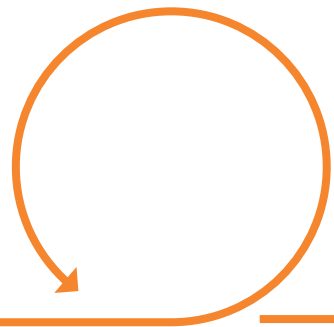


TRANSITION VERS L'AGILITÉ

À L'ÉCHELLE D'UNE ORGANISATION



DEUXIÈME ÉDITION - 2012

AVANT-PROPOS

Présent à l'international, Valtech est un groupe pionnier et leader dans le domaine de l'Agilité, des technologies, et du digital. Avec 1 500 collaborateurs, Valtech accompagne et forme ses clients en mode Agile dans la conception, la réalisation et l'optimisation de projets et de plateformes digitales critiques pour leur croissance. S'appuyant sur une expertise technologique reconnue, Valtech propose une vision novatrice et une mise en œuvre intégrée sur toute la chaîne de valeur digitale avec pour finalité l'accélération du « Time to Market », l'accroissement des revenus et du retour sur investissement pour ses clients.

Créé en 1993 et coté sur l'Eurolist d'Euronext, Valtech est présent dans 8 pays (France, Royaume-Uni, Allemagne, Suède, Danemark, États-Unis, Inde et Corée) avec des nombreuses références de mise en œuvre de l'Agilité, telles que : APEC, AFPA, Banque de France, Club Med, Crédit Agricole, Dassault Aviation, EDF, Mappy, Orange, Pernod Ricard, RTE, Société Générale, Taliance, Thales ...

REMERCIEMENTS



STEPHANE LABATI

// Responsable de l'offre Agile
// Valtech

Ce livre blanc est le fruit de la collaboration de la communauté Agile de Valtech.

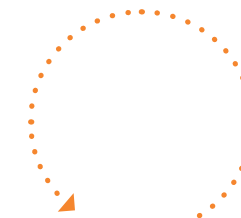
Cette nouvelle mouture enrichit l'édition 2008 avec de nombreux retours d'expérience à l'échelle d'organisations complètes. Elle témoigne de l'envie permanente des consultants de Valtech de partager leur expertise et leur savoir-faire sur les méthodes Agiles.

Merci en particulier à Etienne Charignon, Hélène Granboulan, Nathalie Lopez-Saussier, Thomas Beaugrand, Hubert Gillon, Elisabeth Ducarre, Stéphane Labati, Patrick Le Go, Parijat Sinha, Craig Larman, Jean-Claude Grosjean et Laurent Moulager, qui ont partagé leurs retours d'expérience Agiles dans les domaines aussi variés que les pratiques d'ingénierie, la gestion des exigences, les tests, le pilotage de projet, la conduite du changement et ses facteurs humains, la contractualisation Agile, la qualité logicielle, le lean, la conformité aux standards, l'offshore, l'Expérience Utilisateur ou la création graphique.

L'équipe éditoriale remercie également tous les relecteurs qui, forts de leurs propres expériences, ont largement contribué à améliorer la qualité de ce livre blanc.

Enfin, Valtech remercie ses clients qui lui ont fait confiance en l'associant à leurs défis, renforçant jour après jour son expertise et sa capacité à les aider dans l'adoption de l'Agilité.

SOMMAIRE



1. Les méthodes Agiles « pour les nuls »	11
1.1. Pourquoi adopter les méthodes Agiles ?	12
1.2. Qui est concerné ?	14
1.3. Quelles sont les pratiques Agiles les plus répandues ?	15
2. L'Agilité par la pratique	19
2.1. Témoignage d'un coach Agile Valtech	20
2.2. Étude d'opportunité	22
2.3. Recueil des besoins dans le Product Backlog	25
2.4. Test Driven Requirement : la spécification par l'exemple	28
2.5. TDD, le développement sous contrôle	32
2.6. Suivi et pilotage avec l'Iteration Backlog	34
2.7. Retour d'expérience sur la gestion de projet	36
2.8. Retour d'expérience sur l'automatisation des tests	40
2.9. Résultats obtenus sur des projets réalisés par Valtech	43
3. Le marketing digital Agile	45
3.1. La vision du produit	47
3.2. Personas, vous avez dit Personas ?	48
3.3. La démarche créative	49
3.4. Agilité et Expérience Utilisateur	56

4. La transformation vers l'Agilité	61
4.1. Enjeux et motivations	62
4.2. Le projet de transformation	63
4.3. Définir le projet : le cadrage	64
4.4. Expérimenter : Le pilote	70
4.5. Transformer : Déploiement et optimisation en continu	74
5. Les difficultés à surmonter	81
5.1. Les difficultés couramment rencontrées	82
5.2. La gestion du stress dans les équipes Agiles	84
5.3. La contractualisation Agile	86
5.4. L'externalisation Agile	90
5.5. L'Agilité face aux autres standards	97
5.6. Mettre de l'Agilité dans une démarche CMMI®	98
6. Glossaire des pratiques Agiles	105
6.1. Définitions	106
6.2. Abréviations	108
7. Références bibliographiques	109

SOMMAIRE

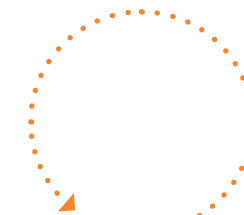


Table des figures

FIGURE 1	Exemple de Burndown Chart	16
FIGURE 2	Présentation schématique du processus de développement Agile basé sur Scrum	17
FIGURE 3	Exemple de pratiques d'ingénierie et de pilotage de projet	24
FIGURE 4	Exemple de Product Backlog	26
FIGURE 5	Gestion des priorités et de la complexité dans le Product Backlog	27
FIGURE 6	Exemple de Burndown Chart	35
FIGURE 7	Planning détaillé d'une itération	37
FIGURE 8	Planning détaillé d'une semaine	37
FIGURE 9	Planning détaillé d'une journée	38
FIGURE 10	Exemple de Persona	48
FIGURE 11	Les étapes d'une transformation Agile	63
FIGURE 12	Une équipe de projet en relation avec son écosystème	65
FIGURE 13	L'accompagnement dans la transformation Agile	72
FIGURE 14	Mesure de maturité Agile	73
FIGURE 15	Le plan de déploiement	74
FIGURE 16	Tableau de Scrum avec les différents types d'histoires utilisateur	92
FIGURE 17	Product Owner en train d'expliquer les nuances des fonctionnalités à l'équipe	94
FIGURE 18	Accueil d'un membre de l'équipe onshore par ses homologues offshore à Bangalore	95
FIGURE 19	Espace de travail ouvert entouré de tableaux blancs	96
FIGURE 20	Cycle de Deming PDCA (Plan, Do, Check, Act)	98
FIGURE 21	Les pratiques Agiles issues de XP et Scrum	106

Table des tableaux

TABLEAU 1	TDR - Données initiales	29
TABLEAU 2	TDR - Affaire versus Convention	30
TABLEAU 3	TDR - Affaire versus Convention	30
TABLEAU 4	TDR - Validation de convention	30
TABLEAU 5	Contexte projet	36
TABLEAU 6	Développement de la vision	65
TABLEAU 7	Les clés de la transformation Agile	68
TABLEAU 8	Critères de sélection d'un projet pilote	71
TABLEAU 9	Dispositif d'accompagnement pour un projet	72
TABLEAU 10	Ordre d'introduction des pratiques Agiles	73
TABLEAU 11	Dispositif d'accompagnement	75
TABLEAU 12	Références bibliographiques	110

1
....

LES MÉTHODES AGILES “ POUR LES NULS ”



OU EN QUOI CONSISTENT LES MÉTHODES AGILES
ET POURQUOI LES ADOPTER

1.1 POURQUOI ADOPTER LES MÉTHODES AGILES ?

- **Les méthodes Agiles consistent en un ensemble de pratiques imaginées pour pallier les difficultés rencontrées dans les cycles de développement en cascade ou en V, encore omniprésentes.**

Les méthodes traditionnelles prônent un enchaînement séquentiel des différentes activités, depuis les spécifications jusqu'à la validation du système, selon un planning préétabli. Elles visent à mieux prédire la façon dont les choses « devraient » se passer. Malheureusement, cette vision rassurante est bien loin de la réalité des projets. Les activités d'ingénierie ne sauraient se succéder strictement sans qu'aucun changement ne vienne perturber un planning qui n'a de durée de vie que le temps de le prononcer.

La conséquence est que plus de 80% des projets exécutés selon ces méthodologies connaissent des retards, des dépassements budgétaires, quand ils ne finissent pas en échec total, pour n'avoir pas su satisfaire les attentes des clients.

Ces problèmes sont liés à plusieurs caractéristiques fondamentales de ces anciennes méthodologies :

- **le rôle joué par le client** : le client intervient principalement au moment du lancement du projet, à quelques jalons majeurs parfois espacés de plusieurs mois, et surtout en fin de projet pour la réception et la recette du système. Cet « effet tunnel » conduit à une solution souvent inadaptée et de piètre qualité.
- **le mode contractuel forfaitaire** qui :
 - durcit les relations entre client et fournisseur,
 - rend le **passage de témoin long** et douloureux à la fin du projet.
- **une trop grande standardisation** des activités d'ingénierie, dont l'enchaînement se révèle souvent inefficace. Formellement, les contrôles d'avancement et de qualité ne peuvent être menés que sur la base de documents dans les premières étapes, et bien des organisations sont devenues des usines à produire de la documentation au lieu de produire de la valeur (fonctions logicielles) pour les clients et les utilisateurs.
- **le passage de relai** entre les phases successives dans lesquelles œuvrent des équipes différentes, généralise une relation de type client-fournisseur et n'encourage ni l'empathie ni l'esprit d'équipe, bien au contraire. Chaque transition se traduit par une perte de temps, de savoir, d'informations ou de responsabilité.

A contrario, les méthodes Agiles préconisent :



L'adoption d'un cycle itératif et incrémental

permettant à une équipe de s'adapter au contexte ainsi qu'aux changements qui ne manquent pas de survenir au cours d'un projet.



L'implication du client

dans le développement, permettant au client et à l'utilisateur de donner leur feedback quant au devenir de l'application en cours de développement, annulant ainsi tout « effet tunnel ».



La définition d'objectifs à court terme

qui permet de maintenir une pression constante mais supportable sur l'équipe, alors qu'au début d'un cycle en V chacun a l'impression d'avoir suffisamment de temps devant lui et subit finalement une pression énorme à l'approche de la livraison.



La collaboration entre les personnes et l'intégration des équipes

qui combat les fameux passages de relais en rassemblant dans un même espace toutes les énergies et la compétence de personnes centrées sur l'application à réaliser. Plus aucune barrière et des tâches définies par l'équipe au meilleur moment, c'est-à-dire quand on en a besoin, plutôt qu'au début du projet.



La livraison d'un produit opérationnel

de bonne qualité parce que souvent testé, doté de la seule documentation strictement nécessaire, et répondant à coup sûr aux vrais besoins des utilisateurs puisqu'il est régulièrement soumis à leur feedback.



HUBERT GILLON
// Delivery Manager
// Valtech

“ Le succès des projets Agiles renforce jour après jour l’engouement des DSI et des équipes informatiques pour des pratiques remettant l’application et l’homme au centre du sujet. Un projet n’est-il pas d’abord une aventure humaine vécue par des hommes pour d’autres hommes ? ”

1.2 QUI EST CONCERNÉ ?

Le Manifeste Agile propose 4 principes fondamentaux :

<p>PRIORITÉ DONNÉE AUX PERSONNES ET AUX INTERACTIONS</p> <p>plutôt qu’au processus et aux outils</p>	<p>PRIORITÉ DONNÉE À LA PRODUCTION DE FONCTIONS</p> <p>plutôt qu’à la documentation</p>
<p>PRIORITÉ DONNÉE À LA COLLABORATION AVEC LE CLIENT</p> <p>plutôt qu’à la négociation contractuelle</p>	<p>PRIORITÉ DONNÉE À L’ADAPTABILITÉ ET À L’ACCUEIL D’ÉVENTUELS CHANGEMENTS</p> <p>plutôt qu’au suivi d’un plan originel</p>

Forts de ces principes, on voit qu’une organisation, un département, une business unit, un projet et même une équipe peuvent adopter l’Agilité avec succès.

Mais qu’en est-il d’un projet déjà démarré ou en difficulté ? L’Agilité peut également dans ce cas améliorer les résultats déjà obtenus et faciliter la résolution de bon nombre des difficultés vécues. Elle va amener les personnes impliquées à :

- mieux collaborer,
- prendre du recul sur l’application en priorisant les actions et en remettant à plat le chiffrage initial,
- donner plus de visibilité aux clients et utilisateurs,
- éliminer « l’effet tunnel » induit par le cycle en V, en le remplaçant par des itérations courtes et maîtrisées.

“

L’idéal serait que toute l’organisation ait conscience de l’intérêt de fonctionner différemment et qu’elle mette dans sa stratégie l’adoption d’un ou plusieurs des principes Agiles.

”

HUBERT GILLON
// Delivery Manager
// Valtech

1.3 QUELLES SONT LES PRATIQUES AGILES LES PLUS RÉPANDUES ?

Il existe de nombreuses méthodes Agiles (XP, Crystal, FDD, Scrum...pour les plus connues) fondées sur les principes évoqués ci-dessus. Chaque méthode apporte son propre lot de techniques et de pratiques, les unes concernant plutôt le pilotage de projet, les autres, plutôt l’ingénierie.

Les pratiques Agiles les plus répandues sont issues de ces différentes méthodes:

- l’implication forte du client à travers le rôle de **Product Owner**,
- l’utilisation d’un **Product Backlog** pour gérer dynamiquement les fonctions du produit à réaliser, et les priorités métier associées ; le Product Backlog est élaboré en début de projet, et révisé autant que nécessaire,
- le **Scrum Meeting**, est une courte réunion quotidienne (environ 15’) qui rassemble tous les membres de l’équipe de développement. Cette réunion permet aux personnes d’échanger des informations sur l’avancement des tâches, signaler les problèmes rencontrés et demander de l’aide si nécessaire,
- le **Retrospective Meeting** est une réunion de fin d’itération, focalisée sur les événements survenus et l’analyse causale des dysfonctionnements, des pertes de productivité et de qualité. Un ou deux axes d’amélioration seront privilégiés de façon consensuelle et se traduiront par des tâches dans le backlog de l’itération suivante,
- l’**Iteration Planning**, en début d’itération, permet à l’ensemble de l’équipe projet de découvrir la liste des fonctions à implémenter, d’identifier et d’estimer les tâches de réalisation,
- la **Vélocité** est un indicateur qui mesure le « volume » de logiciel produit par l’équipe au cours d’une itération. Ce « volume » est estimé préalablement pour chaque fonction (ou User Story),
- le **Burndown Chart** est une représentation graphique de l’avancement des travaux au cours d’une itération : la courbe représente simplement le reste à faire (en charge) tel qu’il est estimé chaque jour par l’équipe. Le point initial représente l’effort total estimé pour l’itération pour l’ensemble de l’équipe, généralement en heures,
- l’**Intégration Continue** consiste à compiler, assembler, vérifier et tester l’ensemble du code source dès qu’un nouvel élément est mis à disposition, soit, idéalement, une à plusieurs fois par jour,
- le **Test Driven Development** (TDD) consiste à écrire les programmes de tests unitaires avant de programmer les fonctions elles-mêmes, puis d’adapter le code source testé unitairement jusqu’à obtenir un code de qualité (Refactoring),
- le **Test Driven Requirement** (TDR) permet de spécifier le logiciel par l’exemple – i.e. les exigences logicielles sont exprimées sous forme de cas de test,
- enfin le **Pair Programming** consiste à programmer en binôme dans le but d’être plus efficace en termes de conception, de revue de code et de transfert de compétences.

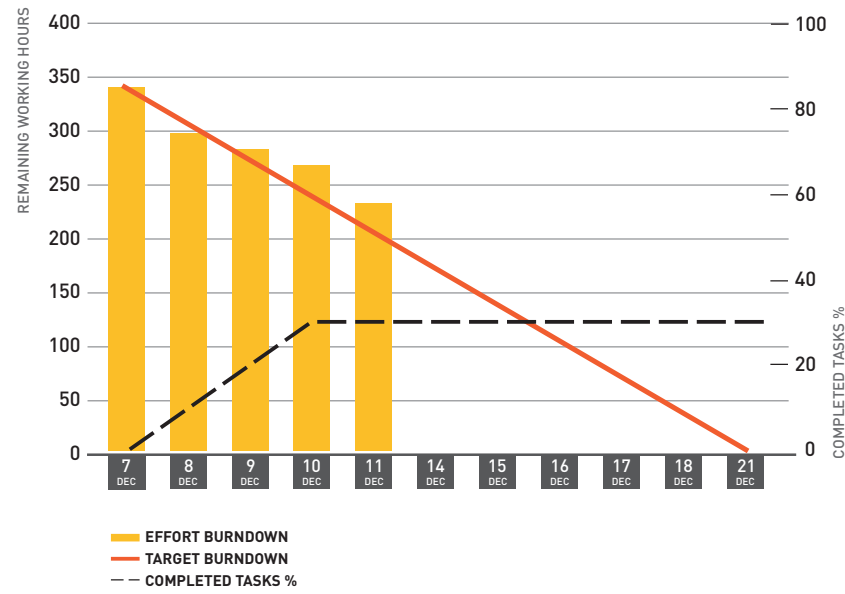


FIGURE 1 Exemple de Burndown Chart (Source Valtech)

L'ESSENTIEL À RETENIR

- Les méthodes Agiles préconisent l'adoption d'un cycle itératif et incrémental.
- L'Agilité prône la collaboration entre les personnes et l'intégration des équipes.
- Les méthodes Agiles mettent l'accent sur l'importance de développer le bon produit.

4 PRINCIPES FONDAMENTAUX :

- priorité aux personnes et aux interactions,
- priorité au développement des fonctions,
- priorité à la collaboration avec le client,
- accueil et adaptation au changement.

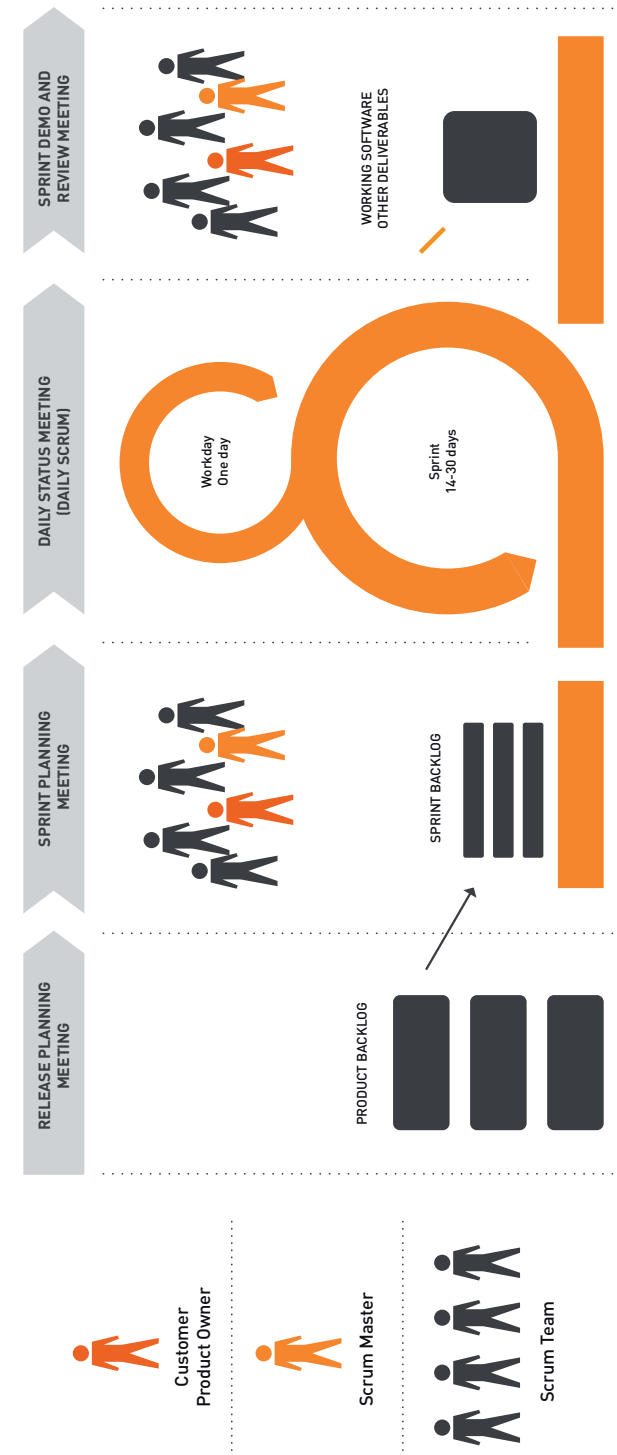


FIGURE 2 Présentation schématique du processus de développement Agile basé sur Scrum (Source Valtech)

2
....

L'AGILITÉ PAR LA PRATIQUE



OU LE RETOUR D'EXPÉRIENCE DE VALTECH

2.1 TÉMOIGNAGE D'UN COACH AGILE VALTECH

Agile, c'est pratique !

L'Agilité peut être vraiment appliquée au quotidien à l'occasion de la réalisation de projets informatiques. Ce n'est pas un doux rêve inatteignable et dans bien des cas, c'est même assez facile.



Les sceptiques vous diront peut-être que l'Agilité n'est qu'un concept, un rêve éloigné des réalités concrètes rencontrées au quotidien. N'entendons-nous pas souvent des « Oui, mais chez nous, ça n'est pas possible ! », ou des « Oui, mais dans la vraie vie, ce n'est pas comme ça ! » ? Ah, oui, mais pourtant, j'existe ! Avant de trouver des projets officiellement Agiles, j'ai fait pendant assez longtemps du développement logiciel Agile en « sous-marin » et, les pratiques que j'ai pu mettre en place ont toujours apporté énormément au projet, voire le succès.



ETIENNE CHARIGNON
// Consultant senior
// Certified Scrum Master
// Valtech

Les pratiques comme le TDD (Développement piloté par les tests : voir en annexe) ou les tests de recette automatisés sont faciles à mettre en place à condition qu'on veuille bien s'en donner la peine.

La qualité est gratuite à condition que l'on veuille bien en payer le prix. Les pratiques que vous voudrez mettre en place feront gagner du temps sur le long terme, mais coûteront quelque chose au début.

Spécifications incrémentales et TDR

Comme on pourrait s'y attendre, les pratiques de développement sont bien plus simples à mettre en place que celles qui font intervenir le client. Pour le faire - lors d'un projet sur lequel nous travaillons actuellement - nous avons planifié une itération zéro de mise en route d'une semaine, durant laquelle nous avons entre autres défini la liste des scénarios d'utilisation mais aussi mis en place FitNesse, un outil de TDR (Test Driven Requirement) basé sur un wiki. **Nous avons conseillé au client de ne plus détailler toutes ses exigences fonctionnelles avant de démarrer les développements.** A la place, nous lui avons demandé la liste des scénarios d'utilisation (le Product Backlog). Puis, pour chaque scénario, il a attribué une valeur métier notée entre 1 et 3. Nous en avons alors estimé la difficulté technique notée de 1 à 13 suivant la suite de Fibonacci. Un tel Product Backlog est ensuite plus facile à manipuler que directement une spécification détaillée.

Mais le travail de spécification ne s'arrête pas là. Au début de chaque itération, nous choisissons les scénarios qui doivent être réalisés pendant l'itération et en définissons les critères d'acceptation. C'est de cette manière que nous détaillons les spécifications. **Pour éviter le gaspillage, les détails ne sont pas prévus entièrement à l'avance** (pas de stock), mais seulement au moment où les développeurs sont prêts à les recevoir.

ETIENNE CHARIGNON

// Consultant senior
// Certified Scrum Master
// Valtech



Ce flux tendu de spécification constitue la « sève du processus Agile ».



« War room »

La première chose à faire est de réunir les gens dans un même lieu, dédié au projet. La mise en place d'autres pratiques s'en trouve grandement favorisée :

- pour suivre le projet, on utilise **les post-it® sur les murs**,
- pour faciliter le travail en binôme et la propriété collective du code, il est préférable d'avoir un ensemble d'ordinateurs non affectés individuellement, mais dédiés à un type de tâche : développement, bureautique... De plus, il est indispensable que les postes de développement soient homogènes.

Dès que vous aurez mis en place tous ces éléments, vous aurez votre « war room ». Il est évident qu'il est plus difficile de le faire lorsque les développeurs sont éparpillés dans un open space.

Serveur d'intégration continue

Votre « war room » peut contenir une **machine dédiée à l'intégration continue**, car il est assez facile de libérer une machine pour la dédier à cette activité étant donné que les développeurs travaillent chaque fois que nécessaire en binôme.

En quelques minutes, vous installez un logiciel comme Hudson ou CruiseControl. Ce type de logiciel est capable d'aller lui-même chercher le code source dans le dépôt de votre système de gestion de version (par exemple Subversion ou ClearCase), de le compiler et d'exécuter une série de tests et de mesures de qualité avec des outils tels que CheckStyle ou Cobertura.

Il reste ensuite à **astreindre plusieurs fois par jours les développeurs à poster leur travail sur le dépôt central** (commit du code dans l'outil de gestion des sources).

2.2 ÉTUDE D'OPPORTUNITÉ

L'étude d'opportunité de l'adoption de l'Agilité par une organisation est le moyen idéal pour s'initier en douceur au monde de l'Agilité. Basée sur l'écoute et l'échange, cette étude permet d'apporter une solution personnalisée et adaptée aux attentes et besoins du client.

L'objectif de cette étude n'est pas de faire un audit projet, mais plutôt d'identifier les **pertes d'énergie** et les éventuels **effets d'inertie** de l'organisation projet, ainsi que d'identifier de **nouvelles pratiques plus efficaces**.

Les étapes de cette étude sont les suivantes :

- état des lieux des pratiques en cours,
- rédaction d'un document de synthèse sur l'existant,
- adoption de pratiques adaptées au contexte du client,
- rédaction et soutenance des pratiques retenues.

État des lieux des pratiques en cours

Cette étape se déroule sous forme d'**entretiens** et de **séances de travail** qui visent à établir la **cartographie des pratiques en cours** auprès d'un échantillon de personnes intervenant sur tout le cycle de vie d'un projet (maîtrise d'ouvrage, maîtrise d'œuvre, cellule qualité, formateurs...).

Cette collecte d'informations est organisée par type d'activité projet telles que :

- le recueil du besoin,
- la gestion de projet,
- le transfert de connaissances,
- les spécifications logicielles (dossier d'analyse),
- la conception et l'implémentation,
- le test logiciel,
- la qualité logicielle,
- le déploiement et la mise en production.

Rédaction d'un document de synthèse sur l'existant

Le document de synthèse des interviews a pour but de mettre en évidence de façon objective et anonyme les différentes remontées d'informations effectuées lors de l'étape précédente.

La synthèse contient:

- la description des périmètres technique, fonctionnel et organisationnel du projet ou du service, candidat à l'Agilité,
- la description des rôles et responsabilités de chaque personne interviewée au sein de l'organisation projet ou service,
- la description de l'état des lieux de chacune des activités précédemment citées,
- la liste des accélérateurs éventuels à l'adoption de l'Agilité,
- l'identification des freins éventuels,
- les incontournables manquants (pratiques indispensables et pourtant absentes).

Ce document sert de base à une nouvelle séance de travail au cours de laquelle **les freins et incontournables manquants sont analysés en détail**. A ce stade, d'autres séances de travail plus ciblées sont réalisées pour **identifier les pratiques les plus utiles**, en s'appuyant sur un catalogue de pratiques Agiles.



NATHALIE LOPEZ-SAUSSIER
// Directeur Général Adjoint
// Valtech Technology

“

Il est important de présenter les pratiques pressenties à l'ensemble des acteurs et de les décrire. Mais il faut également discuter des impacts possibles sur l'organisation afin qu'un sous-ensemble de pratiques puisse être retenu par le client.

”

Adoption de nouvelles pratiques

Les pratiques identifiées précédemment sont synthétisées dans un document où :

- les risques et incontournables manquants sont rappelés avec les pratiques Agiles associées,
- les conditions d'entrée pour la mise en place de toutes les pratiques sont identifiées,
- la description et le mode opératoire de chaque pratique sont détaillés,
- les éventuels Artefacts produits par chaque pratique sont identifiés,
- les conséquences et impacts sur l'organisation et les processus actuels sont décrits,
- les attendus opérationnels de la pratique sont décrits.

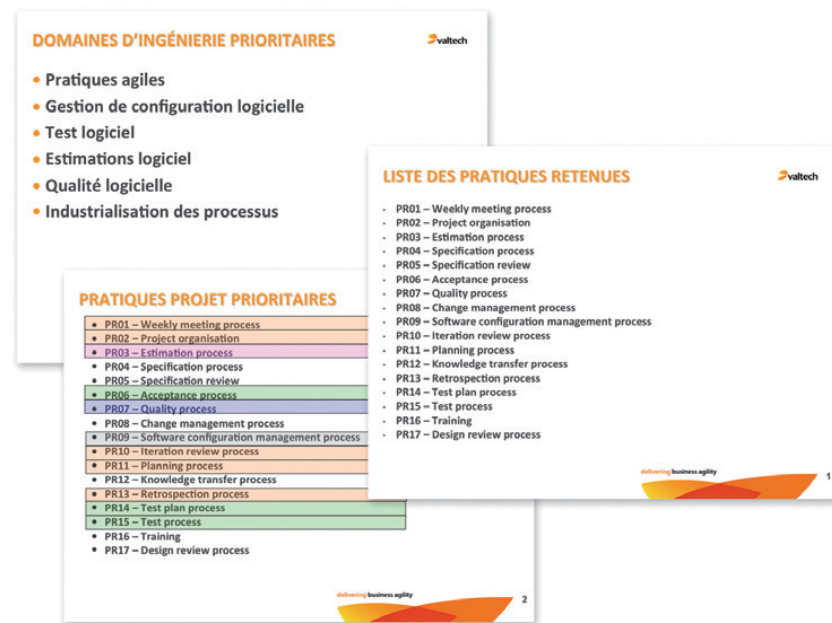


FIGURE 3 Exemple de pratiques d'ingénierie et de pilotage de projet (Source : Valtech)

Rédaction d'un document de synthèse sur les pratiques retenues

Valtech produit le document final de l'étude synthétisant les attendus du client, la démarche, et enfin les pratiques retenues. Un plan d'action et une proposition de planning finalisent le document qui est présenté à l'ensemble des acteurs impliqués dans l'étude.

A l'issue de la soutenance, Valtech propose au client, au choix :

- une prestation de production d'Artefacts issus des pratiques Agiles retenues,
- une mission d'accompagnement dans la mise en œuvre des nouvelles pratiques Agiles sur le projet,
- une formation à l'Agilité.

L'ESSENTIEL À RETENIR

- L'objectif de l'étude d'opportunité est d'identifier les pertes d'énergies, les éventuels effets d'inertie d'une organisation projet ou d'un service ainsi que de nouvelles pratiques plus efficaces et plus Agiles.
- L'étude d'opportunité est basée sur des interviews et des workshops qui permettent de définir la cartographie des pratiques en cours. Cette étude permet ensuite d'identifier les freins à l'adoption de l'Agilité, les incontournables manquants et les pratiques Agiles les plus utiles.
- Un plan d'action et une proposition de planning finalisent l'étude d'opportunité.

2.3 RECUEIL DES BESOINS DANS LE PRODUCT BACKLOG

Le manque de visibilité sur le contenu final probable d'un logiciel est préjudiciable au client mais également aux équipes projet.

Le Product Backlog contient cette description et permet entre autres :

- d'avoir une vision commune sur l'ensemble des fonctions ou cas d'utilisation définissant le périmètre du logiciel à développer,
- de comprendre l'intérêt et les enjeux des développements pour les utilisateurs (appelés également acteurs),
- d'estimer l'avancement du projet sur la base des fonctions ou cas d'utilisation livrés au client,
- de réaliser facilement des macro-estimations en utilisant, par exemple, la méthode des Use Case points,
- de préparer l'identification des tâches du projet en les organisant autour des fonctions ou des cas d'utilisation.

Un Product Backlog a été élaboré, par exemple, chez un de nos clients dans le domaine de l'assurance, pour formaliser de façon synthétique l'ensemble des fonctions attendues par les courtiers, et pour estimer la charge de réalisation de l'application.

La figure suivante présente un extrait du tableau obtenu après deux jours de travail avec la maîtrise d'ouvrage et la maîtrise d'œuvre, ainsi qu'avec deux courtiers en assurance, futurs utilisateurs de l'application.

CAPABILITIES		FEATURES			
Capa.	Functional Capabilities	Lot	Feat.	Features	Actors
C1	Quotation management (Lot 1)	1	F1-1	Quotation creation from scratch	User
		1	F1-2	Quotation creation from an existing version of quotation (same year)	User
		1	F1-3	Quotation creation from an existing version of quotation (previous year)	User
		1	F1-4	Quotation creation from a Petrus Program	User
C2	Date import (Lot 2)	2	F2-1	Select fields to import from loss files	User
		2	F2-2	Import Loss with developments - vertical	User
		2	F2-3	Import Loss with developments - horizontal	User
		2	F2-4	Import Loss files without developments - vertical	User
		2	F2-5	Import the LDFs	User
		2	F2-6	Import the indexes from the index database -European	User
		2	F2-7	Import the indexes from the index database -US	User

FIGURE 4 Exemple de Product Backlog (Source : Valtech)

Les fonctionnalités de haut niveau (Functional Capabilities) de la future application sont des regroupements de fonctions (User Features). Chaque fonction est identifiée de façon unique pour pouvoir ensuite être facilement référencée. Un type d'utilisateur est attribué à chaque fonction.

A la fin de cet exercice, les fonctionnalités de haut niveau sont groupées en lots de livraison.

Une fois ce premier travail réalisé, une priorité de développement a été déterminée et affectée à chaque fonction. Elle est calculée à partir des estimations de complexité et de valeur métier (classées en « High », « Medium » and « Low »). Cette démarche permet de limiter les risques en traitant en priorité les fonctions les plus complexes, et en majorant le ROI (Return on Investment), en livrant d'abord les fonctions apportant le plus de valeur aux utilisateurs. L'extrait du tableau suivant donne une idée du résultat.

FEATURES			PRIORITY			
Feat.	Features	Actors	Complexity	Business Value	Priority	UCP
F1-1	Quotation creation from scratch	User	M	H	Medium	10
F1-2	Quotation creation from an existing version of quotation (same year)	User				
F1-3	Quotation creation from an existing version of quotation (previous year)	User				
F1-4	Quotation creation from a Petrus Program	User				
F2-1	Select fields to import from loss files	User	H	L	Medium	15
F2-2	Import Loss with developments - vertical	User				
F2-3	Import Loss with developments - horizontal	User				
F2-4	Import Loss files without developments - vertical	User				

FIGURE 5 Gestion des priorités et de la complexité dans le Product Backlog (Source : Valtech)

Un nombre de Use Case Points (UCP) a été attribué aux fonctions en raison de leur complexité, ceci afin de servir de base à une estimation globale du projet.



« Finalement, nous avons été capables, en moins de 5 jours, de décrire les besoins client sous la forme d'un Product Backlog, d'identifier l'ensemble des acteurs au sens UML, de définir les priorités d'implémentation, d'estimer la complexité des cas d'utilisation et de chiffrer le projet dont la taille représentait 3.000 hommes-jours. »



HÉLÈNE GRANBOULAN
// Analyste senior
// Valtech

L'ESSENTIEL À RETENIR

- Le manque de visibilité sur le contenu final probable d'un logiciel est préjudiciable au client mais également aux équipes projet.
- L'utilisation du Product Backlog se révèle très efficace à condition d'en maîtriser les concepts et de disposer des personnes compétentes et habilitées à prendre les décisions impactant le futur projet.

2.4 TEST DRIVEN REQUIREMENT : LA SPÉCIFICATION PAR L'EXEMPLE

Le Test Driven Requirement (TDR) est fondé sur le constat que dans bien des cas, il est plus efficace d'expliquer un comportement en décrivant des exemples, plutôt qu'en réalisant une spécification « classique » qui décrit les mécanismes à implémenter.

Le Test Driven Requirement, ou spécification dirigée par les tests, propose :

- de centrer la description et la rédaction des besoins utilisateurs sur des exemples qui constitueront autant de futurs cas de tests de validation,
- de centrer la collaboration entre les équipes du projet sur la compréhension et l'enrichissement de ces exemples.

Le contexte documentaire et collaboratif

Pour un des leaders de services en ressources humaines, Valtech a mis en place la pratique TDR pour un logiciel de gestion qui avait été spécifié en UML. La stratégie de développement avait été axée sur l'efficacité à court terme plutôt que sur la maintenabilité. Les activités de projet étaient confiées à des équipes distinctes (développement / homologation / analyse-gestion de projet / MOA).

Après 6 mois d'exploitation, le client a formulé les remarques suivantes:

- + Les fonctions majeures du logiciel étaient opérationnelles.
- Chaque équipe disposait de ses propres documents, relatifs à son activité, sans les partager avec les autres équipes.
- Les évolutions apportées après la mise en production étaient réalisées sans être spécifiées, engendrant un manque de visibilité sur le contenu fonctionnel réel du logiciel et rendant difficile l'intégration de nouveaux besoins.
- Le périmètre de test déduit des spécifications était incohérent avec les fonctionnalités déjà implémentées.
- L'homologation était de moins en moins souvent réalisée du fait de la difficulté à produire les cas de tests, conduisant ainsi à une baisse de qualité.
- La grande autonomie de chaque équipe se faisait au détriment de la communication.

C'est dans un tel contexte que Valtech a mis en place la pratique du TDR.

Les enjeux du client

Compte tenu du constat réalisé pour parvenir à maîtriser les évolutions logicielles, il a été décidé de modifier les pratiques de spécifications.

Valtech a amené le client à privilégier la description concrète plutôt que la modélisation en incluant des exemples tels que ceux présentés ci-après.

Exemple de spécification TDR

Contexte de l'évolution

Il s'agit de modifier le cycle de vie d'une affaire afin de mettre à jour son état lorsqu'une convention est créée, sans être validée.

Jusqu'à présent, c'est seulement lorsque la convention est validée que l'état de l'affaire est mis à jour.

N.B. Une affaire est une opportunité commerciale détectée pour un compte donné, mais n'aboutissant pas immédiatement à la signature d'une convention avec ce compte. Lorsque l'opportunité commerciale se concrétise, l'affaire est transformée en convention. La convention est définitive à partir du moment où elle est validée. Cette évolution met donc en évidence à la fois le cycle de vie des affaires, celui des conventions et celui des comptes.

Les données initiales

Les « Comptes » et « Affaires » suivants ont été créés :

COMPTES				
Nom du Compte	Matricule Responsable	Entité Responsable	État du Compte	Identifiant du Compte
BONJOUR	0001	E1	1 (prospect sur affaire)	COMPT 01

AFFAIRES				
Nom de l'Affaire	Matricule Responsable	Identifiant du Compte	État du Compte	Commentaire
Affaire A	0002	COMPT 01	1 (ouverte)	Null

TABLEAU 1 TDR - Données initiales (Source : Valtech)

L'affaire est créée dans l'état par défaut « Ouverte » tandis que le compte est créé dans l'état « Prospect sur affaire ».

Transformer une affaire en convention

L'action consiste pour le responsable (matricule 0002 lié à l'affaire A) à transformer l'affaire A en convention.

AFFAIRES – RÉSULTAT				
Nom de l'Affaire	Matricule responsable	Entité responsable	État du compte	Identifiant du Compte
Affaire A	0002	E1	2 (fermée)	Transformée en convention

TABLEAU 2 TDR - Affaire versus Convention (Source : Valtech)

L'affaire est fermée car transformée en convention. Elle ne peut plus être utilisée pour créer une convention.

COMPTES – RÉSULTAT				
Nom du Compte	Matricule Responsable	Entité Responsable	État du Compte	Identifiant du Compte
BONJOUR	0001	E1	1 (prospect sur affaire)	COMPT 01

CONVENTIONS - RÉSULTAT			
Nom de la Convention	Matricule Responsable	État du Contrat	Nom de l'Affaire
Convention issue de l'Affaire A	0002	4 (brouillon)	Affaire A

TABLEAU 3 TDR - Affaire versus Convention (Source : Valtech)

La convention n'est pas encore validée d'où sa création dans l'état « brouillon ». L'état du compte reste inchangé.

Valider une convention

L'action consiste pour le responsable (matricule 0002) à valider la convention.

COMPTES – RÉSULTAT				
Nom du Compte	Matricule Responsable	Entité Responsable	État du Compte	Identifiant du Compte
BONJOUR	0001	E1	2 (client)	COMPT 01

CONVENTIONS - RÉSULTAT			
Nom de la Convention	Matricule Responsable	État du Contrat	Nom de l'Affaire
Convention issue de l'Affaire A	0002	1 (en cours)	Affaire A

TABLEAU 4 TDR - Validation de convention (Source : Valtech)

La validation de la convention entraîne donc :

- le changement d'état du compte qui passe de « prospect sur affaire » à « client »,
- le passage de la convention de l'état « brouillon » à « en cours ».

Une double « nouveauté » : collaboration et format des exigences

Dorénavant, la description des fonctionnalités est affinée de façon collaborative tout au long du processus de développement :

- initiation par la MOA,
- enrichissement par les analystes,
- mise à jour au fil des remarques des équipes de développement et d'homologation.

Cette description s'appuie sur des cas d'exemples qui sont :

- les cas de tests des scénarios standards par la MOA,
- les cas de tests des scénarios d'exception (sans viser l'exhaustivité mais plutôt la vraisemblance) par l'homologation.

Les tableaux utilisés pas à pas dans notre exemple, suite à différentes actions opérateur, permettent de visualiser concrètement les changements d'état successifs. Ils facilitent à la fois la compréhension du fonctionnel mais également l'identification des scénarios de test les plus pertinents.

Finalement, l'ensemble des équipes projet a adhéré à la nouvelle approche TDR. Cette adhésion a été favorisée par le fait que les cas de tests décrits sous forme de tableaux étaient lisibles par des non-informaticiens.

Une expérience riche d'enseignements

Pour en faciliter l'appropriation par les équipes, la méthode TDR a été introduite progressivement sans la nommer, en incluant des exemples dans les spécifications existantes.

La démarche TDR est à l'origine des avancées suivantes :

- L'équipe de développement a pris le réflexe d'enrichir les exemples fournis dans la spécification TDR. Ces exemples ont été utilisés en intégration continue et en homologation.

- Une grande partie des ambiguïtés dans la description des règles métier est désormais levée avant le début des développements.
- Le besoin se stabilise de plus en plus tôt dans le cycle de création d'une fonctionnalité.
- Les erreurs de description ou d'implémentation sont détectées plus tôt et sont donc plus faciles à résoudre.
- L'homologation se déroule désormais normalement.

HÉLÈNE GRANBOULAN

// Analyste senior
// Valtech

“

La formalisation des spécifications selon l'approche TDR permet de produire facilement des spécifications exécutables. Couplé à un outil tel que FIT, FitNesse ou GreenPepper, chaque exemple devient ainsi un cas de test automatique.

”

L'ESSENTIEL À RETENIR

Le Test Driven Requirement (TDR) propose de :

- centrer la description et la rédaction des besoins utilisateurs sur des exemples,
- centrer la collaboration entre les équipes du projet sur la compréhension et l'enrichissement de ces exemples,
- privilégier la description concrète plutôt que la modélisation dans une démarche TDR,
- affiner la description des fonctionnalités de façon collaborative tout au long du processus de développement,
- utiliser les tableaux pas à pas, suite à différentes actions opérateur, permet de visualiser concrètement les changements d'état successifs. Ils facilitent à la fois la compréhension du fonctionnel mais également l'identification des scénarii de tests les plus pertinents.

Le projet de 3 hommes.an dont il s'agit concerne le développement d'une application Java « stand alone » avec une interface Swing et diverses autres parties écrites en C++.

Enjeux client

L'objectif du client consiste à améliorer la qualité et la sécurité des applications sans augmenter les coûts de développement.

Pratiques mises en œuvre

Valtech a aidé le client à mettre en place :

- une approche de développement dirigée par les tests (TDD),
- une démarche d'automatisation des tests de recette.

Difficultés rencontrées

Le projet de développement a suivi le cycle classique en V. Valtech est intervenu à partir de la phase de développement (le bas du cycle en V) et a ainsi hérité d'un document de spécification, fruit de plus d'un an de travail. Il a, dès lors, été impossible de faire réaliser les tests par le client.

Certaines parties de l'application étant développées en Java et d'autres en C++, deux environnements de développement différents et deux outils de tests unitaires ont été utilisés - Eclipse et Visual Studio d'une part, JUnit et CPPUnit d'autre part. Cela a induit un double effort de mise en place des frameworks de tests unitaires.

La couverture des tests unitaires sur la partie C++ s'est avérée difficile à calculer.

Solutions apportées

Test Driven Development

Les composants écrits en C++ étant des bibliothèques utilisées par le code Java et JUnit étant plus facile à utiliser, le code C++ a majoritairement été testé depuis l'environnement Java avec JUnit. Malgré tout, certains tests ont dû rester dans la partie C++, de manière à garder un feedback rapide.

Tests de recette automatisés

L'équipe de développement a écrit les tests de recette, au moyen d'une bibliothèque externe uispec4j, permettant de « scripter » des scénarios d'utilisation de l'interface graphique Swing. Par ailleurs, un simulateur sous MS Windows a permis de simuler les interactions de l'application avec un équipement propriétaire. L'outil Autolt a été utilisé pour piloter ce simulateur depuis la suite de tests en Java.

2.5 TDD, LE DÉVELOPPEMENT SOUS CONTRÔLE

Contexte

Le contexte s'annonce a priori défavorable, voire hostile à l'Agilité : la méthodologie interne prône le cycle en V, totalement ancré dans la culture industrielle de l'entreprise.

Bénéfices obtenus

ETIENNE CHARIGNON

// Consultant senior
// Certified Scrum Master
// Valtech



Le projet a été livré le jour prévu, sans augmentation des coûts. L'équipe de développement s'est montrée très flexible vis à vis des diverses modifications de spécification, le tout sans perte de qualité ni apparition de régression.



L'ESSENTIEL À RETENIR

La mise en œuvre des pratiques Agiles de TDD permet de :

- rester maître de la complexité des développements réalisés,
- livrer dans les temps et le budget impartis.

2.6 SUIVI ET PILOTAGE AVEC L'ITERATION BACKLOG

L'Iteration Backlog a pour vocation de contenir l'ensemble des tâches identifiées et estimées par l'équipe de projet pour l'itération en cours. Grâce à l'estimation initiale et au « reste-à-faire » estimé quotidiennement, une représentation graphique de l'avancement est disponible en permanence (Iteration Burndown Chart) Sur les projets Valtech, les tâches sont associées à des fonctions ou à des scénarios de cas d'utilisation.

- Les tâches sont identifiées pour prendre en compte de nouvelles priorités de livraison et pour minimiser le nombre d'itérations nécessaires à la livraison d'une fonction ou d'un cas d'utilisation. La charge associée à une tâche est généralement comprise entre 4 et 16 heures.
- Chaque tâche est décrite par les informations suivantes :
 - **identifiant** unique (pour la traçabilité avec les fonctions ou cas d'utilisation),
 - **nom** explicite non générique mais spécifique au contexte et au sujet traité,
 - **identification** du membre de l'équipe qui s'est engagé à la réaliser,
 - **effort** estimé en heures par l'équipe lors du planning d'itération (exercice collégial de planification d'itération),

ETIENNE CHARIGNON

// Consultant senior
// Certified Scrum Master
// Valtech



Le Burndown Chart est un outil très puissant pour maîtriser l'avancement des travaux réalisés par une équipe sur une période courte dont les objectifs ont été exprimés en tâches à réaliser.



- Chaque membre de l'équipe projet renseigne quotidiennement le « reste-à-faire » pour les tâches dont il a la charge¹.
- Le **Burndown chart** est mis automatiquement à jour, imprimé et affiché dans l'espace de travail de l'équipe, il est également accessible à distance via un wiki (y compris par le client). Il présente l'effort restant à accomplir en heures, pour finir les tâches allouées à l'itération, le pourcentage de tâches terminées et la courbe idéale de « reste-à-faire ».

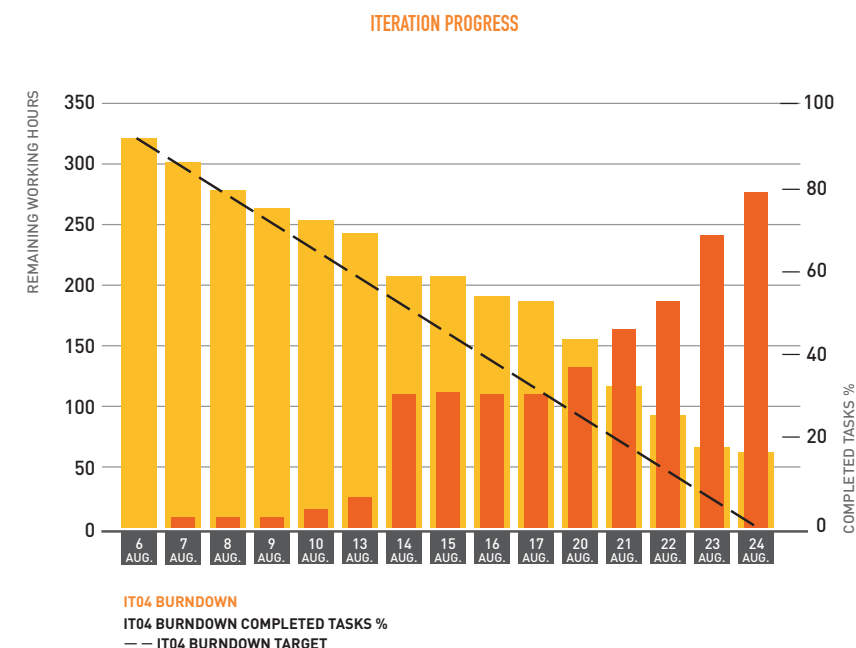


FIGURE 6 Exemple de Burndown Chart (Source Valtech)

En fin d'itération, les métriques suivantes sont relevées et communiquées à l'équipe ainsi qu'au client :

- le **pourcentage du périmètre de l'itération réalisé**. Il est calculé par le rapport entre la taille du logiciel qui devait être livré (poids Fibonacci) et la taille livrée réellement,
- la **Vélocité** de l'itération, c'est à dire le cumul du nombre de points de chaque fonction terminée (codage, test, etc.), et livrée (démontrée, acceptée par le client, prête pour un déploiement éventuel).

¹. Une alternative consiste à nommer chaque semaine et à tour de rôle, un time tracker qui relève ces métriques pour l'ensemble de l'équipe.

L'ESSENTIEL À RETENIR

- L'Iteration Backlog a pour vocation de gérer à court terme l'ensemble des tâches identifiées et estimées par l'équipe de projet sur chaque itération. La mise en place de mesures pertinentes permet, jour après jour, de suivre l'avancement réel et de piloter le projet en conséquence.

2.7 RETOUR D'EXPÉRIENCE SUR LA GESTION DE PROJET

Contexte

Développement d'une application de gestion pour le suivi et la traçabilité des processus de fabrication pour un industriel français de l'aviation.

MODE DE DÉVELOPPEMENT	Duoshore avec une équipe de 5 analystes Onshore sur le site du client et 25 développeurs offshore dans notre centre de développement de Bangalore (Inde).
TAILLE DU PROJET	9 000 hommes.jour
DURÉE DU PROJET	2 ans
Outils utilisés	WSAD, Wiki Confluence, Jira

TABLEAU 5 Contexte projet (Source Valtech)

Enjeux client

Les enjeux pour cette nouvelle application sont :

- offrir un outil souple, robuste et facile à déployer,
- faciliter le travail des utilisateurs par une interface intuitive et simple d'utilisation,
- apporter une solution permettant de mieux maîtriser la traçabilité des procédés de fabrication,
- accroître l'interopérabilité du système avec les autres applications de gestion.

Pratiques mises en œuvre

Les pratiques mises en œuvre sont :

- utilisation de la méthode d'organisation et de suivi Scrum sur les parties France et Inde (2 équipes Scrum),
- mise en place d'un Product Backlog commun aux deux équipes,
- utilisation d'un outil collaboratif (wiki) pour la communication entre les équipes de développement et le client,
- mise en place d'un mode de livraison Inde / France basé sur de l'intégration continue (cf. figure 8). Les différents points de synchronisation France / Inde et MOA / MOE y sont identifiés ci-après.

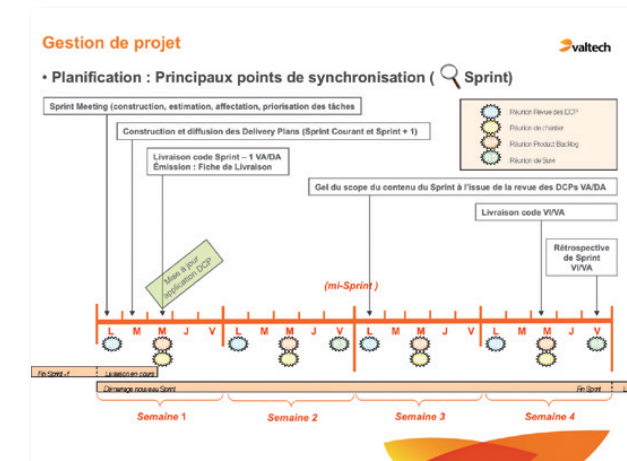


FIGURE 7 Planning détaillé d'une itération (Source Valtech)

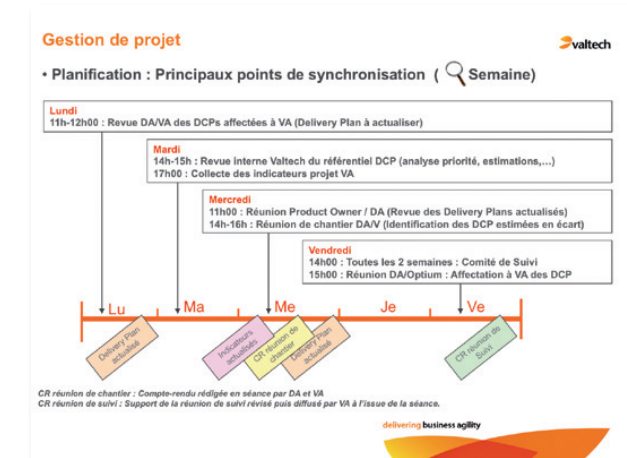


FIGURE 8 Planning détaillé d'une semaine (Source Valtech)

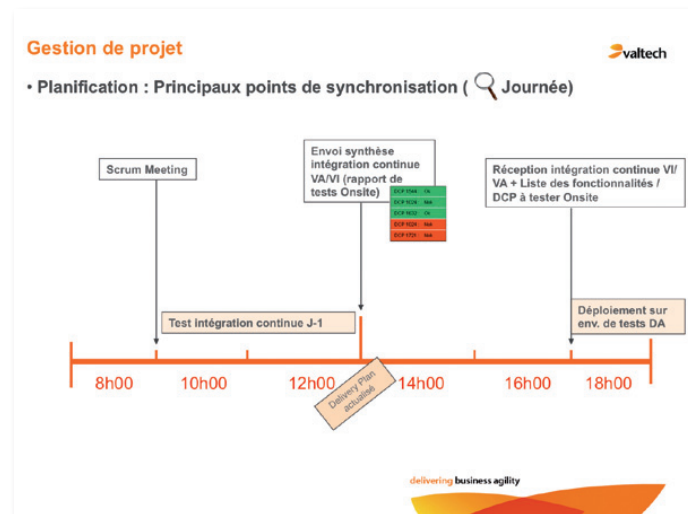


FIGURE 9 Planning détaillé d'une journée (Source Valtech)

Principe de répartition des responsabilités de la maîtrise d'œuvre :

- Onshore (France) :
 - relation avec le client,
 - recueil des besoins,
 - formalisation de documents d'analyse,
 - suivi de la validation de ces documents,
 - transfert de connaissance vers les équipes offshore,
 - support fonctionnel aux équipes offshore,
 - mise en place et contrôle des directives d'architecture du projet,
 - contrôle des flux de traduction Français-Anglais (documents d'analyse) et Anglais-Français (document de conception),
 - développement de fonctionnalités qui ne peuvent être délocalisées,
 - vérifications des scénarios de test,
 - coordination globale du projet.

- Offshore (Inde) :
 - formalisation des documents de conception,
 - travaux de développement,
 - élaboration des scénarios de tests,
 - automatisation des tests,
 - exécution des tests manuels et automatiques,
 - exécution des contrôles de qualité du code (PMD, RSA et FindBugs).

Difficultés rencontrées

Les difficultés rencontrées sont :

- la contrainte d'entrée sur le site et la planification longtemps à l'avance ne permettent pas de faire intervenir des ressources offshore sur le site client,
- l'obligation de planifier les réunions et les workshops bien en amont,
- la clôture de la documentation Word pour le client : en contradiction avec l'approche Wiki qui privilégie l'accès en ligne pour tous,
- la barrière de la langue (l'anglais) pour la maîtrise d'ouvrage,
- le souhait du client de raccourcir la prise en compte des demandes de changement d'une itération à l'autre,
- le cycle initial de validation des livrables documentaires est trop lourd et pas du tout adapté à l'approche itérative,
- les équipes onshore et MOA ne communiquent que par mails.

Solutions apportées

Les solutions apportées sont :

- face à l'impossibilité de traiter les évolutions : **la mise en place d'un volant de jours** pour traiter les évolutions : système d'enveloppe,
- pour réduire le nombre d'anomalies : **mise en place d'un circuit d'intégration continue** entre les développements réalisés en Inde et la plateforme d'intégration sur le site du client,
- l'accélération de la prise en compte des demandes client : **identification d'une provision de charges** pour traiter les demandes de changement en cours d'itération (jusqu'à la mi-itération),
- la réduction du backlog d'anomalies : **constitution d'une provision de charges** pour laisser le temps à l'équipe en Inde de corriger ses anomalies (interne / client) tout en produisant du fonctionnel,
- le rapprochement physique des équipes onshore avec la maîtrise d'ouvrage.

Bénéfices obtenus

Les bénéfices obtenus sont :

- le volant de jours : **fin des tensions** concernant la qualification des anomalies / évolutions,
- **la confiance accrue du client** en la qualité des développements,
- **la visibilité totale sur le contenu du produit et des itérations** qui permet de planifier à l'avance les avenants contractuels pour traiter les nouvelles fonctionnalités majeures,

- le respect des engagements vis-à-vis des autres équipes impliquées dans le processus d'ingénierie (autres équipes de développement, intégration, qualification, tests de performances...),
- réduction du nombre d'anomalies grâce à une meilleure compréhension du fonctionnel par les équipes indiennes et à la découverte au plus tôt des anomalies (intégration continue),
- nouvelles prises de commandes.

L'ESSENTIEL À RETENIR

- Les pratiques Agiles peuvent être appliquées dans un environnement non Agile même si celui-ci a de fortes contraintes. Il faut adapter ces méthodes au contexte du client et les adapter progressivement : une éducation préalable montrant les bénéfices attendus et une mise en évidence régulière des gains obtenus sont impératives.
- Les pratiques Agiles choisies et mises en place avec justesse permettent de préserver un des facteurs clés du succès du projet : la bonne collaboration entre les équipes du fournisseur et celles du client. N'oublions pas que cette collaboration entre équipes est au centre des pratiques Agiles !

2.8 RETOUR D'EXPÉRIENCE SUR L'AUTOMATISATION DES TESTS

Contexte

L'Agilité permet de nombreuses livraisons intermédiaires. Chacune de ces livraisons doit être intégralement testée ; dans le cas contraire, la perte de contrôle sur la qualité croît de manière exponentielle à chaque itération.

Un département de la DSI d'une grande banque de finance souhaite mettre en place l'Agilité au sein de ses équipes de développement. Le département a la responsabilité du développement et de la maintenance d'un ensemble d'applications avec des technologies hétérogènes. Toutes les applications sont déployées en production simultanément, trimestriellement. Chaque déploiement en production est précédé d'une campagne de tests utilisateurs de 2 à 3 semaines visant à s'assurer de l'absence de régression dans l'ensemble des applications.

Enjeux client

La durée des campagnes de tests de non-régression risque de masquer l'effet positif du déploiement des méthodes Agiles. Il est donc impératif de raccourcir la durée des campagnes et d'en augmenter la fréquence.

Pratiques mises en œuvre

Les pratiques mises en œuvre sont :

- automatisation des tests utilisateurs de non-régression (tests fonctionnels sollicitant l'interface graphique),
- mise en place de tests automatisés dans le processus d'intégration continue sur tous les niveaux de tests, depuis les tests unitaires jusqu'aux tests utilisateurs. Les niveaux de tests sont déclenchés à différentes étapes du cycle de construction (build) pour offrir un niveau de réactivité optimal aux équipes de développement : échec d'intégration détecté en moins de 2 min, échec de tests unitaires en moins de 5 min, échec de tests fonctionnels en moins d'une heure.

Difficultés rencontrées

- Les tests utilisateurs de non-régression, ceux qui sollicitent l'interface graphique, sont généralement coûteux à automatiser et à maintenir. La moindre modification d'une interface peut conduire à l'échec d'un script automatisé, même si les services sous-jacents n'ont pas été modifiés : les équipes de développement n'ont pas toujours conscience de cette contrainte et introduisent des modifications sur l'interface, sans reporter cette modification dans les scripts de tests automatisés,
- les tests sont automatisés trop tôt sur des pans fonctionnels non stables, ce qui en général entraîne des coûts prohibitifs de maintenance des scripts de tests,
- l'interface graphique utilise des composants qui se prêtent difficilement à l'automatisation des tests. Ceci est d'autant plus vrai que certains composants proviennent d'éditeurs qui ne fournissent pas le procédé de test de leurs composants,
- le référentiel de données de l'environnement de tests est difficile à contrôler, les données proviennent de copies de la base de production. Les données de tests peuvent donc être perdues ou altérées et impacter le bon fonctionnement des tests de non-régression.

Solution apportée

Valtech s'est focalisé sur :

- l'étude de compatibilité technique entre l'interface graphique et l'outil d'automatisation, afin d'identifier les composants graphiques problématiques et d'amener les développeurs à concevoir des interfaces testables,
- l'étude de ROI, destinée à vérifier l'intérêt économique de l'automatisation (plus un test est exécuté, plus l'automatisation est rentable à court terme),
- la définition de la stratégie d'automatisation visant à :
 - choisir quels tests automatiser pour maximiser la valeur des tests de non-régression par rapport à l'effort investi,
 - définir la façon de gérer le référentiel de données.

ETIENNE CHARIGNON

// Consultant senior
// Certified Scrum Master
// Valtech



L'automatisation des tests dans un contexte itératif et incrémental n'est pas une option. Quel qu'en soit le prix, c'est une obligation.



N.B. Le fait d'itérer implique de rejouer systématiquement certains tests de non-régression. Cela rend économiquement intéressante l'automatisation de ces tests, à condition toutefois que l'effort d'automatisation lié à la technologie utilisée ne soit pas rédhibitoire, d'où la nécessité de s'en assurer.

Bénéfices obtenus

Les campagnes de tests utilisateurs sont progressivement passées d'une fréquence trimestrielle à une fréquence quotidienne. La durée d'un cycle de tests utilisateurs a été réduite de 1 semaine à 2 heures.

La notion de « campagne de tests » perd progressivement de son sens pour laisser place à la notion de « tests en continu ».

L'ESSENTIEL À RETENIR

- Les tests manuels peuvent être comparés à une force de frottement: plus la vitesse et la pression s'accroissent sur un corps en mouvement, plus importants sont les frottements, réduisant l'efficacité des efforts concédés pour accélérer le mouvement.
- L'automatisation des tests réduit ces forces de frottement et évite ainsi l'échauffement et l'usure du dispositif.

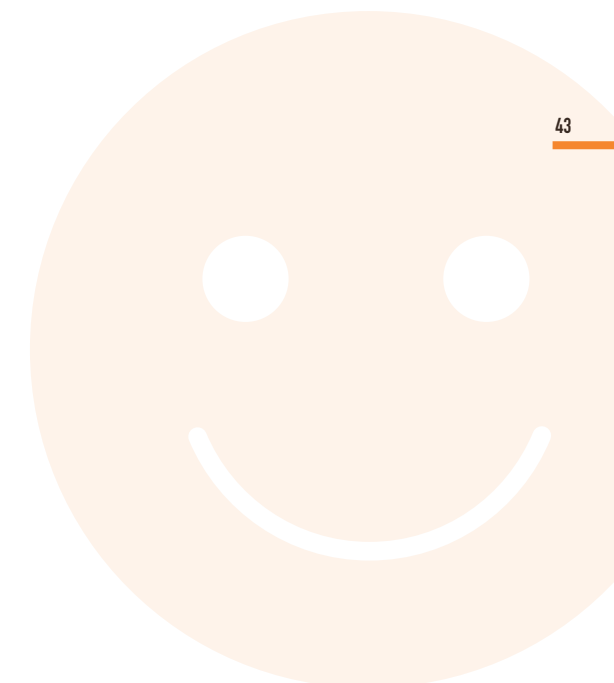
2.9

RÉSULTATS OBTENUS SUR DES PROJETS RÉALISÉS PAR VALTECH

Valtech est la première société en France à avoir proposé en 2002 une formation de conduite de projet dans un contexte itératif et incrémental (à l'époque sur la base de Unified Process). Mais très vite, cette démarche utilisée sur les projets de l'époque a été remplacée par la démarche Scrum, plus Agile et moins contraignante du point de vue de la documentation projet et des livrables.

- **Cette adoption a été renforcée par l'intervention, en 2003, en France mais également en Inde, dans notre centre de développement à Bangalore, de Craig Larman, auteur de nombreux ouvrages de référence en matière de processus d'ingénierie logicielle et d'Agilité.**

Valtech a eu la responsabilité de déployer l'ensemble des principes Agiles sur la totalité des projets, jusqu'à la réorganisation complète des bureaux précédemment organisés en cubicles et transformés en espaces de travail ouverts. Il a également certifié Scrum Master l'ensemble des chefs de projet et a initié la mise en place de nouveaux outils tels que Valtech Cockpit, la plate-forme collaborative Valtech, utilisée sur les projets aujourd'hui.



Cet électrochoc a été réellement salutaire car les résultats sur les projets ont été grandement améliorés sur différents plans:



Le respect des délais

Le time boxing et le fait de procéder par itérations successives a permis aux équipes projet d'augmenter leur productivité.



La qualité des livraisons

Les anomalies étant corrigées au fur et à mesure, avec une priorité plus importante que les fonctionnalités à livrer, la charge de gestion de ces anomalies a diminué pour laisser place à plus de fonctions implémentées.



La confiance de nos clients

Tout nos clients sont restés fidèles et possèdent des équipes à Bangalore, dont le nombre s'accroît chaque année. De plus, de nouveaux clients intéressés par l'offshore et l'Agilité nous ont mis en compétition sur des Proof of Concept, avec à la clef de nouveaux projets sur des domaines jusque là inexplorés par Valtech, comme, par exemple, celui de testeur de puces électroniques.



La satisfaction de nos collaborateurs

Le fait de collaborer plus étroitement avec des équipes distantes et d'une culture différente a motivé de nombreux consultants, les incitant à aller plus souvent en Inde ou à faire des missions de conseil sur l'adoption de l'Agilité dans un contexte onshore mais également offshore.



Un certain confort sur les projets

Souvent le terme projet est synonyme de stress, d'insatisfaction, de pertes financières... Avec la mise en place des pratiques Agiles, la maîtrise des projets s'accroît, le feedback du client et des utilisateurs est réel, la qualité des livraisons et la productivité augmentent nettement.

3

LE MARKETING DIGITAL AGILE



OU POURQUOI ET COMMENT METTRE DE L'AGILITÉ
DANS LA DÉMARCHE DE CRÉATION GRAPHIQUE



LUBOMIRA ROCHET

//Directeur Général Adjoint
//Valtech



Le monde s'accélère. Proposer le bon produit ou service à la bonne personne, dans le bon contexte d'achat, au bon moment est la clé de réussite d'un projet de marketing digital. Supporté par des plates-formes technologiques de plus en plus complexes et des supports d'expériences de plus en plus riches et sophistiqués, le marketing digital est un vecteur de rentabilité fort pour les annonceurs et s'inscrit clairement dans leur exigence de retour sur investissement et d'image.

Le marketing digital personnalisé en temps réel s'inscrit aujourd'hui dans la réalité des usages et des attentes, à la fois des marques et de leurs clients. Mais qui dit temps réel et Web 2.0, dit dialogue continu avec les différents publics, intégration continue des feedbacks des utilisateurs, adaptation constante des objectifs et des stratégies marketing en fonction des changements dans l'écosystème de la marque. Cette nécessaire réactivité en temps réel impose une vraie réflexion sur les méthodologies de gestion des projets marketing et plus généralement sur l'organisation du département marketing lui-même. L'Agilité, qui place l'utilisateur au cœur de sa méthodologie, privilégie l'opérationnel sur la documentation pléthorique, favorise l'échange et la collaboration avec l'écosystème et prône la réactivité permanente plutôt que le suivi strict d'un plan, prend tout son sens dans ce contexte. L'application des principes et des pratiques Agiles au métier du marketing vient donner aux directions marketing les outils de leur efficacité et de leur pertinence, en les aidant à développer leurs produits et à exécuter leurs campagnes plus vite et de manière plus évolutive. Les équipes marketing organisées en mode Agile peuvent ainsi travailler de manière ultra collaborative au développement et à l'amélioration des sites web, des produits ou des services, en tenant compte du feedback des utilisateurs.



3.1 LA VISION DU PRODUIT

À tout produit est nécessairement associée une vision, fixant le cap, donnant du sens et décrivant ce qu'on entrevoit pour le produit à court, moyen ou long terme.

La vision du produit est donc cruciale, structurante ; elle sera le socle de toute l'Expérience Utilisateur du produit.

Parfois peu détaillée, souvent posée en réaction à un problème et pilotée en termes d'objectifs, la vision du produit n'aura de sens que si elle est portée et partagée avec les équipes, pour être en mesure de fédérer et de se projeter efficacement dans le futur.

Construire la vision, la partager puis la porter : trois vrais challenges à relever, notamment pour les Product Owners des projets Agiles, qui utilisent des pratiques hautement collaboratives.

La vision synthétique est indispensable et, de surcroît, facile à mettre en œuvre. On peut, selon les contextes et les objectifs, l'accompagner de l'une voire des deux techniques Product Box.

La « vision synthétique » ou « Elevator Statement » (d'après Moore) est la technique simple et efficace par excellence. Elle permet de structurer, en peu de temps, la vision du produit. Son format est le suivant :

POUR : [public concerné par le produit]

QUI SOUHAITENT : [formulation du besoin des cibles]

NOTRE PRODUIT EST : [ce qu'est le produit]

QUI : [le bénéficiaire majeur, l'utilité de la solution]

À LA DIFFÉRENCE DE : [pratique actuelle, concurrence]

PERMET DE : [éléments différenciateurs majeurs]

Lue à voix haute, la vision synthétique ne doit pas excéder deux minutes. Elle trouve facilement sa place au sein du radiateur d'informations du projet. La méthode des Personas (cf. Encart) pourra venir compléter admirablement les deux premiers éléments de la vision, ceux relatifs à la cible (« Pour ») et à leurs buts (« Qui souhaite »).

— La **Product Vision Box** (d'après Highsmith) qui permet au démarrage d'un projet de construire la vision et la partager, de manière ludique, avec l'équipe chargée de concevoir le produit mais aussi ceux chargés de le vendre. L'équipe crée une boîte, représentation visuelle, concrète du logiciel ou du service qu'elle est censée développer : un nom, une image, un slogan, quelques arguments pour vendre le produit puis sur l'autre face, par exemple, le détail contenant plus de fonctionnalités ou encore les pré-requis.

- La **Product Box** (d'après Hohmann) qui offre une forte connotation « Expérience & Etude Utilisateurs ». L'atelier de travail « Product Box » est résolument orienté clients, tourné vers les utilisateurs du produit final dont on récolte le feedback. Ils créeront en groupe une boîte pour le produit dont ils simuleront ensuite la vente. L'atelier se joue généralement dans une dynamique collective. Plus il y aura de boîtes, mieux ce sera !

3.2 PERSONAS, VOUS AVEZ DIT PERSONAS ?

Un Persona, c'est un utilisateur-type, une représentation fictive des utilisateurs cibles, qu'on peut utiliser pour fixer des priorités, guider nos décisions de conception d'interface et tester les scénarios d'interface les plus prioritaires.

Cette méthode, inventée par Alan Cooper en 1999 dans son best-seller **The Inmates Are Running the Asylum**, permet d'offrir une vision commune et partagée par l'équipe des utilisateurs d'un produit, en insistant sur leurs buts, leurs attentes et leurs freins potentiels, et en proposant un format des plus engageants. Les Personas suscitent en effet de l'**empathie**, un véritable investissement émotionnel : ils prennent très vite une place de choix sur le radiateur d'informations du projet Agile.

Les « rôles Utilisateurs », éléments-clés des user stories, adoptent volontairement un format très synthétique, et restent avant tout sur la relation utilisateur / système : ni éléments fictifs, ni scénario, ni travail d'enquête.

Exemple : « En tant que recruteur, je peux déposer une offre d'emploi pour recevoir plus de candidatures. »

Sur la base de grandes catégories d'utilisateurs (rôles ou segments marketing préalablement identifiés), puis d'ateliers de travail et d'entretiens utilisateurs, la méthode des Personas va donc plus loin, en affinant, en détaillant puis en **personnifiant les rôles utilisateurs**.

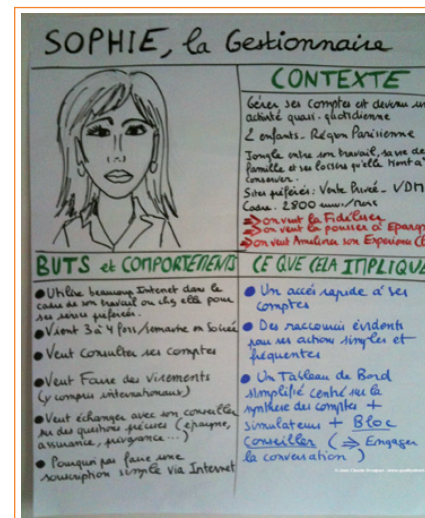


FIGURE 10 Exemple de persona
[Source Jean-Claude Grosjean www.qualitystreet.fr]

La cible du futur produit va donc très vite émerger au travers d'un nombre restreint de Personas et surtout d'un **Persona primaire**, celui pour qui on construit le produit.

La méthode des Personas est une démarche avant tout collaborative qui va mobiliser toute l'équipe : les Personas se construisent **collectivement** au cours de plusieurs ateliers de travail. Cette construction devra impérativement s'appuyer

sur des données issues d'entretiens (parties prenantes et futurs utilisateurs) voire de l'observation directe des cibles, et sur l'épluchage de diverses sources (support, presse, études externes, concurrence ...) : c'est la spécificité et la force de la méthode !

Des éléments de contexte, les buts et comportements et leurs impacts sur le produit : voici au final les constituants de base d'un Persona, travaillés de manière collaborative dans un mode Agile...

3.3 LA DÉMARCHÉ CRÉATIVE

Les paragraphes suivants ont pour but de clarifier le rôle de la création graphique dans un projet. En effet, la phase de création ne consiste pas simplement à mettre en couleur des éléments préfabriqués pendant les spécifications fonctionnelles ou ergonomiques. La réponse graphique a un périmètre plus large, qui complète et enrichit le projet grâce à des solutions d'affichage, de présentation ou de composition.

Conception de l'interface

La conceptualisation de l'interface est avant tout la représentation visuelle d'un concept. Ce concept est élaboré par plusieurs personnes qui travaillent en collaboration. La taille de cette équipe est bien entendu variable selon la complexité du projet.

Elle est constituée généralement de :

- Un directeur artistique (DA) et/ou un directeur de création (DC),
- Un ergonomiste,
- Un consultant fonctionnel ou, consultant métier, dans certains cas.

Ateliers de conception

Conception en ateliers

Le concept général de l'interface n'est pas élaboré de manière cloisonnée. L'équipe de conception s'associe au Scrum Master et au Product Owner pour travailler en ateliers. Il est également souhaitable d'associer, lorsque c'est possible, des utilisateurs finaux de l'outil ou l'application qui seront réalisés.

Ces ateliers sont organisés comme des réunions Agiles, c'est-à-dire que toutes les personnes présentes sont représentatives de l'équipe et chacune a droit à la parole.

Néanmoins, c'est généralement le consultant ou l'ergonome qui dirige les ateliers.

Ces ateliers ont 3 objectifs :

- aider à concevoir le concept visuel et graphique grâce au directeur artistique,
- orienter l'interface vers les besoins utilisateurs et selon les attendus clients grâce à la présence des deux représentants,
- contrôler, orienter ou comprendre la faisabilité pour le Scrum Master et le Product Owner.

Ensemble, ils vont définir les points qui permettront de concevoir le concept et passer en revue :

- la compréhension métier,
- les fonctionnalités,
- la compréhension de l'utilisateur,
- l'Expérience Utilisateur attendue,
- les représentations visuelles et graphiques.

Ces ateliers, organisés très tôt dans le projet, permettent de proposer une représentation visuelle et graphique très rapidement, et cela augmente le niveau de compréhension pour toute l'équipe.

Par exemple, durant ces ateliers, on va concevoir des Personas, donner une attention particulière à une représentation graphique, dessiner les liens entre les actions pressenties, ou certaines interactions particulières...

Les points traités en atelier le sont avec un niveau de détail variant selon la nature des projets. Ainsi, sur certains projets, les équipes peuvent être amenées à traiter des points particuliers liés à la technologie (applications mobiles, interfaces tactiles...) ou à des contextes particuliers (niveaux d'expérience des utilisateurs, contexte d'utilisation...)

Dans tous les cas, ils sont gérés par des priorités et un niveau de complexité associé.

Donner des priorités et les piloter

L'objectif principal est bien entendu de répondre de manière positive à l'utilisateur final. Tout doit être mis en œuvre pour lui proposer une interface simple à comprendre, intuitive. Il ne faut jamais perdre de vue que cette interface devra offrir à l'utilisateur une réponse efficace et adaptée à son besoin...

Le groupe de travail constitué pour ces ateliers est amené à envisager différentes hypothèses de représentation et de modélisation de l'interface.

Dans un premier temps, plusieurs thèmes sont abordés, tels que :

- le type d'interface attendu (aspects graphiques, style...),
- la simplification de l'interface, notamment dans le cas de refontes (raccourcis envisagés, simplification des tâches...),
- les interactions attendues avec l'utilisateur (méthode de saisie, environnement de saisie, niveau de précision...),
- les impacts techniques (faisabilité).

Dans un second temps, chaque point traité est hiérarchisé, et des priorités sont attribuées:

- On détermine les priorités en fonction de tous les paramètres qui impactent le projet : développements techniques, planning, budget, complexité de réalisation...
- On hiérarchise selon les mêmes paramètres, et on prend en compte la faisabilité, les risques sur les innovations, et tous les points particuliers liés aux contraintes graphiques.

Une fois cette étape terminée, la conception des cinématiques et les travaux de création graphique liés à l'ergonomie peuvent débuter.

Décomposer en cinématiques

On distingue 3 types de composants ergonomiques pour lesquels la création graphique va être représentée. Du plus détaillé au plus général :

- une **tâche** : c'est une action utilisateur, composée d'une ou plusieurs actions simples, permettant d'accomplir un objectif unique,
- un **scénario d'usage** (ou cas d'usage) : c'est un ensemble de tâches dépendantes qui permettent à l'utilisateur d'atteindre un objectif identifié,
- une **cinématique** : c'est un ensemble de scénarios permettant l'accomplissement de tâches dans des cas d'usage. Ces actions utilisateurs peuvent être transversales et/ou interdépendantes les unes avec les autres. À l'intérieur d'une cinématique, on retrouve plusieurs scénarios, et plusieurs tâches dans chacun des scénarios.

L'objectif de décomposition en cinématiques est d'identifier et de répertorier les scénarios d'usage les plus importants. Le scénario est simulé sur un document de story-board et sert à identifier une ou des tâches que l'utilisateur doit accomplir.

N.B. les cinématiques peuvent intégrer un certain nombre de tâches qui comportent des niveaux de priorité différents.

La cinématique est matérialisée par un story-board qui présente les écrans de manière filaire, sans habillage graphique (wireframe). Cette représentation donne des indications sur les contenus et la position des éléments de la page. Une attention particulière est donnée aux cinématiques et aux scénarios qui présentent des priorités différentes.

Exemple: une cinématique de saisie et d'enregistrement d'un profil utilisateur, comporte des éléments de faible niveau de complexité (les coordonnées, le nom...), mais elle peut aussi comporter des tâches plus complexes (interfaçage avec un annuaire, vérifications diverses ...)

Cette étape est capitale pour la représentation des données, elle est immédiatement suivie de la réalisation de la représentation graphique.

L'importance d'avoir une vision graphique

La représentation graphique donne à tous les acteurs du projet accès à une représentation proche de la réalisation finale. Elle a de l'importance :

- pour le client, afin de valider ses hypothèses et la conformité avec ses attentes,
- pour l'utilisateur, afin d'anticiper la prise en main et participer à l'amélioration,
- pour l'équipe de développement, afin de conceptualiser le résultat final.

Obtenir une représentation graphique de l'interface très tôt dans le projet permet :

- d'apporter une dimension rarement anticipée par les équipes au départ du projet : elle cristallise des idées, des concepts, des intentions, grâce à la mise en forme et la représentation visuelle de formes, de couleurs, de styles qui seront employés.
- de donner un style, un aspect visuel qui aura un impact sur le niveau de qualité. Ce style pourra être ludique ou plus technique, comporter des orientations plus textuelles, simuler ou s'inspirer d'interfaces tactiles.... Ce style, formalisé graphiquement, est la synthèse visuelle de toutes les idées qui sont issues des ateliers.
- de contribuer à faire évoluer les besoins, grâce à l'apport d'idées innovantes et différenciatrices.
- de fournir des réponses visuelles à des complexités fonctionnelles : une image vaut parfois mieux qu'un long discours !

- **La représentation graphique apporte naturellement des solutions visuelles et interactives qui orientent les solutions techniques et modifient les développements ou les fonctionnalités. L'association des interactions avec l'interface est primordiale pour la réalisation d'une application efficace.**

Intégrer la création graphique dans les projets Agile.

La création graphique est une étape « à risque » dans un projet :

- risque émotionnel lié à la qualité de la création,
- risque de valider une ligne graphique non appropriée, ce qui bloquera les développements,
- difficultés de réalisation des interactions complexes,
- difficultés de mise en production de certaines solutions et les retards que cela peut engendrer.

Dans tous les cas, ce risque se matérialise souvent par une désynchronisation entre les tâches de production graphique et les tâches de production technique.

Avec une approche Agile, ces risques sont limités pour 3 raisons :

- la création est préparée et planifiée en amont du développement,
- les tâches de réalisation et de conception de la création graphique sont décomposées dans le backlog au même titre que les tâches de développement (définition des priorités, indice de complexité),
- l'intervention des équipes graphiques est itérative et permanente tout au long du développement (et pas simplement au début).

Le processus de création graphique

Ce processus comporte quatre phases qui sont préalablement présentées au donneur d'ordre et aux équipes. Chacune des phases apporte un lot de réponses aux problèmes de l'interface et permet de progresser vers la phase d'acceptation.

Pour assurer la réussite du projet, des compromis sont nécessaires : il faut choisir les éléments à réaliser parmi ceux proposés. Néanmoins, il ne faut pas perdre de vue que cette approche doit permettre aussi de fournir des réponses plus facilement.

La représentation graphique entraîne nécessairement des jugements de valeur parfois assez tranchés, il est pourtant indispensable, à ce stade, de suivre le cheminement qui sera proposé par le directeur artistique. Les choix d'interfaces doivent dépasser les prises de position trop émotives (J'aime, je n'aime pas).

La démarche de création consiste à réaliser successivement :



Les pistes graphiques

Ces pistes servent avant tout à défricher des concepts ou des idées, et ne sont pas forcément complémentaire entre elles. (On ne recompose pas une interface avec plusieurs pistes différentes). Elles sont matérialisées par des planches de recherche sur des écrans type ou des écrans qui font état du processus. Ces écrans contiennent la représentation des objets ou des interactions envisagées.

La ligne graphique

La ligne graphique est avant tout un choix d'idées, de concepts et d'états d'esprit plutôt que réellement une des pistes présentées initialement. Ainsi le DA ne va pas nécessairement recomposer une ligne finale par association des pistes initiales, mais plutôt recomposer et enrichir une réponse en tenant compte des remarques reçues.

Le concept graphique

Avec le concept graphique, on aborde une dimension plus technique car les objets créés sont destinés à être utilisés par les infographistes pour produire les éléments nécessaires aux développements.

La charte graphique

La charte graphique est un document de synthèse qui est finalisé à l'issue du projet. Une charte n'est par définie en amont, mais de manière itérative au cours du développement.

Des points de contrôle

Il est important malgré cette décomposition structurée, de se réserver des points de contrôle qui vont agir comme des garde-fous de la création.

Conserver une interface graphique homogène

Le processus itératif permet une décomposition de la création graphique dans le projet Agile. C'est un point positif qui facilite son intégration avec les développements.

Cependant cette décomposition entraîne une fragmentation de la ligne graphique et limite la vue globale de l'interface.

Pour cette raison, il est indispensable d'anticiper le résultat final en positionnant des étapes de consolidation et d'homogénéisation. Ces étapes seront intégrées durant toute la réalisation.

On peut anticiper ces étapes de consolidation dès la première restitution et de manière planifiée, mais il est tout à fait possible de se réserver des consolidations sur demande, en fonction de l'avancement ou face à un problème rencontré.

Ce travail de consolidation de l'interface graphique ne doit pas attendre l'ultime itération. La consolidation régulière permettra en revanche de se replacer dans le contexte des scénarios et d'en vérifier la cohérence, tant du point de vue de l'interface graphique, que des interactions qui en découlent.

Rythmer la progression

Avec le jeu des priorités et des ajustements tout au long du processus Agile, il est nécessaire de faire coïncider les tâches du backlog avec les avancées réelles des développements pendant l'itération.

Les créatifs ne sont pas présents à plein temps avec l'équipe de projet, les points de synchronisation création/développements techniques doivent être positionnés lors de la planification des itérations.

Des interlocuteurs graphiques qui évoluent selon les projets

Comme nous l'avons vu dans les paragraphes précédents, les phases amont sont prises en charge par le directeur artistique, avec pour résultats un concept et une ligne graphique globale. Les tâches de déclinaisons graphiques sont ensuite confiées aux infographistes, sous le contrôle du directeur artistique.

Le DA reste l'interlocuteur privilégié pour la partie graphique. Une fois la production graphique démarrée, il est aussi l'interlocuteur qui travaille en tandem avec l'ergonome pour mettre au point les solutions appropriées liées aux interactions ou aux représentations graphiques.

Lorsque le projet le nécessite et notamment sur les interfaces riches, il est intéressant d'avoir un ergonome possédant une double compétence technique et graphique pour la réalisation des déclinaisons et des interactions de l'interface. La double compétence facilite les échanges avec les membres de l'équipe en charge du développement.

Exemple : dans le cas d'un projet traité en Flash ou Flex, l'ergonome réalise l'interface graphique sur la base des éléments produits par les infographistes, tout en s'interfaçant avec les données du socle technique.

Le concept graphique impacte le design du système

Le concept est indispensable pour bien appréhender l'interface au sens large. Le concept graphique définit des formes, des styles, des couleurs...et tous les éléments indispensables au bon fonctionnement des interactions entre l'utilisateur et le système.

Mais bien que la représentation soit purement graphique, elle influence les solutions techniques.

Le concept graphique ne s'applique pas en fonction de pages isolées, mais en fonction de chemins utilisateurs dans lesquels il agit, interagit et progresse.

Design d'interface et design d'interaction

La création graphique est une représentation visuelle. Mais la finalité n'est pas une représentation figée et statique. Quelle que soit la technologie ou la plate-forme choisie, ce sont bien les interactions avec l'utilisateur qui prévalent.

Il est donc indispensable de penser la création graphique du système de manière itérative, y compris au niveau de chaque tâche utilisateur.

En effet, une action sur la page n'entraîne pas forcément un enchaînement vers une autre page... L'utilisateur peut être amené à actionner différents items sur l'écran pour accomplir son action. Ce sont ces états et ces interactions que la ligne graphique devra aussi anticiper.

D'autre part il est fréquent que des réponses graphiques apportent des solutions d'organisation liées au fonctionnel ou à la technologie et entraînent des changements durant une itération. Graphistes et développeurs doivent donc être particulièrement à l'écoute l'un de l'autre et ne pas hésiter à faire évoluer leurs tâches pour servir le projet.

Ces leviers forts sur lesquels les praticiens de l'Expérience Utilisateur vont pouvoir asseoir leur action sont les suivants :

- les livraisons fréquentes (toutes les deux ou trois semaines),
- l'activité de validation en continu,
- le travail collaboratif,
- la coopération et l'implication forte des clients et utilisateurs, tout au long du projet,
- l'accent mis sur la simplicité.

Malgré tout, Scrum et les méthodes Agiles laissent encore largement de côté les éléments relevant de l'ingénierie des exigences, de l'ergonomie ou encore de l'Expérience Utilisateur.

Ce manque incontestable peut vite se transformer en faiblesse, tant ces dimensions sont devenues cruciales pour assurer une bonne acceptation des produits par les utilisateurs finaux. L'intégration efficace des spécialistes de l'Expérience Utilisateur dans les projets Agile est donc aujourd'hui plus que nécessaire. Celle-ci passe en premier lieu par :

- la diffusion de l'Expérience Utilisateur au sein de l'équipe,
- la défense des utilisateurs, de leur activité et de leurs buts (notamment grâce à la technique des Personas),
- un réel effort autour de la vision du produit.

Ce nouvel élan vers l'Expérience Utilisateur passe aussi par la définition de guidelines et recommandations ergonomiques, et par l'utilisation d'outils de prototypage légers. Autant d'éléments source de valeur pour l'équipe, les clients et les utilisateurs finaux.

Vers une démarche « Agile UX »

Même si l'effort relatif à l'Expérience Utilisateur se poursuit tout au long du projet, notamment au travers de l'accompagnement des équipes de développement et des tests utilisateurs, l'essentiel de l'activité « Agile UX » doit s'enclencher dès les premières itérations.

Ces premières itérations seront le moment idéal pour définir les cibles de l'application à concevoir, avec la technique des Personas, dans le prolongement des user stories.

Une fois la cible définie, la conception graphique et ergonomique de l'application pourra se poursuivre au fil de l'eau, essentiellement autour des cinématiques (enchaînement des écrans de l'application) et d'une activité de storyboarding, c'est-à-dire le prototypage rapide des écrans, dans une approche toujours plus collaborative, au plus juste, et avant tout en support du Product Owner.

3.4 AGILITÉ ET EXPÉRIENCE UTILISATEUR

Des points de convergence évidents

L'Agilité et l'Expérience Utilisateur ont beaucoup de points communs, dont le plus important est le facteur humain. Autre exemple : la recherche permanente du feedback (au travers notamment des pratiques tests) mais aussi la défense de la simplicité, sont communes aux méthodes Agiles et à la démarche ergonomique.

La conception centrée utilisateur peut ainsi bénéficier des conditions très favorables offertes par l'Agilité (des conditions rarement présentes dans les cycles de développement traditionnels).

Le cycle de vie Agile impose pour autant aux spécialistes de l'Expérience Utilisateur d'adapter leur démarche.

Celle-ci va donc se façonner autour de six règles essentielles :

- 1 soutenir le travail d'analyse et de priorisation du représentant des utilisateurs ou de l'équipe métier (Product Owner),
- 2 faire juste ce qu'il faut de recherche utilisateur, de modélisation et de travail sur l'interface, notamment au début du projet,
- 3 travailler sur plusieurs modes à la fois :
 - a. l'anticipation et la conception du contenu des itérations futures (le plus souvent une ou deux itérations maximum),
 - b. l'accompagnement de l'équipe sur le contenu, les user stories en cours, que l'équipe doit livrer à la fin de l'itération,
 - c. le test auprès d'utilisateurs finaux, par exemple du contenu de l'itération précédente, livré par l'équipe,
- 4 proposer plus de réactivité, sur le recueil et la restitution du feedback (par des tests moins formels, progressifs et adaptés),
- 5 faire du prototypage rapide (adapté au contexte et aux destinataires, toujours au plus juste et toujours source de valeur),
- 6 jouer un maximum sur le volet participatif et sur la facilitation en multipliant les ateliers de travail collaboratifs.

Tester toujours plus l'Expérience Utilisateur

Tester, mesurer, tester encore ... Tester l'Expérience Utilisateur en permanence, c'est se lancer dans une démarche d'amélioration continue, et l'affiner au fur et à mesure du cycle projet ou du cycle de vie produit (logiciels, applications web, mobiles, sites internet...). C'est ainsi s'inspirer de nouvelles méthodes, comme le « Guerilla Usability Testing », pour plus de feedback et de valeur.

Les pages d'accueil, les parcours clients et tunnels de conversion, les fiches produit, les processus métier, le design, tout est testable, quel que soit leur degré de maturité: version opérationnelle, concepts, esquisses, pistes graphiques, prototypes haute fidélité ou même prototypes papier... Et l'Agilité avec ses cycles itératifs et ses livraisons incrémentales et régulières offre des contextes très appropriés (ex : RITE encadré) pour ces mesures ponctuelles.

L'idée, dorénavant, est d'aller au devant des utilisateurs, et des clients, et de profiter de toutes les situations dans un contexte où la mobilité gagne du terrain étant donné que les situations d'usage évoluent, y compris pour les applications professionnelles.

Augmenter la fréquence des tests, multiplier les Feedbacks

Moins de participants, moins de tâches mais plus de tests en variant les techniques utilisées, celles qui nécessitent des participants, sans oublier celles dites « expertes » (benchmark, évaluations expertes, activités d'exploration). Dans cette approche, on garde l'esprit Agile en privilégiant les notions: " **Action** ", " **Juste ce qu'il faut de formalisme** " et " **Collaboration** ", notamment en passant plus de temps avec les équipes (workshops, pair designing).

La « Guerilla Usability » se propose par exemple d'initier la démarche de test en interne avant de s'orienter très vite vers la cible du produit en utilisant les contacts clients, l'entourage, les réseaux sociaux, les mailing lists, et ou profitant de formulaires placés sur le site ou de toutes ces occasions au cours desquelles on peut croiser ses clients, comme les salons professionnels.

Comment faire des tests d'ergonomie avec de vrais utilisateurs dans un contexte Agile ? Utiliser le RITE (Rapid Iterative Testing and Evaluation).

L'idée est simple: les modifications sont effectuées dès qu'un problème est détecté avec certitude et que la solution est claire. Autrement dit, une modification peut s'opérer suite au passage du premier participant et être testée, vérifiée avec les suivants : une valeur réelle et immédiate.

Née chez les équipes de développement Microsoft (Games Studio), la méthode RITE innove dans la pratique des Tests Utilisateurs et répond parfaitement aux exigences et à la réalité des projets d'aujourd'hui.

Des Tests Utilisateurs plus efficaces

Le RITE se distingue des Tests Utilisateurs classiques sur la restitution du feedback mais aussi, et surtout, sur son traitement et sa vérification.

Des Tests Utilisateurs moins monotones

Une série de tests avec 8, parfois 12 ou jusqu'à 16 participants, sur une même application, selon un même protocole peut devenir lassante. La méthode RITE, en rupture avec ce modèle figé, permet de se sortir d'une routine dans laquelle on peut vite tomber.

Des Tests Utilisateur tout aussi valides

De la rigueur sur le choix des profils testés, un plan de test bien cadré, des scénarios de test, un protocole verbal (« Penser à voix haute »), un facilitateur... la méthode RITE n'a rien à envier aux Tests Utilisateurs « classiques ».

RITE repose sur une échelle de décision pour qualifier les problèmes rencontrés, et déterminer s'ils seront résolus et testés immédiatement.

4
....

LA TRANSFORMATION VERS L'AGILITÉ



OU POURQUOI ET COMMENT GÉNÉRALISER LES PRATIQUES
AGILES À L'ENSEMBLE DE L'ENTREPRISE

Pour une organisation, le fait de devenir Agile participe à créer la capacité d'adaptation, rapide et permanente, au contexte dans lequel elle évolue. Souvent confinée aux équipes techniques, voire limitée aux équipes de développement logiciel, la mise en œuvre des valeurs et des principes Agiles doit être étendue bien au-delà de ces horizons pour apporter la valeur attendue.

Fort de l'expérience de projets ambitieux de transformation vers l'Agilité, Valtech considère que ce type de projet doit être envisagé de façon systémique : l'organisation, les ressources humaines, la technologie, les processus de développement et certains processus support sont concernés.

Une fois la transformation décidée, et pour en garder l'esprit et le rythme, la vision de l'objectif doit être définie et portée par le bon niveau de l'organisation (sponsor). La transformation doit être menée comme un projet (stratégie, acteurs et indicateurs). Le projet de transformation lui-même est gouverné de façon Agile (itérations, fréquents retours d'expérience, solutions émergentes, etc.). Finalement, le résultat des expériences successives est pérennisé et amplifié (dissémination des leçons apprises, formation de « coachs » internes, mise en place de communautés).

Au-delà de la transformation des équipes de développement, ce chapitre fait le point sur la transformation à grande échelle d'une organisation.

PETER SENGE

// Extrait traduit de l'anglais
// The Fifth Discipline: The art and practice of the learning organization

“

Une organisation apprenante est une organisation qui se nourrit de la variété des expériences, des compétences et des savoirs individuels, grâce à une culture qui encourage les débats, les défis et les mises en doute, au travers d'une vision commune ou d'une intention partagée.

”

4.1 ENJEUX ET MOTIVATIONS

Un projet de transformation, quel qu'il soit, remet en cause des organisations, des processus, mais surtout remet en cause la façon dont les hommes et les femmes travaillent, collaborent, s'épanouissent et adhèrent aux projets de l'entreprise. Dans les transformations d'ampleur, la période d'incertitude et d'instabilité peut être longue et difficile à supporter : les motifs et les enjeux de la transformation doivent être clairs et solides !

Les enjeux et les motivations de la transformation vers l'Agilité sont particuliers à chaque entreprise, ils sont établis sur la base d'éléments objectifs (parts de marché, réductions de coûts, meilleure réactivité, qualité des applications) et subjectifs (satisfaction des clients, qualité des relations dans les équipes, par exemple).

À un niveau de détail plus fin, on citera par exemple :

- réduire le délai entre l'expression d'une demande et la mise à disposition de la solution,
- augmenter la qualité des applications, afin de maîtriser la charge de maintenance et de support,
- limiter les allers-retours entre la maîtrise d'ouvrage et la maîtrise d'œuvre à l'occasion des tests de validation,
- rendre l'avancement des projets plus visible pour les directions opérationnelles en particulier, afin d'adapter les documentations techniques produites aux besoins réels, pour contenir les coûts et les délais,
- faciliter les modifications de planning et de périmètre de livraison, en cas d'environnement très mouvant (produit innovant, besoins instables).

4.2 LE PROJET DE TRANSFORMATION

Une transformation Agile est avant tout un projet de conduite du changement, dont les ingrédients peuvent être identifiés comme :

- une **vision** (ambition, objectifs, horizon temporel, périmètre, stratégie de changement),
- une **trajectoire** (connaissance de l'état actuel, étapes intermédiaires, cible),
- un **sponsor** (de préférence au plus haut niveau de l'organisation),
- une **équipe pluridisciplinaire** (pour la conduite et l'implémentation du changement),
- des **moyens** (budget, infrastructure),
- un **pilotage** (progrès de la transformation, évaluation des effets par rapport aux objectifs).

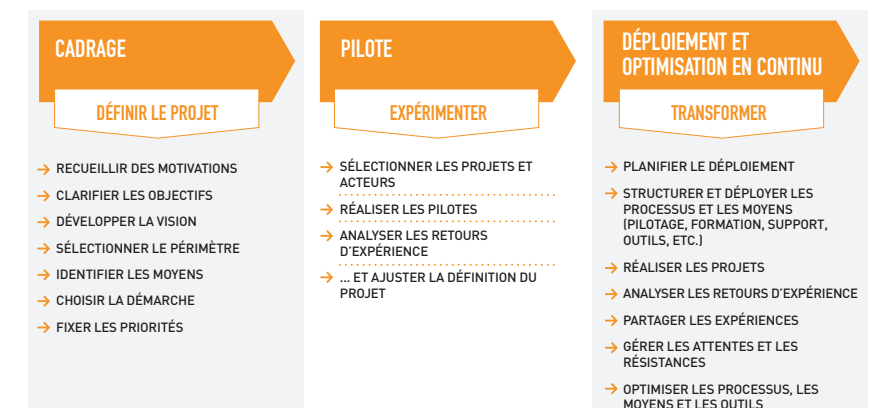


FIGURE 11 Les étapes d'une transformation Agile (Source : Valtech)

4.3 DÉFINIR LE PROJET : LE CADRAGE

Dans les approches Agiles, la définition d'un projet est portée par le Product Owner. Par similarité, le projet de transformation doit être porté par un sponsor, assisté d'autant d'experts de terrain que nécessaire pour définir un projet ambitieux, réaliste et viable parce qu'accepté et compréhensible par toutes les parties prenantes.

- Le cadrage du projet de transformation vers l'Agilité est une activité stratégique. Il est porté par la direction qui en est le sponsor. Valtech préconise une approche combinée et concurrente en « top-down » et « bottom-up ».

Développer la vision

La vision élaborée durant le cadrage doit répondre aux interrogations suivantes ²:

INTERROGATIONS	ÉLÉMENTS DE RÉPONSE
Le sponsor a-t-il le pouvoir de porter le changement ?	Le sponsor doit être conscient de l'ensemble des enjeux. Un sponsor membre de la direction générale est un facteur de succès. S'il n'en n'est pas directement détenteur, le sponsor doit au moins avoir la possibilité d'intervenir sur le budget associé au projet de transformation.
Les managers sont-ils capables d'impulser le changement ?	Les managers doivent être les premiers convaincus, ils doivent : <ul style="list-style-type: none"> · organiser des séminaires d'information, · être formés aux nouvelles techniques de management.
Quel est le degré de changement souhaitable, réaliste ?	L'analyse de la chaîne de valeur (analyse des processus) est un bon outil pour cerner le périmètre souhaitable.
	La transformation concerne les processus de travail, mais également l'organisation : il est en effet souvent nécessaire de remettre en cause les « silos » organisationnels constitués.
Quelles sont les connaissances et les compétences disponibles et celles à acquérir ?	Réaliser une étude d'écart entre le niveau actuel de compétence des équipes en place et la cible, du point de vue managérial, fonctionnel et technique.
Quelles sont les résistances et les contraintes ?	Rechercher les résistances auprès des personnes. Une transformation est une période d'instabilité qui suscite toujours des craintes et des résistances (perte de qualification, changement de rôle et de responsabilités, etc.)
	Communiquer largement et au plus tôt sur les objectifs, qui ne doivent pas uniquement être financiers !
Quel style de conduite du changement adopter ?	Assurément le plus collaboratif possible : le changement est basé sur un certain volontariat au début (phase pilote), avant d'être étendu éventuellement de façon imposée si nécessaire.
Quelle est la démarche à adopter ?	Un déploiement en big-bang est difficilement envisageable. Valtech propose de réaliser le déploiement selon un cycle itératif, dans un esprit d'amélioration continue. A l'échelle de la transformation, on définit quelques jalons par an (par exemple trimestriellement) à la fin desquels une rétrospective globale permet de faire le point sur les progrès réalisés et les points de blocage à lever.

2. D'après Balogun et Hope Hailey 1998

Quels sont les critères de réussite ou d'échec ?	Ces critères sont directement issus des enjeux et des objectifs formulés ; le développement de la vision doit aboutir à la quantification de ces critères.
Comment maintenir la dynamique de transformation dans le temps ?	Capitaliser les leçons apprises durant les projets et partager les expériences en organisant des communautés internes. Impliquer fortement les organisations transverses (méthodes et outils, QA, PMO) qui deviennent, à terme, les promoteurs du changement.

TABLEAU 6 Développement de la vision (Source Valtech)

La vision détermine l'état à atteindre en fin de projet, mais également des états intermédiaires (court, moyen et long terme) qui permettent de mesurer les progrès durant tout le déploiement.

Identifier le périmètre

Envisager la transformation vers l'Agilité de manière systémique



FIGURE 11 Une équipe de projet en relation avec son écosystème (Source Valtech)

Une équipe de projet de développement logiciel ne travaille pas isolément du reste de l'entreprise et de son environnement : les clients, les utilisateurs, le maître d'ouvrage, les sous-traitants éventuels sont parties prenantes du projet.

Dès lors, cet environnement est affecté par le **rythme itératif du projet**, et par la **nécessité de relations de collaboration** plus étroites. L'approche « Lean » propose d'optimiser le système globalement, et ce n'est pas sans raison !

➤ Pour être efficiente, une transformation vers l'Agilité doit dépasser le cadre de l'équipe de développement et s'étendre à la création d'un écosystème Agile.

“

L'approche systémique est intimidante et paraît hors de portée. L'expérience montre cependant que l'on arrive très vite aux « frontières » de l'équipe projet. Nous avons constaté de nombreuses reprises que la volonté de participer au changement de la part des personnes relevant des organisations périphériques était plus forte que prévu.

”



PATRICK LE GO

// Expert et coach Agile
// Valtech

QUI ?	LES CHANGEMENTS MAJEURS	BÉNÉFICES	RISQUES, FREINS ET CONTRAINTES
Équipe projet	Développement itératif et livraisons incrémentales	Motivation portée par les retours des clients et des utilisateurs	Les anciens chefs de projet ont du mal à s'adapter au partage du pouvoir de décision. Il est initialement compliqué pour les équipes de s'adapter au rythme itératif, surtout si les itérations sont courtes
	Développement orienté par les tests	Organisation perçue comme moins procédurière, offrant plus d'imagination et de liberté d'action	Les développeurs voient comme une suspicion d'incompétence le fait de devoir écrire d'abord les tests, tout en assurant, en plus, une couverture très importante
	Auto-organisation	La qualité est sous contrôle permanent	
	Collaboration et esprit d'équipe fort		
Client (Product Owner)	Contractualisation Agile	L'évolutivité des besoins est prise en compte	Nécessite plus de disponibilité
	Rôle de Product Owner : participation au planning d'itération, démonstration et rétrospective d'itération	Meilleure visibilité sur l'avancement	Remise en cause du forfait classique à coût, délai et périmètre figés
	Exigences formalisées tout au long du projet (au lieu d'un cahier des charges initialement complet)	Possibilité offerte d'intervention sur la planification (priorités, mise en production)	Difficulté à trouver le bon Product Owner
		Qualité et adéquation aux besoins	Tentation de la perfection : il faut savoir finir le projet : le client doit apprendre à raisonner en termes de valeur livrée, plutôt qu'en liste de fonctions fournies
		Réduction de la phase d'avant-projet	

QUI ?	LES CHANGEMENTS MAJEURS	BÉNÉFICES	RISQUES, FREINS ET CONTRAINTES
Utilisateur	Revue fréquentes	Mise à disposition de fonctionnalités plus tôt que dans un projet classique	Nécessite plus de disponibilité.
	Déploiement incrémental (optionnel)		
Pré-production et Exploitation	Validation incrémentale à intervalles réguliers	Meilleure visibilité sur les développements, anticipation plus aisée des adaptations des environnements	Nécessite la disponibilité récurrente (potentiellement à chaque itération) des équipes et des environnements.
	Mises en production plus fréquentes.		La planification des activités est très dépendante des projets.
Sous-traitants	Obligation de plus de transparence sur l'avancement, sur les difficultés rencontrées. La visibilité est donnée au moins à chaque itération.	Limitation des erreurs et défauts constatés tardivement, ce qui réduit les risques pour le sous-traitant.	L'obligation de transparence est souvent antagoniste des pratiques des projets au forfait...
	Signature de contrats Agiles		
Méthodes et outils	Passage d'un processus linéaire basé sur des livraisons de documents, à un processus itératif et incrémental, basé sur des livraisons de code opérationnel.	Modernisation des approches et outils.	Un des risques est de construire un nouveau standard détaillé et rigide, et d'oublier les aspects essentiels que sont l'amélioration et l'adaptation continue.
	Acquisition et déploiement de nouveaux outils pour l'environnement de développement et le suivi de projet		
Achats	Modification de la forme et de « l'esprit » des contrats.	La négociation sur un « volume » de fonctionnalités (mesuré en points) est plus simple que la négociation sur une liste de fonctions.	Les contrats Agiles « gagnant-gagnant » apparaissent moins protecteurs que les contrats au forfait.
			Il est inhabituel de s'engager sur un contrat pour lequel le périmètre fonctionnel est reconnu variable.
Ressources humaines	Redéfinition des missions, nouvelles descriptions de postes, prise en compte des changements organisationnels, modification des principes de rémunération variable, etc.	Alignement clair des définitions de rôle sur les missions des intervenants Agiles. Le rôle opérationnel du management intermédiaire est amplifié.	La période de transition peut être complexe à gérer, les équipes Agiles et non encore Agiles étant soumises à des « régimes » différents.

QUI ?	LES CHANGEMENTS MAJEURS	BÉNÉFICES	RISQUES, FREINS ET CONTRAINTES
PMO	Abandon du contrôle d'avancement sur les phases classiques (spécification, conception, développement, tests) au profit d'un suivi par fonctions livrées, adaptation des tableaux de bord de suivi de projet.	Participe à la réactivité de l'organisation (gestion de portefeuille).	Le changement de méthodes de travail et des repères « classiques » est déstabilisant. Le temps d'adaptation aux nouveaux indicateurs et aux outils, peut être conséquent.
	Gestion « Agile » du portefeuille de projet avec utilisation de KANBAN ³		
		Meilleure implication dans les projets, et rapprochement avec les besoins des clients/utilisateurs.	
Assurance Qualité	La qualité est « l'adéquation aux attentes du client » plutôt que « la conformité au cahier des charges » : les comptes-rendus de démonstration deviennent un élément clé.	Meilleure implication dans les projets, le rôle du QA est revalorisé.	Nécessite la remise en cause des savoir-faire
	Prise en compte de nouveaux indicateurs, issus de l'usine de développement logiciel		L'assurance qualité devient plus technique que la seule vérification de documents.

TABLEAU 7 Les clés de la transformation Agile (Source Valtech)

Identifier les moyens

En dehors des moyens organisationnels dédiés au projet de transformation (sponsor global qui porte la vision, pilote le projet de transformation, planifie le déploiement et en évalue les progrès et les résultats), la transformation Agile nécessite un accompagnement par des experts qui permettent de dépasser rapidement le stade du « faire Agile » pour parvenir au stade du « être Agile ».

3. Visuellement, un KANBAN se présente comme un tableau qui porte en colonnes les étapes d'un processus (par exemple étude d'opportunité, pré-étude, développement, qualification, déploiement), et pour chaque étape le nombre maximum de tâches que l'on peut réaliser en parallèle (en fonction de la capacité des équipes). Chaque item (ici un projet) est matérialisé par une « carte » que l'on déplace d'une colonne à l'autre dès qu'une étape est franchie.

Ces experts, généralement externes, ont pour vocation à intervenir :

- auprès du **sponsor**, pour la mise au point de la vision et la qualification des objectifs généraux,
- auprès de la **direction informatique**, et autres directions fonctionnelles (directions métier, marketing, achats, etc.), pour la quantification des moyens, la planification du déploiement, le choix des moyens d'action, les réorganisations potentielles,
- auprès des **niveaux de management** intermédiaires (responsables d'équipe, de département), pour faciliter leur transition vers les nouvelles modalités de suivi et les nouvelles modalités d'action (meneur-facilitateur au lieu de décideur-gestionnaire),
- auprès des **équipes de développement**, ils sont en charge de la transformation sur les aspects processus et sur les pratiques de planification, d'estimation, de suivi d'avancement et de communication au sein de l'équipe,
- auprès des **équipes de développement**, pour la mise en œuvre des pratiques techniques telles que les tests unitaires, la spécification par les tests, l'intégration en continu,
- auprès du **responsable de produit** (Product Owner) et des utilisateurs, pour les accompagner dans leurs nouveaux rôles et faciliter la communication et la collaboration avec les équipes de développement,
- auprès des **entités responsables** des processus support (méthodes et outils, production), pour faciliter les transformations nécessaires.

Des moyens internes doivent également être mobilisés pour amplifier et pérenniser la transformation :

- des « **champions** » **locaux** sont recrutés – sur la base du volontariat, pour entraîner les projets pilotes. Ce sont des opérationnels qui ont la « fibre » Agile, une expérience préalable et qui souhaitent participer activement à la transformation
- des **communautés Agiles**, constituées d'acteurs en provenance des diverses organisations. L'objectif des communautés est de recueillir et formaliser les expériences réalisées par les différents projets, en extraire les meilleures pratiques, partager les observations et résultats et éventuellement proposer de nouveaux axes d'amélioration.
- des **coachs Agiles internes**, formés par exemple à l'occasion des premiers projets par les coachs externes, ils ont vocation à amplifier le mouvement de transformation. Leur connaissance intime de l'entreprise est un atout... mais ils doivent avoir la capacité de remettre en cause les processus établis !

Comme pour tout programme de transformation, la communication est un élément essentiel à toutes les étapes. Le sponsor global et les sponsors locaux sont responsables de la communication de cet aspect.

- Acquérir les savoir-faire par les apports de formateurs, coachs et mentors externes, et partager les succès et les échecs par des communautés internes pour amplifier la transformation.

4.4 EXPERIMENTER : LE PILOTE

La déclinaison rapide du changement au niveau opérationnel, pour le rendre « palpable » est essentielle : l'étape pilote doit démarrer assez rapidement, même si tous les détails d'implémentation ne sont pas connus.

La réalisation des projets pilotes est nécessairement une période d'accompagnement fort et de mutations progressives : il faut expérimenter, adapter les solutions, vaincre les résistances, recueillir les résultats.

La période pilote est l'occasion de mettre en place les structures de relais internes (communautés, sponsors locaux) et les moyens dédiés (wiki, forum, séminaires périodiques) pour le partage des expériences et la propagation des nouveaux savoir-faire.

L'étape pilote permet de détecter concrètement les apports positifs et les difficultés dues à l'organisation et aux processus en place, ces informations sont utilisées pour affiner le plan projet (quel processus ou organisation modifier en premier lieu, comment)

La phase pilote s'étend sur quelques mois, et peut concerner plusieurs projets de développement.

Sélection des projets pilotes et conditions de lancement

La sélection des projets pilotes et des équipes de réalisation de ces projets doit favoriser la réussite de l'expérimentation, et être représentative de la réalité de l'activité de l'organisation. Les critères de sélection incluent :

CRITÈRE	COMMENTAIRES	
Le sponsor est réellement impliqué et disponible, il facilite la mise à disposition des moyens	Le sponsor de l'Agilité est une pièce maîtresse de la transformation	Obligatoire
Le Product Owner est réellement impliqué et disponible, il a la capacité de définir le produit à développer	Le Product Owner est un rôle difficile à tenir, mais il est une pièce maîtresse pour la réussite du projet	Obligatoire
Le Scrum Master et les membres de l'équipe sont motivés pour « devenir Agiles », leur niveau de séniorité est suffisant pour valoriser l'expérience	Durant la transformation, l'équipe peut passer par des phases de doute et d'échecs. La motivation est essentielle (cycle de Kubler-Ross)	Obligatoire

CRITÈRE	COMMENTAIRES	
Le projet se prête à une réalisation en itérations courtes (les fonctions peuvent être découpées de façon à « tenir » dans une itération, les démonstrations sont significatives)	La capacité à obtenir un retour rapide et fréquent est une des clés du développement Agile.	Obligatoire
Les conditions de lancement sont réunies : · le projet démarre en même temps que les activités de coaching · une équipe pluridisciplinaire est constituée · l'environnement matériel est en place · l'effort de montée en compétence est inclus dans le budget du projet	La synchronisation du projet et de la transformation Agile évite un grand nombre de résistances, et évite de perturber un projet qui aurait été lancé préalablement.	Obligatoire
Les contraintes liées à l'organisation et à l'environnement sont limitées.	Commencer les premiers pilotes en environnement simple (pas de sous-traitance, équipe localisée, pas de dépendance avec d'autres projets), et complexifier progressivement durant l'apprentissage	Variable
Le projet pilote apporte une réelle valeur métier.	Le niveau d'implication du Product Owner est lié à la valeur métier apportée par le projet.	Important
Les risques ⁴ du projet sont assez limités (délais, qualité, budget, etc.)	La pression de l'apprentissage en sus de la pression sur le projet peut conduire l'équipe à abandonner l'expérience en cours de route	Important
Le type de projet est représentatif (nouveau développement, intégration, maintenance, développement de services ou de composants)	Favoriser les projets de développement, plus adaptés à une adoption Agile « by the book »	Important
Les périmètres fonctionnel et technique sont représentatifs.	L'outillage de la plateforme de développement est en partie dépendant du langage (Java, C#, C++, etc.)	Important
Le dimensionnement du projet est adapté (durée, effort)	Durée « idéale » comprise entre 3 et 9 mois (pour obtenir des conclusions assez rapidement) Équipe « idéale » entre 5 et 10 personnes	Important

TABLEAU 8 Critères de sélection d'un projet pilote (Source Valtech)

La transformation d'une équipe de projet en équipe Agile

PATRICK LE GO

// Expert et coach Agile
// Valtech

« Une des clés de la réussite de la transformation Agile, à l'échelle d'une équipe de projet, est de laisser l'opportunité et le temps aux personnes qui la composent d'expérimenter et d'apprivoiser « l'état d'esprit » Agile. La recherche d'améliorations rapides dès la première itération est absolument néfaste à moyen et long terme. »

4. À noter qu'un fort niveau de risque est souvent présenté comme un discriminant absolu et réhibitoire. Il faut cependant remarquer qu'une partie des risques « classiques » sont traités directement par les pratiques Agiles. A méditer ?

Schématiquement, la transformation se déroule selon les étapes suivantes :

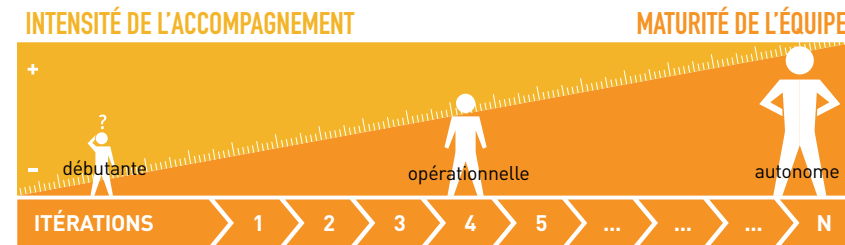
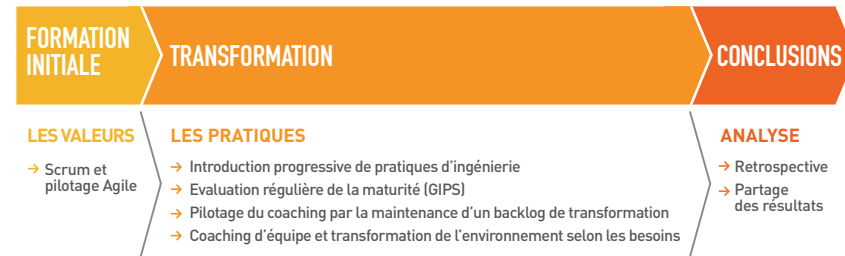


FIGURE 13 L'accompagnement dans la transformation Agile (Source Valtech)

Il est essentiel que tous les acteurs du projet participent ensemble à la formation initiale pour assurer que les valeurs et principes soient connus et partagés par tous⁵; en intra-entreprise, ces sessions sont, en outre, riches d'échanges sur les modes de fonctionnement courants, et les pistes de transformation commencent à y être élaborées avec l'ensemble des parties prenantes.

Le dispositif d'accompagnement pour un projet

INTERVENTION	PUBLIC VISÉ	DURÉE ET RYTHME	SUJETS TRAITÉS
Formation initiale Agile	Tous participants à la transformation	2 jours par session de formation, en début de transformation	Valeurs et principes de l'Agilité, concepts et pratiques du pilotage Agile de projet
Pratiques d'ingénierie Agile	Programmeurs, concepteurs, testeurs	Quelques journées d'intervention par équipe au cours de la transformation	Intégration continue, développement piloté par les tests (TDD), exigences exécutables (TDR) ⁶
Coaching d'équipe de développement	Scrum Master	2 à 4 mois (au minimum 4 itérations) avec un niveau d'intervention qui diminue au fil du temps (80% → 10%)	Mise en place de l'Agilité au sein d'un projet : « l'état d'esprit Agile », itérations, planification, estimation, indicateurs et tableaux de bord, qualité, etc.
	Développeurs, testeurs, analystes, concepteurs, etc.		
Coaching de Product Owner	Product Owner	1 à 2 mois, à raison de quelques jours par itération	Développement de spécifications Agiles (user stories), gestion du backlog, cérémonies Scrum

TABLEAU 9 Dispositif d'accompagnement pour un projet (Source Valtech)

5. Voir le Manifesto Agile
6. La connaissance des pratiques d'ingénierie est préférentiellement transmise par l'exemple pendant le développement du projet (coding dojos, mentoring)

À l'échelle d'un projet pilote, la démarche de transformation est pilotée de façon Agile : au démarrage de la transformation, puis à la suite de chaque rétrospective d'itération, l'équipe et le coach établissent un backlog de transformation qui a toutes les caractéristiques d'un backlog de projet (les items sont estimés, décomposés si nécessaire, priorisés, transformés en tâches). Pour faciliter la démarche, les itérations de transformation sont calquées sur celles du projet lui-même.

Pour mesurer la progression et la maturité de l'équipe, Valtech a développé son propre « GPS équipe Agile » qui représente le niveau d'acquisition des pratiques et techniques Agiles, au moyen de plus de 200 points de mesure, regroupés en 40 pratiques essentielles et projetés sur 7 dimensions.



ADOPTION EVALUATION - 24/09 (orange)
ADOPTION EVALUATION - 29/07 (rouge)
ADOPTION EVALUATION - 10/07 (jaune)

Fundamental maturity 25-45 %
Functional maturity 60-75 %
Fluid maturity 80-95 %

FIGURE 14 Mesure de maturité Agile (Source Valtech)

L'ordre dans lequel les pratiques sont introduites est variable car dépendant des besoins de l'équipe et du projet. Le scénario suivant – non exhaustif – est une base de travail.

ORGANISATION	ENVIRONNEMENT	GESTION DE PROJET	INGÉNIERIE
<ul style="list-style-type: none"> Équipe pluridisciplinaire Membres de l'équipe assignés à 100% au projet Idéalement, l'équipe est co-localisée Participation du responsable de produit (Product Owner) 	<ul style="list-style-type: none"> Gestion de projet visuelle (tableau blanc) Gestion de la configuration logicielle Serveur d'intégration continue Outils collaboratifs (par exemple wiki) Outils de reporting et de suivi d'avancement 	<ul style="list-style-type: none"> Gestion du backlog de produit Estimations relatives Plan de release Définition de « done » et « done done » Réunions quotidiennes Planning d'itération Démonstration Rétrospective Indicateurs : burndown d'itération, vitesse, release burnup, prédictibilité, suivi de la dette technique 	<ul style="list-style-type: none"> Histoires utilisateur et critères d'acceptation (user story cards) Tests unitaires automatisés Remaniement du code (refactoring) Programmation en binôme Standards de code Tests fonctionnels automatisés Déploiement automatisé

TABLEAU 10 Ordre d'introduction des pratiques Agiles (Source Valtech)

4.5 TRANSFORMER : DÉPLOIEMENT ET OPTIMISATION EN CONTINU

Le déploiement à grande échelle des pratiques Agiles est source potentielle de nombreuses transformations dans l'entreprise. Nous abordons ici celles considérées communément comme nécessaires.

Stratégie et organisation

La stratégie de déploiement vise à construire progressivement des ensembles cohérents de plus en plus grands. La construction d'écosystèmes Agiles locaux (par département, par ligne de produit, par zone géographique, par type de produit ou de technologie, etc.) favorise entre autres :

- une gestion Agile des responsabilités des personnes – limitation de l'affectation d'une personne à de multiples projets en parallèle, constitution d'équipes durables,
- une gestion Agile du portefeuille de projets,
- un retour d'investissement rapide sur le coût de la transformation – application directe de pratiques expérimentées dans le même contexte.

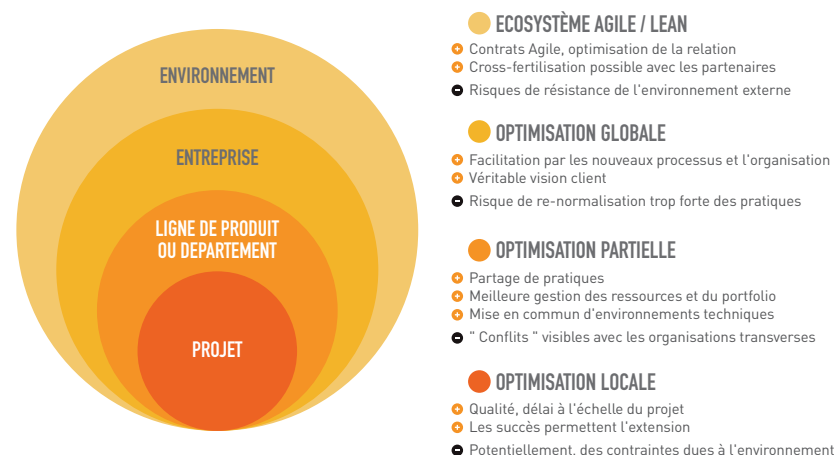


FIGURE 15 Le plan de déploiement (Source Valtech)

Le plan de déploiement est élaboré pour répondre aux enjeux identifiés lors du cadrage, et s'appuie sur le contenu du portefeuille de projets pour sélectionner les projets candidats prioritaires.

Le plan de déploiement tient également compte de facteurs déterminants :

- la disponibilité des personnes mobilisables (équipe de support en formation, coaching, mentoring, mise en place des environnements techniques adéquats),
- le temps nécessaire à l'apprentissage par les équipes de projet (3 à 6 mois pour les pratiques de développement Agiles et de gestion de projet),
- le temps nécessaire à l'adaptation des processus et de l'organisation.

Enfin, le plan de déploiement doit être révisé et mis à jour périodiquement, par exemple sur une base trimestrielle.

Le dispositif d'accompagnement pour la généralisation

Le tableau ci-dessous rassemble les principales interventions identifiées par Valtech dans des contextes variés ; les interventions, dans leur détail (contenu, rythme, durée) sont très dépendantes du niveau de service attendu, de la taille et de la complexité des organisations, et de l'expérience préalable des personnes en place ⁷.

INTERVENTION	PUBLIC VISÉ	SUJETS TRAITÉS
Pilotage de la transformation	Sponsor,	Établissement de la vision
	Responsables de domaines fonctionnels ou techniques,	Plan de déploiement
	DSI ou département R&D,	Suivi du déploiement
	Départements support	
Formation de coach interne (« Coach the coach »)	Futurs coachs internes	Techniques de coaching
		Approfondissement des pratiques Agiles
		Gestion du déploiement Agile
Coaching de responsable d'équipe	Management intermédiaire	Organisation d'équipe (réduction des silos, équipes pluridisciplinaires, équipes durables)
		Pilotage de l'activité (indicateurs de performance)
		Rôle du manager « entraîneur-facilitateur »
Coaching pour la gestion de portefeuille de projets	PMO,	Gestion d'un backlog de projets
	Responsables d'équipes	Mise en place de KANBAN
		Suivi et indicateurs d'avancement, tableaux de bord
Coaching et mise en place de pratiques	Responsables et opérationnels	Le reste de l'écosystème

TABLEAU 11 Dispositif d'accompagnement (Source Valtech)

7. Il est souvent bénéfique de compléter ce dispositif par des interventions de type « coaching personnel » pour faciliter la transition de « manager – gestionnaire » en « leader – facilitateur »

Agir sur l'organisation et sur les processus

Parmi les actions qui contribuent le plus significativement à la réussite d'un programme d'adoption de l'Agilité, on peut citer :

- **La réduction des silos organisationnels au sein des équipes techniques (DSI, R&D).** Une organisation en silos calqués sur le cycle en V est fréquemment rencontrée : des équipes de développement sont « soutenues » par des architectes provenant d'un autre département, les résultats étant testés par des ingénieurs en provenance d'un troisième département. Dans ce cas, chaque département « gère » ses personnels et les alloue temporairement aux projets en fonction des besoins... et surtout des disponibilités ! L'adoption des principes Agiles conduit à considérer les silos comme autant de freins à la communication directe, à la poursuite d'objectifs partagés, à la mise en place d'un planning favorisant la livraison au plus tôt des fonctions.
 - **Créer des équipes pluridisciplinaires intégrées, co-localisées et stables tout au long du projet**

- **La gestion « Agile ou lean » du portefeuille de projets.** Augmenter la valeur créée par les équipes de développement tient entre autres à deux facteurs :
 - la priorisation des projets en fonction de leur valeur métier. Les « clients » doivent participer intensivement à la priorisation du portefeuille de projets.
 - La réduction du nombre de projets menés en parallèle pour éviter que les ressources passent en permanence d'un contexte de projet à un autre.
 - **Prioriser les projets par la valeur métier, réduire le coût lié aux changements de contexte**

- **La transformation du rôle de manager gestionnaire en leader facilitateur.** Un des objectifs fréquemment assigné aux responsables d'équipes est d'optimiser le taux d'utilisation. Dans le rôle du facilitateur, le manager a pour objectif d'optimiser la valeur créée. Dans ce cadre, le manager devient un point d'escalade lorsque l'équipe rencontre des difficultés, et son rôle est d'éliminer ces difficultés pour favoriser la « production » de l'équipe.
 - **Les managers deviennent des acteurs de la production de valeur, au lieu d'en être des gestionnaires**

- **La mise en place de communautés Agiles.** Les communautés sont un vecteur organisé pour assurer la mise en commun des pratiques, des questions et interrogations, des échecs et réussites des projets. Selon la taille de l'organisation, des communautés locales seront installées (par département, localisation, ligne de produit, etc.) avec une communauté globale dont le but est de faire la synthèse des idées.
 - **Une communauté est un lieu d'échange et de génération d'idées. Elle accélère la transformation.**

- **L'adaptation des processus et de l'organisation de l'écosystème au rythme et à l'esprit des principes Agiles**
 - **La validation, la production :** adapter l'organisation et la disponibilité des environnements aux livraisons fréquentes. Cela peut constituer un véritable changement, quand on constate que les environnements de validation sont très souvent partagés entre plusieurs projets concurrents (ils doivent être réservés pour des créneaux fixes et déterminés longtemps à l'avance), et qu'il est courant que les équipes de production soient éloignées du développement.
 - **La sous-traitance, contractualisation :** les principes Agiles prennent tout leur sens dans le contexte de projets où l'incertitude et la variabilité règnent. Dans ce contexte, les relations contractuelles et opérationnelles doivent être adaptées pour permettre de la souplesse (contractualisation Agile), et pour assurer le bon niveau de transparence sur l'avancement et les points de blocage : les contrats au forfait pur ne sont plus adaptés à ces exigences.
 - **Méthodes et outils :** le corpus méthodologique existant – basé sur une démarche en cascade doit être abandonné au profit de la culture Agile. Un objectif majeur est de concevoir un référentiel méthodologique simple, léger et flexible, de ne normaliser que ce qui est nécessaire, et de réévaluer les pratiques en permanence.
 - **La cellule qualité, l'évaluation de la qualité :** dans le même temps le référentiel méthodologique évolue, l'évaluation de la qualité passe du contrôle de l'adéquation au processus (le projet suit-il les phases du cycle en V, la documentation est-elle conforme au standard ?) et du contrôle de l'adéquation aux exigences (le projet a-t-il fourni les fonctions telles que spécifiées ?), à une évaluation de la qualité du service rendu (les fonctions fournies sont-elles acceptées par le responsable du produit, sont-elles toutes utiles, n'en manque-t-il pas d'essentielles ?).
 - **Ressources humaines, évaluation de la performance :** alors que les pratiques Agiles encouragent et nécessitent une approche collaborative, collective, les systèmes de gestion de la performance des membres des projets sont souvent individualistes⁸. Ce domaine très sensible – il conduit au final à déterminer le niveau de rémunération ou l'avancement des personnes – doit être revu pour introduire des indicateurs de performance collective basés sur des critères tels que le nombre de fonctionnalités livrées, la valeur métier livrée, ou des critères Agiles tels que la vitesse, ou la prédictibilité de l'équipe
 - **Créer un écosystème adapté est le passage obligé pour un retour sur investissement significatif**

8. Nous avons rencontrés des cas où la performance est mesurée en nombre de lignes de codes produites, en nombre de défauts détectés ou autres mesures, certes relativement simples à acquérir, mais dont la valeur réelle est très faible !

Adapter l'infrastructure et les environnements de travail

La plate-forme technique de développement est un sujet essentiel, mais ce point ne sera pas détaillé ici car la littérature abonde de descriptions de ce qu'est un environnement de développement Agile.

Par contre, la nécessité d'amplifier la collaboration entre les membres d'un projet apporte des changements notoires du côté des outils non techniques.

Dans le cas d'un projet d'une demi-douzaine de participants, travaillant sur le même site, avec un Product Owner proche et disponible, la conversation face à face, l'utilisation de Post-it® et de schémas sur des tableaux muraux seront suffisants pour assurer la communication.

En environnement distribué, avec une équipe plus importante il devient absolument nécessaire de disposer des outils qui vont fluidifier et sécuriser les flux d'information, et tenter de produire une certaine « illusion de localité ». Au-delà des systèmes de mails ou de messagerie instantanée (l'information n'est pas toujours partagée, les fils d'information ne sont pas structurés ni archivés), une plate-forme collaborative doit être installée.

A minima, une telle plate-forme comprend des **outils de partage/stockage** de documents et informations digitales :



Répertoires partagés, outils de gestion et de partage de contenu Sharepoint, documents Google.



Outils de stockage et de structuration d'informations et de documents tels que les Wiki, forums, FAQ.



Outillage nécessaire à la gestion Agile du projet (backlog, estimations et reste à faire, iteration burndown, release burnup, vélocité, etc.). L'offre est aujourd'hui très étoffée.

Pour remplacer les discussions en face-à-face, les **outils de télécommunication** sont amenés à évoluer :



Téléphonie sur IP, Skype (pour contenir les coûts !)



Visioconférence, éventuellement à base de webcams, outils de conférence sur le Web (tel que Microsoft Office Live Meeting)

Évaluer les progrès et les résultats

La transformation est évaluée sur deux plans. Le premier plan est une mesure des progrès du déploiement ; le second plan est une mesure des effets de la transformation.

Évaluer l'avancement du projet de transformation

L'appréciation de l'avancement du projet de transformation est, par exemple, simplement réalisée par comptage en nombre de projets ou d'équipes, et par une cartographie des départements ou lignes de produits qui ont adopté l'Agilité.

Ensuite, des données plus subjectives sont recueillies par enquête auprès des différents intervenants des projets pour avoir leur perception des effets de la transformation : moral des équipes, perception du niveau de reconnaissance, appréciation sur les nouvelles conditions de travail.

Enfin, si des communautés ont été mises en place, leur activité doit être appréciée pour s'assurer de leur utilité et du bon fonctionnement du dispositif : fréquence des réunions, assiduité, résultats produits, qualité de la communication, etc.

Mesurer les effets de l'adoption de la démarche et des pratiques Agiles

Les résultats escomptés de la transformation seront mesurés en fonction des enjeux et des objectifs de départ ; même s'il est complexe d'établir un ROI pour un tel projet, il n'en reste pas moins qu'il aura une traduction économique.

La mesure de la rentabilité est très hasardeuse lorsqu'elle est fondée sur des hypothèses du type « si nous avons fait le projet comme avant, il aurait coûté X, mais il a en fait coûté Y, donc le Retour sur Investissement est de $X - Y$ ».

Les bénéfices apportés par l'Agilité sont mesurés par l'appréciation des variations de la qualité de service.

Du point de vue des clients, cela se traduit par :

- une amélioration des délais de mise à disposition des fonctions (time to market),
- l'utilité des fonctions développées (plutôt que le respect du cahier des charges initial),
- la visibilité sur l'avancement tout au long du projet,
- une diminution des défauts constatés durant la validation et après la mise en production.

Pour l'entreprise, en fonction des objectifs initiaux, la mesure de la réussite est établie par exemple par :

- la variation des parts de marché,
- la diminution des coûts de maintenance,
- la réduction des appels au support utilisateur (help desk).

Ces résultats sont obtenus sur le moyen ou long terme, par la mise en place de tableaux de bords spécifiques et d'enquêtes de satisfaction auprès d'utilisateurs ou de clients.

Enfin, une synthèse est élaborée pour suivre la progression du nombre de projets réussis, mitigés ou en échec... et les comparer aux études publiées (par exemple l'étude du Standish group) ou à la situation qui prévalait avant la transformation...

L'ESSENTIEL À RETENIR

Nous rappelons les points essentiels de la transformation vers l'Agilité :

- Identifier les enjeux et les objectifs de la transformation (objectifs de l'entreprise, facteurs de succès),
- Définir la vision du changement (périmètre, roadmap, critères de succès),
- S'assurer un support fort et constant au plus haut niveau de l'organisation (sponsor),
- Acquérir les moyens de la transformation (formation, mentoring, coaching, outillage),
- Ancrer la transformation dans la durée (communautés, coachs internes, évolution du management intermédiaire, transformation de l'écosystème),
- Suivre le projet avec un tableau de bord complet (couverture, effets économiques, adhésion des personnes).

5
....

LES DIFFICULTÉS À SURMONTER



OU COMMENT PRÉVENIR LES RISQUES ET LEVER LES RÉTICENCES

5.1 LES DIFFICULTÉS COURAMMENT RENCONTRÉES

L'adoption de l'Agilité est une démarche d'entreprise qui peut se limiter à un projet. Cependant de la même manière que le CMMI® s'adresse à une organisation et non à un projet unique, l'adoption de l'Agilité peut également avoir comme périmètre, l'ensemble des projets d'une organisation.

HUBERT GILLON

// Delivery Manager
// Valtech

“

Une des premières difficultés est le « passage à l'acte » qui suppose que le management soit convaincu des bienfaits d'une telle approche pour les clients et pour les projets délivrant les applications ou les produits.

”

Autre difficulté souvent rencontrée, celle d'adopter l'état d'esprit Agile, qui suppose de ne pas se contenter de faire son marché dans les pratiques Agiles. En effet, l'adoption de l'Agilité est l'affaire de tous et non de certains acteurs qui mettraient en œuvre telle ou telle pratique Agile, pensant que cela limiterait les risques d'échec.

En supposant que tout le monde soit convaincu de l'intérêt d'adopter l'Agilité, la mise en œuvre concrète des pratiques telles que planning games, rétrospectives et backlogs produits suppose une **bonne maîtrise des concepts** sous-jacents ainsi qu'une **expérience concrète des projets**.

- Si l'on prend l'exemple des tâches des backlogs d'itération, il arrive très souvent que les chefs de projet se retrouvent littéralement noyés par le nombre de tâches à gérer. La raison : la granularité des tâches gérées est souvent bien trop fine, ce qui induit évidemment un nombre déraisonnable de tâches : une équipe de 20 personnes définissant à chaque itération de 4 semaines des tâches de 2 heures en charge en moyenne, se traduit par 1600 tâches !
- Le fait de fonctionner en time boxing (durée d'itération fixe quoiqu'il arrive et quelque soit le résultat de l'itération), de livrer des documents partiellement réalisés (contraire à notre culture), ou d'identifier 3 voire 4 améliorations à réaliser sur la prochaine itération à l'issue d'une rétrospective, constituent autant de difficultés et de pièges qu'il faut à tout prix éviter, au risque de discréditer l'Agilité à jamais.

Aussi est-t-il très important de faire intervenir un Scrum Master et des experts des pratiques Agiles pour se faire aider dans la mise en œuvre de l'Agilité.

La granularité des tâches aurait oscillé entre 4 heures et 2 jours par tâche ce qui aurait conduit à 5 fois moins de tâches à gérer, le projet aurait été livré à l'heure, et le nombre d'améliorations aurait été de 1 par itération au maximum.

On peut donc affirmer aujourd'hui que l'Agilité constitue un outil réellement efficace pour favoriser l'utilisation d'équipes distantes en nearshore ou en offshore, capable de réduire la distance entre les hommes au niveau organisationnel, démarche projet et outil, et de créer l'esprit d'équipe si bénéfique à la réalisation des projets informatiques.

L'ESSENTIEL À RETENIR

Pour être Agile :

- il ne faut pas détailler toutes les exigences fonctionnelles avant de démarrer les développements,
- on ne prévoit pas entièrement à l'avance les détails, pour éviter le gaspillage,
- une machine est dédiée à l'intégration continue,
- il faut astreindre les développeurs à poster plusieurs fois par jour leur travail sur le dépôt central,
- il faut adopter collégalement l'état d'esprit Agile,
- la mise en œuvre concrète des pratiques telles que les planning games, rétrospectives et backlogs produit suppose une bonne maîtrise des concepts sous-jacents ainsi qu'une expérience projet concrète,
- il est important de faire intervenir un Scrum Master et des experts des pratiques Agiles pour se faire aider dans la mise en œuvre de l'Agilité.

5.2 LA GESTION DU STRESS DANS LES ÉQUIPES AGILES

Si l'intérêt des méthodes Agiles n'est plus à démontrer, comme l'indique le nombre croissant de projets de ce type, l'expérience montre également les risques de pression et de stress qui peuvent découler de ces pratiques. En tant qu'expert, Valtech est confronté à ces problématiques et souhaite avertir les équipes Agiles et les responsables des dérives potentielles, mais surtout leur apporter des propositions et solutions concrètes.

La problématique

Une des forces de l'Agilité est de mesurer précisément, à court, moyen et long termes, l'avancement et le reste-à-faire d'un projet :

- tous les jours, lors de réunions quotidiennes,
- toutes les 2 ou 3 semaines, lors de démonstrations, rétrospectives et réunions de planification d'itérations,
- tous les 3 ou 4 mois, lors de livraisons de releases.

Cela implique pour chaque membre de l'équipe un engagement de tous les instants. Si cette méthode permet d'améliorer la réactivité aux changements, grâce aux métriques utilisées et à la collaboration sans cesse favorisée, elle peut dans certains cas devenir une source de pression et de stress pour l'équipe ou pour l'individu.

Les dérives et leurs conséquences

Au travers de ses expériences, Valtech a constaté des dérives liées au manque de maîtrise du rythme sur les projets Agiles, en particulier celles exposées ci-après :

La recherche constante de la sur-performance

L'amélioration continue, vertu indiscutable d'un processus Agile, est parfois confondue avec la sur-performance. En d'autres termes, « faire mieux » ne signifie pas forcément « faire plus ». Ainsi, la mesure et le suivi de la vélocité d'une équipe à chaque fin d'itération, peut entraîner l'équipe à se fixer des objectifs toujours plus ambitieux d'une itération à une autre, sans tenir compte de sa capacité réelle. Cela entraîne à terme un épuisement de l'équipe, une baisse de qualité et peut parfois se traduire par un « accident de livraison d'itération » avec une vélocité proche de 0.

Un engagement individuel qui nuit à l'engagement de l'équipe

Certains individus supportent mal de ne pas tenir leurs engagements personnels quotidiens. D'autres encore ont tendance à comparer leurs performances individuelles à celles des autres équipiers au lieu de se préoccuper des objectifs communs et favoriser le travail en équipe. Cela se traduit par une baisse de motivation et de performance.

Une livraison imposée à périmètre constant

Le Product Owner, le client ou le management pousse parfois l'équipe Agile à faire des livraisons à des dates imposées (time boxing) et sans marge de manœuvre sur le périmètre fonctionnel livré. L'équipe perd alors de son autonomie de décision, et est contrainte de détériorer la qualité de l'application en réalisant moins de tests par exemple. La pression en résultant désolidarise l'équipe, et peut impacter la santé morale et physique de certains.

Quelques solutions

Afin d'éviter ces dérives, Valtech a pu mettre en œuvre des solutions permettant d'améliorer la collaboration entre les individus et donc de contribuer à la réussite des projets.

Un rythme régulier soutenu et soutenable

Toutes les parties prenantes d'un projet se doivent d'instaurer un rythme régulier soutenu et soutenable et veiller à son maintien, quitte à se fixer des objectifs moins ambitieux à court terme et à privilégier la performance à moyen et long termes, plutôt que la performance à très court terme.

Des rétrospectives ouvertes et sincères

A la fin d'une itération, chaque équipier doit évoquer sincèrement les problèmes qu'il a rencontrés ; attention aux « fausses rétrospectives » qui ne mettent en exergue que des problèmes superficiels. Afin d'éviter les non-dits et les rétrospectives difficiles quand la pression est trop forte, il est conseillé de faire intervenir une personne externe au projet qui aura une vision plus objective et pourra mieux exprimer les problématiques sous-jacentes.

Pas de sur-engagement

Une estimation de charge trop optimiste ou approximative peut provoquer de mauvaises surprises lors de la réalisation du projet. Les estimations doivent systématiquement être collectives, partagées avec le client et se baser sur les résultats constatés lors des itérations précédentes.

Pas d'engagement contractuel « aveugle »

Un engagement contractuel rigide visant à fixer à l'avance les délais de livraison, les coûts et le périmètre fonctionnel d'un projet, est souvent en décalage avec les engagements opérationnels pouvant être pris par une équipe Agile. Cela peut mener à l'échec du projet. Il faut au contraire veiller à établir des contrats

flexibles, permettant d'une part la satisfaction des objectifs métiers et d'autre part l'adaptation aux capacités de réalisation opérationnelles.

Ces solutions peuvent aider les responsables projets à éviter les dérives, un projet Agile, c'est avant tout une histoire humaine, un travail d'équipe basé sur l'échange et la collaboration. Fort de ces expériences dans la mise en œuvre des méthodes Agiles, Valtech propose à ses clients un accompagnement par le « coaching Agile » permettant de mener à terme leurs projets dans les meilleures conditions pour les équipes et donc pour le client.

L'ESSENTIEL À RETENIR

Pour être sûr de pérenniser la mise en place de l'Agilité dans une équipe, il faut prendre garde à :

- maintenir un rythme soutenu mais soutenable,
- ne pas dissimuler les vrais problèmes lors des rétrospectives,
- éviter le sur-engagement,
- donner de la flexibilité au contrat.

Les attentes de chaque partie prenante

Précurseur dans les méthodes Agiles, Valtech a été le premier à mettre en place la contractualisation Agile dans ses projets. Si jusqu'à présent la confiance des clients envers la société a permis l'acceptation de ces contrats, il s'agit aujourd'hui de proposer des « standards » de contrats Agiles, adaptés à différents contextes. Valtech a travaillé avec des directeurs informatiques, des juristes, des intégrateurs, des experts Agiles indépendants afin d'identifier les exigences des projets et les attentes des différents intervenants et de déterminer les évolutions à apporter aux contrats actuels.

Comment concilier les exigences fonctionnelles et budgétaires du client avec les contraintes opérationnelles et financières du prestataire ? Comment faire collaborer les équipes du client et du prestataire autour d'un projet qui évolue au cours de son développement ? Comment transcrire juridiquement ces différents éléments permettant de rassurer et protéger client et prestataire dans un même objectif de réussite du projet ? Comment faire adhérer les différents services concernés à ce nouveau type de collaboration ?

Malgré certaines divergences inhérentes à la position de chacun, client/prestataire, ou à sa fonction, technique/juridique, il émerge un certain nombre de points d'accord à intégrer dans les contrats Agiles et ce, selon deux grands axes : les concepts et les dogmes.

Les nouveaux concepts à intégrer dans un contrat Agile

Les méthodes Agiles étant principalement axées sur la collaboration et sur les hommes, les éléments qui déterminent un projet Agile sont basés sur de nouveaux concepts à redéfinir afin que les différentes parties aient la même vision du déroulement du projet :

- **Unité de valeur** : indispensable pour déterminer les priorités de développement, la valeur des fonctionnalités du projet est définie en fonction des critères de priorité, d'utilisation et de besoin définis par le client. Cette unité de valeur est spécifique à chaque client et chaque projet. Les clauses contractuelles doivent permettre au client de décider de l'arrêt ou de la poursuite d'un projet dès lors qu'un minimum d'unités de valeur a été produit.
- **Point de complexité** : en parallèle, le point de complexité établit le rapport entre les exigences du client et l'effort de réalisation, afin de mesurer le bénéfice de chaque fonctionnalité par rapport à la tâche que cela représente. Dans un contrat Agile, le prestataire s'engage à réaliser un nombre de points de complexité dans un budget donné.
- **Backlog de produit** : évolutif, ce référentiel contient la liste des besoins fonctionnels (et parfois techniques), ainsi que leur valeur et leur complexité.
- **Transparence et collaboration** : principe de base de la réussite d'un projet Agile, la relation entre les équipes du client et celles du prestataire nécessite un partage et un échange quotidiens pour une collaboration sereine

5.3 LA CONTRACTUALISATION AGILE

➤ Une évolution inéluctable pour toutes les parties prenantes de l'entreprise.

NATHALIE LOPEZ-SAUSSIER

// Directeur Général Adjoint
// Valtech Technology

« Clients, juristes, intégrateurs et experts des méthodes Agiles jugent inéluctable l'adaptation des contrats aux nouvelles pratiques de réalisation Agiles. Ils commencent à poser ensemble les fondements de nouveaux contrats flexibles répondant au mieux aux exigences et contraintes de chaque partie prenante et de l'entreprise. »

Avec l'intérêt croissant des entreprises pour l'utilisation des méthodes Agiles dans les projets IT, il est devenu nécessaire de faire évoluer les contrats classiques en contrats Agiles adaptés à ces nouveaux principes et méthodes de collaboration. Outil indispensable et obligatoire, le contrat ne doit plus être perçu comme un frein mais comme un outil administratif et juridique destiné à accompagner les différentes parties et répondant aux contraintes des méthodes Agiles.

et efficace. La transparence se traduit contractuellement par un accès permanent aux indicateurs de pilotage court, moyen, long termes du projet (itération, release, Product Backlog, indicateurs de vélocité et de qualité). La devoir de collaboration doit être explicite dans le contrat, principalement pour fiabiliser les estimations de complexité.

- **Exploitabilité continue** : à chaque fin de période (itération), les fonctionnalités livrées sont opérationnelles et peuvent être mises en production. Ainsi, leur utilisation immédiate permet de les modifier ou d'adapter le développement au fur et à mesure de l'avancement du projet.

Evolution des dogmes et fondamentaux contractuels basés sur la collaboration



Dans le contrat Agile, les engagements du prestataire et du client portent sur les méthodes de collaboration, les moyens, les outils, la qualité et la vélocité plutôt qu'exclusivement sur le périmètre du produit final, le délai et le budget. Les projets sont caractérisés par des notions d'unité de temps, de lieu et de valeur plutôt que par un objectif fixe et contraignant.



Pour répondre à ces nouveaux aspects contractuels, ce sont les dogmes du contrat qui doivent être adaptés :

- **La réalisation continue et le démarrage « au plus tôt »** : à la différence des projets classiques où les actions sont menées par phases successives (spécification, conception, développement, tests, recette) avec les risques « d'effet tunnel » et de litiges à la livraison que cela peut entraîner, les projets Agiles sont construits et réalisés par itérations courtes (2 à 3 semaines) fournissant des incréments fonctionnels démontrables et favorisant la collaboration entre le client et le prestataire. Le client n'est plus amené à valider des documents intermédiaires techniques mais à accepter directement une partie du produit final. Les actions se mènent de front et de façon continue, les modifications sont intégrées et les priorités sont évaluées au fur et à mesure de la réalisation en fonction des notions de valeurs définies par le client. Ainsi, l'engagement du prestataire est de fournir itérativement des fonctionnalités démontrables ; celui du client est d'assurer un flux continu de demandes et de valider le produit à chaque livraison d'itération.

- **Le copilotage et le coworking** : le projet est mené sur la base d'un véritable co-working qui inclut une réelle collaboration entre les parties au niveau décisionnel et opérationnel. Il n'y a pas d'effet navette entre les équipes du client et du prestataire puisqu'elles travaillent ensemble sur le projet et font le point tous les jours sur l'avancée du développement et les décisions de pilotage.
- **L'engagement du prestataire sur son équipe** : au-delà des compétences de l'équipe, le prestataire s'engage sur la pérennité et la flexibilité de son équipe (diminution de personnel ou montée en charge) afin de s'adapter aux besoins du projet. L'entente entre les hommes et leurs compétences étant indispensable à la réussite du projet, le client a un droit de regard sur le choix des intervenants du prestataire.

Pour que les projets bénéficient des innovations apportées par les méthodes Agiles, il est indispensable d'intégrer ces notions dans le contrat, de mettre en avant méthodologies et outils de collaboration.

La contractualisation Agile : une évolution culturelle

Si la contractualisation Agile est une notion complètement assimilée pour Valtech et ses clients, c'est effectivement une véritable évolution culturelle qui doit être menée dans les entreprises, que ce soit chez les clients ou chez les prestataires. Il s'agit à présent de faire accepter un contrat flexible, où l'**engagement** ne porte plus sur le triptyque périmètre, délai, budget – souvent au détriment de la qualité – mais sur une combinaison subtile de la **vélocité**, du **coût du point de complexité**, de la **qualité**, et de la mise en œuvre de **pratiques itératives et collaboratives** – au service de la satisfaction client.

De ce constat, il ressort la nécessité de mettre en place une communication accrue entre les services concernés, qu'ils soient techniques, « achats » et/ou juridiques, avec une impulsion qui doit venir de la direction. La direction a effectivement la vision et l'autorité nécessaires pour déterminer les objectifs communs aux différents intervenants et engager la société.

Fort de ces constatations et de l'implication de ces partenaires, Valtech poursuit son action dans l'élaboration de contrats standards favorisant le développement des méthodes Agiles.



THOMAS BEAUGRAND

// Avocat spécialiste des contrats et des méthodes Agiles
// Cabinet Staub & associés

L'ESSENTIEL À RETENIR

Pour être certain de pérenniser la mise en place de l'Agilité dans une équipe, il faut prendre garde à :

- maintenir un rythme soutenu mais soutenable,
- ne pas dissimuler les vrais problèmes lors des rétrospectives,
- éviter le sur-engagement,
- donner de la flexibilité au contrat.

5.4 L'EXTERNALISATION AGILE

Fort du retour d'expérience résultant de l'adoption des méthodes Agiles dès leur émergence et de la formation sur la transformation Agile reçue de Craig Larman en 2003, Valtech a étendu les pratiques Agiles de Scrum et XP à son centre de développement offshore à Bangalore, en Inde et a adopté des pratiques de collaboration à distance qui ont grandement amélioré le modèle de développement duo-shore.

Ce chapitre présente un aperçu des pratiques Agiles dans le modèle duo-shore de Valtech.

L'Approche Valtech de la délocalisation

Chez Valtech, nous sommes convaincus que la confiance est au cœur de la réussite de la délocalisation Agile. Une fois la confiance établie, les processus en place sont mieux acceptés par toutes les parties prenantes.

Les équipes Valtech ont introduit de nouvelles pratiques dans leurs activités quotidiennes, pour établir un bon niveau de confiance dans l'équipe de projet et avec les clients, sans compromettre les valeurs et les principes Agiles.

On citera par exemple:

- des modèles contractuels fondés sur la valeur livrée,
- une plus grande collaboration entre les équipes par la mise en place des réunions quotidiennes inter-équipes (Scrum of scrums), de protocoles d'équipes, de salles de réunion virtuelles, de voyages (vers l'Inde et depuis l'Inde) et l'utilisation partagée de wikis et d'outils de gestion du cycle de vie des applications,
- des démonstrations et des livraisons fréquentes, des rétrospectives auxquelles le client est associé,
- le développement de matériel de formation ad-hoc pour optimiser la courbe d'apprentissage des nouveaux membres au sein des équipes projets.

Les sections ci-dessous présentent brièvement certaines de ces pratiques.

Les modèles de livraison

Le contrat Agile

La contractualisation Agile a été adoptée dans le modèle duo-shore. Outre les concepts intégrés dans les contrats Agiles présentés dans un chapitre précédent, le développement en mode duo-shore nécessite les adaptations suivantes :

- un budget de voyage est déterminé lors du lancement du projet,
- la facturation est basée sur les unités de valeur livrées (estimées en points de complexité), plutôt que sur des jalons classiques,
- la désignation par le client d'un Product Owner dédié au projet est impérative,
- la flexibilité des horaires de travail doit être mutuellement acceptée, en raison du décalage horaire potentiel,
- le contrat tient compte des exigences de communication entre les équipes localisées onshore et offshore, et tient compte des frais généraux associés.

Cycles de nouvelles versions et itérations

Pour assurer une bonne prédictibilité de la vélocité de l'équipe⁹, et assurer également que la qualité des fonctions réalisées sera acceptable, il est nécessaire de tenir compte de l'effort nécessaire au transfert de connaissances et de compétences. Malgré les pratiques mises en place, le transfert de connaissance sur un sujet complexe prend plus de temps lorsque l'équipe est à distance, tout simplement parce que la communication directe est moins « fluide » (pas de conversations possibles à la machine à café !)

9. La vélocité mesure la « quantité » fonctionnelle construite et livrée en état opérationnel au cours d'une itération par l'équipe de projet.

Techniques de collaboration

Protocoles d'équipes

Un protocole d'équipe décrit des règles, généralement simples, pour guider la réalisation d'une activité de l'équipe. Certains protocoles peuvent être mis en place au démarrage d'un projet, et d'autres sont issus de l'adaptation de l'équipe à de nouvelles situations.

Exemple 1 : une équipe a mis en place un protocole pour enregistrer les sessions de « chat » entre les membres de l'équipe. Chaque session débute par une ligne « Objet » et se termine par un « Résumé ». Une fois complétée, la totalité de la conversation est publiée dans le wiki de référence.

Exemple 2 : une autre équipe a créé un protocole de gestion des anomalies. Lorsqu'un défaut est découvert, il est ajouté comme une tâche sur le tableau blanc. Si après 4 heures d'attente le défaut n'est pas corrigé, il est considéré comme une anomalie et géré en tant que tel.

Exemple 3 : pour pallier une indisponibilité temporaire du client, une équipe offshore a créé un protocole qui lui permet de poursuivre la définition fonctionnelle de l'application : des membres de l'équipe offshore écrivent des user stories, et les soumettent au client via le wiki du projet. Le client a alors 48 heures pour commenter ou valider la proposition.

Exemple 4 : pour améliorer la lisibilité du tableau des tâches, une équipe a décidé de regrouper les user stories en catégories, et chaque catégorie est caractérisée par des cartes de couleurs différentes.



FIGURE 16 Tableau de Scrum avec les différents types d'histoires utilisateur (Source Valtech)

Les protocoles d'équipe sont généralement applicables localement (ils ne sont pas contractuels), car ils sont la traduction opérationnelle des résultats des rétrospectives de fin d'itération.

Outils légers

Les outils collaboratifs comme un wiki et les outils de gestion de projet comme JIRA et Rally qui sont disponibles au travers d'accès web, se sont avérés des moyens de communication plus efficaces que les outils de communication point-à-point, tels que les e-mails.

Les outils de messagerie instantanée comme Skype ou Google Talk offrent la possibilité, à coût négligeable, d'établir des communications directes avec des outils de partage de documents ou de vidéo conférence.

Nous avons plusieurs outils pour améliorer la collaboration, la productivité et le contrôle de qualité. Certains de ces outils ont été développés par nous-mêmes, d'autres ont été développés par nos prestataires.

JIRA est utilisé pour la gestion de projet, la collecte d'indicateurs de performance, le suivi des exigences et des anomalies.

Le wiki Confluence est utilisé comme plate-forme de partage d'informations pour tous les projets. Les informations concernant un projet (avancement, indicateurs), le produit (spécifications, architecture, etc.) ou l'équipe (contacts, rôles, etc.) sont publiées dans le wiki et rendues accessibles à tous les membres du projet ainsi qu'au client. Avec les fonctionnalités de gestion des versions des pages et des documents, et l'envoi de mails lorsque des modifications de contenu sont introduites, l'utilisation d'un wiki est extrêmement efficace. Pour chaque projet, une Usine de Développement Logiciel (Software Factory) est mise en place pour prendre en charge l'intégration continue (utilisation de CruiseControl ou de Hudson), l'automatisation de l'exécution des tests, et l'analyse statique de code.

Valtech Quality Suite intègre de multiples outils d'analyse de code statique (PMD, Checkstyle, Dependometer, Juca etc.), et fournit un mécanisme de génération de rapport efficace concernant la qualité du code et la couverture de tests.

Salle de réunion virtuelle : les équipes utilisent couramment des outils de vidéo conférence simples (webcam + casque et micro, ou téléphone + flux vidéo projeté sur un écran) lors des réunions avec les membres de l'équipe. La qualité de la communication est très grandement améliorée.

LoFAT est un wrapper pour Selenium et QTP visant à réduire l'effort requis par les testeurs pour créer et maintenir des scripts de test automatisés.

Structure d'équipe

Pour une équipe de taille conséquente (par exemple effectif > 20), il est préférable de diviser l'équipe en groupes pluridisciplinaires centrés sur des unités fonctionnelles du produit. Dans cette organisation (feature team), chaque groupe est responsable du développement et de l'intégration d'une unité fonctionnelle (feature).

Chaque feature team est composée d'analystes, de concepteurs-développeurs, de testeurs, guidés par un Scrum Master.



FIGURE 17

Product Owner en train d'expliquer les nuances des fonctionnalités à l'équipe

Le transfert des connaissances

La mise en place de principes Agiles de collaboration a également amélioré le transfert des connaissances dans le modèle de développement duoshore. Notamment :

- Les sessions de travail en binôme (2 développeurs, 2 testeurs ou 1 développeur / 1 testeur, travaillant ensemble sur une même tâche) permettent le transfert des connaissances par la pratique, une meilleure gestion des risques associés au départ de membres de l'équipe.
- Il a été possible de maintenir la collaboration à distance en utilisant des outils de communication optimisés pour le Web comme WebEx, Interwise ou bien en utilisant des outils de prise de commande de PC à distance, tels que VNC ou RDC, en combinaison avec les méthodes de téléphonie sur le Web tels que Skype. La vidéoconférence s'est également avérée être un outil très efficace pour assurer la collaboration rapide entre les équipes distribuées.
- Une visite du site onshore par le Product Owner et par les architectes de l'équipe onshore au démarrage d'un nouveau projet est le moyen le plus efficace pour partager la connaissance du domaine, des besoins fonctionnels et pour développer l'architecture du produit. Une bon modèle du domaine peut être construit dans ces conditions, pour assurer la bonne compréhension des besoins techniques du projet.

Dimension culturelle



Traditionnellement le voyage se produisait une fois, au début du projet et plus tard, au cours du déploiement. Plusieurs fois, seulement un ou deux membres de l'équipe offshore rendaient visite au client. Il n'y a eu aucune tentative de comprendre les dimensions culturelles de l'autre équipe. Nous avons fait une tentative pour répondre à cette préoccupation en planifiant plus de voyages.

Les voyages fréquents nous ont aidés à approfondir la liaison entre les équipes. Les personnes qui voyagent participent aux activités culturelles et sportives, visitent les lieux historiques avec les membres d'équipe pour mieux comprendre la culture du pays. Ces initiatives d'interactions informelles sur différents sujets nous aident à comprendre la culture des autres.



La fréquence des voyages du client sur le site de développement (Product Owners ou autres parties prenantes) a une influence directe sur la qualité des applications produites. Le coût de ces déplacements est amorti par la réduction des « gaspillages » et des pertes d'efficacité.



FIGURE 18 Accueil d'un membre de l'équipe onshore par ses homologues offshore à Bangalore.

Au cours des visites, la communication directe permet de répondre rapidement à des questions de détail qui n'auraient pas été abordées en communication « à distance », et permet de recentrer l'ensemble des participants sur la vision et les enjeux du projet.

Les échanges directs construisent l'équipe, renforcent la collaboration, chacun étant conscient et au fait des préoccupations et des attentes des uns et des autres.

Infrastructure

En bonne pratique, toutes les activités liées à la mise en place de l'infrastructure sont complétées au démarrage d'un nouveau projet avec toutes les activités dites de calibrage du projet, ce qui comprend le transfert des connaissances du produit. Dans la méthodologie Scrum adoptée par Valtech, cette période de calibrage porte le nom de l'itération de calibrage ou itération-0. Aucune livraison de nouvelle fonctionnalité du produit ne survient durant cette période.

Les activités de mise en place de l'infrastructure suivante sont complétées lors de l'itération 0 :

- mise en place d'un dépôt de code source commun pour onshore et offshore,
- mise en place d'un serveur d'intégration continue du code source qui soit complètement fonctionnel, c'est-à-dire qui soit capable de déclencher une nouvelle construction à chaque modification du code source, d'exécuter les tests unitaires, les tests fonctionnels et les outils d'analyse de code statique, et de signaler les anomalies détectées aux membres de l'équipe,
- mise en place d'outils de gestion de projet et d'un wiki accessible à tous les membres de l'équipe et que chacun sache utiliser,
- chaque membre de l'équipe possède deux moniteurs,
- mise en place d'un espace de travail ouvert pour la circulation libre d'informations entre les membres de l'équipe,
- vérification que toute information partagée soit accessible par l'équipe onshore, les clients, etc.



PARIJAT SINHA
// Practice Head
// Valtech India



FIGURE 19 Espace de travail ouvert entouré de tableaux blancs chez Valtech India

Le gaspillage selon la définition Agile

Les méthodologies Agiles, y compris Scrum, considèrent comme du gaspillage toute tâche qui n'apporte pas de valeur au client. Par expérience, nous identifions les tâches suivantes comme du gaspillage :

- la création de documents élaborés pour le suivi de conformité, concernant par exemple les revues de code. Ils sont avantageusement remplacés par des notes rapides ou des discussions entre développeurs,
- la sur-qualité dans la documentation de conception, utilisation poussée d'UML. Bien souvent, des schémas dessinés au tableau, visibles de tous en permanence, sont appropriés. La documentation nécessaire à la maintenance peut être élaborée après coup,
- la rédaction de comptes-rendus de réunion (réunion quotidienne, rétrospective). Le résultat de la réunion quotidienne est traduit en modifications du task board, les conclusions des rétrospectives peuvent être incluses comme nouvelles tâches, ou formalisées en protocole d'équipe,
- les réunions mal préparées et qui n'aboutissent à aucune conclusion en raison de l'absence d'agenda et d'objectif,
- la duplication d'information,
- la production de multiples rapports d'avancement, chacun destiné à un « lecteur » différent (client, chef de projet, etc.) mais tous basés sur les mêmes données,
- la mise en place d'intermédiaires dans la communication entre l'équipe et le client (fonction de coordinateur),
- la non adhérence au bon protocole pour toute forme de communication. Mécompréhension due aux assumptions et intentions qui ne sont pas transparentes,

- la multiplication de systèmes de contrôle de version (par exemple, un pour les développeurs onshore, un autre pour les développeurs offshore ou pour le client),
- l'utilisation d'e-mails et de rapports comme source primaire d'échange d'information.

L'ESSENTIEL À RETENIR

- Au fur et à mesure que les équipes commençaient à gagner en maturité dans l'Agile, nous avons commencé à trouver nos propres modèles de travail. Notre engagement onshore-offshore est maintenant piloté par les valeurs et les principes Agiles.

5.5 L'AGILITÉ FACE AUX AUTRES STANDARDS

Qualité du produit ou des processus : Agilité ou ISO?

Les méthodes Agiles se décrivent souvent par une approche peu formelle où le focus est mis sur la collaboration entre les personnes et sur le produit à réaliser. A contrario, les méthodes « ISO » sont souvent qualifiées de procédurières voire de contraignantes vis à vis des règles applicables, et plus orientées processus que produit. Ceci semble les opposer, pourtant ...

Le management par la qualité vise à définir des objectifs qualité au niveau de l'entreprise qui se déclinent ensuite par nature d'activité. Il s'agit de planifier et de définir les méthodes nécessaires à la maîtrise des prestations et des produits, et de mesurer le niveau d'efficacité atteint. On traduit souvent cette approche par le concept de « PDCA » (Plan, Do, Check, Act).

Ce mode de management est assez proche de la logique d'itération en Agile avec notamment sa planification des itérations, ses bilans d'itération pour en mesurer les résultats et ses rétrospectives, pour décider des améliorations à réaliser dans les itérations futures. « Agilité » et « ISO » se rejoignent donc sur des valeurs communes de planification opérationnelle et d'amélioration continue.

Ensuite, la dimension majeure de l'ISO 9001 est son approche processus qui vise à maîtriser l'ensemble des activités d'un organisme en cohérence avec la politique qualité de sa direction et les exigences métier et réglementaires pour satisfaire au mieux les clients. Les méthodes Agiles apportent cette même dimension de maîtrise projet et produits avec également une forte implication des clients. « Agilité » et « ISO » se retrouvent donc à nouveau sur ces valeurs.

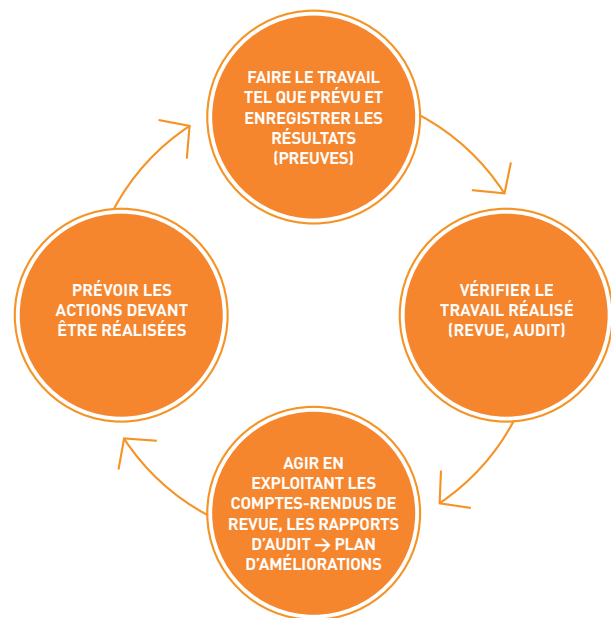


FIGURE 20 Cycle de Deming PDCA (Plan, Do, Check, Act) (Source : Valtech)

HUBERT GILLON

// Delivery Manager
// Valtech



Agilité et ISO concourent à rendre les pratiques de projet plus efficaces au bénéfice des produits livrés et de la satisfaction client.



5.6 METTRE DE L'AGILITÉ DANS UNE DÉMARCHÉ CMMI®

Qui a dit « inconciliable » ?

On a souvent tendance à opposer Agilité et CMMI® en prétendant qu'être Agile signifie faire l'impasse sur tout processus prédéfini qui constituerait un carcan nuisible à la créativité et la productivité. Les détracteurs des modèles qualité tel que CMMI® ont coutume d'affirmer que mettre en place des pratiques Agiles c'est « assurer » la qualité du produit, alors que se conformer à un modèle tel que le CMMI® n'a pour effet que de « rassurer » ceux qui l'appliquent.

C'est oublier un peu vite que le fondement du modèle CMMI® est d'inciter une organisation à s'améliorer de façon continue en structurant cet effort autour d'un ensemble de processus et avec une approche progressive par niveau de maturité. Le cycle d'amélioration continue IDEAL (et plus généralement toute démarche de type PDCA) sous-jacent au modèle CMMI® se conforme donc bien au paradigme d'itération prôné par les méthodes Agiles.

Par ailleurs, CMMI® définit des objectifs (Specific Goals et Generic Goals) visant à garantir que la qualité des produits réalisés ne dépende pas de l'héroïsme des équipes mais de l'efficacité de ses processus. Pour atteindre ces objectifs, CMMI® propose un certain nombre d'exigences structurées par processus (Process Area), décrivant le « quoi faire », et non le « comment faire ». Il y a donc toute liberté à mettre en œuvre des pratiques Agiles pour atteindre ces exigences.

Des pratiques CMMI® Agiles ?

Si les « agilistes » étaient déjà confiants dans la conformité de leurs pratiques avec un grand nombre d'exigences du modèle CMMI®, on constate aujourd'hui que le SEI (Software Engineering Institute) étudie également cette compatibilité, à travers divers articles (CMMI® or Agile: why not embrace both?) ou ouvrages (Integrating CMMI® and Agile Development). Certains Lead Appraisers ont acquis l'expérience de la mise en œuvre des pratiques Agiles et savent donc évaluer une organisation de ce type.

La couverture des pratiques Agiles vis-à-vis des exigences spécifiques du modèle (Specific Goals) est analysée ci-dessous.

Le **niveau 2**, focalisé principalement sur les activités projet, est globalement satisfait par les pratiques Agiles Scrum et XP :

- **REQM** : ce domaine de processus est globalement couvert par :
 - le rôle central du Product Owner, qui permet d'assurer la bonne compréhension des exigences par l'équipe de développement lors des ateliers fonctionnels,
 - le Product Backlog, partagé entre l'équipe et le Product Owner, il permet de définir le besoin et de tracer les changements,
 - la traçabilité entre les exigences et les tests, grâce aux techniques et à des outils de Test Driven Requirement.
- **PP** :
 - le Plan de Release, le Product Backlog et l'Iteration Backlog détaillent la décomposition du travail, l'estimation des charges et le planning,
 - les réunions de planification (Iteration Planning) permettent de s'assurer de l'engagement des membres de l'équipe,
 - par contre, le plan de data management n'est pas couvert explicitement par les pratiques Agiles, mais l'utilisation d'un wiki favorise la bonne maîtrise des informations.
- **PMC** : les réunions de planification d'itération, les scrum meetings, les réunions de fin d'itération et rétrospectives assurent le suivi et le contrôle des activités projets.

- **M&A** : les informations d'avancement collectées lors des Scrum meetings, consignées dans l'iteration backlog et consolidées dans les indicateurs graphiques satisfont à un certain nombre de pratiques de ce domaine.
- **SAM** : la sélection et la gestion des sous-traitants ne sont pas du tout couvertes par les pratiques Agiles. Il convient pour cela d'établir des contrats flexibles innovants (Cf. supra)
- **PPQA** : ce domaine n'est pas couvert par des pratiques Agiles de Scrum ou XP.
- **CM** : la mise en place d'une usine logicielle et de l'intégration continue implique de mettre en place un outil de gestion de configuration. L'outillage de la gestion des changements est également souvent mis en place.

Par ailleurs, alors que le niveau 2 du CMMI® établit des pratiques projet, le niveau 3 vise à l'institutionnalisation de ces pratiques dans l'organisation. Or les pratiques Agiles restent focalisées au niveau projet et ne couvrent donc pas les domaines de processus organisationnels (OPF, OPD, OT).

Citons les pratiques relatives à l'ingénierie :

- **RD** : la plupart des pratiques du domaine sont couvertes avec la planification de l'itération, les ateliers fonctionnels avec une implication du Product Owner pour détailler les fonctionnalités sélectionnées pour l'itération et identifiés les critères de validation.
- **TS** : les pratiques Agiles ne couvrent pas vraiment le processus de choix de la meilleure solution technique au travers de l'étude de solutions alternatives.
- **PI** : l'intégration continue couvre parfaitement cet objectif.
- **VER, VAL** : sont couverts par les pratiques de développement (eXtreme Programming) et par les spécifications pilotées par les tests.

Coté gestion de projet, regardons :

- **RSKM** : il n'y a pas de description explicite de gestion des risques dans Scrum ou XP. Cependant il est reconnu que les pratiques Agiles encouragent les équipes à se concentrer sur les risques les plus critiques le plus tôt possible, qu'il s'agisse de risques techniques ou non. Il convient donc d'ajouter aux pratiques Agiles classiques une gestion des risques qui pourra être mise en œuvre lors des iteration planning ou de réunions dédiées

Les pratiques Agiles ne répondent pas explicitement aux niveaux 4 et 5. On doit cependant mentionner que leur mise en place participe à l'amélioration continue des processus : lors des rétrospectives de fin d'itération, des outils comme les « 5 pourquoi » ou le diagramme d'Ishikawa (Fishbone) sont utilisés dans la recherche de causes racine.

Scrum apparait comme une force pour permettre la couverture de certaines pratiques génériques (Generic Goals), par exemple pour assurer l'engagement des parties prenantes (cas du Product Owner et de l'équipe), pour assurer la planification et le suivi de certains domaines de processus grâce aux backlogs.

Qu'est-ce qu'un « Référentiel Qualité » Agile ?

Tout d'abord, rappelons que l'« Assurance Qualité » au sens du CMMI® couvre à la fois les processus et le produit. Concentrons-nous ici sur le volet processus. **L'Assurance Qualité consiste à s'assurer que l'organisation se conforme systématiquement aux dispositions du « Référentiel Qualité »**, autrement dit au cadre méthodologique et aux procédures afférentes. Les méthodes Agiles introduisent un certain nombre de pratiques dont le succès passe par une application rigoureuse, comme par exemple les différentes « cérémonies Scrum » : iteration planning, Scrum meeting, revue d'itération, rétrospective. Le « contrôle qualité du processus » incombe au Scrum Master, qui reste néanmoins avant tout un facilitateur dans la mise en place et l'adoption des différentes pratiques Agiles.

Le référentiel qualité :

- identifie les diverses **méthodes et pratiques appliquées** sans pour autant les re-détailler. Il suffit pour cela de se référer à l'**abondante bibliographie** qui existe sur le sujet. Chaque projet Scrum pourra simplement indiquer la durée de ses itérations puis utiliser les outils communs de gestion de backlog et de suivi des risques,
- contient divers **modèles de documents** – « Organisation Agile ne signifie pas zéro documentation » – mais en **nombre réduit**. Le principe est de ne produire que la documentation utile - qui apporte de la valeur - et au bon moment. Rédiger un volumineux dossier de conception n'offre aucun intérêt si l'on se conforme déjà à des standards d'architecture et de codage. L'utilisation de frameworks peut aider à imposer l'architecture et la rendre transparente. Il est par ailleurs relativement facile d'automatiser la vérification de telles règles avec des outils d'analyse statique. Par contre, il est important de garder trace des réflexions qui ont amené à ces choix d'architecture et de frameworks, afin de transférer la connaissance du projet et éviter ainsi de se reposer plus tard les mêmes questions.

Un vrai référentiel qualité Agile :

- décrit un ensemble de pratiques et de procédures,
- contient uniquement les procédures qui nécessitent d'être détaillées afin d'être réellement exploitables,
- voit sa bonne **application contrôlée en permanence par les Scrum Masters**.

Qui occupe la tour d'ivoire de l'EPG ?

Un autre des principes Agiles, prôné en particulier par Scrum, consiste à **responsabiliser totalement l'équipe** en lui conférant le pouvoir d'auto-organisation et d'auto-détermination. Par exemple, ce n'est plus un chef de projet qui affecte les tâches aux membres de l'équipe, mais ces derniers qui se les approprient, en fonction de leurs compétences techniques ou fonctionnelles.

De même, on attend de chacun, lors des **rétrospectives**, qu'il identifie les points d'amélioration possibles sur les processus en vigueur et fournisse, si possible, les solutions associées.

Certains considèrent qu'en appliquant le CMMI®, on se repose sur quelques éminents qualitatifs pour promulguer les bonnes pratiques à respecter. Or ce cénacle, l'Engineering Process Group (EPG), est en fait le rassemblement temporaire de certains membres des équipes projets, légitimés par leur expérience et porteurs de l'ensemble des retours d'expérience de leur équipe. Toute l'équipe projet participe ainsi à l'amélioration des pratiques d'une organisation Agile.

L'Agilité façon « Lean »



S'engager dans une démarche d'amélioration continue Lean permet aux maîtrises d'ouvrage (MOA) et aux directions informatiques de concrétiser les bénéfices des méthodes Agiles, en conciliant de façon optimale leurs impératifs de qualité et de réactivité.



ELISABETH DUCARRE

// Expert Lean et CMMI
// Valtech

Les principes Lean

Lean est connu dans le monde automobile par l'expérience Toyota, devenu le premier constructeur automobile, reconnu à la fois pour la qualité et l'innovation de ses produits. Tout le monde s'accorde à reconnaître que ce succès est dû à son système de production Lean. Et les déboires de Toyota en 2009 ne tendent qu'à prouver l'importance de ne pas déroger aux principes énoncés ci-dessous.

Cette approche vise à la fois

- à améliorer la qualité et les délais,
- à réduire les coûts en tirant le meilleur parti des ressources tant humaines que matérielles,
- à éviter toute forme de gaspillage.

Le Lean est basé sur les principes fondamentaux suivants :

- déterminer ce qui **crée de la valeur** pour le client,
- s'assurer que **chaque activité** du processus apporte de la valeur ajoutée ou est indispensable,
- réaliser chaque activité **juste à temps** et **à la demande**, pour que ses résultats soient utilisés sans délai,
- rechercher la **perfection**,
- s'appuyer sur ceux qui réalisent les activités pour rechercher des améliorations.

Ces principes ont été, eux-mêmes, déclinés dans **Lean Software Development** [[Réf.2]] par les principes suivants :

- éliminer les gaspillages,
- favoriser la connaissance,
- construire la qualité intrinsèque,
- reporter la décision,
- livrer rapidement,
- respecter les personnes,
- optimiser le système dans son ensemble.

« Agile is Lean »

On constate que les pratiques Agiles héritées des méthodes Scrum et eXtreme Programming servent directement les principes du Lean Software Development car tous se **focalisent sur le produit** en visant la **satisfaction client**.



Lean Software Development considère toutes les méthodes Agiles comme valides pour appliquer le « Lean Thinking » au monde du logiciel, et en particulier le déploiement de la méthode Scrum.



JEFF SUTHERLAND

// Extrait traduit de l'anglais
// Agile Software Development
with Scrum

- Favoriser la connaissance, Reporter la décision et Livrer rapidement peuvent se traduire par la segmentation en itérations courtes des projets Agiles, ce qui apporte aux entreprises clientes la prédictibilité dont elles ont besoin sur la disponibilité des fonctionnalités en cours de développement. Elles peuvent ainsi planifier l'intégration continue des nouvelles fonctions à leur rythme et en fonction des ressources disponibles, sans surcoût ni surcharge de travail qui induisent de nombreuses sources d'erreur (Concept Lean **Just in time**).
- Un exemple concret de la mise en œuvre du principe **Construire la qualité intrinsèque** (découlant du principe **Rechercher la perfection**) peut se traduire par la technique TDD associée à l'automatisation des tests, qui constituent une approche efficace d'amélioration de la qualité au plus tôt dans le cycle de développement. Cela permet de savoir immédiatement si ce qui est produit possède le bon niveau de qualité et de corriger les défauts au plus tôt.
- Nous répondons également au concept Lean **Stop the line**, en corrigeant les anomalies dès qu'elles sont détectées, ce qui évite de gaspiller des ressources en poursuivant le développement de l'application sur des bases non fiables à 100%. En effet, la priorité de correction des anomalies doit être supérieure la priorité d'ajouter de nouvelles fonctions au logiciel. L'équipe dispose pour cela d'une usine logicielle qui génère quotidiennement une version exécutable de l'application en cours de développement et exécute les tests automatisés afin de détecter au plus tôt les anomalies potentielles.

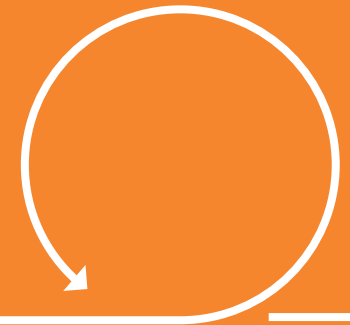
- Enfin, le **Visual Management** est un important concept Lean, largement soutenu dans les méthodes Agiles par une organisation de l'espace de travail. Il permet à toutes les parties impliquées (développeurs, testeurs, chefs de projets, représentants du métier) de visualiser instantanément l'état d'avancement du projet, et ceci en termes simples : avertissement sonore ou lumineux pour alerter d'un « build cassé », consolidation des divers indicateurs d'avancement (couverture de tests, anomalies en cours...) sous forme de diagrammes et symboles de couleur sur un écran ou un tableau à la vue de tous.
- L'ensemble des pratiques citées ci-dessus répondent également au premier principe **Éliminer les gaspillages** : réduire les retards (itérations courtes), se concentrer sur les défauts (TDD), mieux comprendre les exigences (TDR), etc.

L'ESSENTIEL À RETENIR

- « Agilité » et « ISO » se rejoignent sur des valeurs communes de planification opérationnelle et d'amélioration continue.
- Toute organisation souhaite rendre ses processus efficaces et efficients. Certaines déploient des pratiques Agiles pour améliorer leur réactivité et leur satisfaction du besoin client. D'autres se focalisent sur l'homogénéisation et le respect de leurs processus en mettant en œuvre une démarche d'amélioration continue telle que CMMI®, ISO ou Lean. La performance globale de l'organisation passe certainement par la combinaison des deux.
- CMMI® impose des objectifs visant à garantir que la qualité des produits réalisés ne dépend pas de l'héroïsme des équipes mais de l'efficacité de ses processus. Pour atteindre ces objectifs, CMMI® propose un certain nombre de pratiques qui sont des recommandations sur le « quoi faire », et non sur le « comment faire ». Les pratiques Agiles apportent une réponse à ces exigences.

6
....

GLOSSAIRE DES PRATIQUES AGILES



6.1 DÉFINITIONS

Agilité n'est pas synonyme de désordre, au contraire, le monde de l'Agilité comprend un certain nombre de méthodes très formalisées tel que Scrum ou eXtreme Programming (XP) pour ne citer que les plus connues.

Voici un schéma qui est inspiré de la représentation en 3 niveaux de Ron Jeffries (un des fondateurs de l'eXtreme Programming), agrémenté d'autres pratiques Agiles que nous utilisons régulièrement chez Valtech :

- le cercle extérieur contient les **pratiques « liées » au client**,
- le cercle intermédiaire contient les **pratiques de management**,
- le cercle intérieur contient les **pratiques de développement**.

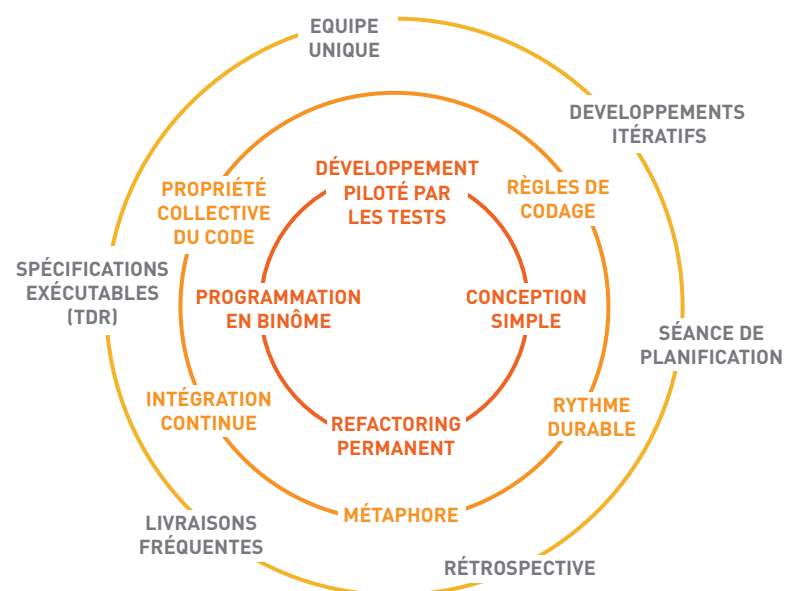


FIGURE 21 Les pratiques Agiles issues de XP et Scrum (Source Valtech)

Cercle client

Développement Itératif : l'activité de développement est organisée en cycles dont la durée est fixée une fois pour toute pour le projet. On appelle ces cycles des itérations ou sprints. Ils définissent le rythme du projet (heart beat).

Equipe Unique (whole team) : il s'agit ici de casser le modèle cloisonné Spécificateur-Développeur-Testeur. Les participants d'un projet de développement forment une seule équipe ou tout le monde participe à la hauteur de ses compétences, et dans le cadre d'un rôle identifié, vers un but commun. Il est important de souligner que cette pratique consiste aussi à rassembler l'équipe sur le plan géographique (dans la mesure du possible).

Livraisons fréquentes (Small Releases) : les livraisons sont effectuées souvent, toutes les 2 à 4 semaines. Cette pratique permet de réduire notamment les difficultés parfois rencontrées au moment de la mise en production.

Rétrospective : à chaque fin d'itération, un temps est prévu pour réfléchir au déroulement de l'itération passée et pour chercher les moyens d'améliorer l'efficacité pour les itérations futures.

Séance de planification (Planning game) : la structure des séances de planification est très codifiée, avec un certain nombre d'étapes identifiées à réaliser avec tous les membres de l'équipe. Ces séances de planification reviennent cycliquement en début de chaque itération, définissant ainsi les spécifications de l'application à produire petit à petit, plutôt qu'en un seul grand coup de canon en début de projet. C'est lors de ces séances que l'on définit ce que l'on appelle dans Scrum le Sprint Backlog ou Iteration Backlog, à partir de la liste des fonctions / scénarios rassemblés dans le Product Backlog et qui suit l'application d'itération en itération.

Tests client automatisés (TDR ou Customer Tests) : on parle aussi de spécifications exécutables ou Test Driven Requirement. Le client définit les critères d'acceptation des scénarios fonctionnels sous forme de cas de test.

Cercle Management

Intégration Continue (Continuous Integration) : le système développé est intégralement assemblé et testé plusieurs fois par jour. Aujourd'hui, cette pratique est grandement facilitée par des outils dédiés (Hudson, CruiseControl...).

Métaphore (Metaphore) : l'équipe doit rechercher et utiliser une analogie comme modélisation du système à développer. Cette technique très répandue dans le monde informatique permet aussi à l'équipe de se construire un vocabulaire commun, notamment pour la communication entre les profils fonctionnels et ceux techniques.

Propriété collective du code (Collective Code Ownership) : tout le code de l'application est accessible en modification par tous les membres de l'équipe. Il n'y a pas de domaine réservé. Cette pratique qui a l'avantage de résoudre le syndrome de l'autobus (qu'est-ce qu'on fait si une personne de l'équipe se fait renverser par un autobus?), permet aussi de fluidifier le travail quotidien de l'équipe. Elle suppose d'avoir une communication très développée sur les détails techniques de réalisation. Les tests unitaires intensifs et la programmation en binôme soutiennent cette pratique de manière significative.

Règles de codage (Coding Standard) : l'équipe se fixe des règles de codage de manière à ce que le code soit homogène et facilement lisible par tous.

Rythme durable (Sustainable Pace) : cette pratique initialement intitulée par les Américains « 40 heures par semaine » et qui n'a jamais signifié grand chose dans la culture française, recommande de ne pas faire d'heures supplémentaires plus de deux semaines de suite. Les membres de l'équipe doivent être en forme pour donner le meilleur d'eux-mêmes. La généralisation des heures supplémentaires est le fléau des équipes mal organisées. L'optimisation des processus apportée par ce type de méthode permet d'améliorer notablement la productivité sans augmenter la charge de travail.

Cercle Développement

Conception Simple (Simple Design) : à tout moment, le design de l'application est le plus simple possible afin qu'il puisse répondre aux exigences rencontrées jusque là. La simplicité ne sous-entend pas de prendre des raccourcis sur la qualité. Le code doit être concis, modulaire, cohérent, lisible et doit passer tous les tests.

Développement Piloté par les Test (TDD : Test Driven Development) : l'activité de programmation suit le processus suivant : écrire un test → écrire le code le plus simple qui puisse compiler → améliorer le code (refactoring) pour introduire l'abstraction nécessaire et éliminer d'éventuelles duplications.

Programmation en binôme (Pair Programming) : elle se caractérise par le fait que le code est écrit par deux personnes, un pilote et un copilote. Les rôles au sein du binôme et les binômes eux-mêmes changent régulièrement, ce qui permet à l'équipe d'avoir une meilleure connaissance du code de l'application.

Refactoring permanent (Mercyless Refactoring) : pratique de développement qui consiste à améliorer le code sans en changer le comportement.

7 RÉFÉRENCES BIBLIOGRAPHIQUES



6.2 ABRÉVIATIONS

- CMMI** : Capability Maturity Model Integration
- DSI** : Direction des Systèmes d'Information
- LSD** : Lean Software Development
- MOA** : Maîtrise d'Ouvrage
- MOE** : Maître d'œuvre
- PDCA** : Plan Do Check Act
- PMD** : Outil d'analyse des flux de données et de recherche des copier/coller, du code mort et des constructions complexes en Java
- ROI** : Return On Investment (retour sur investissement)
- RSA** : IBM-Rational Software Architect
- TDD** : Test Driven Development
- TDR** : Test Driven Requirement

RÉF.	DOCUMENT	AUTEUR	ÉDITEUR	ÉDITION
[Ref.1]	Agile Software Development with Scrum	Ken Schwaber	Pearson Education	2008
[Ref.2]	Implementing Lean Software Development, From Concept to Cash	Mary and Tom Poppendieck	Addison Wesley	2007
[Ref.3]	Gestion de projet : vers les méthodes Agiles	Véronique Messenger	Eyrolles	2007
[Ref.4]	Agile estimating and planning	Mike Cohn	Prentice Hall	2004
[Ref.5]	Gestion de projet – Extreme Programming	Jean-Louis Bénard	Eyrolles	2004
[Ref.6]	User stories applied	Mike Cohn	Addison-Wesley Professional	2004
[Ref.7]	Test-Driven Development	Kent Beck	Pearson Education	2003
[Ref.8]	Agile and Iterative development	Craig Larman	Addison-Wesley Professional	2003
[Ref.9]	Coaching Agile Teams: A Companion for Scrum Masters, Agile Coaches, and Project Managers in Transition	Lyssa Adkins	Addison Wesley	2010
[Ref.10]	Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum	Craig Larman, Bas Vodde	Addison-Wesley	2010
[Ref.11]	Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum	Craig Larman, Bas Vodde	Addison-Wesley	2008
[Ref.12]	The Fifth Discipline - The Art & Practice of the Learning Organization	Peter Senge	Currency Doubleday	1990
[Ref.13]	Gestion de Projet Agile, avec Scrum, Lean, eXtreme Programming, 3eme édition	Véronique Messenger	Eyrolles	2010
[Ref.14]	The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer	Jeffrey K. Liker	McGraw Hill	2004
[Ref.15]	Agile Product Management with Scrum – Creating Product that Customer Love	Roman Pichler	Addison-Wesley	2010
[Ref.16]	The Enterprise and Scrum	Ken Schwaber	Microsoft Press	2007
[Ref.17]	Lean-Agile Software Development: Achieving Enterprise Agility	Alan Shalloway, Guy Beaver, James R. Trott	Addison-Wesley	2009
[Ref.18]	« 10 contract forms for your next Agile project »	Peter Stevens	Munich, 21-Oct-09	2009
[Ref.19]	Five dysfunctions of a team	Patrick Lencioni		
[Ref.20]	Agile software development	Alastair Cockburn		
[Ref.21]	Extreme Programming, embrace change, 2nd edition	Kent Beck		

TABLEAU 12 Références bibliographiques

