

TMMi dans le monde Agile

Version 1.4

Produit par la Fondation TMMi

Auteur: Erik van Veenendaal

Traduction française réalisée par : Françoise BARBEAU, Arnaud CHARPENTIER, Eric LEGAL (traducteurs) et Eric RIOU du COSQUER (relecteur)

Contenus

Table des matières

1. Introduction	7
1.1. Objectifs	7
1.2. TMMi et Agile.....	7
1.3. Intégration du modèle de maturité des tests (TMMi)	9
1.4. Agile	10
1.5. Amélioration des processus de test dans un contexte Agile	11
2. TMMi Niveau 2 : Géré	13
2.1. Domaine de processus Politique et Stratégie de test.....	13
2.1.1. SG 1 Établir une politique de test	13
2.1.2. SG 2 Établir une stratégie de test	14
2.1.3. SG 3 Établir des indicateurs de performance des tests	15
2.2. Domaine de processus Planification des tests.....	15
2.2.1. SG 1 Effectuer une évaluation des risques	16
2.2.2. SG 2 Établir une approche test	16
2.2.3. SG 3 Établir des estimations de test	18
2.2.4. SG 4 Élaborer un plan de test	19
2.2.5. SG 5 Obtenir un engagement sur le plan de test.....	20
2.3. Domaine de processus 2.3 Surveillance et contrôle des tests.....	20
2.3.1. SG 1 Surveiller les progrès des tests par rapport au plan.....	21
2.3.2. SG 2 Contrôler la qualité des produits par rapport au plan et aux attentes	23
2.3.3. SG 3 Gérer les mesures correctives jusqu'à la clôture.....	23
2.4. Domaine de processus 2.4 Conception et exécution des tests	24
2.4.1. SG 1 Analyse et conception de tests à l'aide de techniques de conception de tests	24
2.4.2. SG 2 Effectuer la mise en œuvre des tests	26
2.4.3. SG 3 Exécution des tests	27
2.4.4. SG 4 Gérer les incidents de test jusqu'à la clôture	29
2.5. Domaine de processus 2.5 Environnement de test.....	30
2.5.1. SG 1 Élaborer des exigences en matière d'environnement de test.....	31
2.5.2. SG 2 Mise en œuvre de l'environnement de test	31
2.5.3. SG 3 Gérer et contrôler les environnements de test	31
3. TMMi Niveau 3 : Défini	33

3.1.	Domaine de processus 3.1 Organisation de test	33
3.1.1.	SG 1 Mettre en place une organisation de test	33
3.1.2.	SG 2 Établir des fonctions de test pour les spécialistes des tests.....	34
3.1.3.	SG 3 Établir des programmes de formation en test.....	35
3.1.4.	SG 4 Déterminer, planifier et mettre en œuvre les améliorations du processus de test.....	35
3.1.5.	SG 5 Déployer le processus de test organisationnel et intégrer les leçons apprises.....	36
3.2.	Domaine de processus 3.2 Programme de formation aux tests.....	36
3.2.1.	SG 1 Mettre en place une capacité de formation aux tests dans l'organisation	37
3.2.2.	SG 2 Fournir la formation nécessaire aux tests	38
3.3.	Domaine de processus 3.3 Cycle de vie et intégration du test.....	38
3.3.1.	SG 1 Établir les atouts du processus de test organisationnel	38
3.3.2.	SG 2 Intégrer les modèles de cycle de vie des tests avec les modèles de développement..	40
3.3.3.	SG 3 Établir un plan de test maître	40
3.4.	Domaine de processus 3.4 Tests non fonctionnels.....	41
3.4.1.	SG 1 Effectuer une évaluation des risques non fonctionnels des produits	42
3.4.2.	SG 2 Établir une approche de test non fonctionnel.....	42
3.4.3.	SG 3 Analyse et conception de tests non fonctionnels.....	43
3.4.4.	SG 4 Effectuer la mise en œuvre des tests non fonctionnels	44
3.4.5.	SG 5 Exécution de tests non fonctionnels.....	45
3.5.	Domaine de processus 3.5 Revues par les pairs	45
3.5.1.	SG 1 Établir une approche de revue par les pairs.....	46
3.5.2.	SG 2 Effectuer des revues par les pairs.....	47
4.	TMMi Niveau 4 : Mesuré.....	49
4.1.	Domaine de processus 4.1 Mesure des tests	49
4.1.1.	SG 1 Activités de mesure et d'analyse des tests.....	49
4.1.2.	SG 2 Fournir le résultat de la mesure des tests	50
4.2.	Domaine de processus 4.2 Mesure des tests	51
4.2.1.	SG 1 Des objectifs de projet mesurables pour la qualité des produits et leurs priorités sont établis	51
4.2.2.	SG 2 Les progrès réels dans la réalisation des objectifs de qualité des produits du projet sont quantifiés et gérés	51
4.3.	Domaine de processus 4.3 Examens avancés.....	52
4.3.1.	SG 1 Coordonner l'approche de revue par les pairs avec l'approche des tests dynamiques	52
4.3.2.	SG 2 Mesurer la qualité des produits au début du cycle de vie par le biais de revue par les pairs	52

4.3.3.	SG 3 Ajuster l'approche du test en fonction des résultats de l'examen au début du cycle de vie	52
5.	TMMi Niveau 5 : Optimisation	54
5.1.	Domaine de processus 5.1 Prévention des anomalies	54
5.1.1.	SG 1 Déterminer les causes communes des anomalies	55
5.1.2.	SG 2 Prioriser et définir les actions visant à éliminer systématiquement les causes communes des anomalies	55
5.2.	Domaine de processus 5.2 Contrôle de la qualité	56
5.2.1.	SG 1 Établir un processus de test statistiquement contrôlé	56
5.2.2.	SG 2 Tester à l'aide de méthodes statistiques	57
5.3.	Domaine de processus 5.3 Optimisation du processus de test	57
5.3.1.	SG 1 Sélectionner les améliorations du processus de test	57
5.3.2.	SG 2 Les nouvelles technologies de test sont évaluées pour déterminer leur impact sur le processus de test	58
5.3.3.	SG 3 Déployer les améliorations des tests	58
5.3.4.	SG 4 Établir la réutilisation des ressources de test de haute qualité	58
6.	Vue d'ensemble de l'applicabilité des objectifs et pratiques spécifiques de TMMi	58
6.1.	Évaluations TMMi	58
6.2.	TMMi niveau 2 - Géré	59
6.3.	TMMi niveau 3 - Défini	59
6.4.	TMMi niveau 4 - Mesuré	60
6.5.	TMMi niveau 5 - Optimisation	60
	Références	61

Contributeurs

Asim Ali	(United Arab Emirates)
Katalin Balla	(Hungary)
Clive Bates	(UK)
Jan Jaap Cannegieter	(The Netherlands)
Vahid Garousi	(The Netherlands)
Alon Linetzki	(Israel)
Fran O'Hara	(Ireland)
Jurian van de Laar	(The Netherlands)
Leanne Howard	(Australia)
Poonam Jain	(India)
Tim Moore	(UK)
Alfonsina Morgavi	(Argentina)
Meile Posthuma	(The Netherlands)
Matthias Rasking	(Germany)
Chaobo Shang	(China)
Erik van Veenendaal	(Bonaire – Caribbean Netherlands)
Blaine Webb	(UK)
Karolina Zmitrowicz	(Poland)

Note des traducteurs : Anglicismes conservés

Nous avons pris le parti de laisser certains termes en anglais afin de garantir la cohérence avec d'autres traductions de documents CMMI, TMMi, ISTQB ou IREB. Voici les principaux :

Terme conservé en anglais	Signification ou traduction française possible
User story	Récit/histoire utilisateur
Definition of Done (DoD)	Définition de Terminé
Definition of Ready (DoR)	Définition de Prêt
Product Owner	Représentant des clients et utilisateurs du produit
Backlog	Liste d'éléments à réaliser pour terminer le produit
Burndown Chart	Ne se traduit pas en français. Equivalent d'un tableau d'avancement.
Extreme Programming (XP)	Pas de traduction française, méthode
Scrum	Ne se traduit pas (est le terme anglais pour la mêlée du rugby), cadre de travail pour réaliser des produits complexes
Kanban	Terme japonais qui signifie étiquette. Avec une majuscule, fait référence à un cadre de travail agile.
Test-Driven Development (TDD)	Développement piloté par les tests
Behavior-Driven Development (BDD)	Développement piloté par le comportement
Acceptance Test-Driven Development (ATDD)	Développement piloté par les tests d'acceptation
Mean-Time-Between-Failures	Temps moyen entre défaillances
Goal-Question-Metric (GQM)	Approche par les buts, But-Question-Mesure
Risk Poker	Poker de risques

1. Introduction

1.1. Objectifs

Ce document explique comment combiner une approche Agile pour le développement de logiciels avec le modèle de maturité du processus de test TMMi pour atteindre vos objectifs métier. Il explique comment TMMi peut être utilisé et appliqué de manière bénéfique dans un contexte Agile. Des alternatives éprouvées aux approches de test traditionnelles qui mettent en œuvre les pratiques TMMi, sont fournies tout en maintenant et peut-être même en augmentant l'agilité. Ce document explique comment TMMi peut aider les entreprises Agiles, quel que soit leur degré de maturité, en leur rappelant les pratiques de test critiques qui perdent souvent de leur visibilité au fur et à mesure que l'entreprise se développe et que la pression sur les projets augmente. Chaque domaine de processus de TMMi et ses objectifs spécifiques sont abordés un par un. Il est indiqué comment ceux-ci sont liés à l'agilité et à quoi ressembleront généralement les pratiques. Si une pratique n'est pas censée être exécutée dans un contexte Agile, puisqu'elle n'a pas de valeur ajoutée, cela est également explicitement indiqué.

De nombreuses (petites) entreprises Agiles sont prospères et en pleine croissance, mais manquent souvent de documentations de processus et de formalisation de la transmission. Pour maintenir leur succès et la croissance elles devront être plus rigoureuses dans l'évolution et la formalisation de leurs processus, au risque de perdre la culture Agile, clé du succès initial. C'est l'un des défis pour les initiatives d'amélioration des processus de test TMMi dans un environnement Agile.

Ce document s'adresse à de nombreux publics, les deux principaux étant :

- les entreprises traditionnelles matures qui veulent passer aux méthodes Agiles tout en conservant la maturité de leurs processus ;
- les entreprises Agiles qui ont du succès et qui se développent. En conséquence, elles doivent allier maturité des processus à la flexibilité des méthodes agiles.

Ce document n'est pas destiné à être utilisé indépendamment du modèle TMMi original. Il aide à l'interprétation et à la compréhension des différents objectifs et pratiques de TMMi, tout en donnant des conseils pour l'amélioration des processus de test en environnement Agile. Ce document complète le modèle TMMi original et doit être utilisé en complément.

1.2. TMMi et Agile

Contrairement à une croyance répandue les approches TMMi et Agile ne sont pas incompatibles. Ces deux approches (Agile et TMMi) peuvent non seulement coexister, mais, intégrées avec succès, elles apportent des avantages substantiels. Une autre difficulté, double : envisager les tests différemment, pleinement intégrés au développement agile et les implications dans un contexte d'amélioration du "test". Notez que le "i" de TMMi fait référence au fait que les tests doivent faire partie intégrante du développement logiciel, et non être traités comme un sujet complètement à part. La littérature et les présentations sur les tests dans les projets Agiles ont tendance à se concentrer sur les tests unitaires, l'automatisation des tests et les tests exploratoires, mais le périmètre est plus large ! L'utilisation du modèle TMMi dans un contexte Agile rappelle des pratiques de test essentielles qui sont souvent "oubliées". Ce document montrera, à l'aide d'exemples, que les méthodes TMMi et Agile

peuvent fonctionner efficacement ensemble. Le défi consiste à appliquer les principes des méthodes Agile pour renforcer les pratiques Agiles et faciliter les pratiques TMMi.

Lors de la mise en œuvre de TMMi, il faut tenir compte du fait que le modèle TMMi n'a pas pour but "d'imposer" un ensemble de pratiques à une entreprise, ni d'être appliqué comme une norme à laquelle il faut "prouver la conformité". Utilisé de manière appropriée, le modèle TMMi peut vous aider à localiser les domaines de test spécifiques où le changement peut apporter une valeur ajoutée compte tenu des objectifs de l'entreprise. Cela reste vrai quel que soit le modèle de cycle de vie appliqué. Il est important de toujours se rappeler que les pratiques TMMi sont une composante attendue, mais qu'il est possible de s'écarter des pratiques TMMi définies, en utilisant des pratiques dites « alternatives ». Il faut toujours s'interroger sur le but de la pratique, sa raison d'être et comment elle apporte une valeur ajoutée à l'entreprise. Souvent, dans une culture Agile, l'objectif est déjà atteint, mais par le biais d'une pratique alternative. En général, "toute" solution est conforme tant qu'elle est motivée par les besoins de l'entreprise ! Lorsque vous utilisez TMMi, ne soyez pas trop normatif, TMMi a été conçu différemment. Interprétez toujours les objectifs et les pratiques de TMMi dans votre contexte. En général, en établissant d'abord les besoins en matière de processus dans le contexte spécifique de votre entreprise, des décisions peuvent être prises sur la manière de cibler et de mettre en œuvre les priorités d'amélioration des processus.

La plupart des conflits qui surviennent entre TMMi et Agile sont basés, soit sur une vision historique de TMMi (de ce à quoi une "bonne pratique" devrait ressembler lorsqu'elle est mise en œuvre), soit sur une mauvaise compréhension de la raison d'être des pratiques Agiles, basée sur la manière dont elles devraient soutenir les valeurs Agiles. Les experts TMMi, y compris les responsables des évaluations (Lead assessor), devront reconsidérer et éventuellement repenser les messages qui pourraient être partagés par inadvertance concernant ce à quoi devrait ressembler une "bonne pratique conforme à TMMi" quand elle est mise en œuvre. Lorsque les approches Agiles sont déployées de manière appropriée avec les processus TMMi, cela se traduira par la mise en œuvre effective des pratiques de test, et non par leur suppression. Il convient de noter qu'en plus de pratiques (de test) spécifiques au sein de TMMi, il existe également des pratiques génériques. L'objectif des pratiques génériques est de soutenir l'institutionnalisation d'un domaine de processus, ce qui signifie effectivement que l'entreprise doit disposer d'une organisation capable de soutenir le domaine de processus lorsque de nouvelles personnes sont embauchées ou que d'autres changements interviennent au sein de l'entreprise.

Passer des méthodes traditionnelles de développement logiciel aux méthodes Agiles peut également faire ressortir l'initiative visant à élaguer et alléger les processus tels qu'ils sont définis aujourd'hui. De cette manière, les entreprises basées sur TMMi bénéficieront de l'état d'esprit Agile. Les gens ont tendance à lire dans le modèle TMMi des choses qui n'y sont pas et à créer ainsi des processus et des produits d'activité inutiles et sans valeur ajoutée. En revenant aux origines et aux objectifs d'amélioration et en utilisant le modèle TMMi tel qu'il est prévu, on favorisera l'alignement des processus à réelle valeur ajoutée sur les nécessités et objectifs opérationnels des processus. L'élagage et l'allègement des processus dans un esprit Agile permettront d'obtenir des processus qui reflètent les activités réelles et garantiront que seuls les éléments nécessaires sont utilisés. Un état d'esprit Agile mettra également l'accent sur la simplification, ce qui n'est généralement pas si facile, mais bénéficiera à ceux qui mettent en œuvre TMMi. Les améliorations dans un cadre Agile se feront généralement par le biais de petites équipes responsabilisées qui peuvent agir rapidement, ce qui est une autre façon pour TMMi de tirer profit de son agilité. N'oubliez pas non plus qu'il y a une rupture naturelle entre le niveau 3 et les niveaux 4 et 5 de TMMi. Il est recommandé aux entreprises, en particulier aux entreprises agiles, de choisir et d'appliquer de manière critique les pratiques des

niveaux 4 et 5 de TMMi, qui sont importantes et qui apportent de la valeur. Même si TMMi est complet, les entreprises doivent, pour réussir, identifier les pratiques de test et les améliorations clés sur lesquelles elles doivent se concentrer.

1.3. Intégration du modèle de maturité des tests (TMMi)

Le cadre TMMi a été développé par la Fondation TMMi en tant que ligne directrice et cadre de référence pour l'amélioration des processus de test, en abordant les questions importantes pour les gestionnaires et ingénieurs de test, les développeurs et les professionnels de la qualité des logiciels. TMMi définit le test dans son acceptation la plus large pour englober toutes les activités liées à la qualité des produits logiciels.

TMMi utilise le concept de niveaux de maturité pour évaluer et améliorer les processus. En outre, les domaines de processus, les objectifs et les pratiques sont identifiés. L'application des critères de maturité de TMMi améliore le processus de test et a un impact positif sur la qualité des produits, la productivité de l'ingénierie de test et les ressources nécessaires par temps de cycle¹. TMMi a été développé pour aider les entreprises à évaluer et à améliorer leurs processus de test.

TMMi dispose d'une architecture par étapes pour évaluer et améliorer les processus. Elle contient des étapes ou des niveaux par lesquels une entreprise passe au fur et à mesure de l'évolution de son processus de test, passant d'un processus ad hoc et non géré à un processus discipliné, ajusté, géré quantitativement et en optimisation. L'atteinte de chaque étape garantit que tous ses objectifs ont été atteints et que les améliorations constituent le socle de l'étape suivante.

La structure interne de TMMi est riche en pratiques de tests qui peuvent être apprises et appliquées pour soutenir un processus de test de qualité qui s'améliore de manière incrémentale. TMMi compte cinq niveaux de maturité qui se complètent hiérarchiquement pour donner une trajectoire progressive et évolutive d'amélioration du processus de test. Pour atteindre la maturité de chaque niveau, l'entreprise doit mettre en œuvre l'ensemble de domaines de processus associés. Les domaines de processus pour chaque niveau de maturité de TMMi sont présentés dans la *figure 1*.

L'un des principaux principes sous-jacents de TMMi est qu'il s'agit d'un modèle générique applicable à divers modèles de cycle de vie et environnements. La plupart des objectifs et des pratiques définis par TMMi se sont révélés applicables à des modèles de cycle de vie séquentiels et itératifs, y compris Agiles. Toutefois, au niveau le plus bas du modèle, de nombreuses sous-pratiques et exemples fournis sont (très) différents selon le modèle de cycle de vie. Notez que dans le cadre de TMMi, seuls les objectifs sont obligatoires, pas les pratiques.

TMMi est disponible gratuitement sur le site web de la Fondation TMMi. Le modèle a été traduit en espagnol, en français et en chinois. TMMi est également disponible sous forme de livre publié.

¹ Temps de cycle : de l'émergence de l'idée à sa livraison/utilisation

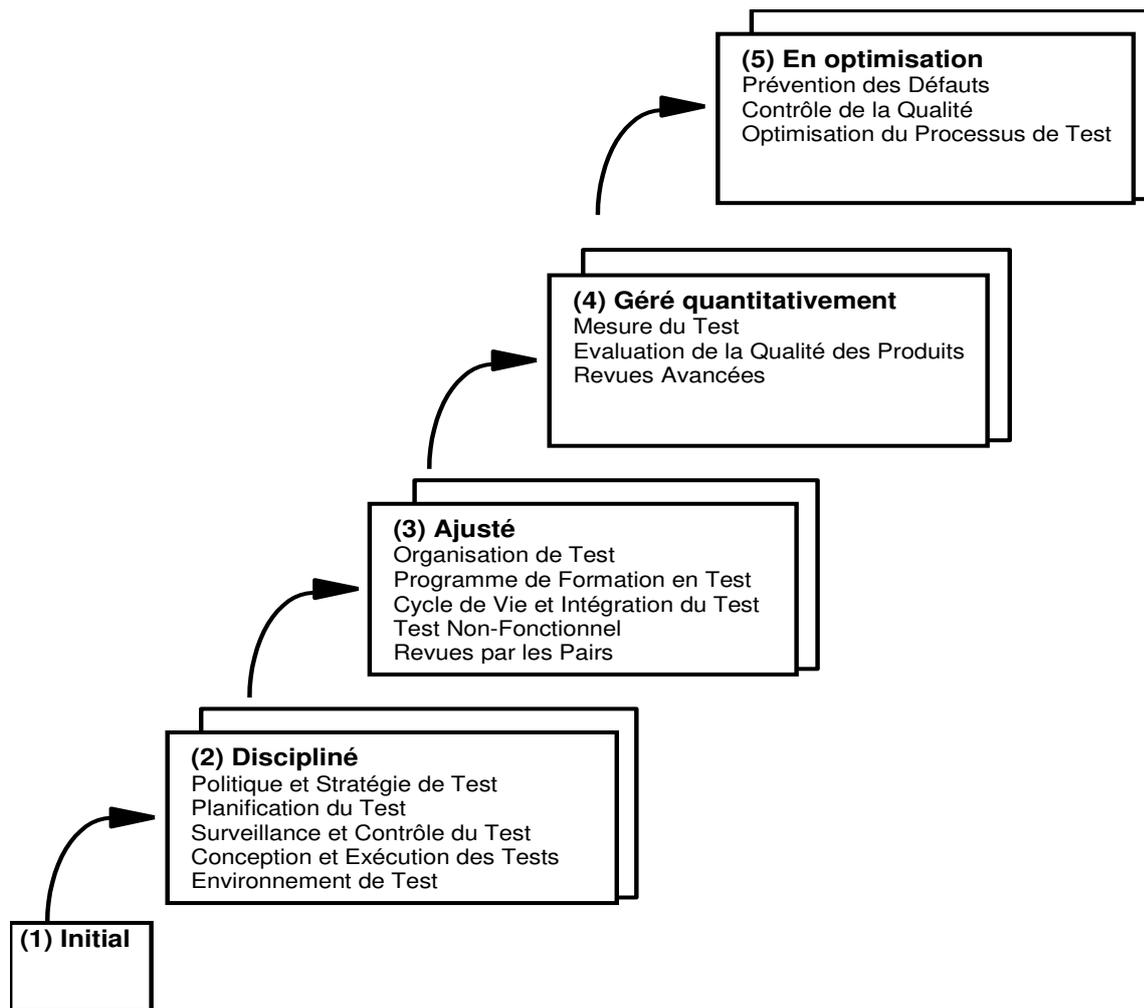


Figure 1 : Niveaux de maturité et domaines de processus de TMMi

1.4. Agile

En 2001, un groupe d'individus, représentant les méthodologies de développement logiciel léger les plus utilisées, s'est mis d'accord sur un ensemble commun de valeurs et de principes qui est devenu connu sous le nom de Manifeste pour le développement logiciel agile ou Manifeste Agile. Le Manifeste Agile contient quatre valeurs. Il s'agit de 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

Le Manifeste Agile reconnaît la valeur des concepts de droite, mais privilégie les premiers (leur accordant une plus grande valeur). Il n'existe pas de « méthode Agile ». Le manifeste a été rédigé et signé par les représentants de plusieurs méthodes, à l'époque qualifiées de légères, parmi lesquelles les plus connues sont *Extreme Programming* (XP) et *Scrum*. Le Kanban (Kanban pour l'IT plus précisément) est arrivé plus tard.

1.5. Amélioration des processus de test dans un contexte Agile

Les méthodes Agiles sont fortement axées sur des équipes autonomes, responsables de leur propre amélioration continue, notamment par des rétrospectives fréquentes. Certaines de ces améliorations peuvent porter sur les tests. Elles peuvent concerner un axe d'amélioration des pratiques de tests pour franchir un niveau de maturité TMMi supérieur, mais peuvent également aborder les thèmes de la méthodologie de test de l'équipe, du processus de test en place dans l'équipe. Il est très important de confirmer que l'amélioration globale du processus de test ne limite pas les actions d'amélioration au sein des équipes Agile, mais qu'elle est au contraire complémentaire.

Certains des principaux aspects à prendre en compte, lorsque l'on examine l'influence de l'Agilité sur le contexte d'amélioration sont les suivants :

- fréquence du cycle d'amélioration ;
- aspects organisationnels ;
- portée des améliorations ;
- sources d'améliorations ;
- niveau de documentation (test) ;
- méthodes d'amélioration.

Dans les projets Agiles, les améliorations ont généralement lieu à l'intérieur de boucles régulières de rétroaction qui permettent d'envisager des améliorations fréquentes du processus de test (par exemple, à la fin d'un Sprint en *Scrum*). Comme la portée est souvent limitée au cycle ou à l'itération précédente, les améliorations apportées sont petites mais fréquentes et se concentrent principalement sur la résolution de problèmes spécifiques au projet. Elles sont rarement axées sur l'apprentissage inter-projets et la formalisation des améliorations au niveau de l'organisation.

Si l'on examine la manière dont l'amélioration du processus de test est organisée et gérée, il est probable que l'on se concentre moins sur un groupe gestionnaire des processus de test au niveau global et plus sur l'autogestion des équipes au sein du projet. Ces équipes doivent généralement modifier le processus de test utilisé au sein du projet pour répondre à leurs besoins, ce qui donne lieu à des processus très personnalisés. Cependant, certaines entreprises utilisent également des réunions hebdomadaires autour du processus pour amener les réflexions et améliorations à un niveau supérieur et transverse.

Étant donné que l'accent est davantage mis sur les spécificités projet de l'amélioration des processus (de test), l'importance accordée aux tests en général au sein de l'entreprise sera probablement moindre. Cela pourrait signifier, par exemple, que les problématiques de tests sont partiellement traitées lorsqu'elles dépassent ce contexte, centré sur le projet et l'équipe. Un exemple typique est l'approche adoptée pour tester certaines exigences de qualité, telles que la performance et la fiabilité. Ces questions peuvent être reportées d'une itération à l'autre parce que les compétences et ressources manquent à l'équipe projet. Dans ces domaines, il est difficile de passer à l'étape suivante sans investissements majeurs. Résoudre les problèmes uniquement au niveau de l'équipe pourrait aussi facilement conduire à une sous-optimisation et à une perte de vue de l'ensemble.

Dans un contexte Agile, l'éventail et le nombre d'idées alternatives d'améliorations à prendre en compte peuvent être sensiblement plus importants que ceux des modèles non Agiles. Comme tous

effectuent des tests dans le cadre du projet, ces idées peuvent provenir de n'importe quel membre de l'équipe. Cela met davantage l'accent sur l'évaluation et la priorisation des suggestions d'amélioration, qui peuvent être davantage un effort d'équipe qu'une tâche assignée à un responsable de l'amélioration du processus de test. Étant donné que ces alternatives peuvent nécessiter des connaissances spécifiques, un expert en processus de test, peut dans ce cas agir en tant que consultant, auprès de l'équipe, à sa demande.

Dans les projets utilisant une méthodologie Agile, ne vous attendez pas à trouver le même niveau de documentation de test que dans des projets utilisant un cycle de vie séquentiel. Il peut n'y avoir qu'un seul « document de test » combiné, couvrant les éléments essentiels d'une politique de test, d'une stratégie de test et même d'un plan de test de haut niveau. Les personnes chargées d'améliorer le processus de test doivent éviter de proposer des suggestions « d'amélioration » qui exigeraient une documentation de test plus rigoureuse et plus complète. L'un des grands principes du Manifeste Agile est que la documentation n'est créée que s'il existe un besoin clair et non ambigu. Non seulement la documentation des tests sera moins détaillée, mais il en sera de même pour la documentation des processus. Souvent, une stratégie appelée "formalisation de l'informel" est appliquée avec succès dans les environnements Agiles. Si quelque chose fonctionne bien, il est inutile de le changer pour le bien de TMMi. Cependant, la documentation peut être partagée et transmise au-delà de l'équipe, ce qui est souvent bénéfique. La stratégie de "formalisation de l'informel" signifie que, si un processus fonctionne, mais qu'il est en partie informel, il est possible de l'enseigner et de le documenter tel qu'il est exécuté. Le fait que le processus soit léger signifie que tous les scénarios d'utilisation possibles ne sont pas abordés dans sa description. Ces processus doivent donc être soutenus par un encadrement et une assistance sur le terrain, en particulier pendant la période de déploiement initial. Par conséquent, pour amener une entreprise Agile à la maturité des processus de test tout en maintenant cette culture, il faut plus - et non moins - de formation. De même, lors d'une évaluation, l'accent sera mis sur la collecte de preuves et sur la réalisation d'un plus grand nombre d'entretiens plutôt que sur l'étude d'artefacts.

Les méthodes utilisées pour proposer des améliorations des processus de test en Agile auront tendance à se concentrer sur les méthodes analytiques pour évaluer les causes profondes des problèmes, comme les diagrammes de cause à effet. Ces méthodes sont particulièrement utiles pour arriver à une attitude orientée solutions, ce qui est important en fin d'itération.

2. TMMi Niveau 2 : Discipliné

2.1. Domaine de processus Politique et Stratégie de test

L'objectif du domaine de processus Politique et stratégie de test est de développer et d'établir une politique de test, ainsi qu'une stratégie de test à l'échelle de l'entreprise ou du programme dans laquelle les activités de test, par exemple les types et les quadrants de test, sont définies sans ambiguïté. L'introduction d'indicateurs de performance permet de mesurer la performance et la valeur des activités de test. Ainsi, les axes d'amélioration peuvent être mis en lumière.

2.1.1. SG 1 Établir une politique de test

Toute entreprise qui se lance dans un projet d'amélioration des tests doit commencer par définir une politique de test qui établit les objectifs généraux de l'entreprise en matière de test, ses buts et sa stratégie concernant les tests et les professionnels du test. Il est important d'aligner la politique de test sur la politique qualité de l'organisation. Les améliorations des tests doivent être guidées par des objectifs métiers clairs, qui à leur tour doivent être documentés dans la politique (d'amélioration) de test. Toutes les parties prenantes au sein d'une entreprise ont besoin d'une politique de test pour atteindre une vision commune des tests et de leurs objectifs, nécessaire pour aligner les activités de test (amélioration des processus) dans toute l'entreprise. Notez que les objectifs de test ne doivent jamais être une fin en soi, ils sont dérivés de l'objectif de plus haut niveau visant à établir la qualité des logiciels et des produits.

Ce qui précède est également vrai dans une organisation qui pratique le développement logiciel Agile. En effet, au sein de nombreuses entreprises, beaucoup de discussions tournent autour de l'évolution du rôle des tests, de leur indépendance, de leur automatisation et des testeurs professionnels dans le développement logiciel Agile. Ces questions et d'autres sont généralement des sujets qui doivent être abordés lors d'une discussion avec la direction et d'autres parties prenantes et documentés dans une politique de test. Toute entreprise, y compris celles qui pratiquent l'Agilité, souhaitant lancer un projet d'amélioration des tests doit identifier et définir les moteurs et les besoins de l'entreprise à l'origine d'une telle initiative. Sinon, pourquoi lancer un projet d'amélioration ? En prenant le temps de cerner les véritables besoins de l'entreprise, on peut fournir un contexte permettant de décider où concentrer les priorités d'amélioration (de test), par exemple sur quel domaine de processus. Notez qu'une politique de test consiste généralement en un document d'une page, en une page web ou un affichage mural concernant toute l'organisation, et non un document au niveau du projet.

L'objectif spécifique de TMMi « Établir une politique de test », y compris ses pratiques spécifiques, est pleinement applicable aux entreprises qui développent leurs logiciels en Agile. Bien entendu, les éléments d'une politique de test peuvent également être intégrés dans une politique de développement. TMMi n'exige pas spécifiquement un document séparé. La politique d'une entreprise développant en Agile pourrait par exemple choisir *Scrum* comme son cadre de gestion, XP comme la principale méthode utilisée et l'adhésion aux valeurs Agile comme un principe essentiel. Un autre principe important qui pourrait être mentionné est que chaque membre de l'équipe est responsable de tout ; la qualité du produit est également une responsabilité collective.

Pendant, en Agilité, des considérations importantes sont à prendre en compte vis-à-vis de la politique de test et spécialement les objectifs définis (d'amélioration) des tests. Même si l'entreprise

s'est dotée d'objectifs généraux d'amélioration des processus de test, ils doivent être équilibrés avec les projets individuels et les équipes Agiles, responsables de l'amélioration de leur propre processus. Pour améliorer les processus Agile, le défi est de guider et encadrer l'amélioration au niveau de l'organisation sans réduire le sentiment d'appropriation de son processus par les équipes.

2.1.2. SG 2 Établir une stratégie de test

La stratégie de test suit la politique de test et sert de point de départ pour les activités de test au sein des projets. Elle est généralement définie à l'échelle de l'entreprise ou du programme et basée sur une évaluation de haut niveau des risques liés au produit et comprendra une description des types, des quadrants et des niveaux de test qui doivent être effectués, par exemple : test unitaire, d'acceptation, de non-régression et de performance. Il ne suffit pas d'indiquer, par exemple, qu'un test unitaire et un test d'acceptation seront effectués dans le cadre d'une itération. Nous devons définir ce que l'on entend par tests unitaires et tests d'acceptation, comment ils sont généralement effectués et quels sont leurs principaux objectifs. L'expérience montre que, lorsqu'une stratégie de test est définie et suivie, les risques de chevauchement entre les différentes activités de test sont moindres, ce qui permet d'obtenir un processus de test plus efficace. En outre, comme les objectifs et l'approche des différents types et niveaux de tests sont alignés, il est probable que la couverture sera meilleure, ce qui se traduira par un processus de test plus efficace et donc par un niveau de qualité plus élevé du produit. Pour établir cet alignement, il est fortement recommandé de mettre en place une stratégie de test globale pour chapeauter à la fois les tests Agiles et les tests non Agiles coexistant au sein d'un même projet. Cependant, si les projets Agiles et non Agiles sont séparés, les deux stratégies le sont aussi.

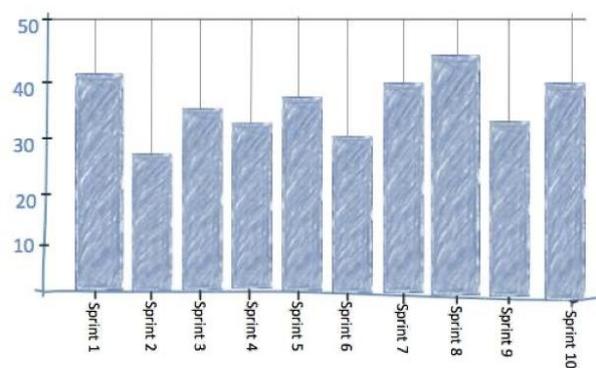
Une stratégie de test est un document essentiel dans un environnement Agile. Elle définit à un haut niveau les tests à effectuer dans les équipes Agiles (équipes d'itération) ; quels types, quadrants et niveaux de tests sont exécutés et à un haut niveau leur approche. Le document décrit comment les tests sont organisés, par exemple, quels tests ont lieu à l'intérieur des équipes Agiles et quels tests ont lieu à l'extérieur. Ce document décrit l'organisation du test, par exemple ce qui sera testé par les équipes agiles et ce qui sera testé en-dehors. Il définira les interactions entre équipes agiles et les tests en-dehors de leur périmètre, tels que les tests d'intégration matériel/logiciel, les tests d'intégration système ou les bêta-tests. Une stratégie de test garantit que tous ceux qui participent aux activités de test en comprennent une vue d'ensemble. Une bonne solution consiste à décrire la stratégie de test allégée dans un document de base qui donne une alternative aux plans de test (de projet) détaillés. La stratégie disponible, au niveau de l'organisation ou du programme, est ainsi discutée et confirmée pendant les planifications de versions (*Release Planning*). Sinon, une stratégie de test dérivée est créée spécialement pour le projet. Elle fournit un cadre de test couvrant toutes les itérations. L'objectif spécifique de TMMi "*Établir une stratégie de test*", y compris ses pratiques spécifiques, est pleinement applicable aux entreprises travaillant en Agile. Notez que les activités de communication, par exemple pour le document de stratégie de test, peuvent être moins pertinentes dans un cadre Agile. Les parties prenantes devraient déjà avoir échangé, lors de discussions antérieures dans le cadre d'une approche globale d'équipe. Cependant, une stratégie de test ne se fait pas au niveau de l'équipe, mais à un niveau plus élevé. Ainsi, un document d'entreprise ou de programme doit encore être communiqué aux équipes et même à d'autres parties prenantes, par exemple celles qui effectuent des activités de test en dehors de l'équipe dans le cas d'un modèle hybride.

2.1.3. SG 3 Établir des indicateurs de performance du test

Les objectifs métiers d'amélioration des tests, tels que définis dans la politique de test, doivent être traduits en un ensemble d'indicateurs clés de performance du test. La politique de test et les indicateurs de performance qui l'accompagnent fournissent une orientation claire et un moyen de communiquer les niveaux de performance du test attendus et atteints. Les indicateurs de performance doivent indiquer la valeur des tests et de l'amélioration des processus de test pour les parties prenantes. Étant donné que les investissements dans l'amélioration des processus nécessitent un soutien du management à long terme, il est crucial de mesurer quantitativement les avantages d'un programme d'amélioration afin de maintenir leur motivation. Attention, cet objectif spécifique de TMMi consiste à définir un nombre limité (par exemple, 2 ou 3) d'indicateurs de performance du test. Il s'agit plus de définir un ensemble d'indicateurs de base qui vous indiquent comment la valeur des tests évolue dans le temps et dans différents environnements de prestation que de mettre en place et d'appliquer un programme de mesure complet.

Dans un cadre Agile, l'accent sera davantage basé sur l'équipe et la réflexion systémique. Cela peut entraîner un élargissement correspondant des indicateurs à l'équipe et au système global plutôt que de les limiter aux seules spécificités du test lui-même. Les indicateurs au niveau 2 de TMMi sont également principalement liés aux résultats des itérations. Il s'agit par exemple des anomalies évitées, de la vélocité, du taux de satisfaction des clients, du ratio effort/gaspillage, du

pourcentage d'automatisation des tests, etc. Le défi consisterait à définir la combinaison appropriée d'indicateurs relatifs à l'approche en équipe et à la pensée systémique tout en donnant une bonne indication des résultats d'un programme d'amélioration des tests basé sur TMMi.



L'objectif spécifique de TMMi "Établir des indicateurs de performance du test", y compris ses pratiques spécifiques, est pleinement applicable, bien que les indicateurs de performance sélectionnés et appliqués puissent avoir une portée plus large et être plus étendus que leur lien aux seuls tests. Bien entendu, ce dernier point rendra l'analyse et l'interprétation des indicateurs de performance plus difficiles. En fait, les indicateurs de performance utilisés peuvent ne pas être appelés indicateurs de performance du test, mais plutôt indicateurs de performance d'équipe ou indicateurs de performance de système. Cela reste acceptable dans un contexte TMMi, tant que cet objectif comporte des éléments liés aux tests et qu'il est utilisé pour évaluer les progrès réalisés dans l'amélioration des tests.

2.2. Domaine de processus Planification du test

L'objectif de la planification du test est de définir une approche des tests basée sur les risques identifiés et la stratégie de test définie, d'établir et de maintenir des plans bien fondés pour effectuer et gérer les activités de test.

Attention : la clé d'une planification du test réussie réside dans la réflexion préalable (« l'activité »), et non dans la définition du plan de test associé (« le document »).

Pour les cycles de vie Agiles, les deux types de planification principales sont la planification des livraisons et la planification des itérations. Le processus de planification des tests au niveau 2 de TMMi se concentre sur les activités de test liées à ces planifications. Pour les livraisons, il s'agit d'anticiper la livraison d'un produit, au démarrage d'un projet. La planification des mises en production doit s'appuyer sur un *product backlog* défini et peut nécessiter un affinage pour transformer de grosses *user stories* en un ensemble de petites. Elle fournit la base d'une approche de test et d'un plan de test englobant toutes les itérations. Les plans de livraison sont plus généraux. Une fois la planification des mises en production terminée, celle de l'itération commence. La planification d'itération anticipe la fin d'une unique itération et porte sur le *backlog* d'itération.

Notez que le domaine de processus planification du test comporte un certain nombre de pratiques spécifiques. TMMi ne précise pas quand ni comment mener ces pratiques, ni ne précise que vous ne pouvez pas planifier de manière incrémentale. L'approche traditionnelle consiste à prendre le plus grand nombre de décisions possible dès le départ, pour pouvoir prendre en compte également les coûts et le calendrier correspondants. Cette approche se justifie par la nécessité de mieux estimer le travail et de réduire le risque de dérive. Les approches agiles partent généralement du principe que nous gagnons en valeur en affinant continuellement le plan sur la base des informations les plus récentes et d'une collaboration permanente avec le client.

2.2.1. SG 1 Effectuer une évaluation des risques

Des tests exhaustifs sont impossibles, et il faut toujours faire des choix et fixer des priorités. Cet objectif de TMMi est donc également applicable aux projets Agiles. Pour les projets Agiles, une évaluation des risques produit de haut niveau doit être effectuée sur la base d'un document décrivant la vision du produit ou à partir d'un ensemble de *user stories* de haut niveau au moment de la planification de la livraison. Pour chaque itération, lors de la session de planification, il faudrait aussi prendre le temps de réaliser une évaluation plus détaillée des risques produit (de cette itération). Le processus d'évaluation des risques sera beaucoup plus léger en Agile si on le compare à ceux qui sont appliqués dans des projets traditionnels suivant un modèle de cycle de vie séquentiel. Le *Risk Poker* [Van Veenendaal] est un exemple d'une technique légère d'évaluation des risques produits. Lors de la planification de la livraison, les parties prenantes connaissant les caractéristiques de la version fournissent un aperçu de haut niveau des fonctionnalités à développer, et toute l'équipe, y compris le(s) testeur(s), participe à l'identification et à l'évaluation des risques.

Lors de la planification de l'itération, l'équipe Agile identifie et analyse les risques produit en se basant sur les *user stories* à développer dans l'itération à venir. Il est préférable que tous les membres de l'équipe Agile, voire d'autres parties prenantes participent à la session sur les risques produit. Le résultat est une liste hiérarchisée des éléments de risque produit identifiant les zones critiques à tester, permettant ensuite de dimensionner l'effort de test à allouer afin de couvrir chaque risque avec suffisamment de tests, et de séquencer ces tests de manière à optimiser l'efficacité et l'efficience du travail de test à effectuer. Les tâches estimées sur le tableau des tâches peuvent être classées par ordre de priorité en partie en fonction du niveau de risque du produit qui leur est associé. Plus les risques sont élevés, plus grand sera l'effort de test impliqué et plus tôt devraient commencer les tâches associées. Plus les risques sont faibles, moins grand sera l'effort de test impliqué et plus tard pourraient commencer les tâches associées.

2.2.2. SG 2 Établir une approche test

Une approche de test est définie pour atténuer les risques produit identifiés et hiérarchisés. Pour une itération donnée, les éléments et fonctionnalités à tester sont identifiés lors de sa

planification. Cette activité est également basée sur le résultat de la session sur les risques produit. La liste priorisée des éléments à tester est généralement liée aux *user stories* à tester pendant l'itération. Les fonctionnalités sont généralement liées, entre autres, aux différentes composantes de qualité du logiciel à tester. De nouveaux risques produit peuvent apparaître au cours de l'itération, nécessitant des tests supplémentaires. Les questions en rapport avec ces nouveaux risques sont généralement vues lors des réunions quotidiennes.

L'approche de test définie au niveau de l'itération pour atténuer les risques peut couvrir par exemple un examen supplémentaire des *user stories* et des critères d'acceptation, un effort de test proportionnel au niveau de risque, la sélection de la ou des techniques de test appropriée(s) en fonction du niveau et du type de risque. L'approche de test au niveau de la livraison se situera à un niveau beaucoup plus élevé et sera basée sur une stratégie de test définie au niveau du programme ou de l'organisation. Souvent, une telle approche est rédigée ou affichée sur le wiki de l'équipe/du projet.

Le risque de régression apparaît toujours comme important dans les approches itératives. L'approche de test doit définir comment le gérer. En général, cela se fait en mettant en place un ensemble de tests de non-régression, de préférence automatisés. Dans ce contexte, la pyramide des tests [Cohn] est utile. Elle montre comment maximiser la valeur des tests de non-régression automatisés, depuis les tests unitaires à la base de la pyramide en passant par les tests par niveau de service jusqu'aux tests d'Interface Utilisateur qui trônent au sommet de cette pyramide. Les tests unitaires sont rapides et fiables. La couche de service permet de tester la logique métier au niveau API ou service sans toucher l'interface utilisateur (UI). Plus les tests sont hauts, plus ils sont longs et instables.

Les critères d'entrée, qui font normalement partie d'une approche de test définie (*pratique spécifique 2.3*), sont susceptibles de ne pas être pertinents pour le développement Agile. Dans le cadre du développement logiciel Agile, les tests font partie intégrante du processus de l'équipe et sont réalisés de façon quasi continue. Par conséquent, il n'est pas nécessaire de disposer d'une liste de contrôle ou d'une passerelle spécifique pour déterminer si les tests peuvent ou non être lancés. Cela s'applique également à un composant qui passe, au sein d'une équipe Agile, d'une étape de test (par exemple, les tests unitaires) à une autre (par exemple, les tests d'acceptation).

Le test des critères de sortie (*pratique spécifique 2.4*) fait partie de la « *Definition of Done* » (« Définition de Terminé ») (DoD). Il est important que la DoD dispose de critères spécifiques liés aux tests, par exemple pour la couverture des tests et la qualité des produits (anomalies). L'itération doit aboutir à la mise en œuvre de l'ensemble convenu de *user stories* et répondre aux critères de sortie (test) tels que définis dans la DoD. En général, les *stories* qui ne répondent pas aux critères de sortie sont mises en attente et peuvent être traitées lors de l'itération suivante. Bien entendu, dans un cadre Agile, un composant qui passe d'une étape de test à une autre n'a pas de critères de sortie. Non seulement une définition de « Done » existe au niveau de l'itération, mais aussi souvent au niveau de la livraison de plusieurs itérations. La DoD au niveau de la livraison contiendra encore une fois des critères liés à la couverture et à la qualité du produit.

La *Pratique spécifique 2.5 Définir les critères de suspension et de reprise* est susceptible de ne pas être pertinente pour les cycles de vie Agiles. Comme les tests font partie intégrante du processus de développement logiciel en Agile, ils ne seront bien sûr pas traités comme une activité séparée et indépendante des autres activités d'itération. En cas d'anomalies bloquantes qui pourraient potentiellement ou réellement menacer la progression des tests, celles-ci sont discutées lors de la réunion quotidienne, moment où l'équipe décidera des actions à entreprendre, le cas échéant, pour

résoudre les problèmes. Ainsi, des critères formels de suspension et de reprise ne seront pas nécessaires et définis, les questions qui s'y rapportent sont traitées dans le cadre du fonctionnement habituel agile qui sert donc de pratique alternative pour cette pratique spécifique.

2.2.3. SG 3 Établir des estimations de test

Les équipes agiles estimeront les tests de manière détaillée pendant la planification des itérations. Les estimations des tests de haut niveau sont faites pendant la planification des livraisons et éventuellement aussi lors des sessions d'affinage du *backlog*. Toutes les estimations sont bien sûr réalisées en équipe et comprennent tous les efforts nécessaires pour livrer chaque *story*. Il est important de s'assurer que les activités de test sont bien prises en compte lors des sessions d'estimation. Cela peut se faire en identifiant les activités de test comme des tâches séparées et par conséquent en estimant chaque tâche individuellement ou - en estimant les *user stories* dans chacune desquelles les tests à réaliser apparaissent de manière explicite - permettant de les prendre en compte. Un testeur, faisant partie de l'équipe Agile, doit participer aux sessions d'estimation. Le *planning poker* ou la *taille de T-Shirt* sont des techniques d'estimation typiques utilisées dans le cadre du développement Agile de logiciels.



Le travail à effectuer pour la prochaine itération est généralement défini dans les *user stories* qui doivent être suffisamment petites (*small*) pour pouvoir être estimées. L'estimation est l'un des critères définis par INVEST [Wake] et s'applique aux *user stories* qui doivent faire partie d'une itération. Lors de la planification des itérations, les équipes agiles identifient et définissent généralement les tâches liées aux tests. Les tâches de test seront reportées sur un tableau de tâches en même temps que celles concernant le développement. Les deux sont utilisées afin de réaliser les

estimations pour l'itération. Cet ensemble de tâches définies sert à structurer la répartition du travail (un peu à la manière des projets séquentiels).

Lors de la planification de la livraison, les *user stories* ou les *epics* seront généralement définies à un niveau plus élevé et ne seront pas encore décomposées en tâches spécifiques. Les estimations en seront bien sûr plus difficiles et moins précises. Comme indiqué, les projets agiles n'utiliseront pas de structure de répartition du travail comme base d'estimation, mais pourront bénéficier, au niveau de la mise en production, d'un simple diagramme qui visualise le produit à développer et permet ainsi d'évaluer correctement le travail.

Bien que les estimations dans les équipes Agiles soient relativement informelles, la logique qui sous-tend les estimations (les facteurs pris en compte) doit être claire. Les échanges basés sur des raisonnements logiques favorisent une plus grande précision des estimations. Généralement, dans les projets agiles, l'estimation est axée sur la taille (en utilisant des jours/homme idéaux comme unité d'estimation) ou l'effort (en utilisant des points de complexité). Les coûts ne sont normalement pas abordés dans le cadre des sessions d'estimation en Agile.

Cet objectif spécifique de TMMi et ses pratiques spécifiques sont donc pleinement applicables, à l'exception de la *pratique spécifique 3.2 Définir le cycle de vie des tests*. L'une des bases du développement logiciel itératif et Agile est de découper le travail. Par conséquent, les tâches identifiées sont généralement suffisamment détaillées pour servir de base à l'estimation (de test). Il

n'est donc pas nécessaire, en Agile, de définir également un cycle de vie pour les activités de tests qui servirait de base supplémentaire pour l'estimation.

2.2.4. SG 4 Développer un plan de test

Il faut le répéter : la planification du test consiste à réfléchir en amont (« l'activité ») et non à définir le plan de test associé (« le document »). Dans un cadre Agile, la plupart des activités de planification du test, telles que définies par cet objectif spécifique de TMMi, seront effectuées lors de la planification des livraisons et des itérations. Toutefois, le résultat de ces activités ne sera généralement pas documenté dans un plan de test, en particulier pour la planification des itérations où il pourrait être reflété sur le tableau des tâches.

Comme les tests font partie intégrante de la planification des versions et de l'itération, les « calendriers » qui en résultent comprendront également des activités de test. Plutôt qu'un calendrier détaillé tel que celui élaboré avec des cycles de vie séquentiels, le calendrier d'un projet Agile ressemble beaucoup plus à un ordonnancement des *user stories* (éléments du *backlog*) et des tâches qui reflète les priorités de la livraison et de l'itération en fonction, par exemple, de la valeur métier souhaitée. Les cartes mentales sont souvent utilisées ici comme technique de soutien. Le tableau des tâches reflétera les priorités d'itération. Ainsi, aucun calendrier explicite (de test) n'est établi ; il est prévu de définir des priorités claires de livraison et d'itération pour les *user stories*, respectivement les tâches à effectuer, y compris les tâches de test.

Au niveau du projet, la constitution de l'équipe intègre des ingénieurs de test ; le besoin de ressources de test ou de ressources polyvalentes est identifié d'emblée.

Lors de l'évolution ou du développement du projet, il est facile d'oublier pourquoi une personne a été, à l'origine, sélectionnée pour le projet / l'équipe. Souvent, les raisons incluent le besoin de compétences ou une expérience spécifique liée au projet ou à l'équipe.

Cela justifie amplement de tracer ces besoins, en sauvegardant l'information liée à la raison de leur sélection dans le projet / l'équipe. Une fois les équipes agiles définies, le nombre de personnes intervenant en test évolue peu. Pendant la planification de l'itération, l'équipe peut, en cas de besoin, débattre (de l'ajout) des ressources et compétences nécessaires pour réaliser les opérations de test pendant l'itération (par exemple les tests non-fonctionnels), dans le but de garantir que l'équipe, les ressources, compétences et le savoir-faire nécessaires pour réaliser les tests requis sont disponibles. Le *Scrum Master* doit s'assurer que le *Product Owner* apporte sa contribution aux tests selon les besoins, par exemple en répondant aux questions et en discutant des *user stories*. Le *Product Owner* doit être suffisamment disponible, ce qui doit à nouveau être organisé en amont.

Une session initiale sur les risques du projet devrait faire partie de la planification de la livraison et de l'itération. La planification des livraisons et des itérations devrait commencer par une session sur les risques projet. L'identification (et la gestion) des risques ultérieurs du projet au cours de l'itération se fait au cours des réunions quotidiennes. Elle est généralement documentée dans le journal des obstacles. Il est important que les problèmes liés aux tests y soient également notés. Les obstacles doivent être discutés lors de la réunion quotidienne, jusqu'à ce qu'ils soient résolus.

Bien qu'aucun plan de test spécifique et détaillé ne soit développé et documenté, la *pratique spécifique 4.5 Établir le plan de test* est toujours pertinente dans un contexte Agile. Les résultats des discussions qui ont lieu dans le contexte de la planification des tests sont cependant susceptibles d'être saisis sous une forme légère, éventuellement une carte mentale.

2.2.5. SG 5 Obtenir un engagement sur le plan de test

En Agile, le processus de création et développement d'une approche et d'un plan de test est un exercice d'équipe, éventuellement guidé par un professionnel du test (en tant que membre de l'équipe). La qualité du produit est une responsabilité de l'équipe. En tant que tel, à partir du moment où l'équipe suit les bons processus, l'engagement sur l'approche et le plan (de test) est déjà un résultat implicite des planifications de livraison et d'itération étant donné que c'est un travail d'équipe. La différence est bien sûr énorme avec la façon de travailler dans un environnement traditionnel où le testeur prépare son plan, et, ensuite, doit obtenir un engagement explicite. Dans les projets Agiles, l'équipe agile (y compris le *Product Owner*) doit comprendre et accepter la liste des risques produit prioritaires et les actions d'atténuation des tests à effectuer. La compréhension et l'engagement peuvent par exemple être obtenus au moyen d'une brève présentation suivie d'un échange lors de la planification des itérations ou des versions expliquant à l'équipe les risques du produit, l'approche de test et sa justification.

Pendant la session d'estimation (voir *SG 3 Établir des estimations de test*), la charge de travail est estimée. Les ressources (de test) pour une équipe sont fixées par la configuration de l'équipe agile. Les *user stories* estimées et sélectionnées pour être développées prennent comme point de départ les ressources disponibles (contrainte). Ainsi, une fois de plus, la réconciliation des niveaux de travail et de ressources n'est pas une activité significative. La pratique spécifique *5.2 Concilier les niveaux de travail et de ressources* n'est donc généralement pas pertinente dans un contexte Agile. Des réunions quotidiennes (parfois « debout ») seront utilisées pour traiter tout problème de ressources pendant l'itération et pour (ré)allouer immédiatement les ressources appropriées, ou pour retirer un livrable d'une itération avant de l'intégrer dans la planification d'une itération ou d'une livraison future.

2.3. Domaine de processus 2.3 Surveillance et contrôle du test

L'objectif de la surveillance et du contrôle du test est de fournir une compréhension de l'avancement des tests et de la qualité des produits afin que des actions correctives appropriées puissent être prises lorsque l'avancement des tests s'écarte considérablement du plan et que la qualité des produits s'écarte considérablement des attentes.

Selon le manifeste Agile et les principes qui l'accompagnent, certaines choses sont fondamentalement différentes du suivi et du contrôle dans les projets Agiles par rapport aux projets traditionnels. Bien que le suivi et le contrôle soient des éléments essentiels d'un projet Agile, cela n'implique pas que l'objectif soit de s'en tenir à un plan rigide, en fait, c'est le contraire qui est vrai, car le manifeste et les principes parlent tous deux d'accueillir le changement. Le suivi et le contrôle des tests pourraient, dans un contexte Agile, être interprétés comme fournissant les meilleures pratiques pour ajuster le plan en continu afin de le maintenir à jour, ce que recommandent les approches Agiles.

Du point de vue de la surveillance et du contrôle du test, cela signifie que nous ne sommes pas guidés par un plan, mais que nous examinons constamment nos progrès et les résultats des tests, et que nous adaptons notre plan et notre approche le cas échéant - de nouveaux risques produit peuvent se révéler. Le plan de test, quel que soit son mode de saisie, est une entité vivante et doit être constamment révisé et mis à jour au fur et à mesure que de nouvelles informations sont disponibles ou que des retours sont donnés. Les projets agiles doivent également garder à l'esprit la

« vue d'ensemble » et assurer le suivi et le contrôle au niveau des itérations mais aussi au niveau « supérieur » des livraisons.

Il est important de noter que, comme le test est un processus qui est pleinement intégré dans le processus global de l'équipe Agile, la surveillance et le contrôle du test font également partie intégrante des mécanismes de surveillance et de contrôle de l'équipe Agile. Par conséquent, les testeurs rendent compte à l'équipe mais pas à un responsable des tests comme dans le cas des projets traditionnels.

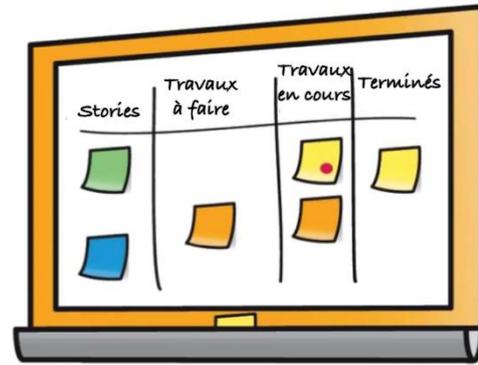
Comme le processus de planification des tests au niveau 2 de TMMi se concentre sur la planification des livraisons et des itérations, il en va de même pour la surveillance et le contrôle du test. Il est prévu que la surveillance et le contrôle du test soient effectués à la fois au niveau de la planification et de la mise en production.

2.3.1. SG 1 Surveiller l'avancement du test par rapport au plan

Les testeurs des équipes agiles utilisent diverses méthodes pour surveiller et enregistrer la progression des tests, par exemple, la progression des tâches de test et des *stories* sur le tableau des tâches Agile, et les *burndown charts*. Ces informations peuvent ensuite être communiquées au reste de l'équipe via des supports tels que les tableaux de bord publiés sur le wiki ou envoyés par courriel, ainsi qu'oralement lors des réunions quotidiennes. Les équipes peuvent utiliser des *burndown charts* pour suivre l'avancement des travaux pour la version et pour l'itération. Un tel graphique sera utilisé par exemple pour comparer la progression à la vélocité attendue et à la liste des fonctions attendues (performance de l'équipe). Parfois, quand les ressources en environnements de test sont rares et vitales, par exemple pour les tests non-fonctionnels, des tableaux spécifiques sont utilisés pour surveiller l'usage des ressources d'environnements de tests par rapport à ce qui a été convenu lors de la planification de l'itération. Une autre pratique répandue consiste à suivre les problèmes d'environnements de test, par exemple avec des autocollants « environnement de test bloqué » posés sur les cartes de *user stories* ou en créant une colonne séparée sur le tableau où sont mises en attente toutes les stories bloquées le temps de les débloquent. Il s'agit de créer une visibilité sur l'impact d'un environnement de test qui bloque la progression.

Pour fournir une représentation visuelle instantanée et détaillée de l'état actuel de l'ensemble de l'équipe, y compris l'état des tests, les équipes peuvent utiliser des tableaux de tâches Agile. Les cartes des *stories*, les tâches de développement, les tâches de test et les autres tâches créées pendant la planification des itérations sont sur le tableau des tâches, leur type est souvent indiqué à l'aide de fiches de couleurs assorties. Au cours de l'itération, la progression est gérée par le déplacement de ces tâches sur les colonnes du tableau des tâches intitulées par exemple : « à faire », « en cours (de réalisation) » et « terminé ». Les équipes agiles peuvent maintenir leurs cartes de *user stories* grâce à des outils qui permettent d'automatiser la mise à jour de l'affichage des tableaux et statuts. Cependant, certaines équipes ne créent pas de tâches spécifiques pour les activités individuelles mais peuvent simplement utiliser la fiche de synthèse et y annoter des commentaires pour les tests, en faisant référence aux outils Agiles ou aux wikis où les tests peuvent être documentés. Les tâches de test sur la carte des tâches sont généralement liées aux critères d'acceptation définis pour les *user stories*. Lorsque les scripts d'automatisation des tests, les tests manuels et les tests exploratoires d'une tâche de test atteignent le statut de réussite, la tâche est déplacée dans la colonne « terminé » du tableau des tâches.

La *Definition of Done* (DoD) sert de critère de sortie qui permet de mesurer les progrès. La DoD doit également être liée aux activités de test et montre tous les critères qui doivent être satisfaits avant que le test d'une *user story* puisse être considéré comme « Terminé ». Notez que les critères de test liés aux tâches de test ne constituent qu'une partie de ce que l'équipe intègre dans la *Definition of Done*. Les critères de la DoD sont généralement appliqués à plusieurs niveaux, par exemple, au niveau de l'itération et au niveau de la livraison. Toute l'équipe examine régulièrement l'état du tableau des tâches, souvent lors des réunions quotidiennes, afin de s'assurer que le rythme de déplacement des cartes est acceptable. Si certaines tâches (y compris les tâches de test) ne progressent pas ou progressent trop lentement, cela déclenche une discussion en équipe pour analyser les problèmes qui peuvent bloquer la progression de ces tâches.



La réunion quotidienne comprend tous les membres de l'équipe agile, y compris les testeurs. Lors de cette réunion, ils communiquent leur état d'avancement et leurs progrès réels au reste de l'équipe. Tout problème susceptible de bloquer la progression des tests est communiqué lors de ces réunions, afin que toute l'équipe soit consciente des problèmes et puisse agir en conséquence, ce qui permet d'y intégrer aussi la gestion des risques. Tous les risques liés au projet, y compris ceux liés aux tests, par exemple le manque de disponibilité des environnements de test, peuvent être communiqués et traités lors des réunions quotidiennes. (Remarque : le suivi des risques liés aux produits fait partie du suivi de la qualité des produits et est donc abordé ci-après dans l'objectif spécifique SG 2 Surveiller la qualité du produit par rapport au plan et aux attentes). Des réunions (*standup meetings*) et une gestion des tâches quotidiennes ainsi que les pratiques de l'équipe agile liées à la correction de l'avancement (des tâches) sont de bonnes techniques éprouvées qui répondent à l'intention des pratiques spécifiques de TMMi dans le domaine de processus de surveillance et de contrôle du test.

Le jalon pour l'inspection, dans un cadre Agile est à la fin de l'itération. Les réalisations des tests, par exemple par rapport à la *Definition of Done*, feront partie de l'examen de l'itération. Des démonstrations sont organisées avec les parties prenantes pour discuter de la valeur métier et de la qualité du produit livré.

Les parties prenantes sont représentées par le *Product Owner* dans la planification des itérations, les revues d'itération (demos) et les rétrospectives. Il est impliqué dans les discussions sur le *Product Backlog* et apportera ses retours et sa contribution à l'élaboration des *user stories* et à la conception des tests tout au long de l'itération. D'autres parties prenantes sont impliquées à la fin de chaque itération lors de la revue d'itération. Aucun suivi spécifique de l'implication des parties prenantes n'est nécessaire car la représentation des parties prenantes par un *Product Owner* est intégrée dans la façon de travailler en Agile. Il est à noter que son implication active dans les projets Agiles est un facteur de succès crucial mais est parfois assez difficile à mettre en pratique. Si c'est le cas, cela sera remonté et discuté lors de la réunion quotidienne et géré comme un risque projet (voir ci-dessus) qui affectera l'efficacité et l'efficience de toute l'équipe.

2.3.2. SG 2 Surveiller la qualité du produit par rapport au plan et aux attentes

Pour le contrôle de la qualité des produits, on utilise en grande partie les mêmes mécanismes que pour le contrôle de l'avancement (voir SG1 ci-dessus). En Agile, pour le suivi des risques liés au produit, l'accent est mis sur l'examen de la liste des risques liés au produit lors de réunions régulières plutôt que sur l'examen de toute documentation détaillée des risques. Les risques produit nouvellement identifiés ou les risques produit modifiés, par exemple à la suite de tests exploratoires, seront discutés et il faudra convenir des mesures de test requises. L'état des différents risques produit est généralement indiqué au moyen de graphiques sur le tableau de bord.

La meilleure pratique consiste à ne considérer aucune fonction comme étant terminée tant qu'elle n'a pas été intégrée dans le système et testée avec succès. Dans certains cas, pour résoudre les anomalies persistantes et d'autres formes de dette technique, des itérations de consolidation ou de stabilisation périodiques sont mises en place. Les métriques utilisées par les équipes agiles pour contrôler et améliorer la qualité du produit sont sensiblement les mêmes que dans les méthodes de développement traditionnelles. Ainsi, on y retrouve par exemple : les taux de réussite et d'échec des tests, les taux de découverte des anomalies, les anomalies trouvées et corrigées. Le nombre d'anomalies découvertes et résolues au cours d'une itération ainsi que le nombre d'anomalies non résolues qui pourraient faire partie du *backlog* de la prochaine itération doivent être contrôlés au cours des réunions quotidiennes de synchronisation (*standup meetings*). Pour contrôler et améliorer la qualité globale du produit, de nombreuses équipes Agiles utilisent également des enquêtes de satisfaction pour savoir si le produit répond aux attentes des clients.

Les critères de sortie des tests, par exemple pour la couverture des tests et la qualité des produits (anomalies), font partie de la *Definition of Done* (DoD). Le respect des critères de sortie convenus est généralement contrôlé via le tableau des tâches, ce qui signifie qu'un article ne peut être considéré comme « terminé » que s'il est conforme aux critères de la DoD.

La réunion quotidienne (*standup meeting*) est le mécanisme utilisé pour effectuer de façon quasi continue des examens de la qualité des produits. En Agile, la fin d'une itération est le jalon de revue de la qualité du produit réalisé. Des démonstrations sont organisées avec les parties prenantes pour discuter de la valeur métier et de la qualité du produit livré, qui est vérifiée et validée par rapport aux critères de qualité définis par la DoD.

Après le domaine de processus Planification du test, les pratiques spécifiques de contrôle des critères d'entrée, de suspension et de reprise sont susceptibles de ne pas être pertinentes. Pour plus d'informations, veuillez-vous référer à l'objectif spécifique *SG 2 Établir une approche de test du domaine de processus Planification du test*, où une explication a été donnée sur les raisons pour lesquelles ces critères sont généralement susceptibles de ne pas être pertinents dans un environnement Agile.

2.3.3. SG 3 Gérer les actions correctives jusqu'à leur terme

Les équipes Agiles remarqueraient très rapidement des problèmes tels que des écarts par rapport aux attentes sur le *burndown chart* et/ou un manque de progression des tâches (de test) et des *stories* sur le tableau des tâches Agiles. Ces problèmes, et d'autres, par exemple les problèmes bloquant la progression des tests, sont communiqués lors des réunions quotidiennes de synchronisation (voir ci-dessus). Ainsi, toute l'équipe en est consciente. Cela permet de déclencher une discussion au sein de l'équipe pendant laquelle les problèmes sont analysés. Le résultat pourrait

être que la base de référence a besoin d'une mise à jour (par exemple, la suppression d'une ou plusieurs *user stories* du *backlog* d'itération), que la *Definition of Done* est peut-être trop stricte ou que des changements devraient être apportés à la manière de travailler. L'équipe décidera des mesures correctives à prendre en collaboration avec le *Product Owner*. Dans les cas où le *backlog* d'itération est modifié, le *Product Owner* présente un *backlog* d'itération mis à jour à l'équipe et les éléments supprimés du *backlog* d'itération peuvent être reportés à une itération suivante et discutés lors de la planification de l'itération.

La gestion des actions correctives dans les projets Agiles est principalement une responsabilité de l'équipe auto-organisée. L'équipe peut définir et mettre en œuvre des actions correctives appropriées, ou soumettre tout problème au *Scrum Master* en tant qu'« obstacle ». Les responsabilités du *Scrum Master* incluent généralement d'aider l'équipe à gérer les problèmes jusqu'à leur résolution. D'habitude, les actions correctives sont évoquées dans les réunions quotidiennes ou les séances de rétrospective. Les actions correctrices sur lesquelles l'équipe s'est accordée, peuvent être suivies jusqu'à leur fermeture en tant que « tâches » ou « éléments de *backlog* » à travers le *Product Backlog* ou (le temps de l'itération) grâce au tableau des tâches.

2.4. Domaine de processus 2.4 Conception et exécution des tests

L'objectif de la conception et de l'exécution des tests est d'améliorer la capacité du processus de test pendant leur conception et leur exécution en établissant une spécification de conception, en utilisant des techniques de conception, en exécutant un processus d'exécution des tests structuré et en gérant les incidents de test jusqu'à leur clôture.

Bien que l'objectif sous-jacent (« atténuer les risques et tester le logiciel ») soit le même pour un projet Agile et un projet séquentiel traditionnel, l'approche adoptée sur la manière de tester est généralement très différente. Dans un projet Agile, la flexibilité et la capacité à réagir au changement sont des points de départ importants. En outre, l'analyse et la conception des tests, la mise en œuvre et l'exécution des tests ne sont pas reportées dans des phases ultérieures, mais sont plutôt effectuées en parallèle, en se chevauchant et de manière itérative. Le niveau de détail de la documentation des tests établie est une autre différence essentielle. Généralement, les techniques basées sur l'expérience et les anomalies sont plus souvent utilisées dans les projets Agiles, bien que les techniques de conception de tests basées sur les spécifications puissent également être applicables et utilisées. En mettant davantage l'accent sur les tests unitaires et d'intégration, les techniques boîte blanche telles que les diagrammes d'état et de décision sont également beaucoup plus répandues. Une dernière différence majeure est le niveau du risque de régression, qui nécessite davantage de tests de régression aux différents niveaux de test. Dans l'idéal, les tests de régression sont hautement automatisés. Il existe de nombreuses différences, mais en fin de compte, dans le contexte du processus de conception et d'exécution des tests, il s'agit toujours de créer des tests, d'atténuer les risques produit, d'exécuter des tests et de trouver des anomalies.

2.4.1. SG 1 Analyser et concevoir les tests en utilisant des techniques de conception de tests

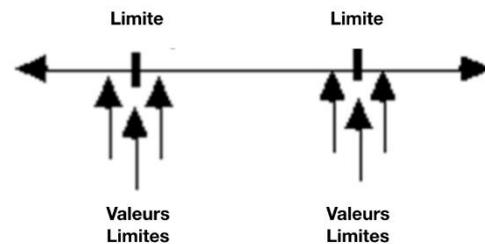
En Agile, les activités d'analyse, de conception et d'exécution des tests se soutiennent mutuellement et se déroulent généralement en parallèle tout au long d'une itération. Dans les projets à cycle de vie séquentiel, l'analyse des tests est effectuée par les testeurs qui examinent la base de test, par exemple les exigences, et en évaluent la testabilité une fois qu'elle a été créée. Dans les projets agiles, les testeurs font partie d'une équipe qui crée et affine les *user stories* de manière

collaborative. De fréquentes révisions informelles sont effectuées pendant le développement des exigences, y compris les critères d'acceptation pour chaque *user story*. Les métiers, les développeurs et les testeurs collaborent pour définir ces critères. Généralement, la perspective unique du testeur améliorera la *user story* en identifiant et en rendant testables les détails manquants. Un testeur peut apporter sa contribution en posant aux représentants métier des questions ouvertes sur la *user story* et ses critères d'acceptation et en proposant des moyens de la tester. L'analyse des tests n'est donc pas une activité distincte explicite, mais plutôt une activité implicite que les testeurs effectuent dans le cadre de leur rôle dans le développement collaboratif de la *user story*.

Sur la base de l'analyse des *user stories*, des conditions de test² sont identifiées. Dans une perspective de test, la base de test est analysée afin de voir ce qui pourrait être testé - ce sont les conditions de test. Ces conditions, (parfois appelées situations de test) sont essentiellement une identification de « choses » qui doivent être testées/couvertes [Black, Van Veenendaal]. À condition que les critères d'acceptation définis soient

suffisamment détaillés et clairs ils pourraient reprendre le rôle des traditionnelles conditions de test. Les conditions de test sont ensuite traduites en tests, nécessaires pour assurer la couverture des critères d'acceptation définis et convenus. Il est également souvent utile d'effectuer des analyses de test à un niveau supérieur plutôt que de se contenter de simples *user stories*. Par exemple, analyser une *feature*, une *epic* ou un ensemble de *user stories* permet d'identifier des conditions de test plus abstraites que celles des *user stories* et qui couvrent également plusieurs *user stories*. Les techniques de conception de test basées sur des spécifications sont généralement utiles pour dériver les conditions de test à partir des *user stories* et des critères d'acceptation. Cependant, dans un contexte Agile, ces techniques de conception de test sont le plus souvent utilisées de manière plus implicite qu'explicite, les testeurs ayant maîtrisé les techniques sur la base de leur expérience, sont capables de les utiliser avec souplesse dans leur contexte. Les conditions de test seront documentées dans un format léger, contrairement à la manière plus traditionnelle de travailler où elles sont documentées dans le cadre d'un document de spécification de conception de test. Parfois, en particulier dans les projets où les tests exploratoires sont largement utilisés, les conditions de test sont simplement identifiées au moyen d'un brainstorming et sont documentées en tant qu'idées de test (à intégrer dans les chartes de test). La définition des conditions de test sert de base pour établir une traçabilité horizontale (voir ci-après) et gérer la couverture de l'ensemble de tests de non-régression (automatisés), dans le cadre desquels d'autres tests automatisés sont développés pour couvrir des conditions de test spécifiques.

Le principe du « tester d'abord » (test first) étant appliqué en Agile, les tests qui couvrent l'ensemble des conditions de test seront identifiés (et éventuellement automatisés) avant le



² Le terme "condition de test" est utilisé dans ce paragraphe sur la base des pratiques TMMi, bien que ce ne soit pas un terme Agile courant. TMMi définit une condition de test comme "Un élément ou un événement d'un composant ou d'un système qui pourrait être vérifié par un ou plusieurs cas de test, par exemple une fonction, une transaction, une caractéristique, un attribut de qualité ou un élément structurel". Il n'est pas question ici d'imposer une couche supplémentaire de documentation de test à une équipe Agile. Toutefois, ce qui est important, c'est que l'équipe comprenne clairement quels éléments (ou événements) doivent être testés. Dans certains projets, de nombreux tests existent sans que l'on comprenne ce qu'ils couvrent réellement et ce qu'ils doivent couvrir. Dans un contexte Agile, l'objectif des pratiques TMMi sur les conditions de test est généralement atteint par des pratiques alternatives telles que les critères d'acceptation (à condition qu'ils soient spécifiés au niveau de détail requis) et les chartes de test dans les tests exploratoires.

développement du code ou au moins en parallèle. Pour les tests unitaires automatisés, une approche telle que le développement piloté par les tests (*Test Driven Development*, TDD) peut être envisagée. Pour les niveaux de test plus élevés, le développement piloté par le comportement (*Behavior-Driven Development*, BDD) et le développement piloté par les tests d'acceptation (*Acceptance Test-Driven Development*, ATDD) sont des approches de développement Agiles appréciées qui prennent également en charge les tests. Le BDD et l'ATDD sont également très liés à l'automatisation des tests.

La plupart des tests manuels, seront identifiés/affinés au fur et à mesure que l'équipe progresse dans leur exécution. Ils sont rarement documentés de manière aussi détaillée que dans les projets traditionnels, mais plutôt sous la forme d'idées de tests utilisant des tests exploratoires. Pour les domaines complexes et critiques, une approche plus traditionnelle de la conception des tests et du développement de cas de test utilisant des techniques de conception de tests formels peut être la meilleure façon de couvrir les risques. Toutefois, cette approche permet également de limiter la quantité de documentation par rapport aux tests effectués dans un environnement traditionnel de cycle de vie séquentiel. La priorisation des tests suit la priorisation de la *user story* qu'ils couvrent. La priorité des *user stories* est basée sur la valeur métier ; plus elle est élevée, plus la priorité le sera. Les tests doivent (c'est important) être établis de façon à couvrir non seulement les risques fonctionnels et non fonctionnels mais aussi et surtout en Agile spécifiquement couvrir le risque de régression

Les données de test spécifiques nécessaires pour soutenir les conditions de test et l'exécution des tests sont identifiées. Toutefois, dans les environnements agiles, contrairement aux environnements traditionnels, les données de test nécessaires ne sont généralement pas spécifiées au préalable dans un document de spécification de test. Au contraire, si les outils et/ou les fonctionnalités nécessaires sont disponibles, les données de test sont immédiatement créées pour permettre un démarrage presque immédiat de l'exécution des tests manuels. Cependant, pour les tests automatisés, les données devront généralement être spécifiées au préalable.

La traçabilité entre les exigences, les conditions de test et les tests doit être établie et maintenue. Les équipes doivent indiquer clairement qu'elles ont couvert les différentes *user stories* et les critères d'acceptation dans le cadre de leurs tests. Alors qu'un outil formel de gestion des exigences n'est peut-être pas nécessaire, l'équipe a besoin de soutien pour organiser et gérer ses exigences de façon à pouvoir attribuer des identifiants à chaque critère. Ces identifiants peuvent être utilisés pour veiller à ce que l'achèvement des tests respecte les critères convenus. Dans beaucoup d'organisations Agiles, les *user stories* servent de base pour développer un ensemble de critères d'acceptation et par là-même, de tests liés. Une fois ces tests mis en place, ils deviennent eux-mêmes les exigences détaillées convenues. L'utilisation de cette approche peut suffire pour atteindre l'objectif de la pratique spécifique de TMMi sur la gestion des exigences liées à la traçabilité horizontale.

2.4.2. SG 2 Implémenter les tests

L'implémentation des tests consiste à mettre en place les prérequis nécessaires pour commencer l'exécution des tests. Généralement, le développement de la documentation des tests, par exemple les procédures de test, pour soutenir l'exécution des tests est réduit au minimum. Des scripts de test (de régression), plutôt automatisés, sont développés et priorisés. La préparation de tests de régression démarre aussi dès que possible, en parallèle des autres activités de test.

Dans un environnement Agile, il est rare de détailler les procédures de test. En travaillant au sein d'une équipe compétente, les tests seront très probablement documentés à un niveau d'abstraction beaucoup plus élevé. Cela suffira à ceux qui exécutent les tests, puisqu'on s'attend à ce qu'ils aient le niveau requis de connaissances techniques et métier pour effectuer les tests. Ceux qui

exécutent les tests travaillent au sein de l'équipe, de sorte qu'ils auront une meilleure compréhension de ce qui est codé et comment, et de la manière dont la fonctionnalité est adaptée à l'objectif. Étant donné que le niveau de changement au sein des itérations et entre celles-ci est généralement élevé, l'élaboration d'une procédure de test détaillée entraînerait également un travail de maintenance important. Généralement, une grande partie des tests est effectuée sous la forme de tests exploratoires. Dans les tests exploratoires, aucune procédure de test détaillée n'est élaborée, mais plutôt des chartes de test de haut niveau d'une page, décrivant les idées de test et les conseils à l'intention du testeur. Cependant, les données spécifiques nécessaires à la réalisation des tests doivent bien sûr être créées au préalable.

De nombreuses équipes Agile utilisent des tests automatisés. Les approches typiques utilisées sont le développement piloté par les tests (*Test-Driven Development*, TDD), le développement piloté par le comportement (*Behavior-Driven Development*, BDD) et le développement piloté par les tests d'acceptation (*Acceptance Test-Driven Development*, ATDD). En utilisant une ou plusieurs de ces approches, des scripts de test automatisés seront créés dans le cadre de l'activité de mise en œuvre des tests.

L'intégration continue est une pratique clé pour les projets Agiles. Le test est une activité parallèle et totalement intégrée qui doit être réalisée par l'équipe. Cette activité ou phase n'est ni indépendante, ni séparée. La *pratique spécifique 2.3 Spécifier la procédure de test d'admission* devient donc sans objet au sein d'une équipe Agile. Il n'est pas nécessaire et, en fait, il n'est pas logique de spécifier et d'effectuer un test d'admission formel et spécifique. Toutefois, dans le cas où certains tests, par exemple, le test d'intégration du système ou le test d'intégration matériel/logiciel, sont effectués en dehors du champ d'intervention de l'équipe agile, ils peuvent spécifier un test d'admission pour évaluer la qualité de la livraison de l'équipe agile et déterminer si le produit qui a été livré est prêt pour leur test.

Dans les environnements agiles, aucun calendrier spécifique d'exécution des tests ne sera créé et, par conséquent, la *pratique spécifique 2.4 Développer un calendrier d'exécution des tests* ne sera probablement pas pertinente. L'ordre dans lequel le test sera exécuté au cours d'une itération est très souple, bien qu'il soit guidé par la priorité des différentes *user stories*. Les différents tests à exécuter seront identifiés lors de la planification de l'itération et gérés comme des tâches de test par le biais du tableau des tâches Agiles.

2.4.3. SG 3 Exécution des tests

Il faut bien sûr tester les logiciels développés dans l'itération. Selon le manifeste Agile, comme pour les autres objectifs dans ce domaine de processus, l'intensité de documentation sera généralement bien moindre que dans un projet séquentiel traditionnel. Souvent, ni procédure de test détaillée ni journal de test ne sont produits. Les tests, en particulier les tests de régression, devraient être, dans la mesure du possible et de leur faisabilité, automatisés.

Dans un projet Agile, les logiciels devraient être intégrés, au minimum, tous les jours. Les tests font partie intégrante du processus de développement Agile dans lequel le travail est effectué en coopération mutuelle pour livrer un produit de qualité. Équipes de test et de développement ne sont ni dédiées, ni séparées. Par conséquent, un test d'admission spécifique tel que défini dans la pratique spécifique 3.1 *Effectuer un test d'admission* est susceptible de ne pas être pertinent dans un environnement Agile pur. Toutefois, lorsque certains tests sont effectués en dehors de l'équipe Agile (comme c'est souvent le cas), un test d'admission sera généralement effectué par l'équipe responsable des tests en dehors de l'équipe Agile. Il faut garder à l'esprit que certaines activités

doivent être achevées avant que ne commencent les tests d'acceptation sur une *user story* spécifique au sein de l'équipe Agile. Ces activités, (telles que, par exemple, achever et réussir la quantité convenue de tests unitaires), font généralement partie de la *Definition of Done*.

L'exécution des tests est effectuée conformément aux priorités définies lors de la planification des itérations. Certains tests peuvent être exécutés en utilisant une procédure de test documentée, mais, en général, de nombreux tests seront exécutés en se basant sur un cadre de travail utilisant des tests exploratoires et des tests basés sur des sessions. Dans les tests exploratoires, la conception et l'exécution des tests se font en même temps, guidées par une charte de test préparée, qui fournit les objectifs et les conditions de test à couvrir pendant une session de test limitée dans le temps. Ainsi, les résultats des tests les plus récents guident les tests suivants.

Avec le développement itératif, il y a un besoin accru d'organiser et de structurer les tests de régression. Cela se fait parfois manuellement, mais de préférence à l'aide de tests de régression automatisés et d'outils support. Les tests de régression, en particulier les tests unitaires et d'intégration, font souvent partie d'un processus d'intégration continue, pratique importante dans le cadre du développement logiciel Agile. Il s'agit essentiellement d'un processus de construction et de test automatisé qui a lieu au moins une fois par jour et qui détecte rapidement et précocement les anomalies. L'intégration continue permet d'exécuter régulièrement des tests de régression automatisés et d'envoyer rapidement à l'équipe un retour d'information sur la qualité du code et la couverture du code obtenue.

Les incidents constatés lors des tests peuvent être consignés et signalés. La question se pose au sein des équipes Agile de la nécessité d'enregistrer tous les incidents trouvés. En particulier dans le cas d'une équipe colocalisée, les testeurs et développeurs ont besoin d'échanger. Les incidents/anomalies qui peuvent être corrigés immédiatement et inclus dans la prochaine version peuvent ne pas avoir besoin d'être enregistrés, ils doivent juste être corrigés. Dans un cadre Agile, tous les incidents détectés ne sont généralement pas consignés et gérés en conséquence. Comme indiqué, beaucoup sont simplement soulevés de manière informelle par un membre de l'équipe (qui peut être un testeur) et corrigés immédiatement par le développeur qui a injecté l'anomalie. Certaines données sont perdues, mais les coûts sont réduits en même temps. Certaines équipes n'enregistrent que les incidents qui échappent aux itérations, d'autres les enregistrent si le problème ne peut pas être résolu le jour même, d'autres encore n'enregistrent que les incidents de haute priorité. Si tous les incidents trouvés ne sont pas enregistrés, des critères doivent être disponibles pour déterminer quels incidents doivent être enregistrés et lesquels ne doivent pas l'être. N'oubliez pas que l'objectif est de corriger correctement l'anomalie et non de consigner un incident. Voici un exemple de ce critère : "Si une anomalie peut être résolue avant la prochaine mise en place et construction quotidienne, elle n'a pas besoin d'être documentée. Toute anomalie identifiée qui ne peut être résolue en un jour, ou qui affecte une partie prenante extérieure à l'équipe, doit être documentée et saisie".

Parfois, les incidents sont enregistrés sur le tableau des tâches Agiles, soit sous forme d'autocollant sur une tâche, soit comme tâche distincte, en visualisant qu'elle bloque l'achèvement d'une story et de ses tâches. Il est acceptable d'utiliser des post-it sur un tableau de tâches afin de gérer vos anomalies, c'est souvent la méthode choisie pour les équipes colocalisées. L'équipe Agile travaillera ensuite sur les incidents et les gèrera jusqu'à leur clôture (voir l'objectif spécifique 4 de ce domaine de processus ci-après). Certains choisissent d'enregistrer et de documenter les incidents trouvés dans un outil, par exemple de gestion ou de suivi des anomalies. Un tel outil doit être utilisé

de la manière la plus simple possible et ne doit pas forcer la soumission d'éléments de rapports d'incidents de tests sans ou avec trop peu de valeur ajoutée.

Les équipes Agiles considèrent aussi l'enregistrement des données pendant l'exécution des tests comme une bonne pratique afin de déterminer si le ou les éléments testés répondent aux critères d'acceptation définis et peuvent effectivement être qualifiés de « terminés ». Cela est également vrai lorsque l'on applique des techniques empiriques, comme les tests exploratoires. Les exemples d'informations qu'il peut être utile de documenter sont la couverture des tests (ce qui a été couvert et ce qui reste à tester), les observations faites pendant les tests, par exemple, le système et la *user story* testés semblent-ils stables, la liste des risques (quels risques ont été couverts et lesquels restent parmi les plus importants), les anomalies et autres problèmes constatés, enfin les éventuelles questions ouvertes.

Les informations enregistrées doivent être saisies et/ou résumées dans une forme quelconque d'outil de gestion de statut (par exemple, outils de gestion des tests, outils de gestion des tâches, tableau des tâches), pour faciliter la compréhension par l'équipe et les parties prenantes de l'état actuel de tous les tests effectués. Bien entendu, l'équipe n'a pas seulement besoin d'un statut pour les tâches de test isolées, mais elle doit aussi connaître le statut global de la *user story*. Elle est incomplète tant que toutes les activités connexes ne sont pas terminées, pas seulement les tests.

2.4.4. SG 4 Gérer les incidents de test jusqu'à la clôture

Comme indiqué ci-dessus, dans les environnements agiles, il arrive souvent que tous les incidents constatés ne soient pas enregistrés. Cet objectif spécifique s'applique uniquement aux incidents enregistrés et qui doivent donc être gérés jusqu'à leur clôture. En principe, la gestion des incidents dans un environnement Agile est simple. Dans le cas où l'incident est enregistré sur le tableau des tâches Agile, soit sous la forme d'un autocollant sur une tâche ou d'une tâche séparée, il est visualisé comme une anomalie empêchant l'achèvement d'une story et de ses tâches. La façon de faire habituelle en Agile sera appliquée comme pour tout autre obstacle ou tâche bloquant la progression. L'équipe évoquera l'incident, qui lui est affecté pour résolution, lors de réunions quotidiennes. Une fois résolu, l'incident est enlevé du tableau des tâches.

Si la décision est prise de reporter l'incident constaté à une autre itération, il devient simplement une autre option (ou un autre changement) souhaitée pour le produit. Ajoutez-le au *backlog* et établissez des priorités en conséquence. Lorsque la priorité est suffisamment élevée, il sera repris par l'équipe lors de la prochaine itération. Notez que même si la priorité est généralement définie par le métier, il faut parfois accorder une certaine priorité aux incidents liés à ce que l'on appelle la dette technique. Certaines équipes ont dédié des itérations pour nettoyer les incidents/anomalies liés à la dette technique. Toutefois, il ne s'agit pas d'une pratique standard recommandée en Agile, la dette doit être remboursée régulièrement.

Pour illustrer la manière dont les incidents peuvent être traités dans un environnement Agile, imaginez d'abord que l'équipe est en plein milieu d'une itération. Alors qu'elle travaille sur l'une des *user stories*, l'équipe détermine qu'elle ne peut pas la « terminer », parce qu'un des critères d'acceptation n'est pas rempli, par exemple, à cause d'une anomalie introduite pendant le développement. Les équipes agiles ont tendance à corriger ce genre d'anomalies le plus tôt possible dans l'itération en cours, sinon elles reporteraient le travail non terminé (équivalent de l'inventaire rationalisé dans le lean). Ce type d'incidents sera visualisé sur - et géré par le tableau des tâches.

Deuxièmement, supposons que toutes les *stories* de l'itération soient terminées selon la « DoD » à la fin de l'itération. Mais si, pendant la revue, lors de la démonstration d'une *story*, quelque chose ne se comporte pas comme le client le voudrait, on peut considérer que le comportement inattendu n'a pas vraiment été discuté avec le métier avant le début de l'itération. Dans ce cas, la fonctionnalité logicielle produite fonctionne, et ce qu'il faut probablement faire, c'est traduire l'incident en une *story* supplémentaire et l'ajouter au *Product Backlog*.

C'est une bonne pratique de surveiller le nombre d'anomalies trouvées et résolues au cours d'une itération ainsi que le nombre d'anomalies non résolues pouvant intégrer le *backlog* de la prochaine itération, au cours de la réunion quotidienne (*stand-up*) et également lors des rétrospectives.

Dans un environnement traditionnel de cycle de vie séquentiel, il existe souvent un comité de contrôle des configurations (*Configuration Control Board CCB*) spécifique qui se réunit, examine et décide de la suite à donner à l'incident. En Agile, ce rôle est dévolu à l'équipe, responsabilisée. Bien entendu, le représentant du métier, par exemple le *Product Owner*, a un rôle important à jouer lorsqu'il s'agit de décider de la suite à donner à un incident. La gestion des incidents est un travail de longue haleine, qui détourne l'attention et les efforts. Les coûts augmentent au fur et à mesure que nous essayons de formaliser un processus pour répertorier, les qualifier, ordonner les incidents et, finalement, planifier leur résolution. Un tel processus complexe de gestion des incidents est un pur gaspillage. Il vaut mieux énoncer quelques principes de base sur la manière dont les incidents sont traités et gérés par les équipes agiles et agir en conséquence.

2.5. Domaine de processus 2.5 Environnement de test

L'objectif de l'environnement de test est d'établir et de maintenir un environnement adéquat, y compris des données de test, dans lequel il est possible d'exécuter les tests d'une manière gérable et reproductible.

Avec le domaine de processus Environnement de test, un environnement de test adéquat, incluant les données de test génériques est établi et maintenu. Il y est possible d'exécuter les tests d'une manière gérable et reproductible. Bien entendu, dans un projet de développement logiciel utilisant le cycle de vie Agile, un environnement de test adéquat est indispensable. Etant donné la durée courte du cycle d'une itération, l'environnement de test doit être très stable et disponible. Les problèmes d'environnement de test auront toujours un impact immédiat sur le déroulement et les résultats de l'itération. Une gestion adéquate de la configuration et des modifications de l'environnement et des données de test est donc également de la plus haute importance.

Les environnements de test se présentent sous de nombreuses formes qui dépendent notamment du système testé mais, en général, la spécification, la mise en œuvre et la gestion de l'environnement de test se font parallèlement au projet de développement logiciel Agile. La mise en œuvre d'un environnement de test prend souvent beaucoup de temps et peut être très complexe, ce qui rend presque impossible l'exécution de cette tâche dans le temps de cycle limité d'une itération [Van der Aalst et Davis]. Elle implique également différents acteurs et ingénieurs qui constituent généralement une équipe agile. Les méthodologies Agiles ont été développées pour soutenir le développement de logiciels, et non pour le développement et la gestion d'environnements de test ou d'infrastructures. Elles sont simplement moins adaptées et la manière dont l'environnement de test ou le développement et la gestion de l'infrastructure sont effectués reste largement la même que

dans un environnement traditionnel, bien qu'il puisse y avoir quelques changements dans le processus lorsque l'équipe pratique l'agilité.

2.5.1. SG 1 Développer les exigences d'environnement de test

La spécification des exigences d'environnement de test est effectuée au début du projet. La spécification des exigences est examinée pour s'assurer de son exactitude, de sa pertinence, de sa faisabilité et de sa représentation précise d'un environnement opérationnel « réel ». La spécification précoce des exigences a l'avantage de donner plus de temps pour acquérir et/ou développer l'environnement de test requis et les composants tels que les simulateurs, les bouchons ou les pilotes.



Dans certains projets de développement Agile, une itération initiale (dite itération 0) est appliquée pour obtenir et spécifier les exigences de l'environnement de test.

2.5.2. SG 2 Implémenter l'environnement de test

Parfois (une partie de) la mise en œuvre a lieu dans l'itération initiale en plus d'autres activités, par exemple, la formation, l'évaluation de haut niveau des risques produit et l'établissement d'une définition de terminé (DoD – *Definition of Done*). De cette manière, l'objectif est de déjà disposer d'un environnement de test (partiellement) fonctionnel dès les premières itérations de développement logiciel. Les actions ou les problèmes environnementaux techniques peuvent faire partie du *backlog* produit.

La définition de l'environnement de test (y compris les données de test génériques) pourrait se faire comme pour un démarrage séquentiel du cycle de vie au moyen d'un plan complet, mais il est souvent préférable de commencer par la mise en œuvre dès que possible, et de disposer d'une version initiale de l'environnement de test dès le début de la première itération.

En effet, il est difficile d'être générique sur les environnements de test. Dans certains domaines, les bouchons et les pilotes sont rapidement remplacés par la virtualisation des services, ce qui permet généralement une mise en œuvre beaucoup plus rapide des environnements de test. Dans certains projets agiles, la fourniture des environnements de test est hautement automatisée. Les environnements de test peuvent être mis en service en quelques minutes ou heures, plutôt qu'en quelques jours. De nos jours, la virtualisation des machines, des serveurs et des services est également beaucoup plus utilisée. Le cloud peut également contribuer à fournir les environnements requis, ce qui accélère encore l'accès.

2.5.3. SG 3 Gérer et contrôler les environnements de test

La gestion et le contrôle de l'environnement de test auront lieu tout au long d'un projet Agile. Parfois, l'équipe Agile doit mettre en place et maintenir elle-même l'environnement de test, mais le plus souvent, celui-ci est géré en dehors de l'équipe Agile. Lorsque les activités sont effectuées par l'équipe Agile elle-même, cela implique bien sûr qu'elle dispose des connaissances techniques suffisantes pour pouvoir effectuer ces tâches. Cela signifie également que les tâches deviennent une partie du processus Agile, elles doivent, par exemple, être abordées lors d'une session de planification, apparaître sur le tableau des tâches et éventuellement être discutées lors de réunions quotidiennes en cas de problème.

La principale conclusion du *domaine de processus Environnement de test* est que les objectifs et les pratiques spécifiques sont toujours applicables et ne changent pas en substance, à l'exception de leur calendrier dans le cycle de vie.

3. TMMi Niveau 3 : Ajusté

3.1. Domaine de processus 3.1 Organisation de test

L'objectif du domaine de processus Organisation de test est d'identifier et d'organiser un groupe de personnes hautement qualifiées, responsables des tests. En plus des tests, le groupe de test gère également les améliorations et les actifs du processus de test de l'entreprise en se basant sur une compréhension approfondie des forces et des faiblesses du processus de test actuel de l'entreprise et de ses actifs.

3.1.1. SG 1 Etablir une organisation de test

L'organisation de test est un domaine de processus souvent mal compris. Beaucoup lisent que TMMi exige un groupe de test indépendant ou même un département indépendant qui effectue des tests. Bien que cela soit possible, d'autres modèles d'entreprise répondent aux exigences de TMMi. Outre l'organisme de test indépendant qui prépare et exécute les tests, le centre de compétence de test est également une option. Les tâches et responsabilités typiques d'un centre de compétence en matière de tests consistent à établir, gérer et améliorer les processus de test, et à fournir des ressources aux projets ayant une expérience, des connaissances et des compétences en matière de tests. Un centre de compétence en matière de tests est généralement propriétaire des méthodologies et des processus de test. Notez qu'un testeur au jour le jour fait généralement partie de l'équipe agile, mais il est fréquent qu'un testeur fasse en plus partie d'une organisation de test, comme le centre de compétence de test ou la guilde de test.

Une organisation de test nécessite un leadership dans les domaines liés aux tests et aux questions de qualité. Les membres du personnel d'un tel groupe sont appelés spécialistes des tests. L'option du centre de compétence de test s'intègre bien à l'Agile et garantit que le test est préservé en tant que discipline et traité avec le sérieux qui lui est dû. Bien sûr, certains leaders d'opinion Agiles ne sont pas favorables au maintien de disciplines particulières. Bien que cette option puisse être parfois préférée, la réalisation de TMMi exige de traiter et de conserver les tests en tant que discipline. Il va sans dire que cela ne devrait être fait que si cela a du sens du point de vue métier. Par conséquent, lorsque cela n'a pas de sens, TMMi n'est probablement pas la voie d'amélioration à privilégier.

En Agile, l'organisation des tests peut prendre la forme d'une guilde de test. C'est généralement un groupe informel et autogéré de testeurs, membres de différentes équipes agiles. Les activités importantes d'une guilde de test peuvent être le partage des connaissances, l'apprentissage en groupe, l'observation des tendances, etc. L'adhésion à une guilde se fait volontairement. Dans la pratique, une guilde de test se réunit souvent deux fois par semaine. Elle peut également se voir confier des tâches plus formelles comme l'établissement de parcours de carrière pour les tests, l'organisation de formations et la détermination, la planification et la mise en œuvre d'améliorations des processus de test. Cette guilde doit rester largement informelle et autogérée. Une contrainte importante pour la réussite d'une guilde de test est que ses membres se voient attribuer du temps à consacrer à ses différentes activités.

Les testeurs indépendants sont souvent plus efficaces pour détecter les anomalies. Trouver un équilibre entre indépendance et agilité est un défi. Certaines équipes Agiles conservent des équipes de test totalement indépendantes et distinctes, et affectent des testeurs à la demande pendant les

derniers jours de chaque sprint, ce qui permet de préserver leur indépendance. Ces testeurs peuvent ainsi fournir une évaluation objective et impartiale du logiciel. Cependant, cette approche est souvent problématique du fait des contraintes de temps, du manque de compréhension des nouvelles fonctionnalités du produit et des problèmes de relations avec les parties prenantes et les développeurs. Une autre option consiste à mettre en place une organisation de test indépendante et distincte, où les testeurs sont affectés à des équipes Agiles sur le long terme, dès le début du projet, ce qui leur permet de conserver leur indépendance tout en acquérant une bonne compréhension du produit et en développant des relations solides avec les autres membres de l'équipe. En outre, l'organisation indépendante de test peut avoir des testeurs spécialisés en dehors des équipes Agiles pour travailler sur des activités à long terme et/ou indépendantes des itérations, comme le développement d'outils de test automatisés, la réalisation de tests non fonctionnels, la création et la maintenance d'environnements de test et de jeux de données et la réalisation de niveaux de test qui pourraient ne pas s'intégrer dans un Sprint, (par exemple, les tests d'intégration de système). Notez qu'il est également possible de pratiquer des tests indépendants au sein d'une équipe Agile. L'indépendance existe tant que le testeur du produit est distinct du créateur du code et du créateur des actifs. Les développeurs peuvent donc tester le code des autres via des revues de code et/ou des tests unitaires, ou des analystes métier peuvent contribuer aux tests du point de vue de l'utilisateur en utilisant par exemple différents profils d'utilisateurs (personas). Enfin et surtout, L'indépendance peut être mise en œuvre à travers une structure qui l'organise et en définit la responsabilité, mais il s'agit probablement avant tout d'avoir le bon esprit critique.

3.1.2. SG 2 Établir des fonctions de test pour les spécialistes de test

Les tests sont considérés comme une profession et une discipline appréciée. L'équipe Agile a besoin de connaissances et de compétences en matière de tests. Le spécialiste des tests les fournira et accompagnera également les autres membres de l'équipe lors des activités de test. Les fonctions et les parcours professionnels sont définis et soutenus par un programme de formation. Les connaissances et les compétences requises pour un testeur Agile typique sont différentes de celles nécessaires dans une organisation traditionnelle. Le testeur aura bien sûr toujours besoin de connaissances et de compétences « traditionnelles » en matière de tests, mais aussi des connaissances et des compétences pour

l'automatisation des tests et pour la réalisation de tâches en dehors des tests, comme, par exemple ; la rédaction de scripts ou l'ingénierie des exigences. Étant donné qu'un testeur Agile travaille en équipe, les compétences relationnelles sont tout aussi importantes que les compétences techniques. Le profil de compétence du testeur Agile est dit en « T ». Cela doit se refléter dans la description de ses fonctions. L'Organisation de test est composée de personnes dotées des compétences et de la motivation nécessaires pour être un bon testeur dans un contexte Agile. Ces personnes sont affectées à une fonction de test spécifique. Dans un contexte Agile, il est particulièrement important, puisque l'ensemble de l'équipe effectuera des activités de test, d'élaborer d'autres descriptions de fonctions non spécifiques au test qui listent des attentes concernant les activités de test, les rôles et les connaissances et compétences requises en matière de test.



3.1.3. SG 3 Établir des parcours de carrière en test

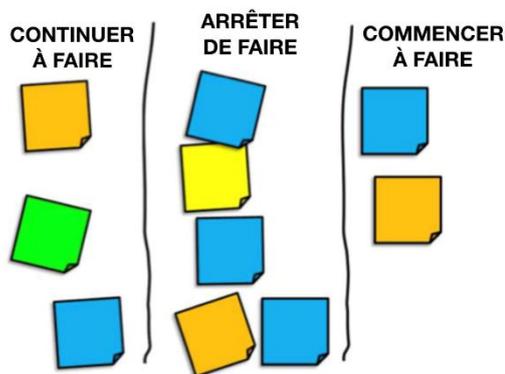
Les programmes de formation en test sont établis pour permettre aux professionnels des tests d'améliorer leurs connaissances, leurs compétences, leur statut et leur rémunération, et ainsi leur permettre de progresser dans leur carrière. Sur la base des programmes de formation définis, des plans de développement de carrière personnels sont élaborés et mis à jour pour chaque membre de l'organisation de test. Cet objectif spécifique et les pratiques spécifiques qui l'accompagnent sont également pleinement applicables dans un contexte Agile. L'accent étant mis sur l'aspect humain, il s'agit clairement d'un objectif spécifique important. L'équipe agile a besoin de testeurs qui comprennent l'exigence de leur fonction, peuvent coacher les autres membres de l'équipe et sont motivés. Beaucoup de ces besoins sont pris en compte dans les programmes de formation des testeurs.

Toutefois, la définition réelle des plans de carrière dans le test sera généralement quelque peu différente dans une organisation Agile par rapport à ceux d'une organisation qui suit un cycle de vie séquentiel. Alors que, dans une organisation traditionnelle, les plans de carrière dans le test sont souvent axés sur une croissance verticale (du testeur au gestionnaire de test), les plans de carrière dans le test au sein d'une organisation Agile ont tendance à se concentrer sur une croissance horizontale (du testeur junior au testeur senior et peut-être au coach de test). La diversité des fonctions et rôles hiérarchiques, comme la gestion des tests est généralement moins nécessaire qu'une évolution en tant que *Product Owner* ou *Scrum Master*. Il convient de noter que de nombreuses tâches traditionnelles de gestion des tests sont toujours pertinentes, mais qu'elles sont reprises par d'autres (par exemple, *Scrum Master*, équipe Agile).

3.1.4. SG 4 Déterminer, planifier et mettre en œuvre des améliorations du processus de test

Le processus d'amélioration tel que décrit par cet objectif spécifique de TMMi et ses pratiques spécifiques sont largement couverts par la réunion rétrospective effectuée par une équipe Agile. La rétrospective d'itération est une opportunité pour l'équipe Agile de s'auto-inspecter et de définir les actions d'amélioration à mettre en œuvre lors de la prochaine itération. Une rétrospective se produit généralement après la revue d'itération et avant la planification de la prochaine itération.

Pendant la rétrospective, l'équipe discute de ce qui s'est bien passé dans l'itération, ce qui pourrait être amélioré, et ce qui sera engagé pour améliorer la prochaine itération. Au cours de chaque rétrospective, l'équipe Agile identifie et planifie les moyens pour accroître la qualité des produits, par exemple en améliorant les processus de travail ou en adaptant la DoD, le cas échéant. Les rétrospectives peuvent donner lieu à des décisions d'amélioration axées sur l'efficacité, la productivité et la qualité des tests. Ils peuvent également porter sur la testabilité des applications, des *user stories*, des fonctionnalités ou des interfaces système. L'analyse des causes racines des anomalies peut amener à des améliorations des tests et du développement. Les testeurs jouent un rôle important dans les rétrospectives. Ils apportent leur point de vue unique en tant que membre de l'équipe. Tous, qu'ils soient testeurs ou non, peuvent apporter leur contribution aux activités de test et de non-test.



A la fin de la rétrospective, l'équipe agile devrait avoir repéré des améliorations (du test) à apporter lors de la prochaine itération. La mise en place de ces améliorations lors de l'itération représente l'adaptation suivant l'inspection de l'équipe agile elle-même. L'équipe vote afin de choisir, pour la prochaine itération les améliorations les plus importantes. Dans ce cas, les améliorations (des tests) peuvent rester dans le *backlog* pendant plus d'une itération, même si elles finiront par être mises en place. Il est important que l'équipe sélectionne uniquement les améliorations qu'elle peut faire tenir dans l'itération à venir, soit un maximum de deux ou trois par itération. Les testeurs devraient assumer l'amélioration des tests pour garantir qu'elle soit effective dans l'équipe. La rétrospective est un mécanisme important qui permet à une équipe d'évoluer et de s'améliorer constamment tout au long de la vie d'un projet.

Comme la portée est souvent limitée au cycle ou à l'itération précédente, les améliorations apportées sont petites mais fréquentes et se concentrent principalement sur la résolution de problèmes spécifiques au projet. Souvent, ces améliorations ne sont pas axées sur l'apprentissage inter-projets et l'institutionnalisation des améliorations. Si l'on regarde comment l'amélioration du processus (test) est organisée et gérée, il est probable que l'on se concentre moins sur un groupe de processus (test) au niveau global et plus sur l'autogestion des équipes au sein du projet. Bien que ce ne soit pas une mauvaise chose, il est important de créer un mécanisme permettant de partager les améliorations locales testées avec le reste de l'entreprise et d'aborder les domaines d'amélioration qui dépassent le pouvoir de l'équipe Agile. Une solution possible pourrait consister à faire participer un représentant de l'organisation de test aux réunions rétrospectives locales afin de s'assurer que les améliorations sont, le cas échéant, partagées et institutionnalisées dans toute l'entreprise. Le responsable de l'amélioration du processus de test peut également éliminer des problèmes fondamentaux ou plus larges affectant la manière de réaliser les tests à travers l'organisation.

3.1.5. SG 5 Déployer le processus de test organisationnel et intégrer des retours d'expérience

Les projets Agiles doivent aussi avoir accès et pouvoir réutiliser les processus standard de test et leurs actifs, autant qu'ils en ont besoin et que ça leur apporte de la valeur. L'organisation de tests veillera à ce qu'ils soient déployés dans tous les projets. (Voir le domaine de processus Cycle de vie et intégration du test pour plus d'informations sur le processus de test standard de l'entreprise, ses actifs et leur rapport aux projets agiles).

La mise en œuvre du processus de test standard de l'entreprise et l'utilisation de ses actifs dans les projets Agiles peuvent être suivies par les équipes auto organisées elles-mêmes, par exemple en abordant ce point en rétrospective. La présence à ces réunions d'un représentant de l'organisation de test, par exemple d'un responsable de l'amélioration du processus de test, permet à cette personne de suivre l'état d'avancement de la mise en œuvre et de traiter les questions inter-projets. Cette personne peut également rapporter à l'organisation de test les précieux enseignements tirés et les améliorations des tests de l'équipe Agile. Les leçons apprises et les améliorations des tests peuvent ensuite être soumises sous forme de propositions d'amélioration du processus de test et gérées en conséquence. Les enseignements tirés de l'équipe Agile sont ainsi, le cas échéant, intégrés dans le processus standard (Agile) (test) de l'organisation et dans ses actifs.

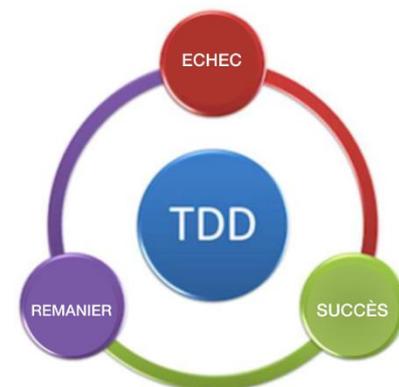
3.2. Domaine de processus 3.2 Programme de formation en test

L'objectif du domaine de processus Programme de formation en test est d'élaborer un programme de formation qui facilite le développement des connaissances et des compétences des personnes, afin que les tâches et les rôles des tests puissent être exécutés de manière efficace et efficiente.

L'objectif principal du programme de formation en test est de fournir la formation (test) nécessaire aux testeurs et aux autres membres de l'équipe. En raison de l'énoncé du Manifeste Agile « les individus et les interactions plus que les processus et les outils », les processus dans un environnement Agile seront généralement documentés de manière légère et n'aborderont pas tous les scénarios possibles. Dans ce cas, la formation des personnes devient alors un facteur de succès critique qui doit être visé et qui est presque comme une contre-mesure pour les processus légers. Un programme de formation de qualité garantit que les personnes impliquées dans les tests continuent à améliorer leurs connaissances et leurs compétences en matière de tests, et acquièrent les connaissances nécessaires dans le domaine et d'autres connaissances et compétences liées aux tests. Puisque la qualité et les tests sont désormais des responsabilités d'équipe, on s'attend à ce que la formation liée aux tests ne soit pas uniquement dispensée aux professionnels des tests, mais plutôt à l'ensemble de l'équipe Agile.

3.2.1. SG 1 Etablir une capacité organisationnelle de formation en test

Le processus commence par l'identification des besoins stratégiques de l'entreprise en matière de formation en test. En plus des besoins traditionnels de formation des testeurs, par exemple sur les techniques de conception des tests et les mesures de couverture, il est maintenant nécessaire d'acquérir des connaissances et des compétences sur les spécificités des tests Agiles. Parmi les exemples, citons le manifeste Agile et ses principes, les cadres tels que *Scrum*, Kanban et XP, les concepts de test Agile (par exemple, la pyramide de test et les quadrants de test), *Test Driven Development (TDD) (Développement Piloté par les Tests)*, *Behavior Driven Development (BDD)(développement piloté par le comportement)*, *Acceptance Test-Driven Development (ATDD)(Développement Piloté par les Tests d'Acceptation)*, rédaction de *user stories* - qui comprend le soutien du *Product Owner* pour rédiger les critères d'acceptation en s'assurant qu'ils restent testables, l'automatisation des tests et les tests exploratoires. En outre, certains domaines de connaissances et de compétences typiques de la gestion des tests, tels que la planification, l'estimation (par exemple, le *poker planning*), le suivi et l'analyse des risques (par exemple, le *poker de risque*), deviendront désormais des domaines qui doivent faire partie de l'ensemble des connaissances et des compétences de la plupart des testeurs travaillant dans une équipe agile et pas seulement du gestionnaire de test. L'identification des besoins stratégiques de formation en test ne se limite pas aux seuls testeurs, il est de la plus haute importance de spécifier également les besoins de formation liés aux tests pour les autres membres des équipes Agiles.



Comme dans toute autre entreprise, les besoins de formation en test seront traduits dans un plan de formation (léger), qui pourra également répondre à certains besoins spécifiques de formation en test de projets. Le plan de formation doit être revu et des engagements doivent être pris. Notez que, en particulier dans le domaine de l'agilité, certaines compétences sont transmises de manière efficace et efficiente par des moyens informels (par exemple, formation sur le lieu de travail, sessions

de formation à l'heure du déjeuner, coaching et mentorat) alors que d'autres compétences nécessitent une formation formelle. L'approche appropriée pour satisfaire les besoins spécifiques de formation en test doit être déterminée par domaine de connaissances et de compétences, par exemple, la formation sur le lieu de travail aux tests exploratoires étant généralement un aspect important.

3.2.2. SG 2 Fournir les formations en test

Globalement, les pratiques spécifiques du SG2 sont les mêmes dans une organisation traditionnelle et Agile. Sur la base du plan de formation de l'entreprise, les membres de l'équipe sont identifiés comme devant recevoir la formation nécessaire pour être en mesure de remplir efficacement leur rôle de test. La formation est ensuite programmée, y compris les ressources nécessaires, et dispensée. Cela semble évident, mais certaines organisations Agiles ont du mal à trouver du temps pour la formation. Les équipes Agiles se consacrent entièrement à terminer avant l'échéance de la fin de l'itération. Il est bien sûr nécessaire de prendre en compte le temps de formation pendant la planification de l'itération, puisque le participant y sera moins disponible.

Comme indiqué ci-dessus, les moyens de formation informelle sont souvent utilisés dans une organisation Agile. Par exemple, en plus de la formation formelle, le mentorat et le coaching sur le lieu de travail sont généralement encouragés et utilisés en permanence. En fait, le mentorat est la responsabilité de chacun, il est attendu à tous les niveaux d'une entreprise. Le travail en binôme est un autre moyen couramment utilisé par les équipes Agiles pour l'amélioration des compétences et le transfert de connaissances. Lorsque l'on utilise *Scrum*, coacher l'équipe sur les pratiques de *Scrum* est de la responsabilité du *Scrum Master*.

Les formations dispensées, qu'elles soient formelles ou informelles, seront enregistrées et leur efficacité évaluée. Une formation (test) suivie peut également être évaluée dans le cadre d'une réunion rétrospective au cours de laquelle on examine si les connaissances et les compétences des participants sont désormais en meilleure adéquation avec l'exécution des tâches de test.

3.3. Domaine de processus 3.3 Cycle de vie et intégration du test

L'objectif du cycle de vie et intégration du test est d'établir et de maintenir un ensemble utilisable d'actifs de processus de test dans l'organisation (par exemple, un cycle de vie de test standard) et de normes d'environnement de travail et d'intégrer et de synchroniser le cycle de vie des tests avec le cycle de développement. L'intégration du cycle de vie garantit une implication précoce des tests dans un projet. L'objectif du cycle de vie et de l'intégration du test est également de définir une approche de test cohérente à travers plusieurs niveaux de test, basée sur les risques identifiés et la stratégie de test définie, et de fournir un plan de test global, basé sur le cycle de vie des tests défini.

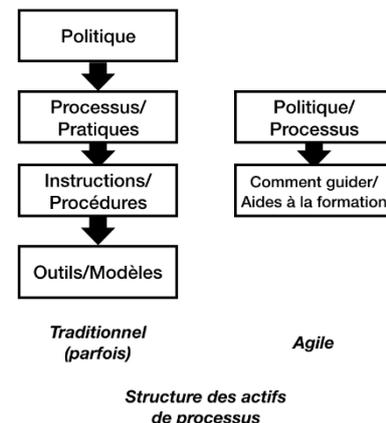
3.3.1. SG 1 Établir les actifs du processus de test organisationnel

Une responsabilité importante de l'organisation de test est de définir, de documenter et de maintenir un processus de test standard, en accord avec la politique et les objectifs de l'organisation. Il contient, entre autres, une description des différents types et niveaux de tests à exécuter. Bien que TMMi n'impose pas de tests détaillés, définis et documentés, souvent des modèles ciblés, élaborés avec des exemples, constituent une excellente approche pour établir une méthode de travail commune. Voir également les trois piliers de Bob Galen sur la qualité et les tests Agiles, où il identifie les normes comme étant des checklists, des modèles et des référentiels comme l'un des piliers des

tests Agiles de logiciels [Galen]. Un modèle contient implicitement, dans sa structure, l'identification des activités nécessaires pour recueillir des informations et documenter les résultats. Un processus n'a pas besoin d'être une séquence stricte d'activités. Lorsque des dépendances existent, elles peuvent être matérialisées par des notes dans le modèle, ou des champs dans un formulaire qui ne sont pas accessibles tant que d'autres champs obligatoires ne sont pas remplis. La valeur pratique des modèles réside dans leur transmission de l'intention réelle d'un processus. Les modèles permettent également d'éviter les ambiguïtés que l'on trouve généralement dans les documents "verbeux" des processus. En tant qu'organisation Agile, si vous avez des processus efficaces, vous pouvez simplement les documenter de manière légère, et peut-être y ajouter quelques éléments. Ci-après, quelques directives clés souvent utilisées dans une organisation Agile lors de l'établissement de processus :

- Les procédures « à faire » et les lignes directrices sont présentées séparément
- Aucun processus n'excède deux pages (objectif, règle non contraignante)
- Les processus ne contiennent pas d'informations sur le « comment » ou les outils, sauf si certains ont été rendus obligatoires pour tous les projets, quelles que soient leur taille et leur échelle.
- Des lignes directrices distinctes contiennent des options d'adaptation/planification et des informations pratiques. Notez que les « modes d'emploi » peuvent contenir des références à des ouvrages ou des documents existants.

Il n'est pas rare, dans de nombreuses grandes entreprises, de voir quatre niveaux d'actifs de processus tels que la politique, les processus/pratiques, les instructions/procédures et les outils/modèles. Ce n'est dû à des exigences TMMi, mais à la façon dont de nombreuses grandes entreprises ont choisi de mettre en œuvre leurs actifs de processus. Bien que le choix des actifs de processus incombe à chaque entreprise, la plupart des entreprises Agiles ont constaté que deux niveaux d'actifs de processus sont suffisants. Pour ce faire, il faut consolider une déclaration de politique avec la description de processus associée qui résume le « minimum à faire » pour exécuter le processus. Le second niveau contient des guides sur le « comment faire » et sur la personnalisation. Ce niveau peut être considéré comme une aide à l'adaptation du processus, donnant à l'équipe l'autonomie nécessaire pour décider comment travailler exactement dans le cadre défini. Il comprend généralement des modèles comme support. Dans la plupart des organisations agiles, les procédures pas à pas sont remplacées par des guides d'outils et des formations/mentorats. Comme indiqué ci-dessus, un modèle, de charte de test, de fiche de session de test ou de plan de test, peut servir de processus, les activités de processus requises étant implicites dans le modèle. Il s'agit d'une technique courante observée pour l'élaboration de descriptions de processus Agiles efficaces. [McMahon]



L'ensemble des processus de test standard de l'entreprise peut être adapté par les projets afin de créer leurs processus spécifiques définis. Dans le contexte de la SP 1.3, établir des critères et directives d'ajustement, la plupart des entreprises utilise aujourd'hui une approche dite « sur mesure » par suppression de processus. Cette approche exige que l'on s'efforce d'expliquer pourquoi quelque chose n'est pas nécessaire. Plus les projets sont simples, plus le travail d'adaptation est important, ce qui n'est pas une approche Agile. Quand on adapte, jusqu'où peut-on aller ? Y a-t-il un minimum défini ? Cependant, si vous partez d'un minimum où le minimum est ce que chacun doit faire, vous éliminez le risque de personnaliser un « minimum à faire ». Une approche sur mesure par ajout est beaucoup

plus cohérente avec les approches Agiles, et totalement conforme à TMMi. L'adaptation est un mécanisme extrêmement puissant pour les organisations Agiles qui souhaitent maintenir leur agilité ou pour les entreprises très matures (sur le test) qui cherchent à augmenter leur agilité. Un ensemble de critères simples pourrait aider à poser les bonnes questions lors de l'adaptation. Sans critères pour les aider, certaines ont tendance à trop s'adapter.

Pour soutenir le déploiement du processus de test standard, une bibliothèque de ressources de processus de test sera créée et maintenue, souvent sur un wiki, dans laquelle les testeurs pourront trouver par exemple des modèles, des meilleures pratiques et des checklists, qui les aideront dans leur travail quotidien. La création d'une bibliothèque de ressources de processus de test est également utile dans un contexte Agile, à condition l'utilisation des ressources fournies apportent une valeur ajoutée aux équipes pendant leurs activités de test. Le même raisonnement s'applique plus ou moins à une base de données de processus de test dans laquelle les données de test, par exemple les estimations, la couverture, les mesures de qualité, sont collectées et mises à la disposition des équipes. Il faut s'assurer que tout ce qui est collecté et mis à disposition pour permettre l'apprentissage et le partage d'expériences entre les équipes leur apporte de la valeur.

Bien entendu, même dans un contexte Agile, des normes d'environnement de travail sont utilisées pour guider la création d'environnements de travail de projet.

3.3.2. SG 2 Intégrer les modèles de cycle de vie de test aux modèles de développement

Le modèle standard de cycle de vie de test définit les principales activités et les produits livrables pour les différents niveaux de test. Le modèle standard de cycle de vie de test est aligné sur les modèles de cycle de vie de développement afin d'intégrer les activités de test en termes de phases, de jalons, de produits livrables et d'activités. L'intégration du cycle de vie se fait de manière à garantir une participation précoce des tests dans les projets. On dirait qu'il n'y a rien à faire dans le cas où un cycle de vie Agile a été adopté ; on s'est occupé de tout. En Agile, le développement et les tests doivent être totalement intégrés dans le cycle de vie, et les tests doivent déjà avoir eu lieu dès que possible.

Bien que cela soit vrai dans de nombreuses entreprises qui utilisent des méthodes Agiles, certaines luttent avec cet aspect. Leur itération ressemble à un mini cycle en V où les tests ne commencent que vers la fin de l'itération, sont souvent effectués sous la pression du temps, parfois même interrompus. Bien entendu, il ne s'agit pas d'une mise en œuvre d'un cycle de développement et de test Agile intégré comme cela devrait être le cas. Il montre cependant que pour certaines entreprises qui utilisent des méthodes proches de l'agilité, cet objectif spécifique est un domaine à améliorer. Il est important de créer une compréhension commune en définissant la manière dont les tests peuvent être intégrés et effectués dès le début de l'itération.

Cet objectif spécifique est donc également applicable à une entreprise qui fait du développement Agile de logiciels. S'il est réalisé de manière appropriée, le développement et les tests devraient être totalement intégrés et il existe une compréhension commune de la manière dont cela est fait. Dans ce cas, il n'y a rien de plus à faire. Dans d'autres circonstances, comme décrit ci-dessus, cet objectif spécifique est un domaine d'amélioration très important et devrait conduire à une bien meilleure harmonisation des activités de développement et de test réalisées dans une itération.

3.3.3. SG 3 Établir un plan de test maître

Au niveau 3 de TMMi, les tests concernent la planification des tests principaux. Elle traite de la coordination des tâches, des responsabilités et de l'approche de test à tous les niveaux. Cela permet d'éviter les redondances inutiles ou les omissions de tests entre les différents niveaux et peut augmenter de manière significative l'efficacité et la qualité du processus de test global. Le plan de test maître décrit l'application de la stratégie de test pour un projet particulier, y compris les niveaux particuliers à réaliser et la relation entre ces niveaux. Une planification TMMi de niveau 2 (test) est déjà traitée à la fois au niveau de la mise en production et au niveau de l'itération dans le cadre du domaine de processus planification du test. Une équipe Agile effectue généralement plusieurs niveaux de test qui doivent tous être traités de manière appropriée dans le cadre de la planification de la mise en production et des itérations.

Cependant, certains projets qui fonctionnent selon un modèle de cycle de vie Agile ont adopté une approche quelque peu hybride, en ce sens qu'en plus des tests au sein d'une équipe Agile, certains tests sont organisés en dehors de l'équipe. Dans de telles circonstances, un plan de test maître définira et gèrera la relation entre les équipes Agiles et les niveaux de test effectués en dehors de leur champ d'application, par exemple, les tests spécifiques non fonctionnels (sécurité, performance), les tests d'intégration matériel/logiciel, les tests d'intégration système ou les bêta-tests. Toutes les pratiques spécifiques liées à cet objectif spécifique seront également applicables. En particulier, au fur et à mesure que les entreprises se développent, et dans le cadre de projets plus importants, documenter, de manière cohérente, un plan de test maître et les décisions connexes commence à avoir une plus grande valeur. Cela est particulièrement vrai lorsqu'une entreprise utilise des cadres d'agilité à l'échelle.

Si tous les tests ont lieu avec les équipes Agiles, un plan de test maître spécifique n'a pas de valeur ajoutée, cet objectif TMMi sera donc probablement dans ce cas non pertinent. Si c'est le cas, la planification au niveau des mises en production et des itérations devrait couvrir tous les aspects des tests nécessaires.

3.4. Domaine de processus 3.4 Test non fonctionnel

L'objectif du domaine de processus test non fonctionnel est d'améliorer les capacités du processus de test pour inclure les tests non fonctionnels pendant la planification, la conception et l'exécution des tests.

L'importance des tests non fonctionnels est indépendante du cycle de vie appliqué. Le produit en cours de développement dictera les aspects non fonctionnels pertinents pour les tests. Cela est bien sûr vrai pour les modèles de cycle de vie séquentiels et les modèles de développement Agiles. Les tests non fonctionnels sont donc également applicables au développement de logiciels en Agile.



Toutefois, selon la nature des aspects non fonctionnels et l'approche de test utilisée, certains tests non fonctionnels, par exemple les performances et la fiabilité, lorsqu'ils sont testés de manière traditionnelle, ne peuvent pas être testés en tant que tels en une courte itération. Comme ces tests non fonctionnels peuvent, selon les méthodes et techniques utilisées, prendre des semaines à mettre en place, à organiser et à exécuter, leur temps de passage est encore moins compatible avec le rythme existant au cœur des itérations. Il est donc souvent nécessaire d'adopter une approche différente pour tester les attributs non fonctionnels. C'est une des améliorations

fondamentales de l'Agile, dans la mesure où les attributs de qualité sont déjà testés tôt tout au long des itérations, et non laissés à la fin comme avec un projet avec un cycle de vie traditionnel. Tous les composants/fonctions peuvent ne pas être disponibles dès la première itération pour effectuer un test non fonctionnel complet ; cependant, les résultats de ce test peuvent être utilisés pour donner des indications précoces sur les problèmes de qualité.

Les *user stories* doivent porter à la fois sur des éléments fonctionnels et non fonctionnels afin de garantir que le produit développé correspond aux besoins des utilisateurs et des clients. Les caractéristiques de qualité ISO 25010 peuvent aider à structurer les exigences et les testeurs doivent tenir compte de ces éléments non fonctionnels [ISO25010].

3.4.1. SG 1 Effectuer une évaluation des risques produit non fonctionnels

La ou les sessions sur les risques produit pour les projets agiles, tels qu'identifiés et décrits dans l'objectif spécifique « SG1 Effectuer une évaluation des risques produit » dans le cadre du processus planification du test, seront désormais explicitement étendues pour inclure également les aspects et risques non fonctionnels. Au niveau de la livraison, l'évaluation des risques, maintenant étendue aux tests non fonctionnels, peut être effectuée sur la base de la vision du produit. Au niveau de l'itération, elle est réalisée en utilisant les *user stories* qui définissent les exigences non fonctionnelles comme une contribution majeure.

Il est préférable que tous les membres de l'équipe, y compris le *Product Owner*, et éventuellement d'autres parties prenantes participent aux séances sur les risques produit. Pour certains domaines non fonctionnels, l'aide de spécialistes peut être nécessaire. Les séances sur les risques liés aux produits débouchent généralement sur une liste documentée des risques (non fonctionnels) liés aux produits, classés par ordre de priorité. Comme indiqué dans le domaine du processus de planification des tests, dans les projets agiles, le processus utilisé et la documentation qui en résulte seront beaucoup plus légers qu'un projet traditionnel suivant un modèle de cycle de vie séquentiel.

3.4.2. SG 2 Établir une approche de test non fonctionnel

Une approche de test pour les caractéristiques de qualité non fonctionnelles pertinentes est définie pour atténuer les risques non fonctionnels identifiés et hiérarchisés des produits. Pour une itération spécifique, les éléments non fonctionnels à tester sont identifiés lors de la planification de l'itération. La liste prioritaire des éléments non fonctionnels à tester est généralement liée aux *user stories* à tester dans cette itération. De nouveaux risques non fonctionnels liés aux produits peuvent apparaître au cours de l'itération et nécessiter des tests supplémentaires. Les questions telles que les nouveaux risques produit non fonctionnels nécessitant des tests supplémentaires sont généralement discutées lors de réunions quotidiennes.

L'approche de test non fonctionnel qui est définie au niveau de l'itération pour atténuer les risques non fonctionnels couvrira généralement l'identification de méthodes de test non fonctionnelles appropriées et de technique(s) de test basée(s) sur le niveau et le type de risques non fonctionnels. En général, elle portera également sur l'utilisation d'outils de soutien et sur l'approche d'automatisation des tests non fonctionnels. L'approche pour les tests non fonctionnels au niveau de la mise en production sera à un niveau beaucoup plus élevé et devrait être basée sur la stratégie de test définie au niveau du programme ou de l'organisation, ainsi que sur le plan de test principal défini, s'il existe. (Voir SG3 Etablir un plan de test maître dans le domaine de processus Cycle de vie et

intégration du test pour la justification de l'établissement ou non d'un plan de test maître). L'approche de test non fonctionnel au niveau de la livraison et de l'itération fait partie de l'approche de test globale et sera maintenue ou affichée sur le wiki de l'équipe/du projet.

Les critères de sortie non fonctionnels font partie de la « *Definition of Done* » (DoD). Il est important que la DoD ait des critères spécifiques liés aux tests non fonctionnels, par exemple, le *Mean-Time-Between-Failures* (MTBF)³ ou « les pages web du front ont été testées pour la liste des 10 principaux risques de l'OWASP⁴ ». L'itération devrait aboutir à la mise en œuvre de l'ensemble convenu de *user stories* non fonctionnelles (y compris leurs critères d'acceptation) et répondre aux critères de sortie non fonctionnels (test) tels que définis dans la DoD. Notez que les attributs non fonctionnels peuvent également faire partie des critères d'acceptation des *user stories* fonctionnelles et ne doivent donc pas nécessairement être spécifiés comme des *user stories* non fonctionnelles distinctes. Une *Definition of Done* existera non seulement au niveau de l'itération, mais souvent aussi au niveau d'une livraison couvrant plusieurs itérations. La DoD, pour la mise en production, peut également contenir des critères non fonctionnels.

3.4.3. SG 3 Analyse et conception de tests non fonctionnels

Cet objectif spécifique suit en grande partie les mêmes pratiques, mais maintenant d'un point de vue non fonctionnel, comme pour l'objectif spécifique SG1 Effectuer l'analyse et la conception des tests. Il s'agit d'utiliser les techniques de conception des tests du domaine de processus de conception et d'exécution des tests. Au cours de l'analyse et de la conception des tests, l'approche des tests non fonctionnels est traduite en conditions et tests tangibles. En Agile, l'analyse et la conception des tests et leur exécution sont des activités qui se soutiennent mutuellement. Elles se déroulent généralement en parallèle tout au long d'une itération. Cela est également vrai pour la plupart des tests non fonctionnels. L'analyse des tests non fonctionnels n'est donc pas une activité explicite séparée, mais plutôt une activité implicite que les testeurs effectuent dans le cadre de leur rôle dans le développement collaboratif de *user stories*.

L'analyse des *user stories* non fonctionnelles permet d'identifier les conditions de test non fonctionnelles. Les conditions de test sont essentiellement une identification des « choses » qui doivent être testées/couvertes. Si les critères d'acceptation définis sont suffisamment clairs et détaillés, ils reprennent généralement le rôle des conditions de test traditionnelles. Les critères d'acceptation sont ensuite traduits en tests non fonctionnels. Dans le cas de tests non fonctionnels, il est souvent avantageux d'effectuer des analyses de test à un niveau supérieur plutôt que de se contenter de simples *user stories*. Par exemple, en analysant une fonctionnalité, une *epic* ou un ensemble de *stories* pour identifier des conditions de test non fonctionnelles plus abstraites que celles des *user stories* ou qui couvrent également plusieurs *user stories*. Le principe de « tester d'abord » étant appliqué en Agile, les tests non fonctionnels qui couvrent l'ensemble des conditions de test non fonctionnelles seront identifiés (et éventuellement automatisés) avant, ou au moins en même temps que, le développement du code.

Pour la plupart des tests manuels non fonctionnels, les tests seront identifiés/affinés à mesure que l'équipe progresse dans l'exécution des tests non fonctionnels. Le plus souvent, les tests ne sont pas documentés de manière aussi détaillée que dans les projets traditionnels, mais plutôt sous forme d'idées de tests lorsqu'on utilise des tests exploratoires. Pour les domaines non fonctionnels

³ *Mean Time Between Failure* : temps moyen entre les défaillances (pannes)

⁴ Top Ten [OWASP](#) : projet de la « Open Web Application Security Project », pour recenser les dix risques de sécurité applicatifs web les plus critiques

complexes et critiques, une approche plus traditionnelle de la conception des tests et du développement de cas de test utilisant des techniques formelles de conception de tests non fonctionnels peut être la meilleure façon de couvrir les risques. Toutefois, cette approche permet également de limiter la quantité de documentation par rapport aux tests non fonctionnels dans un environnement traditionnel de cycle de vie séquentiel. La priorisation des tests non fonctionnels suit généralement la priorisation de la *user story* qu'ils couvrent. Toutefois, la hiérarchisation peut également être dictée par le temps nécessaire à la préparation et à la réalisation d'un certain test non fonctionnel.

Les données de test spécifiques nécessaires pour soutenir l'exécution des tests non fonctionnels sont identifiées. Typiquement, en Agile, on ne commence pas par spécifier l'intégralité des données de test nécessaires dans un document de spécification de test. Lorsqu'elles sont spécifiées, elles sont souvent enregistrées dans le cadre de la *user story*. Cependant, si les outils et/ou les fonctionnalités nécessaires sont disponibles, les données de test nécessaires pour soutenir les tests non fonctionnels sont le plus souvent créées instantanément pour permettre un démarrage rapide de l'exécution des tests manuels non fonctionnels. En revanche, pour les tests non fonctionnels automatisés, les données doivent généralement être spécifiées à l'avance.

La traçabilité entre les exigences non fonctionnelles, les conditions de test et les tests doit être établie et maintenue. Les équipes doivent indiquer clairement qu'elles ont couvert les différentes *user stories* non fonctionnelles et les critères d'acceptation dans le cadre de leurs tests. Dans de nombreuses organisations Agiles, les *user stories* servent de base pour développer un ensemble de critères d'acceptation et de tests. Cette approche permet d'obtenir une traçabilité horizontale des exigences aux tests.

3.4.4. SG 4 Implémenter les tests non fonctionnels

L'implémentation des tests consiste à mettre en place tout ce qui est nécessaire pour commencer l'exécution des tests. Généralement, le développement de la documentation des tests, par exemple les procédures de test, pour soutenir l'exécution des tests est réduit au minimum. Des scripts de test plutôt automatisés (régression) sont développés et classés par ordre de priorité.

L'implémentation des tests non fonctionnels suivra bon nombre des pratiques déjà décrites dans l'objectif spécifique SG2 « Implémenter les tests », dans le cadre du domaine de processus Conception et d'exécution des tests.

Ce qui doit être fait spécifiquement et la manière dont la mise en œuvre des tests non fonctionnels est effectuée dépendent largement de l'approche définie, des techniques utilisées et du niveau des caractéristiques non fonctionnelles qui doivent être testées. Par exemple, les tests exploratoires, qui nécessitent moins de préparation, peuvent convenir aux tests d'utilisabilité, mais sont souvent moins adaptés aux tests de fiabilité et de performance approfondis.

Pour certaines caractéristiques de qualité non fonctionnelles, la disponibilité des données de test est essentielle et doit être gérée pendant l'implémentation du test. Cette procédure est en grande partie la même que pour les projets traditionnels. Ils peuvent néanmoins être documentés de manière succincte, ce qui permet de les réutiliser pour des tests de régression.

Bien entendu, la mise en œuvre et la préparation des tests commenceront dès que possible, parallèlement à d'autres activités (de test). Il ne s'agit pas d'une phase distincte, mais plutôt d'un ensemble d'activités (énumérées sur le tableau des tâches) qui doivent être réalisées pour permettre une exécution efficace et efficiente des tests.

3.4.5. SG 5 Exécuter les tests non fonctionnels

Comme pour l'objectif précédent dans ce domaine de processus, nous nous référerons largement à l'objectif spécifique connexe du SG 3 « Exécuter les tests », qui fait partie du domaine de processus Conception et exécution des tests, dont il a été question plus haut dans le présent document. Les pratiques pour l'exécution de tests non fonctionnels, le signalement d'incidents de test et l'écriture de journaux de test dans un environnement Agile sont fondamentalement les mêmes que pour les tests fonctionnels dans un environnement Agile. La documentation sera généralement rédigée de manière moins intensive que dans un projet traditionnel. Souvent, ni procédure de test détaillée ni journal de test ne sont produits.

L'exécution des tests non fonctionnels se fait conformément aux priorités définies lors de la planification des itérations. Certains tests peuvent être exécutés en utilisant une procédure de test documentée, mais, généralement, de nombreux tests non fonctionnels seront également exécutés et encadrés par des tests exploratoires et des tests basés sur des sessions. Avec le développement itératif, il y a un besoin accru d'organiser et de structurer les tests de régression. Cela se fait parfois manuellement, mais de préférence en utilisant des tests de régression automatisés et des outils de soutien. Bien entendu, les tests de régression s'appliquent également aux aspects non fonctionnels du système qui ont été identifiés comme importants à tester.

Les incidents non fonctionnels découverts lors des tests peuvent être enregistrés et signalés par l'équipe. Généralement, au sein des projets Agile, l'équipe discute pour déterminer si tous les incidents trouvés doivent effectivement être enregistrés. Certaines équipes n'enregistrent que les incidents qui échappent aux itérations, d'autres les enregistrent s'ils ne peuvent pas être corrigés le jour même, d'autres encore n'enregistrent que les incidents de haute priorité. Si tous les incidents trouvés ne sont pas consignés, des critères doivent être disponibles pour déterminer quels incidents doivent être consignés et lesquels ne doivent pas l'être. Parfois, les incidents sont consignés sur le tableau des tâches Agiles, soit sous forme d'autocollant sur une tâche, soit sous forme de tâche distincte, en indiquant qu'elle bloque l'achèvement d'une *story* et de ses tâches. Certains choisissent d'enregistrer et de documenter les incidents trouvés à l'aide d'un outil, par exemple un outil de gestion ou de suivi des anomalies. Un tel outil doit alors être utilisé de manière aussi légère que possible et ne doit pas forcer la soumission d'éléments de rapports d'incidents de test non fonctionnels qui n'ont pas ou trop peu de valeur ajoutée.

Pour savoir si les éléments testés satisfont les critères d'acceptation définis et peuvent ainsi être ainsi être considérés comme terminés, les équipes Agiles considèrent que c'est une bonne pratique d'enregistrer des informations pendant les tests non-fonctionnels. Les informations consignées doivent être saisies et/ou résumées dans une forme quelconque d'outil de gestion de statut (par exemple, outils de gestion des tests, outils de gestion des tâches, tableau des tâches), pour qu'il soit facile pour l'équipe et les parties prenantes de comprendre l'état actuel de tous les tests effectués.

3.5. Domaine de processus 3.5 Revues par les pairs

L'objectif du processus de revues par les pairs est de vérifier que les produits d'activité répondent aux exigences spécifiées et d'éliminer les anomalies de certains produits d'activité de manière précoce et efficace. Un effet corollaire important est de développer une meilleure compréhension des produits d'activité et des anomalies qui pourraient être évités.

3.5.1. SG 1 Établir une approche de revue par les pairs

Les équipes agiles ne procèdent généralement pas à des revues formelles par les pairs, en ce sens qu'elles n'ont généralement pas de moment précis où les personnes se rencontrent pour donner leur avis sur un produit d'activité. Cependant, de telles révisions, moins formelles, sont effectuées en continu, tout au long du développement, ce qui permet d'atteindre l'objectif de révision par les pairs. Cette pratique est courante dans de nombreuses organisations Agiles. Cependant, il faut encore faire preuve de discipline dans la conduite de ces activités. Cet objectif de TMMi couvre les pratiques qui permettront d'établir une approche de revue par les pairs au sein d'un projet. Une approche de revue définit comment, où et quand les activités de revue doivent avoir lieu et si ces activités sont formelles ou informelles. L'établissement d'une approche de revue par les pairs est également applicable aux projets Agiles, cependant, les techniques d'examen appliquées et la manière dont les examens sont organisés sont généralement très différentes.

Exemples de revues par les pairs généralement réalisés dans le cadre de projets Agiles :

- En organisant régulièrement des séances d'affinage et de mise au point (*grooming*) des spécifications (par exemple, des *user stories*) avec l'équipe et les parties prenantes de l'entreprise tout au long d'une itération ;
- Des réunions quotidiennes avec les autres membres de l'équipe pour discuter ouvertement des produits d'activité, par exemple le code ou les tests en cours de développement et fournir un retour d'information.
- La démonstration fréquente des produits aux clients dès le début, et a minima à la fin d'une itération pendant la revue d'itération ;

Les projets échouent souvent à cause de mauvaises spécifications. Les problèmes de spécification peuvent résulter d'un manque de compréhension des besoins réels des utilisateurs, d'une absence de vision globale du système, de caractéristiques redondantes ou contradictoires, et d'autres erreurs de communication. Dans un cadre Agile, les *user stories* sont rédigées de manière à saisir les besoins du point de vue des représentants métier, des développeurs et des testeurs. Dans le développement séquentiel, cette vision partagée d'une fonctionnalité est réalisée par des révisions formelles après la rédaction des exigences ; dans le développement Agile, cette vision partagée est réalisée par des révisions informelles fréquentes pendant l'établissement des exigences. Ces séances d'examen informelles sont souvent appelées soit séances d'affinage de *backlog* soit *backlog grooming*. Au cours des réunions d'affinage, le représentant métier et l'équipe de développement (et la partie prenante si elle existe) utilisent des techniques de révision pour trouver le niveau de détail nécessaire à la mise en œuvre et pour clarifier les questions en suspens.

L'examen presque continu par l'équipe des produits d'activité en cours d'élaboration fait partie de l'approche globale de l'équipe. Il est effectué dans le but d'identifier rapidement les anomalies et aussi d'identifier les possibilités d'amélioration. Les méthodes et techniques agiles telles que XP (*Xtreme Programming*) ou le travail en binômes incluent également des revues par les pairs comme pratique de base pour créer des boucles de feedback pour les équipes. Bien entendu, en cas de complexité ou de risque élevé, l'équipe peut choisir d'appliquer une technique d'examen semi-formelle ou formelle, comme une inspection. Dans ces cas, il est clairement justifié de consacrer des efforts à un examen plus formel et d'appliquer une méthode de travail plus disciplinée. Une partie des critères de l'approche d'examen peut être définie pour déterminer quand utiliser une technique d'examen plus formelle.

Alors que l'ingénierie des exigences met l'accent sur la validation des exigences par des méthodes telles que les revues informelles, la revue pas à pas, la lecture basée perspective (de sources, guides, normes établis), les méthodes Agiles s'efforcent plutôt de valider les exigences par des retours d'information fréquents et précoces afin de mettre en œuvre rapidement des incréments de produit de valeur. La nécessité d'une validation formelle précoce est réduite en montrant des résultats rapides sous la forme d'incrément de produit intégrés. Si l'incrément ne répond pas complètement aux exigences des parties prenantes, le delta est réintégré dans le *backlog* des produits sous la forme de nouvelles exigences et est classé par ordre de priorité avec tous les autres éléments du *backlog*. Une démonstration de ce qui a réellement été construit au cours d'une itération est un moyen simple et très efficace de dynamiser une discussion à propos de la validation autour de quelque chose de concret. Rien n'apporte plus d'intérêt à la conversation que de pouvoir voir comment quelque chose fonctionne réellement. La démonstration est une activité qui est effectuée pendant la revue de l'itération. Les fonctionnalités sont démontrées aux parties prenantes et discutées avec elles, les adaptations nécessaires sont apportées au *backlog* produit ou au plan de lancement afin de refléter les nouveaux enseignements de la discussion.

3.5.2. SG 2 Effectuer des revues par les pairs

Bien sûr, comme toute approche, elle ne doit pas seulement exister en tant qu'approche définie et convenue, une adhésion est nécessaire. L'équipe doit consacrer beaucoup de temps aux séances d'amélioration du *backlog* au cours d'une itération, en appliquant des examens informels (et éventuellement formels) dans le cadre de sa routine quotidienne, et montrer régulièrement l'avancement aux parties prenantes. Des démonstrations aux parties prenantes et aux clients sont attendues au moins à la fin de chaque itération.

Les critères d'entrée ne sont généralement applicables qu'avec des examens formels, et donc susceptibles d'être moins ou pas pertinents dans le cadre des projets Agiles. Toutefois, l'utilisation de critères de sortie est applicable et présente une valeur ajoutée. INVEST [Wake] est un exemple d'un ensemble de critères de sortie qui sont couramment utilisés dans les projets Agiles lors de la révision et de la mise à jour des *user stories*. INVEST est un acronyme qui englobe les concepts suivants qui constituent une bonne *user story* :

- Indépendant
- Négociable
- Valorisable
- Estimable
- Suffisamment Petit
- Testable

Bien entendu, d'autres critères de sortie liés à la qualité existent et peuvent également être utilisés de manière avantageuse



Il est important que le testeur, faisant partie de l'équipe Agile, participe aux sessions de revue. Cela est particulièrement vrai pour les sessions au cours desquelles les produits d'activité sont discutés et revus et qui servent de base aux tests tout au long de l'itération, par exemple lors des sessions d'affinage du *backlog*. Il est recommandé d'avoir au moins un développeur et un testeur présents lors de ces sessions, afin de garantir la présence d'autres points de vue sur le système. Généralement, la

perspective unique du testeur améliorera la *user story* en identifiant les détails manquants ou les exigences non fonctionnelles. Un testeur peut notamment aider à identifier et à définir les critères d'acceptation d'une *user story*. Le testeur contribue en posant aux représentants métier des questions ouvertes sur la *user story*, en proposant les moyens de la tester et de confirmer ses critères d'acceptation.

La Pratique spécifique 2.3 Analyser les données des revues par les pairs est particulièrement pertinente dans le cas des revues formelles. Dans le cas des revues formelles, beaucoup d'efforts sont déployés pour s'assurer qu'ils le sont de manière efficace et efficiente. Les données de la revue par les pairs sont recueillies et communiquées à l'équipe afin d'apprendre et d'améliorer le processus de revue. Comme indiqué, les revues formelles, par exemple l'inspection, sont moins courantes dans les projets Agile. Alors que la collecte de données est une partie essentielle des revues formelles, elle est beaucoup moins courante dans les revues informelles. Par conséquent, on peut affirmer que la collecte, l'analyse et la communication de données détaillées des revues par les pairs sont moins ou même pas pertinentes dans un contexte Agile. Pour choisir de collecter ou non des données lors des revues par les pairs, posez les questions suivantes :

- Qui va utiliser ces données si elles sont collectées ?
- Comment ces données sont-elles liées à nos objectifs métier ?

Si personne ne veut utiliser les données de manière significative, alors ne gaspillez pas de précieuses ressources pour les collecter. En conclusion, pour la plupart des projets Agiles, la *pratique spécifique 2.3 Analyser les données des revues par les pairs* sera considérée comme probablement non pertinente. Notez que certaines données de base précieuses seront toujours collectées et utilisées dans le contexte des pratiques génériques, telle que la pratique générique 2.8 Surveiller et contrôler le processus.

4. TMMi Niveau 4 : Géré quantitativement

Comme indiqué précédemment dans ce document, il est important de se rappeler qu'il existe une rupture naturelle entre le niveau 3 de TMMi et les niveaux 4 et 5 de TMMi. L'intérêt de faire passer une entreprise aux niveaux 4 et 5 de TMMi continue de faire l'objet d'une importante controverse. Il est notamment recommandé aux entreprises Agiles d'évaluer ce choix de manière critique et de ne mettre en œuvre que les pratiques des niveaux TMMi 4 et 5 qui ont de l'importance et de la valeur ajoutée. Bien que TMMi soit complet, pour réussir, les entreprises doivent identifier les pratiques de test et les améliorations clés sur lesquelles il faut se concentrer. Il est intéressant de noter que Jeff Sutherland, co-fondateur de *Scrum*, a publié un document sur l'intérêt d'utiliser *Scrum* avec des pratiques de plus haut niveau de maturité [Sutherland et al]. Ci-après, des informations seront fournies sur les niveaux 4 et 5 de TMMi, la manière dont on peut valoriser ces niveaux plus élevés de TMMi et leurs pratiques, grâce à une utilisation moins formelle dans une approche Agile.

Tout d'abord, contrairement à ce qui a été dit, il est aussi envisageable d'utiliser déjà de manière sélective les pratiques TMMi de niveau 4 et 5 avec des techniques Agiles pour répondre aux principaux objectifs métier, tout en essayant d'atteindre le niveau 2 ou 3 de TMMi. Vous n'avez pas besoin d'attendre et vous ne devriez pas. Soyez flexible et ne restez pas bloqué sur les pratiques de test de niveau 2 et 3 de TMMi. Utilisez le modèle basé sur les moteurs de votre entreprise et utilisez les domaines de processus, les objectifs et les pratiques qui ont le plus de valeur. Ainsi, utilisez TMMi de manière plus continue dans une certaine mesure, et n'utilisez pas les niveaux de maturité de manière trop stricte.

4.1. Domaine de processus 4.1 Mesure du test

L'objectif de la mesure de test est d'identifier, de collecter, d'analyser et d'appliquer des mesures pour aider une entreprise à évaluer objectivement l'efficacité et l'efficience du processus de test, la productivité de son personnel de test, la qualité du produit qui en résulte et les résultats de l'amélioration du test. En tant que tel, l'organisme de test développera et maintiendra une capacité de mesure de test utilisée pour soutenir les besoins d'information de la direction.

Les approches agiles aident à obtenir l'engagement de ceux qui doivent effectuer et mesurer avec précision le travail dans les projets. Toutefois, les approches agiles ne fournissent pas beaucoup de soutien lorsqu'il s'agit de prendre du recul pour s'assurer que les bonnes mesures sont utilisées dans toute l'entreprise, qu'elles sont alignées sur les objectifs de mesure dérivés des besoins de l'entreprise. Ce sont là des atouts que le domaine de processus TMMi Mesure du test peut apporter aux organisations Agiles.

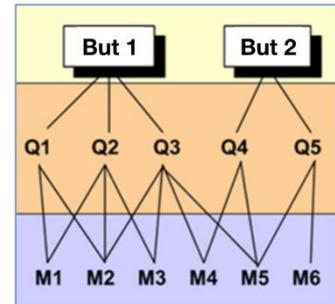
4.1.1. SG 1 Aligner les Activités de mesure du test et d'analyse

Trop souvent, les programmes de mesure semblent viser à collecter le plus de données possibles. Un programme de mesure doit être très ciblé, seules les choses qui comptent doivent être collectées. Commencez par définir des objectifs clairs pour le programme de mesure et revoyez-les régulièrement, par exemple : "Cela devrait nous permettre de mieux réussir les tests dans nos projets". Ensuite, discutez avec les parties prenantes des mesures de test qui permettront vraiment

de soutenir ces objectifs. Sur la base de ce brainstorming, un ensemble limité de mesures de base sera défini.

La méthode *Goal-Question-Metric* (GQM) (approche par les buts, But-Question-Mesure) [Van Solingen et Berghout] est une méthode très pratique qui soutiendra cette approche. Des métriques spécifiques, adaptées au contexte, sont nécessaires, en fonction des besoins des entreprises. Les métriques standard de l'entreprise sont le plus souvent insuffisantes pour une amélioration réelle des processus

(de test). L'objectif des mesures est de guider les décisions. Utilisez de petites équipes autonomes pour en tirer des mesures significatives, puis examinez-les et affinez-les à l'intérieur de cycles courts.



Avec un programme de mesure de test ciblé partant de la valeur ajoutée (métier), le domaine de processus de mesure de test TMMi est totalement conforme au principe de simplicité Agile. Assurez-vous qu'il existe une stratégie de mesure de test et de collecte de données allégée au sein de l'entreprise. Il convient de souligner que l'on peut faire évoluer un programme de mesure du test en commençant par quelques objectifs de mesure ciblés et quelques mesures clés, et en les élargissant ou en les modifiant ultérieurement en fonction de l'évolution et des besoins de l'entreprise.

Il est à noter que dans un cadre Agile, l'accent sera davantage mis sur le travail en équipe et la réflexion systémique. Cela peut entraîner un élargissement correspondant de certaines mesures à l'équipe et au système global plutôt que de les limiter uniquement aux spécificités des tests eux-mêmes. Cela pourrait bien conduire à davantage de défis lors de la phase d'analyse des mesures.

Bien sûr, les paramètres qui seront définis et pour lesquels les données seront collectées dépendront des objectifs de mesure, mais voici quelques exemples de paramètres de test typiques pour les équipes agiles :

- Temps de cycle des anomalies ; les équipes agiles doivent s'efforcer de réparer les anomalies le plus rapidement possible. En fait, l'un des principaux objectifs de l'approche Agile collaborative est de réparer les anomalies plus rapidement afin que les logiciels soient mis en production plus tôt.
- Débordement des anomalies (nombre d'anomalies reportées à une livraison ultérieure) ; les équipes agiles visent à produire un logiciel fonctionnel à chaque itération. Le débordement d'anomalies mesure les anomalies qui ne sont pas corrigées au cours d'une itération ou d'un sprint donné en comptant simplement les anomalies restant à la fin de chaque sprint ou itération. Dans certaines entreprises agiles, on appelle cela la dette technique.
- Nombre d'anomalies constatées au cours d'une itération
- Nombre d'anomalies constatées après la remise à la production/clients
- Nombre d'anomalies constatées par des personnes extérieures à l'équipe
- Nombre de demandes au support clients
- Pourcentage de couverture des tests automatisés
- Pourcentage de couverture du code.

4.1.2. SG 2 Fournir les résultats de mesure du test

Une fois que les mesures pertinentes sont définies en fonction des besoins des entreprises, les pratiques spécifiques à cet objectif précis sont généralement toutes applicables comme au sein

d'une entreprise traditionnelle. La seule différence est que cela se fera avec un état d'esprit différent, en gardant les pratiques aussi légères que possible.

Les données nécessaires à la mesure des tests sont collectées et contrôlées quant à leur exhaustivité et leur intégrité, puis les données sont analysées comme prévu et les résultats sont communiqués à toutes les parties prenantes concernées.

4.2. Domaine de processus 4.2 Evaluation de la qualité des produits

L'objectif de l'évaluation de la qualité des produits est de développer une compréhension quantitative de la qualité des produits et de soutenir ainsi la réalisation des objectifs de qualité des produits de projets spécifiques.

4.2.1. SG 1 Etablir les objectifs du projet pour la qualité des produits et leurs priorités

En substance, contexte Agile ou environnement traditionnel, il n'y a pas de différence réelle entre l'identification et la fixation d'objectifs prioritaires mesurables pour la qualité des produits. L'objectif global est de contribuer à satisfaire les besoins et les désirs des clients et des utilisateurs finaux en matière de produits de qualité. Il existe de multiples options pour y parvenir dans un contexte Agile. Certaines utilisent la *Definition-of-Done* pour définir des objectifs mesurables pour la qualité du produit, d'autres utilisent des fonctions, des *epics* ou des *user stories* pour les définir, en utilisant également des critères d'acceptation pour les rendre mesurables. Le choix (la façon de travailler) dépend entre autres de la nature de l'objectif de qualité du produit. La norme [ISO25010] donne un aperçu des attributs de qualité.

La *Definition of Done* (DoD) est une liste complète des activités d'équipe et/ou des produits de travail nécessaires qui garantit que seules les *user stories* réellement terminées, non seulement en termes de fonctionnalités mais aussi en termes de qualité du produit sont livrées. L'avantage d'indiquer des objectifs de qualité du produit dans la définition des tâches est qu'ils restent visibles et donc réalisables par l'ensemble de l'équipe au fur et à mesure de l'élaboration des incréments. Il est évident que les objectifs de qualité devront être globalement cohérents avec l'application pour être mentionnés au sein de la DoD. Si les objectifs de qualité ne sont pas applicables à l'échelle de l'organisation, la meilleure solution consiste probablement à les définir comme une fonction ou une *epic*.

Au début, parfois "seulement" une liste de haut niveau des attributs de qualité pertinents du produit est spécifiée pour la mise en production. Ces attributs seront progressivement développés au cours de la session d'affinage, sous la forme d'une fonction, d'une *epic* ou d'une *user story*. Par la suite, au début d'une itération, l'équipe examinera si un ou plusieurs des objectifs de qualité du produit spécifiés feront partie de l'objectif pour cette itération spécifique.

4.2.2. SG 2 Quantifier et gérer le progrès vers l'atteinte des objectifs du projet en termes de qualité des produits

Une fois que les objectifs quantitatifs de qualité des produits ont été fixés, ils seront surveillés, contrôlés et éventuellement ajustés si nécessaire. Étant donné que la qualité du produit sera définie dans le cadre de la DoD et/ou en tant que fonctionnalités, *epics* ou *user stories* avec des critères d'acceptation mesurables, les mêmes méthodes et techniques peuvent être utilisées pour suivre et

gérer la progression vers la réalisation des objectifs de qualité du produit comme déjà expliqué et décrit dans le *domaine de processus 2.3 Surveillance et contrôle du test* au niveau 2 de TMMi. Les progrès seront mesurés quotidiennement et discutés lors de la réunion de suivi quotidienne qui rendra compte des progrès, de la qualité et des risques. En outre, l'évaluation de la qualité des produits est généralement soutenue par le processus de mesure des tests, qui fournit une infrastructure de mesure.

4.3. Domaine de processus 4.3 Revues par les pairs avancées

L'objectif des Revues par les pairs avancées, qui s'appuient sur les pratiques des Revues par les Pairs du domaine de processus de niveau 3 de TMMi, est de mesurer la qualité des produits au début du cycle de vie et d'améliorer la stratégie et l'approche de test. Cela se fait en alignant et coordonnant les revues par les pairs (tests statiques) et les tests dynamiques.

4.3.1. SG 1 Coordonner l'approche des revues par les pairs avec l'approche des tests dynamiques

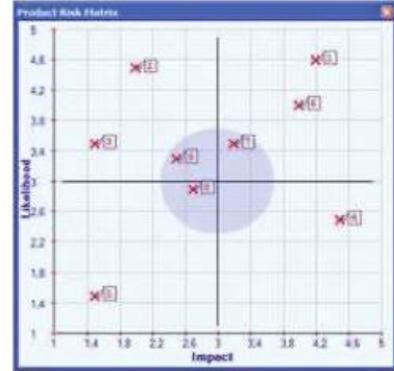
Alors que dans les projets traditionnels, les tests statiques, par exemple les revues, ne sont le plus souvent pas liés aux tests dynamiques effectués par une équipe de test indépendante, on s'attend à ce que ce soit totalement différent dans un contexte Agile. Dans un environnement Agile, l'approche visant à garantir la qualité du produit doit être une démarche dans laquelle toutes les activités liées aux tests, tant statiques que dynamiques, sont organisées comme une seule approche cohérente et coordonnée. Il est bien sûr de la responsabilité de l'équipe Agile d'intégrer les tests statiques et dynamiques et de trouver la combinaison optimale. L'équipe Agile est censée prendre l'initiative d'établir une approche intégrée cohérente et efficace. L'approche pour les tests statiques (revue par les pairs) sera donc alignée et coordonnée avec celle pour les tests dynamiques.

4.3.2. SG 2 Mesurer la qualité du produit tôt dans le cycle de vie grâce aux revues par les pairs

Au niveau 4 de TMMi, l'entreprise fixe des objectifs quantitatifs pour les produits logiciels et les produits d'activités connexes. Les revues par les pairs jouent un rôle essentiel dans la réalisation de ces objectifs. Alors qu'au niveau 3 de TMMi, les revues par les pairs sont principalement effectuées pour trouver des anomalies, l'accent est maintenant mis sur la mesure de la qualité des produits et des documents afin de contrôler la qualité des produits au début du cycle de vie. Les pratiques de revue sont améliorées pour inclure des pratiques comme l'échantillonnage, l'application de critères de sortie et les règles de prescription. C'est en effet ce que Tom Gilb appelle l'inspection agile, il s'agit de passer du nettoyage à l'échantillonnage, la mesure, la motivation et la prévention des anomalies [Gilb]. Cette pratique conviendra clairement à une entreprise Agile de grande maturité. Un autre exemple serait d'utiliser les règles INVEST [Wake] pour les *user stories* comme *Definition-of-Ready* (DoR) et de mesurer la qualité des *user stories* en les comparant à ces règles. Avoir une *Definition-of-Ready* signifie que les stories peuvent être immédiatement commencées.

4.3.3. SG 3 Ajuster l'approche du test sur la base des résultats des revues tôt dans le cycle de vie

Lorsque les résultats des revues par les pairs et les tests dynamiques sont coordonnés, les résultats et les données des premières revues peuvent influencer les risques produits et l'approche des tests. Comme déjà indiqué dans l'objectif spécifique 1 de ce domaine de processus, en Agile, la qualité est un travail d'équipe et les résultats des activités basées sur la vérification et la validation seront discutés lors de réunions d'équipe. Le principe consistant à utiliser les résultats des premiers tests pour influencer les activités de test successives correspond donc presque parfaitement à la manière de travailler au sein des équipes et des projets Agile.



5. TMMi Niveau 5 : En Optimisation

Les objectifs et les pratiques du niveau 5 de TMMi sont issus d'approches très matures de gestion de la qualité et d'amélioration des processus de haute maturité. À ce haut niveau de maturité, les objectifs et les pratiques sont presque indépendants du cycle de vie du logiciel qui est appliqué. Bien sûr, une entreprise lean- agile abordera les choses différemment des entreprises traditionnelles hiérarchisées. Néanmoins, les pratiques essentielles autour du contrôle de la qualité, de l'optimisation des processus et de la prévention des anomalies resteront largement inchangées. La raison en est que ces pratiques sont considérées comme des pratiques de gestion de la qualité, généralement effectuées à un niveau plus élevé que le niveau de développement opérationnel du produit. Ces pratiques ne sont donc pas des pratiques directement liées aux activités principales de développement et/ou de test de logiciels dans le cadre d'un projet ou d'un flux de livraison de produits.

5.1. Domaine de processus 5.1 Prévention des défauts

L'objectif de la prévention des défauts est d'identifier et d'analyser les causes communes des défauts tout au long du cycle de vie de développement et de définir des actions pour empêcher que des défauts similaires ne se reproduisent à l'avenir.

Les entreprises Agiles se concentrent généralement déjà sur la prévention des défauts, ce qui constitue une base importante sur laquelle les objectifs et les pratiques spécifiques de ce domaine de processus peuvent être construits. Les entreprises Agiles efficaces se concentrent sur prévenir les défauts plutôt que simplement les découvrir. Une culture de prévention des défauts consiste à séparer les défauts du « travail en cours » de ceux qui s'échappent de l'itération dans laquelle les *user stories* sont élaborées.

La rétrospective agile est une pratique qui permet aux équipes de réfléchir, d'apprendre et de s'améliorer continuellement dans ce qu'elles font. Bien qu'ils soient le plus souvent utilisés pour explorer la manière actuelle de travailler, les défauts peuvent également être utilisés pour étudier les problèmes de qualité ou pour convenir d'actions susceptibles d'améliorer la qualité des logiciels fournis.

Le domaine de processus "Prévention des défauts" concerne les pratiques d'identification et d'analyse des causes communes de ces défauts dans les équipes, les produits et les flux de valeur, et la définition d'actions spécifiques pour éliminer les causes communes de ces types de défauts à l'avenir. Tous les défauts, qu'ils soient constatés au cours du développement, des tests ou sur le terrain, relèvent du domaine de processus. Comme la prévention des défauts à ce niveau nécessite des données de mesure, la prévention des défauts s'appuie sur les pratiques de mesure de niveau 4 du TMMi et sur les données de mesure disponibles concernant le développement, les tests et la qualité des produits.

Il existe essentiellement deux types distincts d'activités de prévention des défauts dans les entreprises Agiles : au niveau de l'équipe et au niveau inter équipes. Les activités de prévention des défauts liées à l'équipe sont de la responsabilité de l'équipe. Elles qui ont déjà été traitées au niveau 3 de TMMi, les activités de prévention des défauts inter-équipes sont effectuées par un groupe de processus de test ou une guilda de test et ce sont celles qui sont spécifiquement traitées par ce domaine de processus.

5.1.1. SG 1 Déterminer les causes fréquentes de défauts

En Agile, les améliorations, y compris les améliorations des processus de test, ont généralement lieu sur la base de boucles de retour fréquentes. Comme la portée est souvent limitée à l'itération précédente, les améliorations apportées sont petites mais fréquentes et se concentrent principalement sur la résolution de problèmes spécifiques locaux. Souvent, ces améliorations ne sont pas axées sur l'apprentissage inter équipes et l'institutionnalisation des améliorations. La résolution de problèmes uniquement liés à l'équipe pourrait aussi facilement conduire à une sous-optimisation et à perdre de vue le tableau d'ensemble. C'est là que ce domaine de processus TMMi apporte une valeur ajoutée, en s'appuyant sur les pratiques Agiles existantes, la prévention des défauts est désormais également appliquée au niveau organisationnel (entre les équipes, les produits et les flux de valeur) et généralement gérée par un groupe de processus (de test) ou une guilde de test (*voir paragraphe 3.1.1*).

La sélection des défauts à analyser doit être basée sur divers facteurs, notamment le risque et la valeur métier. Il convient de se concentrer sur les domaines où la prévention des défauts présente la plus grande valeur ajoutée (généralement en termes de réduction des coûts ou des risques) et/ou les endroits où les défauts sont les plus critiques. Bien que le domaine de la prévention des défauts soit couramment utilisé pour les défaillances, il peut également être utilisé pour les questions relatives à la vitesse de développement.

Bien qu'un groupe de processus de tests puisse coordonner les activités de prévention des défauts, cela doit être fait par une équipe pluridisciplinaire, qui contiendrait par exemple des représentants de l'ingénierie des exigences, de l'ingénierie des systèmes et/ou du développement de logiciels, car les actions d'amélioration affecteront souvent d'autres disciplines. Bien entendu, cela est déjà courant dans les méthodes et les équipes de livraison Agile. Assurez-vous cependant que toutes les disciplines requises sont impliquées, conformément aux actions d'amélioration. Les méthodes analytiques d'évaluation des causes racines des défauts, telles que les diagrammes de cause à effet, les diagrammes en arête de poisson ou les 5 pourquoi, sont généralement utilisées pour la prévention des défauts et les pratiques Agiles déjà courantes.

Les trois pratiques spécifiques (*1.1 Définir les paramètres de sélection des défauts*, *1.2 Sélectionner les défauts à analyser* et, *1.3 Analyser les causes des défauts sélectionnés*) telles que définies par le modèle TMMi pour cet objectif spécifique sont également applicables à une entreprise Agile.

5.1.2. SG 2 Prioriser et définir des actions pour éliminer systématiquement les causes racines des défauts

Il n'y a pas de différences réelles dans l'exécution des pratiques spécifiques dans le cadre de cet objectif spécifique, qu'elles soient pratiquées dans un cycle de vie Agile ou séquentiel. Bien entendu, la solution proposée et la proposition d'amélioration ultérieure seront différentes puisqu'elles doivent s'adapter à la méthode de travail Agile, par exemple en créant des éléments supplémentaires dans le *backlog* au lieu de propositions d'action formalisées avec des approbations. Des mesures appropriées sont prises pour minimiser la probabilité que le problème se reproduise. Souvent, cela signifie qu'il faut combiner l'amélioration des processus avec un encadrement et/ou une formation supplémentaire pour les membres de l'équipe et les autres membres du personnel. Les questions relatives aux personnes et aux processus doivent être abordées de manière intégrée, plutôt que d'essayer de les séparer artificiellement.

Les entreprises Agiles s'appuient également sur des équipes autogérées. Cela implique que les équipes peuvent choisir les améliorations à mettre en œuvre et/ou à rejeter. Ainsi, lors de la priorisation et de la définition des actions, il faut impliquer une représentation appropriée des équipes pour discuter de la valeur ajoutée réelle que ces améliorations leur apportent.

La manière dont les améliorations seront déployées sera également différente, elle sera moins descendante et ressemblera davantage, pour les équipes autogérées, à une proposition basée sur le volontariat. Pour en savoir plus sur le déploiement des améliorations des tests, voir la section "Optimisation du processus de test" ci-après.

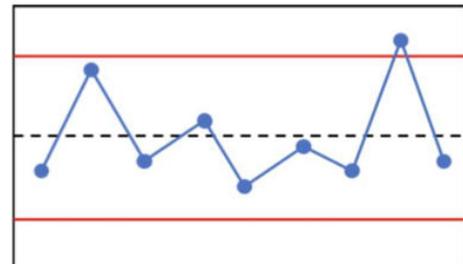
5.2. Domaine de processus 5.2 Contrôle de la qualité

L'objectif du contrôle de qualité est de gérer et de contrôler statistiquement le processus de test. À ce niveau, les performances du processus de test sont entièrement prévisibles et stabilisées dans des limites acceptables. Les tests au niveau du projet sont effectués à l'aide de méthodes statistiques basées sur des échantillons représentatifs afin de prévoir la qualité du produit et de rendre les tests plus efficaces.

5.2.1. SG 1 Établir un processus de test statistiquement contrôlé

Les principes et les éléments essentiels du contrôle de qualité sont les mêmes quelle que soit la méthode de cycle de vie appliquée. Le contrôle de qualité des processus consiste à établir des objectifs pour la réalisation du processus de test standard. Sur la base des mesures prises par les équipes sur les performances du processus de test, une analyse est effectuée et des ajustements sont faits pour maintenir les performances du processus de test dans des limites acceptables. Quand ces performances sont stabilisées dans les limites définies, ces mesures et leurs limites deviennent les bases de référence utilisées pour contrôler statistiquement la performance du processus de test.

La capacité du processus de test standard de l'entreprise, c'est-à-dire la performance du processus de test qu'une équipe peut espérer atteindre, est maintenant pleinement comprise et connue. Par conséquent, les écarts par rapport à ces attentes peuvent être corrigés au sein d'une équipe dès le début et de manière cohérente afin de garantir que les performances restent dans les limites acceptables.



Il est important de sélectionner pour ce contrôle statistique les sous-processus essentiels pour l'entreprise ayant connu des problèmes dans le passé. L'expérience montre que le plus souvent, les sous-processus qui répondent le mieux à ces critères ne sont pas limités à des équipes individuelles, mais traversent plutôt plusieurs équipes, projets et flux de livraison de produits. Lorsque l'on utilise des méthodes et techniques Agiles, le travail est divisé en courtes itérations, où le travail effectué à chaque itération implique la réalisation de multiples activités de développement et de test. Par conséquent, le développement Agile soutient, plus efficacement que le développement traditionnel, le suivi des sous-processus les plus précieux pour le contrôle statistique, car les cycles sont beaucoup plus courts, ce qui permet un retour d'information presque instantané sur les performances des processus de test. Un retour d'information rapide est généralement très précieux pour les équipes et constitue un principe Agile important.

5.2.2. SG 2 Réaliser le test en utilisant des méthodes statistiques

Le contrôle de qualité des produits s'appuie sur des profils opérationnels [Musa] et des modèles d'utilisation du produit dans son environnement prévu pour faire des inférences statistiquement valables et aboutir à un échantillon représentatif de tests. Cette approche, particulièrement utile au niveau du système, utilise des méthodes de tests statistiques pour prédire la qualité du produit sur la base de cet échantillon représentatif. En d'autres termes, lorsque l'on teste un sous-ensemble de toutes les utilisations possibles représentées par le profil d'utilisation ou de fonctionnement, les résultats des tests peuvent servir de base à des conclusions sur les performances globales du produit.

Il s'agit fondamentalement d'une approche qui s'intègre bien à l'Agile, parce qu'elle est légère par nature et motivée par le retour d'informations sur l'utilisation opérationnelle et les sessions de revue du produit. Cependant, elle exige qu'une entreprise Agile ait le bon niveau de maturité pour pouvoir être appliquée. Cette approche se construit essentiellement à partir des données et mesures obtenues, dont elle dépend fortement.

Une entreprise utilisant des méthodes statistiques dans ses tests est en mesure de quantifier les niveaux de confiance et la fiabilité (voir le cadre TMMi pour plus d'informations sur ces termes). Quand ces méthodes sont appliquées, ces deux niveaux sont utilisés en tant que définition de terminé.

5.3. Domaine de processus 5.3 Optimisation du processus de test

L'objectif de l'optimisation du processus de test est d'améliorer en permanence les processus de test existants utilisés dans l'entreprise et d'identifier les nouvelles technologies de test (par exemple, les outils ou les méthodes de test) qui peuvent être appropriées et de gérer leur déploiement dans l'entreprise. L'optimisation du processus de test soutient également la réutilisation des actifs de test dans l'ensemble de l'entreprise. Les améliorations soutiennent les objectifs de l'entreprise en matière de qualité des produits et de performance des processus de test, tels qu'ils découlent des objectifs métier de l'entreprise.

5.3.1. SG 1 Sélectionner des améliorations du processus de test

Les propositions d'amélioration des tests sont collectées et analysées. Elles proviennent de sources internes, généralement, d'équipes, de projets et de produits différents. Les sources internes comprennent les questions et les idées provenant des activités de prévention des défauts, des revues d'équipe, des rétrospectives, des chapitres et des guildes. Après une analyse des coûts et des bénéfices, les propositions d'amélioration intéressantes sont expérimentées pour évaluer les changements majeurs nouveaux et non prouvés avant qu'ils soient, le cas échéant, déployés dans d'autres équipes, à l'échelle de l'entreprise. Enfin, une sélection d'améliorations à tester est faite pour un déploiement dans toute l'entreprise.

5.3.2. SG 2 Evaluer de nouvelles technologies de test afin de déterminer leur impact sur le processus de test

Dans le cadre de l'optimisation des processus de test, le groupe de processus de test organisé de manière centralisée (ou guildes de test) recherche de manière proactive sur le marché les nouvelles technologies, telles que les outils, les méthodes, les techniques ou les innovations techniques, qui peuvent améliorer la capacité de test des équipes Agiles. En se tenant au courant des innovations technologiques liées aux tests, en les évaluant et en les expérimentant systématiquement, l'entreprise, en consultation avec les représentants des équipes, sélectionne les technologies de test appropriées pour améliorer la qualité de ses produits et la productivité de ses activités de test. Des tests pilotes sont réalisés au sein des équipes pour évaluer les technologies de test nouvelles et non éprouvées avant qu'elles ne soient intégrées dans la pratique standard.

5.3.3. SG 3 Déployer des améliorations des tests

Des améliorations des tests et de nouvelles technologies de test appropriées sont déployées dans les équipes Agiles pour améliorer les tests. Leurs avantages sont mesurés et les informations sur les nouvelles innovations sont diffusées dans les entreprises. Une différence existe bien sûr entre le déploiement dans une entreprise traditionnelle et dans une entreprise Agile. Dans une entreprise traditionnelle, le déploiement se traduit le plus souvent par de nouvelles pratiques qui doivent être appliquées par toutes les personnes concernées. Dans une entreprise Agile, l'équipe a plus d'autonomie, elle est autogérée. Cela implique également un processus de déploiement différent. Les améliorations des tests et les nouvelles technologies de test appropriées sont incorporées dans les actifs du processus de test organisationnel selon les besoins, la formation et le soutien sont organisés et mis à disposition. Toutefois, en fin de compte, l'équipe Agile doit décider elle-même si une amélioration repérée sera incorporée ou non à ses méthodes.

5.3.4. SG 4 Établir la réutilisation d'actifs du processus de test de haute qualité

Au niveau 3 de TMMi, une certaine réutilisation des logiciels de test entre les équipes, les projets et les flux de produits peut déjà avoir lieu ; cependant, la réutilisation des ressources de test devient un objectif majeur au niveau 5 de TMMi. L'équipe ne doit pas réinventer la roue, mais plutôt tirer parti de l'expertise et des ressources déjà disponibles, ce qui permet d'économiser du temps et des efforts. Les ressources de test de haute qualité qui peuvent probablement être réutilisées dans toute l'entreprise sont identifiées, par exemple lors de réunions rétrospectives, de sessions sur les leçons apprises, de discussions au sein de la guildes de test ou d'évaluations de test. Les ressources de test réutilisables sont ensuite sélectionnées et ajoutées au référentiel de test central dans un format modifiable pour être réutilisées dans le cadre de livraisons de produits et de projets futurs.

6. Vue d'ensemble de l'applicabilité des objectifs et pratiques spécifiques de TMMi

6.1. Évaluations TMMi

Pour aider les évaluateurs, (Lead) Assessors ou non, dans leurs analyses, ce chapitre donne un aperçu de l'applicabilité des domaines de processus, des objectifs spécifiques et des pratiques

spécifiques dans un contexte Agile. Dans les chapitres précédents, un nombre limité de pratiques spécifiques ont été identifiées qui devraient être moins ou peut-être même non pertinentes. Elles sont énumérées ci-après. Au cours d'une évaluation TMMi, un évaluateur devra explicitement déterminer la pertinence de ces éléments dans le contexte spécifique Agile. S'il est décidé qu'un élément est effectivement moins ou même non pertinent, cela doit être clairement documenté avec une justification à la fois dans les processus de l'entreprise et dans le rapport d'évaluation.

6.2. TMMi niveau 2 - Discipliné

Domaine de processus 2.2 Planification du test

SG2 Etablir une approche de test	Applicable
- SP2.3 Définir les critères d'entrée	<i>Probablement moins/non pertinent</i>
- SP2.5 Définir les critères de suspension et de reprise	<i>Probablement moins/non pertinent</i>
SG3 Établir des estimations de test	Applicable
- SP3.2 Définir le cycle de vie du test	<i>Probablement moins/non pertinent</i>
SG5 Obtenir un engagement sur le plan de test	Applicable
- SP5.2 Concilier les niveaux de travail et de ressources	<i>Probablement moins/non pertinent</i>

Domaine de processus 2.3 Surveillance et contrôle du test

SG2 Surveiller la qualité du produit par rapport au plan et aux attentes	Applicable
- SP2.3 Surveiller les risques produit	<i>Probablement moins/non pertinent</i>
- SP2.5 Surveiller les critères de suspension et de reprise	<i>Probablement moins/non pertinent</i>

Domaine de processus 2.4 Conception et exécution des tests

SG2 Implémenter les tests	Applicable
- SP2.3 Spécifier la procédure de test d'admission	<i>Probablement moins/non pertinent</i>
- SP2.4 Planifier l'exécution des tests	<i>Probablement moins/non pertinent</i>
SG3 Exécuter les tests	Applicable
- SP3.1 Effectuer un test d'admission	<i>Probablement moins/non pertinent</i>

6.3. TMMi niveau 3 - Ajusté

Domaine de processus 3.5 Revue par les pairs

SG2 Effectuer des Revues par les pairs	Applicable
- SP2.3 Analyser les données des revues par les pairs	<i>Probablement moins/non pertinent</i>

6.4. TMMi niveau 4 – Géré quantitativement

Tous les domaines du processus TMMi de niveau 4, les objectifs spécifiques et les pratiques spécifiques sont applicables. Aucune pratique spécifique n'est indiquée comme étant probablement moins pertinente/non pertinente

6.5. TMMi niveau 5 – En optimisation

Tous les domaines du processus TMMi de niveau 5, les objectifs spécifiques et les pratiques spécifiques sont applicables. Aucune pratique spécifique n'est indiquée comme étant probablement moins pertinente/non pertinente

Références

[Black, Van Veenendaal] Rex Black, Erik van Veenendaal and Dorothy Graham (2012), Foundations of Software Testing ISTQB Certification (3rd edition), Cengage

[Cohn] M. Cohn (2009), Succeeding with Agile: Software Development using Scrum, Addison-Wesley

[Galen] R. Galen (2015), Three Pillars of Agile Quality and Testing: Achieving Balanced Results in your Journey

Towards Agile Quality, RGCG, LLC

[Gilb] T. Gilb (2004), Agile Specification Quality Control, International Council on Systems Engineering (INCOSE)

[ISO25010] ISO/IEC 25010 (2011), Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) System and Software Quality Models, International Organization for Standardization

[Sutherland et al] C.R. Jakobsen, K. Johnson and J. Sutherland (2007), Scrum and CMMI level 5: The Magic Portion for Code Warriors in: IEEE Agile 2007 Conference, Washington, D.C.

[McMahon] Paul. E. McMahon (2011), Integrating CMMI with Agile Development – Case Studies and Proven Techniques for Faster Performance Improvement, Addison Wesley

[Musa] J. Musa (1998), Software Reliability Engineering Testing, McGraw-Hill Education

[Van der Aalst and Davis] L. van der Aalst and Cecile Davis (2013), TMap NEXT in scrum – Effective testing in agile projects, Sogeti Nederland B.V.

[Van Solingen and Berghout] R. van Solingen and E. Berghout (1999), The Goal/Question/Metric method, McGrawHill

[van Veenendaal] E. van Veenendaal (2014), PRISMA: Product Risk Assessment for Agile Projects, in: Testing Experience, Issue 04/12, December 2012

[Wake] B. Wake (2003), INVEST in Good Stories, and SMART Tasks, xp123.com/articles/invest-in-good-stories-and-smart-tasks