. en identifiant les exigences applicables pour une itération donnée);



- La priorisation des exigences en les affectant à une itération donnée et dans une logique de Minimum Viable Product ;
- La présentation régulière du produit à l'utilisateur final afin d'obtenir son point de vue sur l'adéquation à son besoin et donc implicitement de valider la bonne compréhension et prise en compte des exigences.

4.2 Avantages et difficultés

En sécurité fonctionnelle, les notions d'identification progressive, de priorisation et de MVP seront à utiliser avec certaine limite puisque le client-utilisateur final attend un produit respectant l'intégralité des exigences de sécurité. Elles pourront cependant avoir un fort intérêt pour les livraisons entre équipes de développement, OEM.

Dans le périmètre SOTIF, les notions d'identification progressive des exigences, de priorisation, d'exposition à des contextes réels pour lever les risques résiduels semblent tout à fait adaptées.

Dans les deux cas, les caractéristiques des exigences devront être respectées et la traçabilité assurée.

Les points clés à adresser sont donc les suivants :

- Intérêts et limites à la notion de MVP, de priorisation en approche AGILE sur un périmètre de Sécurité Fonctionnelle ;
- Structuration des exigences ;
- Traçabilité des exigences de sécurité fonctionnelle en approche agile ;
- Gestion de la configuration des exigences,
- Qualité de la documentation afin de prouver la couverture des exigences.

4.3 CONDITIONS DE SUCCES

Concernant l'écriture et la gestion des exigences, certaines conditions de succès sont propres au développement en mode agile. D'autres, également présentes dans un mode de développement plus classique, sont rappelées ci-dessous comme point d'attention à ne pas oublier.

4.3.1 CONDITIONS DE SUCCES SPECIFIQUES AU MODE AGILE

4.3.1.1 Intérêt et limite à la notion de priorisation d'exigences et de MVP :

Dans le domaine de la sécurité fonctionnelle, les notions d'identification progressive des exigences, de priorisation et de MVP ne semblent pas avoir d'intérêt vis-à-vis du client-utilisateur final qui attend un produit respectant l'intégralité des exigences de sécurité. Cependant, elles peuvent en avoir un pour les équipes ou partenaires de développement d'un système afin de prioriser le travail, de rapidement tester et lever des risques identifiés. Elle permettra de faire coexister des développements en logique MVP et les développements de la sécurité fonctionnelle associée.

Dans le périmètre SOTIF, les notions d'identification progressive des exigences, de priorisation, d'exposition à des contextes réels pour lever les risques résiduels semblent par contre plus adaptées.

Les méthodes et critères pouvant aider à la priorisation des exigences sont discutés en §2.3.2.

4.3.1.2 Gestion et traçabilité des exigences

- **Gestion des exigences fonctionnelle et safety :**

Dans un contexte de développement Agile, la décomposition du travail en incréments fonctionnels doit préserver la cohérence et la complétude requises pour les exigences safety. Ce découpage doit prendre en compte l'agrégation et l'intégration ultérieures des éléments.

Pour cela il est recommandé :

- D'intégrer exigences fonctionnelles et exigences de sécurité fonctionnelle dans un même backlog,
- De spécifier tous les niveaux liés à une fonctionnalité dans une même itération ou dans des itérations adjacentes,
- Adopter une approche « verticale » lors de la définition puis l'implémentation des exigences safety. Cela consiste à couvrir au sein d'une itération produit l'ensemble des exigences depuis le Safety Goal jusqu'aux éléments matériels et logiciels en passant par les exigences de sécurité fonctionnelle (FSR-Functional Safety Requirements) et techniques (TSR-Technical Safety Requirements). Cette approche permet de garantir la cohérence globale et facilite la démonstration progressive de la conformité safety à condition d'avoir des hypothèses de safety concept robustes (voir §3.3.1 Pre GAME),
- Construire les documents d'interface, par exemple les DIML Dossier d'Interface Matériel et Logiciel, de manière interactive et itérative.

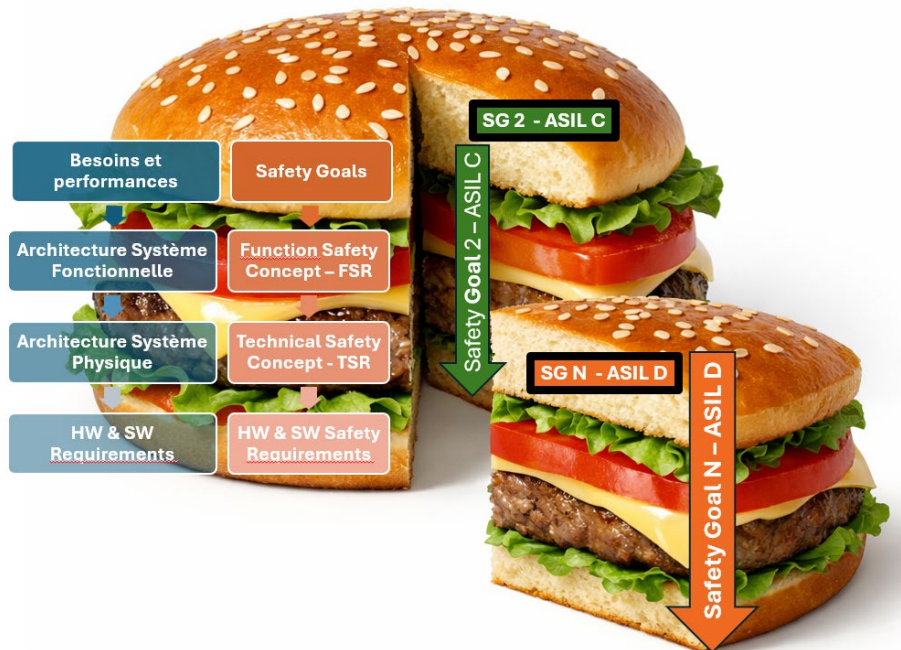


Figure 49 - Portion de Hamburger implémentant un Safety Goal

- **Structuration des exigences safety :**

La structuration des exigences safety est nécessaire depuis le niveau des Safety Goals, en passant par les exigences du Safety Concept, jusqu'au HSR-Hardware Safety Requirements, SSR-Software Safety Requirements.

- **Traçabilité des exigences safety :**

Une fois les exigences identifiées, il est nécessaire d'être capable de relier les exigences entre-elles :

- Par une Traçabilité bidirectionnelle sur les exigences Safety,
- Pour la gestion des évolutions, de s'assurer d'un lien bidirectionnel depuis les demandes d'évolutions jusqu'aux exigences impactées (cf ASPICE).

Pour cela les outils de ticketing traditionnels ne sont pas suffisants, ils ne distinguent en général pas une activité (caractère volatile) d'une exigence (caractère persistant) et ne sont pas conçus pour satisfaire aux requis rappelés ci-dessus.

Pour satisfaire à ces exigences, ils doivent :

- Soit intégrer des fonctionnalités de gestion d'exigence ;
- Soit permettre une liaison avec des outils de gestion d'exigence en assurant le maintien en cohérence des deux.



Une pratique qui consisterait à adapter un outil de ticketing (exemple : tag « Safety Requirement ») sans vérifier que la solution satisfait à l'entièreté des exigences de traçabilité ne peut pas être retenue.

L'utilisation de deux outils distincts sans fonctionnalité permettant un maintien automatique de la cohérence entre les deux pourrait conduire à :

- une gestion manuelle avec des risques d'incohérence, et donc à des risques de non-conformité,
- une perte d'efficacité sur la gestion de ces exigences.

- **Gestion de la configuration des exigences :**

Le respect des règles sur la structuration et la traçabilité permettront de répondre plus facilement aux requis de gestion de la configuration :

- Identification d'ensemble d'exigences (baseline/référentiel d'exigences),
- Gestion des évolutions par le moyen de versionnage et de capacité à identifier les exigences amonts et avalés impactées en cas d'évolution.

4.3.1.3 Rigueur de la maintenance documentaire

En approche itérative, et donc avec une fréquence d'évolutions élevée, il faudra être particulièrement vigilant à la maîtrise de la configuration et de la définition des exigences afin que malgré les nombreuses évolutions, il soit toujours possible d'assurer la traçabilité et la cohérence de l'ensemble, ainsi que l'accessibilité à l'ensemble des configurations ayant été générées durant le cycle de développement.

3.3.2 QUELQUES POINT D'ATTENTION A NE PAS OUBLIER

Organisation du travail :

Les développements en mode agile officialisent le traitement par "lots/paquets" de fonctionnalités du plus haut niveau jusqu'à la réalisation et la validation. Dans ce contexte, la décomposition du travail en incréments fonctionnels doit préserver la cohérence et la complétude requises pour les exigences safety. Ce découpage doit prendre en compte l'agrégation et l'intégration ultérieures des éléments. Pour évaluer la maturité atteinte à un instant donné, il est crucial de maîtriser la structuration des éléments – leur capacité à s'intégrer de manière fiable pour réaliser les exigences safety. Par exemple, pour savoir si l'on a atteint 50% ou 80% du développement du système, il faut s'assurer que les 50% ou 20% restant ne vont pas entraîner une reconception.

Lors de la structuration du travail en mode agile, il est recommandé de spécifier les différents niveaux liés à une fonctionnalité dans une même itération ou des itérations adjacentes, depuis le niveau système jusqu'aux éléments matériels et logiciels. Ceci permet d'avancer au même rythme et en cohérence avec les Objectifs de Sécurité, les Functional Safety Requirements (FSR), Technical SR, Hardware SR et Software SR, d'aller rapidement du concept haut niveau à la démonstration de la réalisation. On évitera donc de cadencer indépendamment les différentes couches de conception.

On gardera aussi la cohérence entre le fonctionnel et le dysfonctionnel. Pour ce faire, on évitera de traiter la safety dans un backlog séparé. Les documentations de synthèse, comme le DIML pourront être construites de manière collaborative, itérative et incrémentale. On décrira avec attention les interfaces à enjeux de safety entre le matériel et le logiciel liées au lot de l'incrément en cours en atteignant le niveau de maturité cible sur ce périmètre, les autres interfaces pouvant être complétées par la suite.

Dynamique et maturité :

Concernant les mesures de confirmation au sein de la norme ISO 26262, le développement en mode agile soulève le défi de la maturité nécessaire pour conduire ces « revues ». Il n'est pas nécessaire d'attendre qu'un livrable soit entièrement finalisé pour engager une mesure de confirmation dont l'objectif est de valider la conformité d'un point de vue de la Sécurité Fonctionnelle et ISO 26262. Par exemple si le développement agile privilégie la conception de certaines parties d'une architecture, il est possible que l'analyse safety complète des safety concepts (fonctionnel et technique) ne puisse être menée qu'à un stade plus avancé du projet.

Notre recommandation est de ne pas attendre cette complétude mais de construire l'analyse de manière incrémentale en menant également des mesures de confirmation périodiques. Le processus agile structuré autour de rituels réguliers offre une opportunité d'intégrer ces revues d'une manière efficace notamment pour les fonctions transverses telles que la Sécurité Fonctionnelle ou la Qualité. Cela permet de maintenir une synchronisation rapprochée et efficace avec les équipes de développement.

Au niveau matériel, le fait de ne développer qu'une partie du produit tout en souhaitant démontrer sa faisabilité peut entraîner un besoin accru de prototypage ou la nécessité de s'appuyer encore plus sur la modélisation.

D'un point de vue de la Sécurité Fonctionnelle et dans la continuité de la notion de cohérence « verticale » évoquée pour les exigences, nous recommandons que le DIML qui décrit les interfaces critiques entre matériel et logiciel soit construit d'une manière itérative. Les éléments nécessaires à un incrément produit doivent atteindre un niveau de maturité suffisant tandis que les autres peuvent être définis ultérieurement.

La réussite de cette approche réside sur la priorisation rigoureuse des sujets à fort enjeu en termes de ressources afin de garantir une construction incrémentale sans remise en cause des éléments déjà validés.

Dans un développement logiciel, les exigences safety doivent également intégrer les contraintes de l'usine qui attend une maturité suffisante pour garantir la fabrication de véhicules conformes. Cette attente représente un défi supplémentaire notamment face à des modifications de logiciel parfois tardives. Le cadre Agile vient formaliser ce mode évolutif et interactif et peut donc faciliter la prise en compte de l'usine comme une partie prenante des incréments produit afin de soutenir la démarche de Sécurité Fonctionnelle. Cela permet de réduire les risques de défauts logiciels critiques impactant la chaîne de production et d'assurer une réactivité élevée en cas de problème.

5 SYNTHÈSE ET POINTS OUVERTS

En synthèse, les principaux apports des Approches Agiles sur le périmètre de la Sécurité Fonctionnelle sont les suivants :

- Travailler en milieu incertain,
- Optimiser le temps de développement,
 - Les activités de différentes strates d'ingénierie débutent en parallèle,
- Anticiper les risques et problèmes,
 - Les validations sont anticipées,
- Gérer les évolutions,
 - La prise en compte de modifications tardives est facilitée, et cadrée par l'Approche AGILE pratiquée (« protection » des équipes des changements intempestifs pendant l'itération), néanmoins le processus Safety de gestion des évolutions doit être respecté (analyse d'impact),
- Réduire les effets tunnels.

Vis-à-vis des contraintes et limites de mise en œuvre, nous retiendrons les points suivants :

- La logique MVP (Minimum Viable Product) ne peut s'appliquer pour le client final sur la Functional Safety :
 - Une fonction livrée au client final doit être sûre,
 - Une livraison fonctionnelle interne sans (tous) les mécanismes de sécurité reste possible
- La logique MVP qui s'appliquerait sur la partie Fonctionnelle entrainerait la livraison du scope Functional Safety associé ainsi que la vérification de la non régression de l'existant.
- Afin de pouvoir tester plus tôt et de manière plus fréquente, la disponibilité des moyens de validation physique devra être anticipée et budgétée.
 - Risque pour réaliser les tests d'intégration et de validation avec des moyens représentatifs de la solution finale,
 - Nécessité d'automatiser les tests.

Les recommandations pour mises en œuvre des Approches Agiles:

- Assurer des fondations culturelles communes Agile et Safety,
- Synchroniser les développements Fonctionnels et Safety,
- Maîtriser le lancement de la phase itérative et la priorisation des activités,
- Adresser les contraintes Safety de traçabilité des Exigences & Gestion de configuration,
- Adapter l'intégration des rôles et fonctions Safety aux équipes d'équipes et équipes Agiles,
- Renforcer l'usage de tests virtuels et de la simulation afin d'identifier au plus tôt des risques ou non conformités vis-à-vis des tests physiques finaux,
- Intégrer les Vérifications et Confirmations Measures au Backlog.

Une prochaine version viendra compléter ce chapitre avec les éléments suivants :

- Conception et validation,
- Processus Métier,
- Documentation.

Annexe 1 : Liste des figures et des tableaux

LISTE DES FIGURES

Figure 1- Besoins Internes et Externes du monde Automobile.....	11
Figure 2 - Canevas de classification des techniques et méthodes [Guérineau et al.,2018]	12
Figure 3 – Flux d'activité selon Scrum Framework @ Scrum.org	16
Figure 4 – Flux d'activité KANBAN représenté selon un Tableau de bord KANBAN	18
Figure 5 - Répartition de l'usage des Frameworks par entreprise selon StateOfAgile 2022	20
Figure 6 - Organisation du Scrum of Scrums (Altclassia)	21
Figure 7 - Framework Scrum@Scale @ 2022 Scrum Inc.....	21
Figure 8 - Déclinaison des configurations de SAFe aux différents niveaux de l'Entreprise.....	22
Figure 9 - Différents modes d'hybridation possible entre Agile et Waterfall @PMI 2022.....	24
Figure 10 - Domaines du Framework Cynefin.....	25
Figure 11- Matrice de Stacey adaptée aux managements projets logiciels.....	26
Figure 12- Grille d'identification des points de vigilance liés aux approches Agile ou Classique.	27
Figure 13 - Les 4 leviers principaux pour booster les chances de réussite de votre transformation AGILE selon l'étude McKinsey - The Impact of agility www.mckinsey.com	29
Figure 14 - Thèmes des problématiques "Agile et Qualité"	32
Figure 15 - "Multiple horizon planification" - Industrial DevOps	38
Figure 16 – Éléments de Planification en approche AGILE : Synchronisation et Releases.....	39
Figure 17 - Extrait du Manifeste AGILE.	48
Figure 18 - Processus général de Maitrise des risques.....	49
Figure 19 - ROAM board proposé par SAFe (@Scale Agile, Inc)	51
Figure 20 - Possible complément des activités SCRUM avec les rituels d'identification, de gestion et traçabilité des risques	53
Figure 21 - Cycle en V dans le développement de logiciel automobile proposé par [Pfeffer, 2019].....	55
Figure 22 - Modèle de développement en V.....	56
Figure 23 – Etapes de Validation Virtuelle "au plus tôt"	60
Figure 24 - Complémentarité Numérique / Physique - Becamel & Di Valentin - SIA 2023	61
Figure 25 - Enrichissement des modèles numériques par les données de test - Becamel & Di Valentin - SIA 2023.....	61
Figure 26 - Build incrementally with fast feedback - SAFe - Scale Agile Inc.	62
Figure 27 - Processus de conception IHM @Veethree Group	63
Figure 28 - Résultat IHM à chaque étape du processus de conception @Veetree Group	63
Figure 29 - Fréquente intégration "end-to-end" - SAFe - Scale Agile Inc.....	64
Figure 30 - Exemple de définition de « Scaled DoD » selon SAFe	64
Figure 31 - Interactions entre MBD et Validation du modèle final.	65
Figure 32 - Représentation visuelle du « T-shaped », du site www.taleez.com	74
Figure 33 - Matrice des compétences	76
Figure 34 - Contribution des principales normes automobiles à l'obtention de Système Hautement Sûr.....	81
Figure 35 - Thèmes des problématiques "Agile et Sécurité"	84
Figure 36 - Structuration des Exigences selon ISO26262 (Figure 2 part 8).....	88
Figure 37 - Rappel des activités principales SOTIF.....	89
Figure 38 - Niveau d'incertitude des développements ISO26262 et SOTIF.....	90
Figure 39 - Enchaînement des activités décrites par ISO21448.....	91
Figure 40 - Schéma itératif en Sécurité Fonctionnelle selon SCA3DA	92
Figure 41 - Enchaînement des activités décrites par l'ISO21448 (partie 1/2)	96

Figure 42 - Enchaînement des activités décrites par l'ISO21448 (partie 2/2)	98
Figure 43- Modèle des zones des scénarios SOTIF.....	98
Figure 44 - Identification des KDD selon le Framework CESAF.....	100
Figure 45 - Revue de vérification et de confirmation selon l'ISO26262.....	102
Figure 46 - Organisation dynamique des équipes en approche projet traditionnelle	106
Figure 47 - Organisation dynamique des équipes en approche Agile.....	107
Figure 48 – Evaluation et Qualification d'un Software Tools (ISO26262 - Part 8 - §11 - Fig. 4) ..	108
Figure 49 - Portion de Hamburger implémentant un Safety Goal	112

LISTE DES TABLEAUX

Tableau 1 - Point clé du Framework SCRUM.....	17
Tableau 2 - Point clé du Framework KANBAN	19
Tableau 3 - Comparaison des Frameworks LeSS et SAFe.....	23
Tableau 4 - Vue d'ensemble de sujets Safety adressés par différentes normes.	80
Tableau 5 - Domaine de déploiement des Approches AGILES.....	82
Tableau 6 – Intégration des Fonctions Safety dans les équipes AGILES	104
Tableau 7 - Détermination du niveau TCL - Tool Confidence Level (selon ISO26262 - Part8 - Table 3)	108
Tableau 8 - Exemple de méthodes de qualification possibles pour un niveau TCL3	108

Annexe 2 : Références bibliographiques

- Berthou, V. (2022). *Les deux racines conceptuelles de l'agilité*, Cahiers COSTECH numéro 5. <http://www.costech.utc.fr/CahiersCOSTECH/spip.php?article151>.
- Hackman, J. R., & Vidmar, N. (1970). *Effects of size and task type on group performance and member reactions*. American Sociological Association.
- Henrik Kniberg, A. I. (2012). *Scaling Agile @ Spotify*. <https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf>.
- Intacs. (2022, July). *Agile SPICETM*. Récupéré sur intacs® website: <https://intacs.info/index.php/component/rsfiles/download-file/files?path=SPICE%2BDocuments%252FAgile%2BSPICE%252FAgile%2BSPICE%2BPAM%2Bv1.2%2BReleased.pdf&Itemid=750>
- Koné, T. F. (2021). *Contribution à la démonstration de la sécurité du véhicule autonome, basée sur une stratégie de génération de scénarios, modélisée par niveaux d'abstraction et orientée par la sensibilité du VA, pour une validation par simulation*. Université de Lorraine. HAL Open Science.
- Leite, A. I. (2023). *A method to the specification of safety requirements in agile contexts*. Sao Paulo: ICMC Sao Carlos.
- Matthew Skelton, M. P. (2019). *Team Topologies*. IT Revolution.
- PMI. (2022). *PMBOK Guide V7*. Switzerland.
- Reinertsen, D. G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Pub.
- Scaled Agile, I. (2021, December). *Agile Contracts*. Récupéré sur Scaled Agile Framework: <https://www.scaledagileframework.com/agile-contracts/>
- Scaled Agile, I. (2023, February). *Lean UX*. Récupéré sur Scaled Agile Framework: <https://scaledagileframework.com/lean-ux/>
- Schwaber, K. (1995). *SCRUM Development Process*. Advances Developments Methods.
- Stacey, R. D. (2011 (6th edition)). *Strategic Management and Organisational Dynamics: the challenge of complexity to ways of thinking about organisations*. London: Pearson Education - ISBN 978-0-273-70811-7.
- Sutherland, J. (2001). *Agile Can Scale: Inventing and Reinventing SCRUM in Five Companies*.
- Valentin, M. (2020 (2nd édition)). *Le modèle Tesla: du Toyotisme au Teslisme : la disruption d'Elon Musk*. DUNOD.
- W BECAMEL L DI VALENTIN - Renault Group & Stellantis. (2023). *La simulation numérique au service des projets*. SIA.
- LeSS "Architecture & Design " <https://less.works/less/technical-excellence/architecture-design#ArchitectureDesign>
- Jack Reeves (2005 essay): "One of Code As Design: What is Software Design?"
- Craig Larman (1992 essay): "that final source code is the real software design"
- Craig Larman (2016 book): "Large-Scale Scrum: More with LeSS" ISBN 0321985710
- Craig Larman (2016 book): "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development" ISBN 0131489062
- Patrick Debois (2007): "DevOps"
- Patrick Debois (2008 conference) "Agile 2008 Toronto Agile Infrastructure and Operations Presentation"
- [Michaël Valentin](#) (2020): "Le modèle Tesla - 2e éd. - Du toyotisme au teslisme : la disruption d'Elon Musk: Du toyotisme au teslisme : la disruption d'Elon Musk" ISBN 2100806017
- Matthew Skelton et Manuel Pais (2019) : "Team Topologies " ISBN 978-1942788812

William BECAMEL - Renault Group | Laurent DI VALENTIN - Stellantis (2023) "La simulation numérique au service des projets"; "<https://www.sia.fr/publications/752-simulation-service-projets>"

Annexe 3 : Grille d'identification des points de vigilance liés aux approches Agiles ou Classiques

Bénéfices et risques différenciants associés aux approches agiles ou classiques											
		Faible						Fort			
		Caractérisation du périmètre produit ou du projet									
ENJEUX PRODUIT	1	Instabilité des exigences et des demandes produit	👍 x								++
	2	Capacité de modifier à faible coût le Produit et à livrer partiellement	👍	x							
	3	Taux de nouveauté vs. standardisation	👍							x	
	4	Enjeu de rupture planning								x	
	5	Quantité d'innovations peu matures à intégrer								x	
ENJEUX ORGANISATIONNELS	6	Autonomie de compétences de l'équipe pour délivrer le produit ou le projet	👍	x							
	7	Capacité à avoir des lots techniques indépendants sur le projet	👍				x				
	8	Capacité de «design change» sans impact contractuel	👍								
	9	Autonomie de décision sur le terrain et subsidiarité	👍						x		
	10	Possibilité d'optimiser le niveau d'exigence documentaire	👍				x				
	11	Capacité du processus et des moyens de validation à valider en boucle courte les changements	👍						x		
	12	Maturité digitale de l'organisation	👍				x				

x caractérisation de l'environnement du projet

👍 / 👍
Bénéfice / risque projet classique
👍 / 👍
Bénéfice / risque projet agile