



HAL
open science

Implémentation du modèle ALM Capgemini dans un service SAP-BW

Manuel Collet-Beillon

► **To cite this version:**

Manuel Collet-Beillon. Implémentation du modèle ALM Capgemini dans un service SAP-BW . Génie logiciel [cs.SE]. 2013. dumas-01280483

HAL Id: dumas-01280483

<https://dumas.ccsd.cnrs.fr/dumas-01280483>

Submitted on 29 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL ASSOCIE DE LYON

MEMOIRE

Présenté en vue d'obtenir

le DIPLOME D'INGENIEUR CNAM

SPECIALITE : INFORMATIQUE

OPTION : SYSTEMES D'INFORMATION

par

Manuel COLLET-BEILLON

Implémentation du modèle ALM Capgemini dans un service SAP-BW

Soutenu le 17 Janvier 2013

JURY

PRESIDENT : **Mr Christophe PICOULEAU** *Professeur responsable CNAM Paris*

MEMBRES : **Mme Valérie LESAFFRE** *Manager ALM Capgemini*
 Mr Bertrand DAVID *Professeur des Universités Ecole
Centrale – Responsable filière informatique CNAM Lyon*
 Mr Claude GENIER *Administrateur supérieur au CERN -
co-Responsable filière informatique CNAM Lyon*
 Mr Fabrice DUBOST *Responsable qualité et environnement
Sogeti Rhône-Alpes - Maitre de conférence, responsable des relations
industrielles de Polytech Grenoble.*

Remerciements

Travaillant au sein de l'équipe SAP de Caggemini Montbonnot depuis plus de 6 ans, j'ai eu la chance de pouvoir évoluer dans différentes phases du cycle de vie des applications. En commençant par le support, en passant par le développement, l'expression du besoin ou encore la conduite de projet, j'ai pu acquérir une vision d'ensemble des problématiques de gestion d'un patrimoine applicatif. En tant que pilote de l'équipe BW, je suis maintenant impliqué dans son organisation et dans sa coordination avec son écosystème.

Je tiens à remercier en premier lieu la responsable du service, Madame Valérie Lesaffre, qui m'a accompagné dans ma mission ainsi que dans la rédaction de ce mémoire.

Merci à Monsieur Claude Genier, pour son aide et son encadrement sur ce mémoire.

Je remercie également le responsable de production du pôle BW, Monsieur Ludovic Piron, pour son soutien quotidien et sa sagacité, ainsi que tous mes collègues du service SAP.

Merci également à Fabrice Dubost et toute l'équipe de Sogeti, sans qui cette aventure n'aurait pas eu lieu.

Enfin, un merci spécial à Katell Bouvet, Maëlle et Thomas Collet-Beillon pour m'avoir supporté pendant ces 3 ans de CNAM.

Liste des abréviations

Acronyme	Signification
ALM	Application Lifecycle Management (Gestion de Patrimoine Applicatif)
API	Application Programming Interface (Interface de Programmation)
AUP	Agile Unified Process (Processus Unifié version Agile)
CAB	Change Advisory Board (Comité de gestion des changements)
CEE	Centre Etude Emploi
CHSCT	Comité hygiène sécurité et conditions de travail
CRM	Customer Relationship Management ou gestion des relations clients
CVS	Chinese Value Survey (Enquête sur les valeurs Chinoises)
DMT	Demand Management Tool (Outil de gestion de la demande)
DSDM	Dynamic software development method (Méthode dynamique de développement de logiciels)
FAQ	Foire Aux Questions
FDD	Feature Driven Development (Développement piloté par les fonctionnalités) : méthode agile
GPA	Gestion de Patrimoine Applicatif
ITIL	Information Technology Infrastructure Library (Bibliothèque pour l'infrastructure des technologies de l'information)
PAQ	Plan d'Assurance Qualité
PDCA	Plan Do Check Act (Préparer Développer Contrôler Ajuster)
PIR	Post Implementation Report (notion ITIL : compte rendu d'implémentation d'un changement)
QG	Quality gate (Etape de validation de qualité)
RAD	Rapid Application Development (Development rapide d'application) : méthode agile
RFC	Request for Change (Demande de changement)
RUP	Rational Unified Process (Processus Unifié version Rational)
SDM	Service Delivery Manager (Responsable de production)
SLA	Service Level Agreement (Accord de Niveaux de Service)
TDD	Test Driven Development (Développement piloté par les tests) : méthode agile
TMA	Tierce Maintenance Applicative (Externalisation de Maintenance Applicative)
UML	Unified Modeling Language (Langage de Modélisation Unifié)
UP	Unified Processes (Processus Unifiés) : méthode de développement logiciel
VSM	Visual Stream Mapping (Cartographie du flux de valeur)
WVS	World Value Survey (Enquête sur les valeurs mondiales)
XP	eXtreme Programming : méthode agile

Glossaire des termes techniques

Terme	Signification
Administrator workbench	Environnement de développement BW
ASARIS	Application BW pour Schneider-Electric (domaine logistique)
Business Explorer	Environnement de création d'état BW
BW	Business Warehouse (Entrepôt de données) : Module décisionnel de SAP
CRM	Customer Relationship Management (Gestion des Relations Client)
EBP	Enterprise Business Partner (Application de gestion des partenaires)
ERP	Enterprise Resource Planning (Progiciel de gestion d'entreprises)
ETL	Extraction Transformation Loading (Extraction Transformation et Chargement de données)
HP	Hewlett-Packard : Société Américaine d'édition de logiciel et de matériel informatique
HPQC	Hewlett-Packard Quality Center (Outil de gestion de tests)
iDOC	Intermediate Document (objet structuré d'échange d'information)
Infocube	Objet BW : Ensemble de tables organisées selon un schéma en étoile et servant à stocker et à restituer des données
Infogérance	Externalisation de la gestion d'un système informatique
Info-objet	Elément de base de BW servant à définir les données
MAP-CCO	Application BW pour Schneider-Electric (domaine contrôle de gestion)
Middleware	Intergiciel, logiciel d'échanges d'information entre deux systèmes
ODS	Object Data Store (Objet de stockage de données dans BW)
Offshore	Equipe informatique délocalisée
Onshore	Equipe informatique physiquement proche du client
PSA	Persistent Staging Area (Zone de Stockage Intermédiaire Persistente)
SAP	Société Allemande d'édition de progiciels de gestion
Source system	Source de données pour une application BW
SSRM	Application BW pour Schneider-Electric (domaine achat)
VENUS	Application de gestion des fournisseurs pour Schneider-Electric
XI	Middleware (Intergiciel) de SAP

Table des matières

Remerciements.....	3
Liste des abréviations	5
Glossaire des termes techniques	7
Table des matières	9
Introduction	11
1. Problème posé et travaux antérieurs.....	13
1.1. Mise en place du centre de service BW.....	13
1.1.1. Généralités	13
1.1.2. Une nouvelle gestion de la connaissance	16
1.1.3. Les difficultés de la multi culturalité.....	18
1.2. Le modèle ALM Capgemini	20
1.2.1. Définition de l'ALM	20
1.2.2. L'industrialisation	24
1.2.3. Approche offshore.....	30
1.2.4. Synthèse de l'ALM et adaptation au centre de service BW	31
1.3. Agile.....	32
1.3.1. Généralités	32
1.3.2. Analyse des méthodes SCRUM et XP	34
1.3.3. Autres méthodes.....	39
1.3.4. Analyse de l'agilité dans un contexte de maintenance.....	41
1.3.5. Synthèse de l'agilité et adaptation au centre de service BW	45
1.4. Lean	46
1.4.1. Définition du Lean.....	47
1.4.2. Les pratiques du développement de logiciel Lean.....	50
1.4.3. Comparaison avec les méthodes agile.....	56
1.4.4. Critiques du Lean.....	57
1.4.5. Adaptation du Lean au contexte du centre de service	58
2. Solutions proposées.....	61
2.1. Amélioration continue.....	61
2.1.1. Définition et planning	61
2.1.2. Objectifs.....	64
2.1.3. Indicateurs	64
2.2. Organisation.....	67
2.3. Management.....	68
2.3.1. Management visuel	69

2.3.2.	Carte de l'amélioration continue	71
2.4.	Cartographie des processus	72
2.4.1.	Gestion des demandes d'évolution	73
2.4.2.	Gestion de la demande de support	75
2.5.	Cartographie interculturelle	77
2.5.1.	Indices culturels	78
2.5.2.	Cartes d'identités	84
2.6.	Modèle de gestion de la connaissance.....	86
2.6.1.	Gestion de la documentation.....	86
2.6.2.	Mise en place du kit du nouvel arrivant.....	88
2.6.3.	Capitalisation.....	89
3.	Résultats	91
3.1.	Indicateurs	91
3.2.	Production Opérationnelle	95
3.2.1.	SAP BW	95
3.2.2.	Catalogue de demandes.....	99
3.2.3.	Exemple de demandes de changement : Projet « Preferred ».....	101
3.2.4.	Deuxième exemple de demande : Modification des volumes achat.....	106
3.3.	Résultats méthodologiques	109
3.3.1.	Modèle de pilotage de service.....	109
	Conclusion	113
	Bibliographie	115
	Ouvrages imprimés.....	115
	Sites web	115
	Illustrations.....	119
	Tableaux	121
	Résumé.....	122

Introduction

Le centre de Capgemini à Montbonnot gère des services informatiques externalisés. Historiquement, cette entité était centrée autour de son partenariat avec Schneider-Electric, mais l'évolution stratégique de Capgemini a conduit à l'ouverture de ce centre à de nouveaux clients.

L'arrivée de nouveaux contrats comme Rio-Tinto a permis de compenser la décroissance des affaires avec Schneider-Electric et donc de maintenir un volume d'activité sensiblement identique.

Cette mutation a néanmoins engendré un besoin de transformation des services existants pour les faire migrer d'un monde mono-client à un monde plus dynamique, multi-clients, source d'enrichissement et d'intérêts mais également de défis et de difficultés.

A l'intérieur du centre Capgemini nommé ALM (Application Lifecycle Management) se trouve le service SAP qui est lui-même découpé en fonction des périmètres métiers gérés : logistique, production, finance et décisionnel.

C'est dans ce dernier domaine, que j'ai été amené à intervenir pour accompagner la mutation du service.

Le service décisionnel gérait jusqu'alors trois applications différentes et était organisé avec des personnes dédiées à chaque application. La prise en charge au début de l'année 2012 des applications SAP de Rio-Tinto France a engendré une réorganisation du service BW pour mutualiser l'activité existante. Le but était de dégager une bande passante suffisante pour gérer les nouvelles applications avec les mêmes ressources, dont une partie d'entre elles était située sur un site délocalisé à Mumbai, en Inde.

L'objectif principal de ce mémoire est donc de définir et formaliser un modèle de management permettant de répondre aux nouveaux défis proposés par cette organisation :

- Maintien du niveau de service sur le périmètre existant (Schneider-Electric).
- Réussite de l'intégration du nouveau périmètre (Rio-Tinto).
- Configuration de l'équipe pour la rendre facilement extensible à un nouveau périmètre potentiel (prospects).

Pour atteindre cet objectif, je me suis basé sur les pratiques internes en vigueur au sein de mon entité ainsi que sur des méthodes actuelles comme l'Agile ou le Lean.

En effet, ces méthodes étaient, à mon sens, les mieux adaptées pour nous aider à gérer un nouvel environnement plus turbulent, avec plus de changements ou de conflits de priorités mais disposant également de plus de libertés et de souplesse pour agir.

De plus, afin de garantir le succès de cette mission, il m'a semblé nécessaire de me pencher sur la collaboration avec nos collègues indiens pour tenter d'améliorer notre synergie et nous permettre de réussir ce challenge en équipe.

Ce mémoire présente donc le cheminement effectué tout au long de cette année et les réflexions menées pour répondre à la question « Comment allons-nous nous organiser pour répondre aux enjeux d'un service informatique externalisé multi-clients, multiculturel et multi-sites ? »

Le premier chapitre décrit le contexte du mémoire et détaille la problématique soulevée. Dans un second temps, ce chapitre présente le modèle ALM Caggemini puis les méthodes Agile et la philosophie Lean en analysant à chaque fois les parties applicables à notre contexte.

Le second chapitre présente les solutions qui ont été mises en œuvre dans le domaine de l'organisation pour soutenir l'activité.

Enfin, le troisième chapitre présente les résultats obtenus sur le plan méthodologique et un aperçu des effets des changements organisationnels sur la production technique effectuée au cours de l'année. La figure 1 modélise le concept du mémoire.

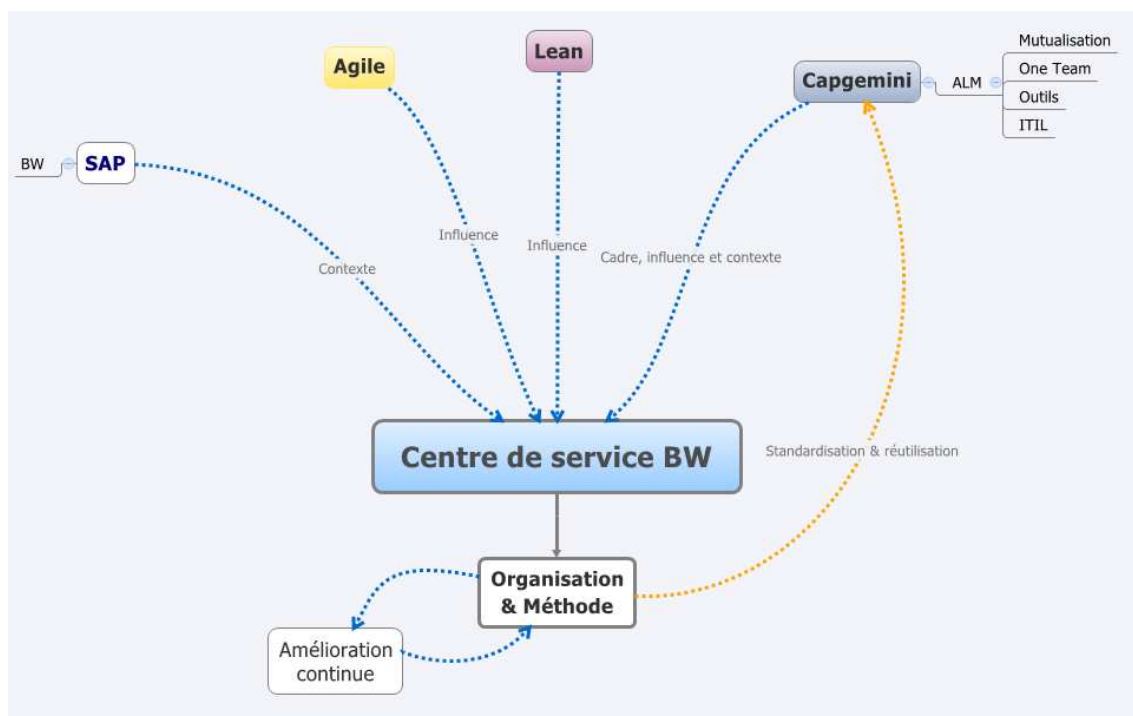


Figure 1 - Modèle conceptuel du mémoire

1. Problème posé et travaux antérieurs

Cette partie décrit la problématique générale de la mission qui m'a été confiée. Elle présente également une étude bibliographique des méthodes permettant de répondre au besoin. Cette étude porte sur les pratiques internes à la société ainsi que sur les méthodes actuelles du marché : Agile et Lean.

1.1. Mise en place du centre de service BW

Notre centre de service effectue des missions pour le compte de clients externes sur le domaine SAP BW. En amont de la mission, le périmètre concernait uniquement le client Schneider-Electric et était séparé entre trois applications :

- ASARIS

Il s'agit d'une application basée sur des données de vente et de distribution issues d'un SAP transactionnel. Elle sert à donner des informations pour le pilotage de la logistique comme le suivi des durées de livraison ou le statut des portefeuilles de commande.

- MAP-CCO

Cette application de contrôle de gestion présente la marge des différentes entités de distribution de Schneider-Electric.

- SSRM

Cette dernière concentre les données de volumes d'achat issues des différents systèmes de gestion du groupe ainsi que le suivi de la qualité des fournisseurs.

Chaque application était gérée de manière indépendante par des personnes dédiées et les rares interactions entre les membres de l'équipe étaient uniquement liées à l'usage d'une même technologie.

1.1.1. Généralités

L'arrivée d'un nouveau client (Rio-Tinto) a suscité le besoin d'une mutualisation avec les équipes existantes. En effet, le volume d'activités prévues pour ce client ne permettait pas de lui affecter une ressource à temps complet, ce qui pouvait poser des problèmes pour la gestion des absences et limiter la flexibilité de la production lors des périodes de surcharge, ou au contraire, de sous-activité.

De plus, cette arrivée concrétisait la politique d'ouverture vers de nouveaux clients. Il était important de disposer d'un outil productif capable d'absorber de nouveaux périmètres de manière homogène avec les comptes déjà sous notre responsabilité. Il a été décidé de fusionner tous les services en un seul pour traiter les demandes indépendamment de leur origine en termes d'application ou de client et d'utiliser un modèle de pilotage commun. La figure 2 illustre cette fusion.

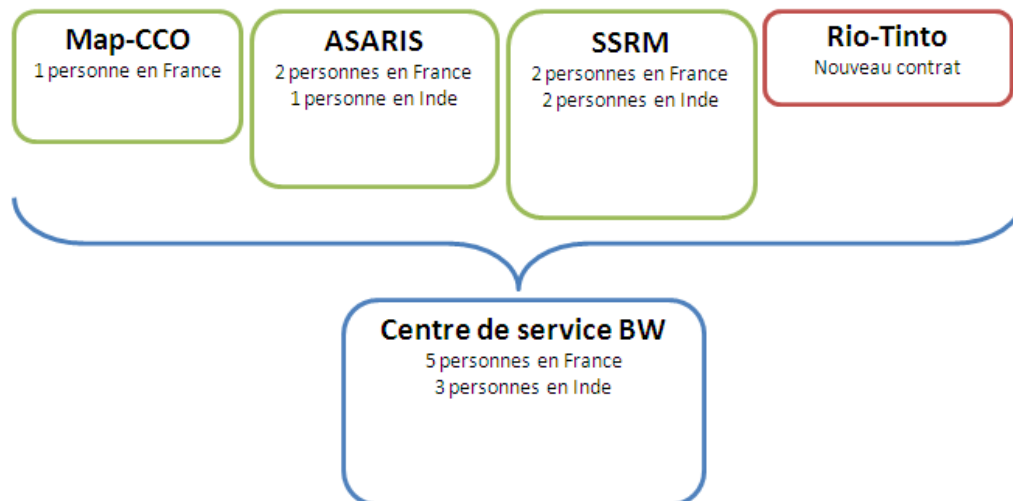


Figure 2 - Fusion des services

Cette nouvelle approche présentait divers avantages et divers risques :

Avantages :

- Lissage de l'activité sur plusieurs domaines

Le fait de mutualiser des ressources sur plusieurs périmètres applicatifs permet de mieux gérer les périodes de surcharge ou au contraire celles de sous-charge. En effet on peut estimer que dans cette configuration, le pire scénario est une phase de surcharge de chaque application à la même période. Dans ce cas, le service affronte les mêmes difficultés qu'auparavant lorsque chaque application était indépendante. Au contraire si une phase de sous charge de l'application A correspondait à une phase de surcharge de l'application B, il était aisé, en mode mutualisé, de basculer des ressources d'une application vers l'autre pour répondre à la demande.

- Meilleure gestion des remplacements (période de congés ou autres absences)

La gestion des remplacements est facilitée par le fait que l'équipe a une meilleure connaissance globale de chaque application. Le passage de compétence est donc limité aux problèmes spécifiques de la période.

- Augmentation de la facilité de rotation des effectifs

Le fait d'étendre la compétence d'une application à plusieurs personnes et de limiter l'apparition d'experts irremplaçables, induit la possibilité d'intégrer et de libérer des consultants plus aisément. Cette possibilité est un avantage à la fois pour le management de l'équipe et pour les collaborateurs qui ne sont plus verrouillés à un applicatif.

- Réduction des coûts

Comme dans le cadre de fusions de sociétés, il est possible d'effectuer certaines économies d'échelle en rationalisant les processus. Ainsi la part de pilotage se trouve diminuée. Le lissage vu précédemment est un autre axe de diminution des coûts. En effet le dimensionnement d'une équipe est généralement défini par rapport à sa charge moyenne pondérée d'éventuels pics de charges connus à l'avance. Le lissage de ces pics sur plusieurs applications permet d'ajuster le dimensionnement au plus près de la charge moyenne.

- Augmentation générale de la connaissance du produit SAP BW

Pouvoir travailler sur la même technologie mais dans des contextes fonctionnels différents permet de varier les types de demandes et donc d'augmenter la connaissance générale de chacun.

- Facilitation de l'arrivée de futurs clients.

La trajectoire du service, autrefois dédié à Schneider-Electric est maintenant tournée vers l'accueil de nouveaux clients. Le passage à une organisation mutualisée prépare les collaborateurs à un mode de fonctionnement permettant d'intégrer de nouveaux périmètres naturellement en utilisant les processus et les bonnes pratiques existantes.

Risques :

- Dilution de la connaissance spécifique de chaque application

L'organisation mutualisée limite la production d'experts connaissant l'intégralité de l'application. Ces experts peuvent présenter l'intérêt de rassurer le client et d'être capable de répondre rapidement à toutes les problématiques du domaine. La mutualisation, elle, implique de gérer la connaissance de l'application au niveau de l'équipe. Selon les sujets le client a alors à faire à un collaborateur moins expérimenté dont le niveau de réponse est moindre que celui de l'expert.

- Fragilisation des membres de l'équipe

Pour les personnes ayant été habituées à ne travailler qu'avec un seul interlocuteur client et sur une seule application depuis plusieurs années la mutualisation pouvait s'avérer être un facteur déstabilisant. Le fait de travailler sur un nouveau domaine fonctionnel donnait le sentiment d'être moins compétent et de ne pas être en mesure de maîtriser son action.

- Activité éclatée sur différents clients et dossiers

La multiplication des clients entraîne la multiplication des dossiers par collaborateur, il faut donc pouvoir gérer la liste des tâches de manière à ne pas en laisser de côté. En particulier, les activités annexes au développement ou les actions de documentation souvent considérées comme secondaires par les collaborateurs ne doivent pas souffrir de cette organisation.

De plus, l'intervention sur des sujets différents pouvait engendrer plus d'interruptions dans le déroulement d'une tâche et générer des pertes liées à la bascule d'une activité à l'autre et au temps nécessaire à retrouver le niveau de concentration atteint avant l'interruption.

- Gestion des priorités

Le fait d'avoir plusieurs clients implique d'avoir un risque de conflit de priorités. Il est donc nécessaire de disposer d'instances et d'outils permettant de réduire l'impact de ce type de risque. De plus il faut que les collaborateurs soient capables de faire preuve d'une plus grande flexibilité et d'une plus grande adaptabilité afin d'augmenter le nombre de solutions possibles à un problème donné.

- Risque d'effondrement de l'équipe

Dans une organisation en silo, les risques sont limités à une seule application. Une surcharge ou une crise n'impactent pas les autres domaines. Dans une organisation mutualisée, il faut veiller à ce qu'un problème sur une application ne remette pas en cause la production des autres.

1.1.2. Une nouvelle gestion de la connaissance

La transformation du service engendrait un besoin accru d'une bonne gestion de la connaissance. Pour maximiser les avantages, il fallait permettre à chaque personne de l'équipe de pouvoir traiter des sujets de plusieurs domaines fonctionnels, l'idéal étant que tout le monde soit capable de traiter tous les sujets. Néanmoins, cet idéal nécessitait un effort de formation qui pouvait être trop important et donc contre-productif. Il était donc nécessaire d'ajuster la balance entre un coût de gestion de la connaissance trop élevée d'une part et une absence de partage d'autre part.

Pour un consultant, le fait d'intervenir sur un nouveau domaine, impliquait d'aller interroger plus souvent la documentation. De plus la recherche risquait d'être plus fastidieuse si les conventions de classement ou le type de documentation n'était pas uniforme. Dans un environnement mutualisé, la gestion de la connaissance et de la documentation devenait donc un enjeu crucial si l'on voulait pouvoir permettre à des personnes ayant la connaissance technique mais pas l'expérience fonctionnelle d'intervenir avec un maximum de rapidité.

Pour illustrer le propos, nous pouvons prendre l'exemple de l'application ASARIS.

Cette application a été développée en 2005 par Schneider-Electric avec l'aide de consultants internes. Le niveau de documentation de l'application était alors à son maximum car toutes les fonctionnalités ont été décrites en détail dans le cadre du projet. La documentation était alors stockée dans un outil de gestion documentaire, propriété de Schneider.

A la suite du projet, la phase de maintenance a amené son lot de modifications à l'applicatif. Une importance moyenne a alors été accordée à la mise à jour des documentations projet. Une partie des modifications a été uniquement décrite dans de nouveaux documents de maintenance sans reprendre les dossiers originaux. Pour certaines fonctionnalités il devenait alors nécessaire de compiler plusieurs documents pour avoir une vue globale.

L'activité a ensuite été externalisée, ainsi que l'ensemble du système d'information Schneider-Electric, la maintenance passant sous responsabilité Capgemini.

Par la suite, des équipes indiennes ont été intégrées au modèle de production. Les collaborateurs de ces équipes mettaient à jour les documents en anglais à partir des traductions des documentations originales, et les nouveaux documents étaient créés en langue anglaise.

Enfin, pour permettre de partager la documentation entre les consultants indiens et français, l'ensemble des documentations avait été basculé sur un nouvel outil de gestion de la documentation plus efficace et plus puissant, offrant une meilleure gestion des versions. L'outil qui a été choisi était l'outil privilégié par le groupe Capgemini : Teamforge de Collabnet.

On voit que la documentation a subi plusieurs phases distinctes qui ont contribué à la rendre hétérogène, morcelée voire obsolète pour certaines parties.

Il devenait donc nécessaire d'identifier de nouveaux moyens de gérer la documentation pour permettre de la rendre efficace rapidement et lui conserver une cohérence tout au long de la vie de l'application. Pour ce faire, il fallait mettre en place un cercle vertueux de la connaissance afin de conserver, renouveler et enrichir le savoir collectif (Pillou, 2006). Ces moyens devaient en outre être suffisamment simples pour ne pas diminuer la productivité de l'équipe.

1.1.3. Les difficultés de la multi culturalité

Le groupe Capgemini s'appuie fortement sur des équipes « offshore » situées dans plusieurs pays dit « à bas-coût » dont l'Inde. Actuellement le groupe dispose d'une force de travail de plus de 35 000 collaborateurs dans ce pays sur un effectif mondial de près de 120 000. À ce jour, l'Inde dispose d'une des meilleures offres d'offshore et continue de progresser grâce à une amélioration de son système éducatif (Gartner, 2011a).

La combinaison de ces équipes offshore avec des collaborateurs localisés permet de tirer partie des avantages des deux modèles :

- L'utilisation de l'offshore permet de disposer d'un groupe de collaborateurs important et d'être compétitif en termes de coûts (Gartner, 2011b).
- L'utilisation de ressources françaises permet d'être proche des clients sur le plan géographique, culturel et linguistique.

Cette approche implique donc l'existence d'équipes multiculturelles au sein de Capgemini et donc une prise en compte de cet aspect dans le management.

Dans le cadre du service SAP, la notion de management interculturel a été gérée au début de la relation avec l'Inde grâce à l'organisation de formations ou de voyages d'échange. Malheureusement, cet aspect n'a pas été suivi dans le temps et, rapidement, les équipes indiennes ont été abordées comme des fournisseurs de services. Ils étaient dirigés par un mode de gestion rationnelle fondé sur l'idée que l'utilisation de processus communs permettrait de garantir la coordination des équipes et donc la productivité. Les valeurs culturelles de l'entité « dominante » (client) ont donc été appliquées à celles de l'entité « dominée » (fournisseur) sans prendre en compte les risques de blocages ou d'incompréhensions liés aux différences de cultures.

Cette approche s'est avérée superficielle et il est dangereux de croire que toutes les organisations et tous les individus de la planète peuvent être régis selon les mêmes règles et les mêmes schémas (Meier, 2010).

De plus, les gains attendus par un projet d'offshore sont souvent minorés par un certain nombre de coûts cachés liés, en partie, au management des ressources externalisées.

En effet, la part de management est accrue par la distance physique et culturelle des équipes. Cette distance culturelle est d'ailleurs rarement prise en compte dans la gestion des ressources ce qui contribue à réduire l'efficacité de ce management. De manière générale, la complexité et la particularité des relations interculturelles sont négligées car des mécanismes de stéréotypes, de jugements de valeur ou de préjugés dominent.

- Les **stéréotypes** sont des simplifications de la réalité ayant pour but de classer les individus selon des catégories figées et donc de nier les différences de chaque personne. Concernant l'Inde, le fait d'associer le pays à la religion hindouiste est un stéréotype. En effet, près de 20% de la population, soit près de 250 millions de personnes, ne pratique pas cette religion (Wikipedia, 2012).

- Un **préjugé** est un jugement fait à l'avance et qui nous est souvent imposé par une opinion ou une croyance collective. A la différence du stéréotype, le préjugé a une connotation subjective et très souvent négative. Par exemple, pour l'Inde le préjugé se matérialise souvent par une perception médiocre de la qualité des développements ou par un doute sur la capacité des Indiens à comprendre le besoin exprimé.
- **Les jugements de valeurs** sont, par oppositions aux jugements de faits, des appréciations de valeurs subjectives. Ces appréciations portent sur des critères qui peuvent être discutables selon le point de vue mais que l'on cherche à rendre universelles. Les jugements de valeurs portent souvent sur la beauté, la morale ou les notions de bien et de mal. Ils ne sont ni vrais ni faux, mais leur acceptation, ou au contraire, leur refus, revient à valider ou à rejeter l'essentiel des croyances et des valeurs qui les ont générées. Par exemple, considérer comme injuste le système de caste en Inde est un jugement de valeurs.

Les relations interculturelles sont une des difficultés importantes du management moderne et les tentations de réduire leur complexité en utilisant des artifices comme les préjugés ou les stéréotypes sont grandes. L'utilisation de filtre et la négation des différences peuvent conduire à perdre la richesse potentielle d'une collaboration internationale, et dans le pire des cas générer des conflits coûteux, susceptibles de réduire fortement les gains attendus d'une utilisation de l'offshore. Dans ce cas, c'est la compétitivité des entreprises françaises qui s'en trouve menacée.

Il semblait donc important d'étudier, en parallèle des pratiques Agile et Lean, la gestion culturelle de l'équipe afin d'identifier et d'assumer les différences de chacun et de trouver un socle commun permettant de bâtir une vraie relation entre les différents collaborateurs.

Ce biais était une des pistes d'amélioration des conditions de travail de chacun en limitant les conflits et les incompréhensions, sources de stress. Il était espéré que cette augmentation de la synergie entre les équipes puisse avoir un impact sur la productivité globale même si cela a pu être difficile à mesurer.

1.2. Le modèle ALM Capgemini

Dans le cadre du problème posé, l'étude des solutions existantes et des travaux antérieurs devait au préalable se porter sur les pratiques en vigueur au sein de la société. En effet le centre de service BW est intégré dans un centre SAP, lui-même compris dans un ensemble multi technologique regroupant plusieurs clients. A ce titre, le centre de service doit respecter les pratiques en vigueur en s'inspirant du cadre défini pour l'intégralité du domaine.

1.2.1. Définition de l'ALM

ALM est une offre portée par l'entité « Application Services » qui est un des services de Capgemini.

La figure 3 présente l'organigramme global de la société.

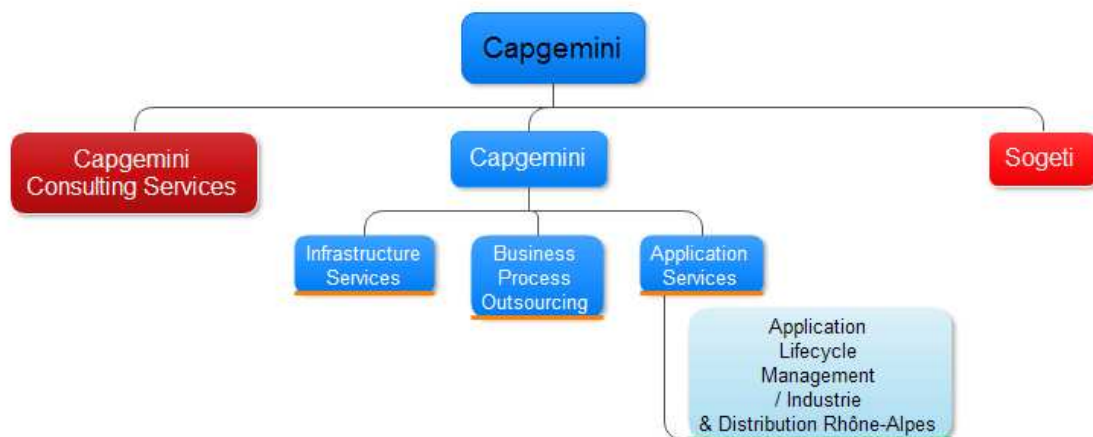


Figure 3 - Organigramme général Capgemini

Généralités

Le modèle historique de maintenance applicative externalisé est nommé TMA pour Tierce Maintenance Applicative (maintenance d'application par un tiers). L'approche traditionnelle était de fournir un service dédié à un client correspondant à une brique du système d'information. Ce service étant composé d'outils, de ressources humaines et d'un pilotage de la prestation, généralement sur des périodes de plusieurs années. Le client ne gère plus alors son service informatique mais uniquement le respect des engagements en termes de qualité et de production.

Les contrats de TMA intègrent donc la mise en place de plans d'assurance qualité (PAQ), mais aussi de SLA (Service Level Agreement) dans lesquels les deux parties s'accordent sur des délais de réponse, des exigences de disponibilité, et d'autres critères de performance. Enfin des indicateurs sont définis pour mesurer le respect de ces objectifs.

Généralement, un contrat de TMA se limite à un aspect du service, par exemple le support utilisateur, ou le développement.

Une première évolution du modèle TMA est la GPA (Gestion du Patrimoine Applicatif). Dans cette approche, le contrat d'infogérance ne concerne plus des applications gérées de manière autonome mais des groupes d'applications pouvant représenter des processus clients complets (Achats, Commerce, etc.). Une GPA peut aller jusqu'à l'externalisation intégrale d'un système d'information chez un prestataire, qui gère alors l'ensemble de la vie des applications :

- Support
- Maintenance évolutive (de la réception du cahier des charges jusqu'à la mise en production)
- Maintenance corrective

L'objectif de ce type de contrat est de réaliser des économies d'échelle par la mise en commun d'un certain nombre d'outils ou de services, par exemple, la gestion de la connaissance ou la gestion du changement.

L'autre objectif fondateur d'une GPA est la construction de plans d'évolution visant à transformer les systèmes d'information pour les tourner vers d'avantage de simplification et d'optimisation. Ceci représente donc un second levier de réduction de coût pour l'entreprise utilisatrice qui négocie généralement des contrats dégressifs dans le temps.

Enfin une GPA propose une vision « industrielle » du processus de gestion du système d'information. Pour un client, c'est l'assurance de modes de fonctionnements homogènes sur l'ensemble du périmètre couvert, ce qui facilite les interactions entre les utilisateurs et le prestataire. L'industrialisation apporte également une standardisation des niveaux de service.

D'après une étude menée conjointement par HP et Capgemini en Mars 2011, il apparaît que les entreprises ont besoin de rationaliser leurs systèmes d'informations. En effet, toujours selon cette étude, 85% d'entre elles affirment qu'elles ont un besoin de refonte de leur architecture et près de la moitié pensent que 50% de leur portefeuille d'application est obsolète (Capgemini, 2011). Cette attente forte est le moteur du modèle ALM qui propose aux utilisateurs d'externaliser des pans entiers de systèmes d'information et non plus de gérer une somme d'applications dans un modèle TMA. Cette approche permet d'aider le client à transformer son architecture grâce à une vision globale. Le modèle s'appuie également sur des compétences multi-technologiques et méthodologiques permettant de couvrir l'ensemble des besoins concernés.

Contrairement au modèle traditionnel (voir figure 4), l'ALM chez Capgemini est un modèle de GPA qui inclut, en plus, une mutualisation des services sur plusieurs clients comme l'illustre la figure 5.

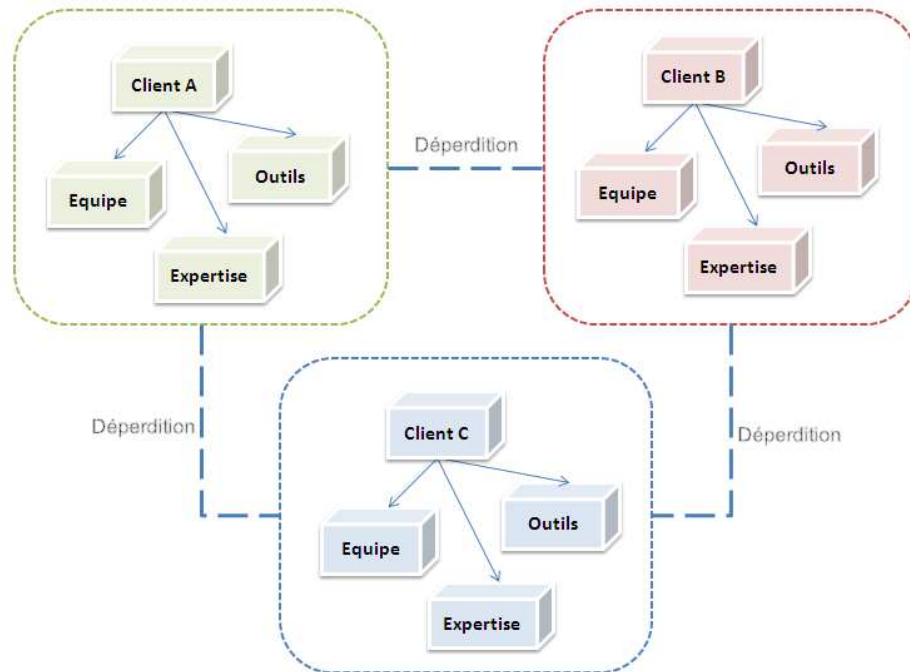


Figure 4 - Modèle traditionnel

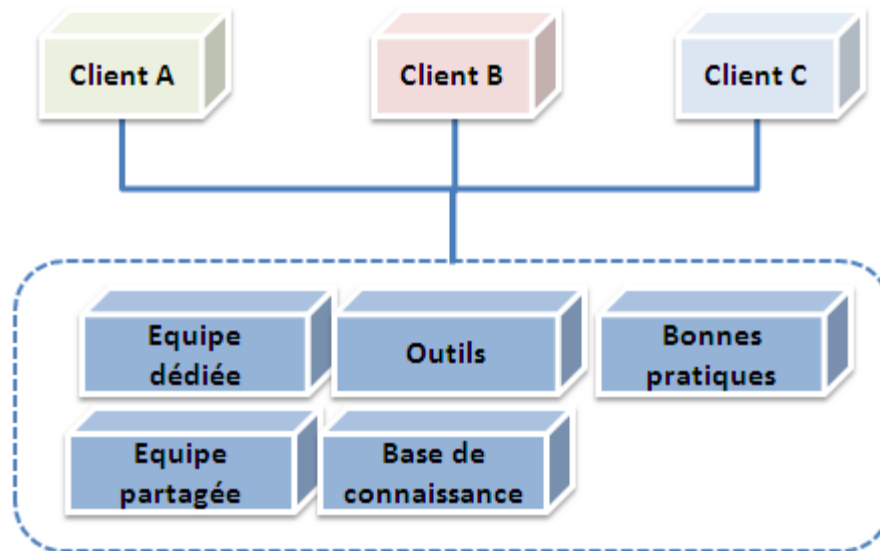


Figure 5 - Modèle d'équipes mutualisées

L'extension du périmètre d'intervention permet évidemment de multiplier les économies d'échelle, mais également de proposer plus de services aux utilisateurs. En effet, en mutualisant les ressources, il est plus facile de financer l'intervention d'un consultant expérimenté ayant des compétences spécifiques qui se serait avéré trop cher pour un seul client.

Aujourd'hui, le pôle ALM de Grenoble gère plusieurs clients : Schneider-Electric, Rio-Tinto, Martinet, Lennox etc.

En ce qui concerne BW seul Schneider-Electric et maintenant Rio-Tinto sont concernés mais les équipes commerciales travaillent sur de nouveaux prospects qui pourraient inclure des prestations BW.

ALM SAP de Montbonnot

Voici le schéma d'organisation général de l'ALM SAP de Montbonnot. Le service est découpé en trois pôles organisés par branches métiers. Le pôle Finance, BW, et « Autres comptes » regroupe des activités hétérogènes dont notre centre de service BW. Le compte Rio-Tinto est rattaché au domaine de la production, sauf pour sa partie BW.

La figure 6 présente l'organigramme de l'ALM SAP de Montbonnot.

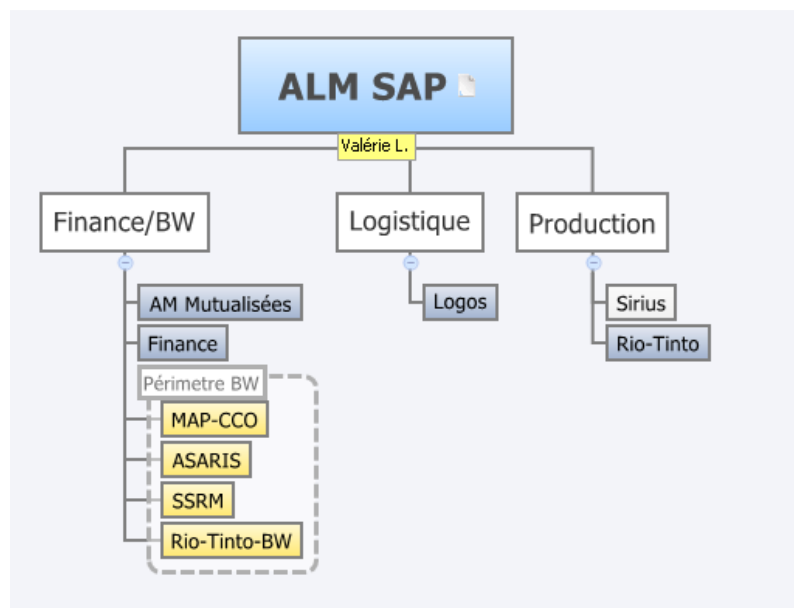


Figure 6 - Organigramme ALM SAP Montbonnot

Chaque pôle est dirigé par un SDM (Service Delivery Manager) qui porte la responsabilité de la production des différents comptes du pôle. Le SDM est également le garant de l'engagement de Capgemini vis-à-vis de ses clients.

1.2.2. L'industrialisation

L'industrialisation en place au sein de l'ALM repose sur plusieurs aspects destinés à assurer une gestion cohérente des services : le reporting, l'outillage, le référentiel ITIL et le pilotage interne des comptes.

Reporting

Le reporting est une partie importante de l'industrialisation. La possibilité de disposer de tableaux de bords cohérents sur l'ensemble des domaines permet d'avoir une vision rapide et synthétique de la situation, quel que soit le périmètre concerné.

Le pilotage des SLA et des niveaux d'engagement est ainsi effectué grâce à un outil Qlickview développé et maintenu par une équipe de reporting. Cette équipe se charge d'extraire les données des différents outils de gestion des demandes et d'en présenter une vue agrégée par client. La figure 7 présente un exemple de tableau de bord pour les SLA applicatifs gérés sur le client Schneider-Electric :

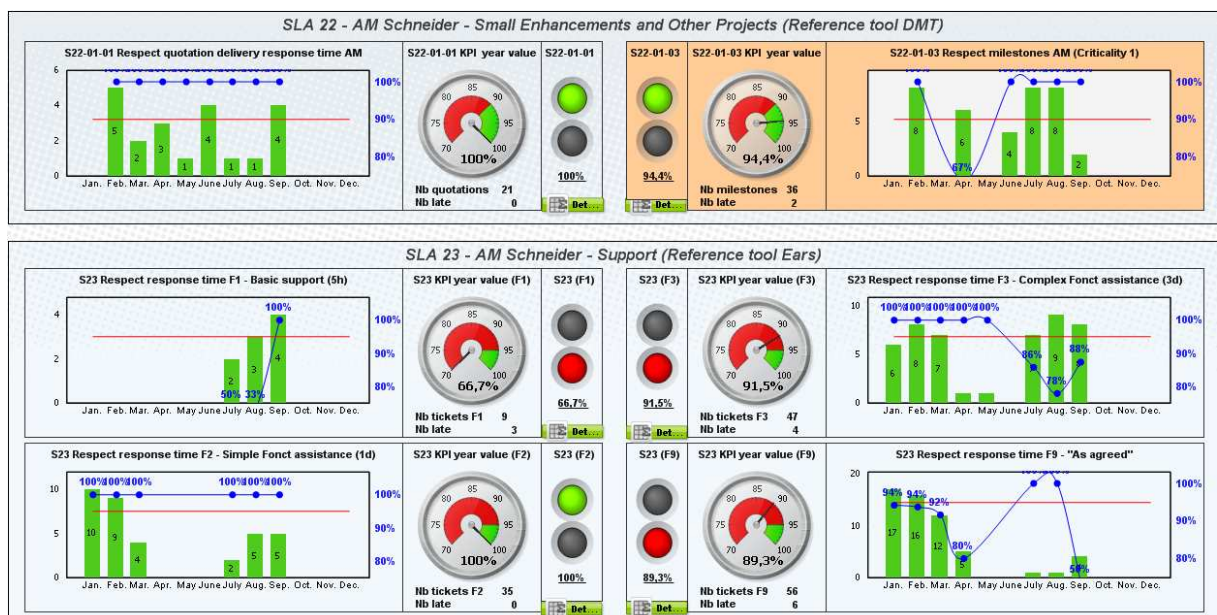


Figure 7 - Exemple de tableaux de bord SLA

Ces tableaux de bords présentent l'état mensuel des indicateurs de services ainsi que l'évolution au fil de l'année. Ces niveaux sont partagés avec le client et des plans d'actions sont mis en place en cas de dérive des résultats.

L'équipe reporting produit également en interne un tableau des consommations portant sur les demandes d'évolution. Ces tableaux mensuels permettent de comparer les jours consommés aux jours « vendus » pour une demande d'évolution et de détecter une éventuelle dérive donnant lieu à un plan d'action. Cet indicateur est standardisé au niveau de tout le contrat Schneider-Electric mais n'est pas déployé sur Rio-Tinto car la facturation est pour l'instant effectuée au temps passé.

Enfin, il existe un troisième niveau de reporting de type financier qui est standardisé pour tout Capgemini. Ces états présentent les résultats sur les différents contrats (chiffre d'affaires, marge, etc.).

Outillage

L'outillage est très présent dans le modèle ALM et permet de rationaliser bon nombre de tâches. Les outils disponibles sont en outre des garants de l'application des processus en place.

Par exemple, pour la gestion des demandes d'évolution dans le cadre du contrat Schneider-Electric, un outil appelé DMT (Demand management tool – outil de gestion de la demande) a été développé. Celui-ci permet de suivre l'évolution d'une demande de manière partagée entre le client et Capgemini. Pour ce faire, l'application dispose d'un workflow interne. La clôture de chaque phase requiert des informations ou des livrables spécifiques qui jalonnent la vie de l'évolution.

L'intérêt de cet outil réside dans le fait que sa manipulation est partagée entre Schneider-Electric et Capgemini. Chaque partie est responsable de l'avancement du workflow pour les phases qui lui incombent. Le processus de gestion de la demande est ainsi piloté par l'outil de manière équivalente sur l'ensemble des applications du contrat.

Pour la **gestion de la connaissance**, l'application Teamforge de Collabnet a été déployée. L'utilisation standardisée de cet outil à travers tout le service SAP permet de posséder un référentiel de documentation interne cohérent sur l'ensemble des contrats.

Clarity est l'outil de **gestion des temps** des collaborateurs. Il permet de mesurer le temps passé sur les différentes tâches (gestion des demandes, support, ou autres). Les données stockées dans l'outil sont ensuite extraites pour servir de bases aux comités de maintenance et suivre, notamment, la productivité du service sur la gestion des demandes.

Pour la **gestion des tests**, l'application HPQC (Hewlett-Packard Quality Center) permet de capitaliser sur la préparation et l'exécution des tests. Chaque campagne de test est préparée dans l'outil avec la liste des étapes à dérouler en fonction du composant à tester. L'équipe dispose ainsi d'un référentiel permettant de savoir quels tests appliquer dans quelles situations.

Le plan de test est découpé par modules fonctionnels cohérent. Un développeur modifiant une partie de ce module a ainsi la liste des tests pertinents pour vérifier la non-régression

D'autre part, l'outil permet de stocker le résultat d'une implémentation d'un plan de test dans la partie appelé « Test Lab. ». Il permet ensuite d'éditer un état garantissant la bonne exécution des scénarios de test et le résultat de ceux-ci.

Grâce à la liste de tous les tests et de leurs résultats attendus ou avérés, cet outil devient une base de connaissance importante pour décrire le fonctionnement d'une application. De plus, HPQC permet de capitaliser sur les problèmes rencontrés. En effet, l'outil possède un module « Defect » (défauts) qui permet d'enregistrer la liste des retours faits pendant les phases de recette. Ceci permet de conserver une vision des erreurs rencontrées pour éviter de les reproduire à la prochaine évolution, mais également de disposer d'un indicateur de qualité sur les développements effectués.

L'intérêt de l'outil réside dans la qualité des tests et des résultats de tests documentés. Il n'est pour l'instant utilisé que pour Schneider-Electric, Rio-Tinto n'ayant pas souhaité l'implémenter. Le processus de test chez Rio-Tinto est géré manuellement mais assez bien documenté dans des fiches de test remises au moment du cahier des charges par le client et étoffées pendant l'analyse jusqu'à la livraison d'une spécification.

Processus ITIL

Trois processus du référentiel ITIL sont en place au niveau de l'ALM et sont donc applicables dans le cadre du centre de service BW pour Schneider-Electric. Ces processus, la gestion des incidents, la gestion des problèmes et la gestion des changements sont décrits ci-dessous.

Pour Rio-Tinto, les processus de gestion des incidents et des problèmes ont été gardés en interne chez le client, et celui-ci n'a pas souhaité mettre en place de gestion des changements sur le périmètre applicatif. Il s'agit néanmoins d'une piste d'amélioration que Capgemini cherche à mettre en place chez Rio-Tinto.

Gestion des changements

Dans le référentiel ITIL la gestion des changements est un processus incontournable. On estime que 80% des incidents survenant dans un système sont liés à un changement mal maîtrisé (Chamfrault, Durand, 2011).

Dans ITIL V3 la gestion des changements se concentre sur les demandes de changement non standard, c'est-à-dire ne faisant pas partie d'un catalogue prédéfini et nécessitant une qualification préalable.

Dans ITIL V3 le processus est basé sur une RFC (Request For Change ou demande de changement) et se présente comme ceci :



Figure 8 - Processus de gestion des changements

La gestion des changements chez Capgemini ALM reprend la définition d'ITIL : lorsqu'un changement non standard est identifié, une RFC est créée. Elle est positionnée dans le statut QG1 (Quality Gate 1). Les Quality Gate (ou passage qualité) sont des étapes identifiant l'avancement d'une demande. Elles sont au nombre de 5 :

1. QG1 : La demande de changement est créée
2. QG2 : Une date d'implémentation est validée
3. QG3 : Le changement est prêt à être implémenter
4. QG4 : Le changement a été mis en place. Un compte rendu d'implémentation nommé PIR (Post Implementation Report) est créé.
5. QG5 : Le changement a été validé par le client et peut être clos.

Le passage d'une QG à une autre se fait dans une instance de décision nommée CAB (Change Advisory Board). Ce comité se réunit de manière hebdomadaire en présence d'un représentant de chaque type de service (poste de travail, gestion de l'infrastructure, gestion des applications et gestion du réseau) ainsi que du gestionnaire des changements qui est le garant du respect du processus. Le management valide ainsi le passage des sujets dans les différents statuts et s'assure que tous les critères de déploiement soient remplis.

Gestion des incidents

Le second processus ITIL déployé pour le client Schneider-Electric est la gestion des incidents. L'objectif de ce processus est de rétablir au plus vite un niveau de service tel qu'il est défini dans le SLA.

Il existe plusieurs enjeux pour le fournisseur de service dans ce processus. Il faut d'abord déterminer la gravité et l'impact de l'incident et ensuite en déduire la priorité et les moyens à affecter. Vis-à-vis du métier, l'objectif de ce processus est de partager de manière transparente l'état du système et l'avancement de la résolution.

La figure 9 illustre le processus :



Figure 9 - Processus de gestion des incidents

Le but de la gestion d'un incident n'est pas de déterminer la cause de celui-ci mais bien de rétablir le service au plus vite, l'identification de la cause étant du domaine de la gestion des problèmes. A ce titre, Capgemini est engagé sur un certain nombre de niveaux de services relatifs à la gestion des incidents :

- Délai de communication d'un incident
- Délai de résolution d'un incident
- Délai d'escalade d'un incident

Une instance hebdomadaire de partage et de revue des incidents est organisée avec le client afin de valider les gravités déterminées à chaud et la responsabilité de l'incident.

Gestion des problèmes

Ce processus est le dernier mis en place dans le cadre du contrat Schneider-Electric. Il vise à identifier les causes d'un incident afin d'en empêcher la réitération.

Une bonne gestion des problèmes permet d'améliorer la productivité des équipes de support et de minimiser l'impact négatif des incidents sur le métier. Elle permet également de réduire le nombre d'incidents.

Voici la représentation du processus :

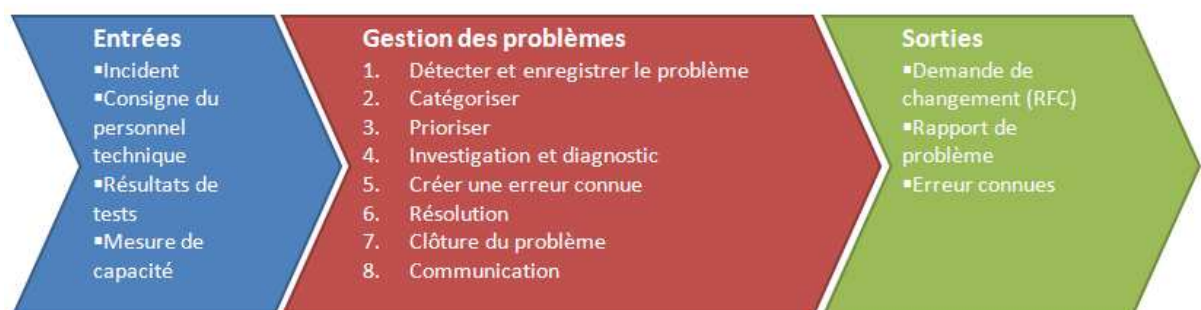


Figure 10 - Processus de gestion des problèmes

Il est important de noter que dans ITIL, le déclencheur d'un problème n'est pas nécessairement réactif, mais il peut être proactif.

La détection et l'enregistrement se fait par l'équipe de gestion des problèmes qui est en étroite communication avec l'équipe de gestion des incidents. Le problème est ensuite suivi à travers ses différentes phases, de la détermination de la cause d'origine à la résolution proprement dite du problème. L'état d'avancement est partagé avec Schneider-Electric de manière hebdomadaire car ce processus est bénéfique pour les deux parties (réduction des perturbations pour le métier et limitation du risque de pénalités lié aux incidents pour Capgemini).

Pilotage des comptes

Le pilotage d'un compte client est en partie standardisé sur l'ensemble des contrats. Il se décompose en trois instances de niveaux différents.

1. Comité opérationnel

Le comité opérationnel est mené entre le responsable technique du client et le responsable opérationnel Capgemini. Il est hebdomadaire ou bimensuel et traite de l'avancement des sujets de terrain. Selon les cas, il peut inclure un utilisateur ou un développeur pour échanger sur un sujet particulier.

Ce comité opérationnel est la seule instance qui n'est pas contractualisée et qui n'est donc pas en place sur tous les domaines.

2. Comité de maintenance

Le comité de maintenance se tient de manière mensuelle en compagnie du responsable client et du SDM Capgemini mais également des représentants commerciaux du domaine.

Le comité de maintenance aborde les points suivants :

- Suivi de l'avancement des budgets (vision financière).
- Respect du SLA.
- Gestion des risques.
- Prévisions d'activité.
- Revue de la satisfaction client (donnant lieu à une appréciation).

Suivant la nature des risques ou des résultats, le comité de maintenance peut donner lieu à des plans d'actions en vue de traiter les différentes difficultés.

3. Comité de pilotage

Ce comité est biannuel et regroupe les responsables de service, aussi bien du côté système d'information que du côté métier. Il permet de donner les grandes directions stratégiques pour le prochain semestre. Les décisions prises dans ces comités débouchent généralement sur des plans d'actions tant organisationnels qu'opérationnels.

1.2.3. Approche offshore

Dans l'approche traditionnelle de l'offshore (voir figure 11) qui est utilisé par de nombreuses sociétés, les tâches entre les différents services sont extrêmement compartimentées. La transmission d'information ne se fait que dans un sens et les possibilités de rétroaction sont faibles. Ce type de fonctionnement est simplement l'implémentation d'un cycle en cascade avec une équipe de développement éloignée.

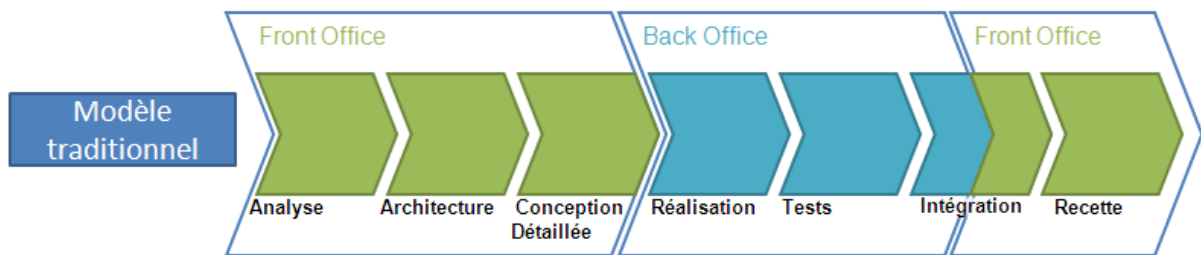


Figure 11 - Modèle traditionnel d'offshore

On associe généralement deux équipes :

- Un « front office » (terme qui peut se traduire par guichet ou réception) et qui est proche du client en matière de langue, de culture, ou d'horaires de travail.
- Un « back office » (qui peut se traduire par « arrière boutique ») est, par opposition la partie du dispositif qui n'est pas visible par le client.

Ce modèle permet de réduire les coûts de fonctionnement d'un service informatique sans contraindre le client à devoir gérer des différences culturelles, linguistiques ou de fuseau horaire avec son prestataire de service. Ces tâches sont en effet du domaine du Front Office, sorte de médiateur organisationnel.

Le nouveau modèle de gestion de l'offshore chez Capgemini (en place depuis 2009) vise à réduire les incompréhensions entre les équipes front office et les équipes back office, de la même manière que les méthodes agiles visent à réduire les incompréhensions entre développeur et client.

Comme le montre la figure 12, dans le modèle d'offshore baptisé « One Team », la frontière entre front-office et back-office se réduit pour aboutir à une gestion conjointe des dossiers. L'avancement dans les phases ne se fait plus par un passage de relais, source de perte d'information, mais par un accompagnement progressif.

Cette méthode permet une meilleure gestion des risques (déformation du besoin, solution technique inadaptée) en les identifiant plus tôt dans la vie d'une évolution.

Comme son nom l'indique, cette organisation a pour objectif de fusionner front office et back office dans une seule équipe projet avec une vision commune.

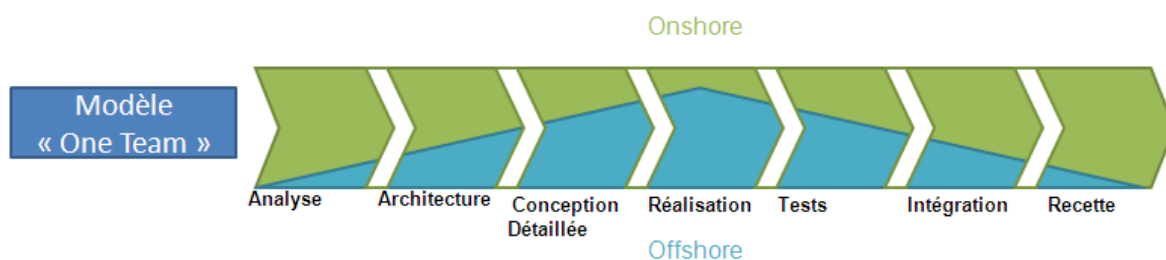


Figure 12 - Modèle "One Team"

Pour ce faire, il est nécessaire d'appliquer les principes suivants :

- Les outils et les méthodes doivent être partagés et appliqués de manière transparente qu'on soit sur site ou offshore.
- Une demande doit être gérée de bout en bout et non pas avec des relais.

1.2.4. Synthèse de l'ALM et adaptation au centre de service BW

Par définition, le centre de service BW s'insère dans le modèle ALM. Toutes les caractéristiques d'un centre ALM ont été appliquées, sous réserve de compatibilité avec les demandes contractuelles des clients.

Le tableau I illustre l'application des différents items du modèle ALM selon chaque client du centre de service BW.

Tableau I - Synthèse de l'application du modèle ALM par client

	Schneider-Electric	Rio-Tinto
Reporting Standardisé	✓	✓
Outil de gestion des demandes de support	ITSM	OVSD
Outil de gestion des demandes d'évolution	DMT	OVSD
Gestion de la connaissance	Teamforge	Teamforge
Gestion des tests	HPQC	Manuel
ITIL	Gestion des Incidents Gestion des Problèmes Gestion des Changements	Gestion en interne Rio-Tinto
Comité opérationnel	✓	✓
Comité de maintenance	✓	✓
Comité de pilotage	✓	✓
Offshore One Team	✓	Pas d'offshore pour Rio-Tinto

1.3.Agile

Un environnement composé de 5 applications, deux clients (au sens de l'entreprise) et multiculturel est nécessairement considéré comme étant agité. Or une des principales missions des méthodes Agile est justement de répondre à ce type d'environnement.

Cette section définit les principes généraux de l'agilité et analyse quels éléments ont pu être utilisés dans le cadre du centre de service BW.

1.3.1. Généralités

La philosophie Agile est une des réponses apportées pour répondre aux problèmes de développement de logiciels et améliorer le taux de réussite des projets informatiques (Hibbs et al, 2009). Matérialisé par la célèbre étude CHAOS de 1994 présentant des statistiques affolantes : seulement 16% de projets terminés dans les temps respectaient le budget et le périmètre définis au départ (Standish Group, 1994). L'échec des projets informatiques est principalement imputé au modèle de la cascade (figure 13) qui exclut le client dès les phases de spécifications et n'autorise pas de retour en arrière une fois une étape terminée.

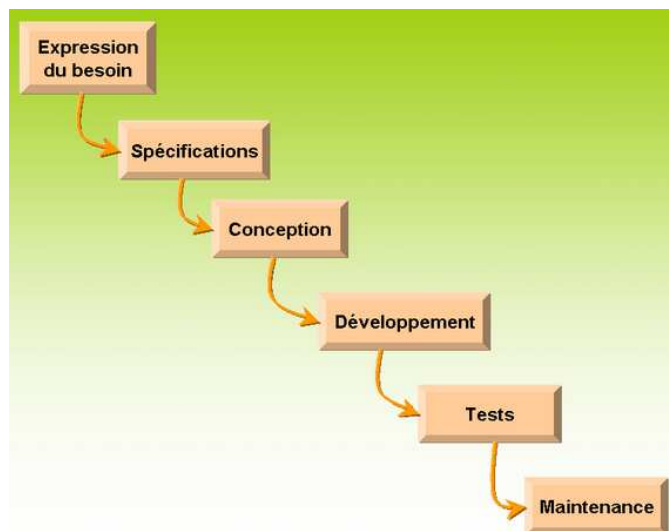


Figure 13 - Cycle en cascade

Une méthode Agile se caractérise par une approche itérative et incrémentale. Parmi ces méthodes on peut citer les plus célèbres : SCRUM (qui signifie mêlée en anglais), RAD (Rapid Application Développement), FDD (Feature Driven Development), Agile UP, ou XP (Extreme Programming).

Les principes généraux des méthodes agiles ont été regroupés dans une liste appelée « Manifeste agile » comprenant 12 principes (Agile Alliance, 2012) :

1. La plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à forte valeur ajoutée.
2. Le changement est accepté, même tardivement dans le développement, car les processus agiles exploitent le changement comme avantage compétitif pour le client.
3. La livraison s'applique à une application fonctionnelle, toutes les deux semaines à deux mois, avec une préférence pour la période la plus courte.
4. Le métier et les développeurs doivent collaborer régulièrement et de préférence quotidiennement au projet.
5. Le projet doit impliquer des personnes motivées. Donnez-leur l'environnement et le soutien dont elles ont besoin et faites leur confiance quant au respect des objectifs.
6. La méthode la plus efficace de transmettre l'information est une conversation en face à face.
7. L'unité de mesure de la progression du projet est un logiciel fonctionnel (ce qui exclut de comptabiliser les fonctions non achevées).
8. Les processus agiles promeuvent un rythme de développement soutenable (afin d'éviter la non-qualité découlant de la fatigue).
9. Les processus agiles recommandent une attention continue à l'excellence technique et à la qualité de la conception.
10. La simplicité et l'art de minimiser les tâches parasites sont appliqués comme principes essentiels.
11. Les équipes s'auto-organisent afin de faire émerger les meilleures architectures, spécifications et conceptions.
12. A intervalle régulier, l'équipe réfléchit aux moyens de devenir plus efficace, puis accorde et ajuste son processus de travail en conséquence.

D'abord conçue pour le développement de logiciels, la philosophie agile s'est propagée à la gestion des organisations sous le terme de « Management Agile ».

Cette philosophie veut promouvoir la responsabilisation de chaque collaborateur en lui affectant des tâches étendues (par opposition à la simplification des tâches vues dans le Taylorisme). On fait ainsi confiance aux ressources opérationnelles pour appréhender la complexité des tâches qui leur incombent. En retour, les collaborateurs sont plus motivés car leurs missions sont plus intéressantes et ils ont plus de levier d'action pour résoudre les problèmes.

Enfin le management Agile utilise l'amélioration continue des processus. En déployant cette activité jusqu'au niveau les plus fins de la hiérarchie, l'entreprise est capable d'optimiser sa production et de détecter les gaspillages même minimes qui, une fois consolidés, peuvent produire de grandes économies de temps.

1.3.2. Analyse des méthodes SCRUM et XP

Les méthodes agiles sont nées du constat que les temps de développement des méthodes classiques sont trop longs, que les applications livrées ne correspondent pas toujours au besoin, que les développeurs sont peu impliqués dans la conception et que personne n'a de vue complète d'un système développé.

SCRUM

SCRUM est aujourd'hui la méthode agile la plus utilisée. 52% des méthodes agiles utilisées en 2011 étaient du SCRUM (Versionone, 2011).

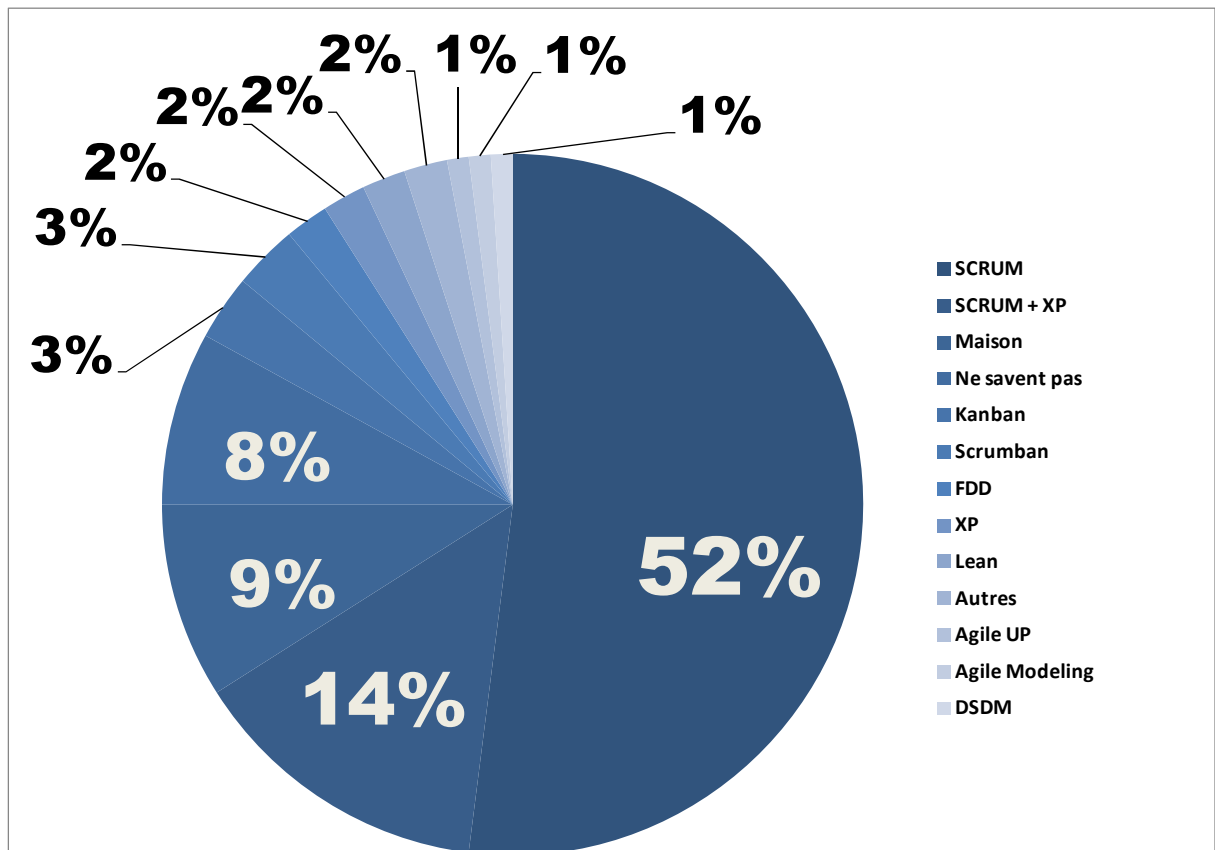


Figure 14 - Utilisation des méthodes Agiles (Versionone, 2011)

D'après le guide SCRUM, la méthode s'appuie sur trois piliers (SCRUM, 2012) :

1. Transparence

La transparence a pour but de clarifier la vision du processus de développement. La transparence doit être réciproque entre les clients et les développeurs. Cette notion permet de partager les problèmes au plus tôt et donc d'en limiter le coût et les conséquences.

2. Adaptation

L'adaptation permet de corriger ou d'améliorer les processus défailants ou ne produisant pas un résultat satisfaisant.

3. Inspection

L'inspection correspond à la revue régulière des productions comparées aux objectifs affichés.

La méthode SCRUM s'appuie également sur une organisation particulière, composée d'un propriétaire de produit, que l'on peut assimiler au client, d'un SCRUM Master et d'une équipe de développement.

La particularité de l'équipe de développement est qu'elle est auto-organisée et pluridisciplinaire. Elle peut donc elle-même décider quel est la meilleure marche à suivre pour réaliser le résultat attendu. L'équipe possède également en interne toutes les connaissances et les capacités pour résoudre les problèmes rencontrés.

La condition nécessaire à la réussite de cette organisation est la motivation de l'équipe. En effet, les membres d'une équipe SCRUM doivent être complètement impliqués dans leur mission car les tâches demandées dépassent celles d'un développeur standard. Ils doivent notamment réfléchir à comment produire et prendre de la hauteur par rapport aux tâches techniques, ce qui nécessite un réel intérêt pour la mission.

La motivation est également une conséquence puisque les ressources gagnent en autonomie et en valorisation de leur travail. Ce ne sont plus de simples exécutants mais ils sont des acteurs du résultat.

De plus, SCRUM ne reconnaît aucune différence à l'intérieur d'une équipe de développement. Même si certains peuvent être spécialisés sur différents domaines, il n'y a pas, au sein des équipes, d'assignations à des tâches précises (analyse, tests.. etc.)

Comme évoqué précédemment, la méthode SCRUM définit des acteurs particuliers en plus de l'équipe de développement. Il s'agit du propriétaire de produit et du « SCRUM master ».

Le propriétaire de produit est la personne qui exprime et priorise les demandes de développements (appelés items du carnet de produit). Son objectif est de maximiser la valeur du produit développé et donc de s'assurer par ses demandes que le travail de l'équipe de développement est bien utile pour le métier. En supplément, la méthode SCRUM a dicté certaines règles encadrant la mission du propriétaire de produit en précisant notamment que ce rôle ne peut être confié qu'à une personne par équipe et qu'il est seul maître de la hiérarchisation des demandes.

Le « SCRUM master » a le rôle de faire appliquer les principes de SCRUM et de veiller à ce que tout le monde adhère aux valeurs de la méthode. Il est au service de l'équipe et du propriétaire de produit pour faciliter la communication. Il sert également de formateur pour aider les différents développeurs à tendre vers la qualité attendue. Il est également utile pour identifier et corriger les obstacles existants dans l'organisation et susceptibles de limiter l'efficacité de l'équipe.

En matière de fonctionnement, la méthode SCRUM est organisée autour de « sprints » qui constituent des unités temporelles (au maximum un mois) à l'intérieur desquelles des incréments livrables de l'application sont produits. Un sprint est composé d'une planification, d'une période de réalisation, d'une revue et d'une rétrospective. Le cycle de vie SCRUM, qui utilise divers artéfacts, est illustré par la figure 15.

On trouve en entrée d'un sprint la liste des items qui restent à développer dans le produit (Backlog) ainsi que la liste de ce qui a été achevé (Burndown).

Après la phase de planning, un sprint est lancé, rythmé quotidiennement par un état d'avancement et un suivi du reste à faire (Backlog Sprint) et de l'avancement (Burndown Sprint).

En sortie un produit livrable est présenté et une analyse du sprint est menée (Rétrospective).

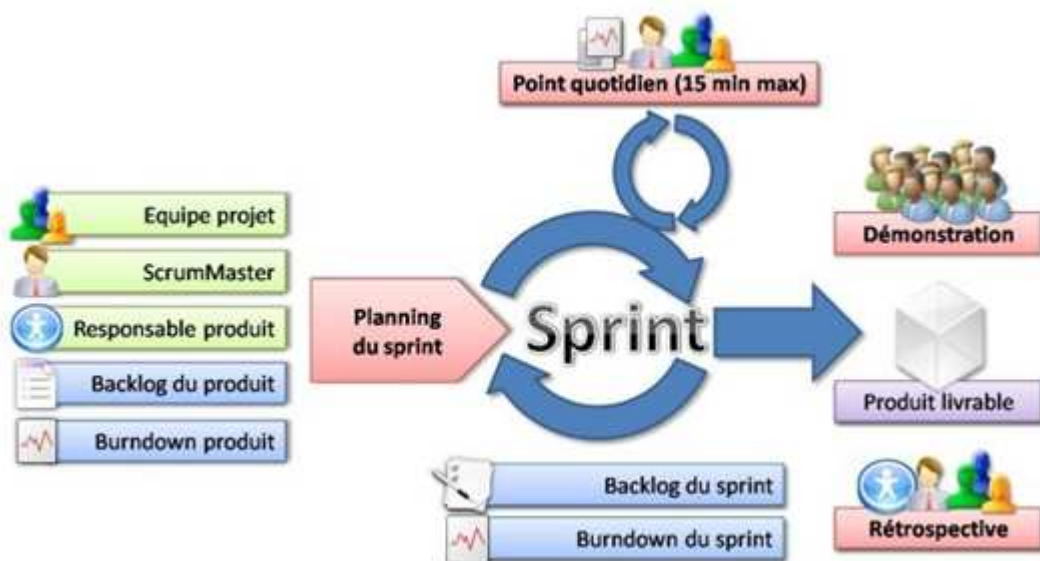


Figure 15 - Cycle de vie SCRUM (MCNext, 2010a)

XP

La méthode XP est la seconde méthode agile la plus utilisée, quand elle est associée à SCRUM (voir figure 14).

L'objectif d'XP est de réduire les risques relatifs à un développement logiciel. La méthode s'attaque à la non-qualité, aux retards ainsi qu'à l'incompréhension entre métier et développeurs.

La méthode XP repose sur 5 piliers :

1. La communication :

XP fait de la communication un élément prioritaire d'un projet informatique en partant du principe que l'absence de communication est un risque critique. La communication est encouragée entre développeurs, avec les managers et avec les clients.

2. La simplicité :

Partant du postulat qu'un code simple, minimaliste et propre est plus facile à maintenir et à faire évoluer, XP encourage à développer des fonctionnalités au plus près des besoins métiers et à n'intégrer que ce qui est réellement nécessaire.

3. Les retours :

La méthode prévoit des boucles de retours à plusieurs niveaux de manière à renvoyer de l'information selon un cycle le plus court possible. Pour les développeurs le cycle est de l'ordre de la minute avec les tests unitaires, pour les clients de quelques jours avec les tests fonctionnels. XP encourage donc à avoir des cycles d'itérations très brefs et une forte implication du client permettant d'avoir un retour rapide sur les actions menées côté développement. Pour le client cette organisation permet d'avoir en permanence une vision de l'adéquation de l'outil avec ses besoins.

4. Le courage :

Le courage est nécessaire dans la méthode XP pour pouvoir se remettre en question en permanence. Si cela s'avère nécessaire en fonction des retours, un développeur doit savoir repenser son architecture ou même tout reconstruire si la qualité ou la valeur n'est pas au rendez vous. XP donne également de la responsabilité à chacun, par exemple les développeurs sont chargés de se répartir les tâches, ils sont également encouragés à exprimer clairement leurs problèmes.

5. Le respect

Cette valeur a été rajoutée dans la deuxième édition du livre de Kent Beck « Extreme Programming Explained » (Agile software development, 2007).

Le respect recommandé dans XP peut prendre plusieurs formes. Il s'agit de respecter le client et son expertise, de respecter les membres de l'équipes en les considérant avec estime et en prenant en compte leurs contributions. Il s'agit également de respecter le projet en délivrant toujours la meilleure qualité, et en mettant la même application pour chaque action. Enfin le respect doit être établi entre le management et les collaborateurs de manière réciproque.

L'enjeu du respect est de permettre qu'aucun membre de l'équipe ne se sente dévalorisé ou laissé de côté. Cet aspect est essentiel afin d'atteindre le niveau de motivation et d'engagement nécessaire à la réussite d'XP.

Outre ces 5 piliers, la méthode XP est organisée autour de 12 pratiques :

1. Le « planning game » (jeu de planification) est la phase de planification d'une release. Dans cette phase, le client décrit ses besoins sous forme de « user stories » (histoires utilisateurs). L'équipe de développement attribue ensuite des points à chaque histoire en fonction de la complexité et de la charge de développement. Le client choisit donc sur cette base les histoires qui seront traitées dans la release.
2. XP suggère de produire des releases les plus petites possibles tout en livrant des produits complets et avec le maximum de valeur ajoutée.
3. XP recommande l'utilisation de métaphores pour décrire l'architecture du système et ainsi permettre un dialogue plus simple entre personnes de différentes cultures professionnelles.
4. Comme cela a été décrit dans les piliers de la méthode, XP prône une conception tournée vers la simplicité en ne développant que le strict nécessaire.
5. Les tests unitaires sont écrits par les développeurs et les tests fonctionnels par les clients. XP recommande d'écrire les tests avant le développement de l'application. Idéalement, XP prône l'automatisation des tests, de manière à pouvoir tester fréquemment des grandes parties de l'application et valider le fonctionnement général.
6. La révision du code ou « refactoring » vise à retravailler fréquemment des parties de codes pour améliorer leur lisibilité ou leur simplicité dans le but de faciliter des évolutions futures. Cette révision ne doit avoir aucun impact sur les fonctionnalités du produit et ne doit entraîner aucune régression.
7. XP préconise une programmation en binôme. Chaque paire est composée d'une personne devant le clavier : le « driver » (conducteur) qui implémente la fonctionnalité demandée et le « Partner » (partenaire) qui supervise le code en ayant une vision plus globale. Malgré les apparences, cette approche génère une bonne productivité car le code développé est livré plus vite et de meilleure qualité.
8. Selon XP, l'intégralité du code d'un produit doit être maîtrisée par l'ensemble des membres de l'équipe de manière à permettre à chacun de travailler sur n'importe quelle partie de l'application, même s'il ne l'a pas écrite.

9. Le code est intégré de manière continue, c'est-à-dire plusieurs fois par jour, et doit garantir à chaque instant la cohérence fonctionnelle de l'application.
10. En termes de charge de travail, l'objectif d'XP est de lisser au maximum les efforts pour éviter les périodes de surcharge que peuvent connaître les méthodes traditionnelles notamment dans les jours qui précèdent une livraison ou un jalon attendu. L'effet recherché est de limiter la baisse de productivité et de qualité observée lors de périodes de trop grande activité.
11. Une implication forte du client est souhaitée par la méthode qui va même jusqu'à préconiser la présence du client sur site pour favoriser les échanges et la communication.
12. Enfin, XP requiert l'utilisation de standards de code et de programmation pour faciliter la lecture du code par tous les membres de l'équipe

Cycle de vie XP :

Le cycle de vie d'XP (voir figure 16) démarre du côté client par la description des histoires utilisateurs et du côté du développement par une revue de l'architecture destinée à étudier les meilleures solutions techniques possibles. Par la suite les évolutions sont planifiées et les itérations de développement sont lancées avec une rétroaction permanente entre tous les niveaux (Planification, Itération, Recette) en fonction : de l'avancement, des bugs rencontrés ou des nouveaux besoins utilisateurs.

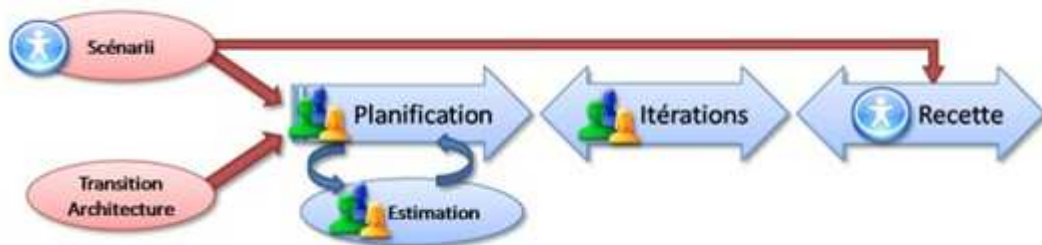


Figure 16 - Cycle de vie XP (MCNext, 2010b)

1.3.3. Autres méthodes

Il existe de nombreuses autres méthodes qui sont qualifiées d'agiles mais qui sont beaucoup moins utilisées que SCRUM et XP. Voici une liste non exhaustive des méthodes actuelles.

- **TDD : Test Driven Development**

Cette méthode est basée sur de très courts cycles de développements qui placent l'automatisation des tests comme principe fondateur. Le développeur commence par implémenter un test automatique vérifiant le résultat de la méthode qu'il doit coder. Ce test doit échouer à sa première exécution puisqu'il est créé avant même le code fonctionnel.

Le développeur implémente ensuite la méthode qui doit répondre au besoin du métier. Finalement il ré-exécute son test pour vérifier qu'il est cohérent. Il relance ensuite tous les tests existants pour vérifier que l'évolution n'a pas généré de régression. Le cycle se répète ensuite indéfiniment en fonction des besoins du métier.

Cette méthode confère beaucoup de confiance aux développeurs qui peuvent tester en permanence l'intégralité de l'application. Les retours des projets ayant utilisé cette méthode font état d'une productivité accrue grâce à des gains de fiabilité. De plus cette approche oblige le développeur à se questionner sur comment l'application va être utilisée pour pouvoir écrire son test. Il en résulte généralement une meilleure compréhension du besoin et un meilleur design.

- **RAD : Rapid application development**

La méthode RAD est probablement la plus ancienne des méthodes apparentée au monde Agile. Elle est découpée en 5 phases.

1. Une « initialisation » où le périmètre et l'organisation globale du projet est défini.
2. Une phase de cadrage où les besoins des clients sont recensés.
3. Une phase de design à laquelle participe activement les utilisateurs.
4. Une phase de construction basée sur des cycles itératifs et des prototypes validés par les utilisateurs de manière régulière.
5. Une phase de recette et de déploiement.

Le RAD se différencie des méthodes agiles car il n'apporte pas de notion de flexibilité par rapport au besoin des utilisateurs. Les spécifications étant figées dans la phase de design.

- **DSDM : Dynamic software development method**

Cette méthode a été développée en Angleterre en 1994 et a été utilisée par de grandes entreprises comme British Airways. Elle est fondée sur 9 principes :

1. Implication des utilisateurs.
2. Les équipes DSDM doivent être autorisées à prendre des décisions.
3. Le produit est rendu tangible aussi souvent que possible.
4. Le produit développé doit répondre impérativement aux besoins du métier.
5. Utilisation d'un développement itératif.
6. Toute modification doit être facilement réversible.
7. Un schéma directeur fixe les grandes lignes du projet.
8. Les tests sont intégrés au projet.
9. L'esprit de coopération entre tous les acteurs est primordial.

Le cycle de vie est quant à lui découpé en 5 phases : une étude de faisabilité, une étude du métier pour comprendre l'impact du projet sur les processus, une modélisation fonctionnelle qui peut engendrer aussi bien des modèles d'analyse que du code exécutable, une phase de conception et de construction itérative et enfin une phase de déploiement

- **Agile UP**

La méthode RUP (Rational Unified Process) propose déjà certains principes d'agilité comme l'approche itérative, la collaboration avec le client ou la tolérance aux changements. Par contre la méthode est très centrée sur la modélisation UML, ce qui lui donne une certaine lourdeur.

Agile UP est une version simplifiée de cette méthode qui s'attache moins à la modélisation que sa grande sœur (Ambysoft, 2006). Les disciplines d'études de besoin, d'analyse et de design et d'implémentations sont regroupées dans une phase unique d'implémentation qui, selon les principes agiles, s'emploie à livrer rapidement des modèles exécutables aux utilisateurs.

1.3.4. Analyse de l'agilité dans un contexte de maintenance

Selon l'intelligence collective, les méthodes agiles sont plutôt destinées aux petits projets de développement liés aux nouvelles technologies. La démarche d'utiliser de l'agilité sur un projet de maintenance d'un ERP pouvait donc paraître inappropriée ou risquée.

Pourtant les objectifs de l'agilité sont intéressants pour n'importe quels projets informatiques : développer plus vite, mieux comprendre les besoins clients, augmenter la tolérance au changement...

La mise en application de certains principes agiles semblait donc être un bon moyen d'innover pour améliorer l'efficacité d'une équipe dans le cadre d'un contrat de maintenance d'applications.

D'ailleurs il a été démontré que l'agilité pouvait également être applicable dans le cadre de la mise en place d'un ERP comme SAP (Bluefin solution, 2012), à condition d'adapter la méthode à l'environnement.

Dans ces conditions, voici parmi les principes du manifeste agile, ceux qui ont été retenus pour servir de base à la construction du centre de service BW.

Dans Agile, la priorité maximum est mise sur la **livraison rapide et fréquente de fonctionnalités à forte valeur ajoutée**. Ce principe a été retenu car il est applicable dans le cadre d'un projet de maintenance. En effet les demandes d'évolutions émanent directement de besoins métier exprimés par les utilisateurs eux-mêmes ou leurs représentants. Ces demandes sont généralement issues de leurs problématiques quotidiennes et peuvent présenter des retours sur investissement à très court terme.

Malheureusement, sur la notion de fréquence de livraison nous sommes ici contraints par les exigences du client, et par exemple dans le cas de Schneider-Electric qui demande des releases trimestrielles. La justification de cette exigence est de limiter les utilisateurs dans leurs demandes d'évolutions et de les contraindre à une réflexion approfondie sur les sujets qui doivent passer à chaque release. Les sujets non prioritaires devant attendre une release ultérieure ayant ainsi toutes les chances d'être abandonnés ou 'oubliés'. De l'autre côté les mises en production Rio-Tinto se font au fil de l'eau ce qui est plus dans l'esprit de l'agilité, mais pose d'autres problèmes comme la difficulté de planifier l'activité et de lisser la production sur l'année.

L'acceptation du changement dans le développement est une valeur difficile à appliquer. De manière générale, chez les développeurs, le changement est souvent perçu de manière négative: soit comme une remise en cause de leurs compétences, si le changement est perçu comme issu d'une mauvaise compréhension des besoins, soit comme une instabilité ou un manque de compétence du client. Le coût financier du changement est également un frein car la négociation sur ce point peut s'avérer source de conflits.

Cette valeur de l'agilité peut présenter plus de difficultés à être mise en place dans un contexte de maintenance que dans celui d'un projet de développement car les méthodes en place sont souvent anciennes et rigides et donc plus dures à changer.

Pourtant, même si cet aspect de l'agilité nécessite une vraie remise en question des acteurs tant côté client que côté prestataire, il peut être bénéfique pour les deux parties. En incluant une marge de changement dans chaque évolution on donne au client un droit d'erreur ou simplement le droit de changer. Une autre manière de limiter les impacts est de permettre au client de tester très tôt les évolutions, par exemple dans l'environnement de développement. Si celui-ci peut ainsi manipuler ou visualiser une partie de l'évolution et soumettre ses remarques en avance dans le cycle de vie de l'évolution, le coût d'un changement est alors plus faible et moins risqué que dans les phases de recette utilisateur.

En termes de communication le manifeste agile insiste fortement sur la **proximité entre le client et les développeurs**. Ce principe là s'applique également parfaitement dans le cadre d'une maintenance applicative. Même si l'externalisation n'offre pas la possibilité de communication en face à face, il est important de privilégier au minimum des communications téléphoniques permettant des échanges rapides contrairement à l'utilisation d'email qui retardent et peuvent brouiller la communication. Cette pratique est applicable à n'importe quel projet informatique, quelle que soit sa taille ou son contexte.

La **motivation des équipes et leur autonomie** est également un point applicable à beaucoup de projets informatiques. Toutefois, dans le cadre d'un projet de maintenance, l'architecture étant déjà en place, la liberté d'innover est plutôt réduite. Pourtant, l'idée d'impliquer l'équipe dans la réflexion sur la manière de travailler peut amener des réactions positives et aider à fédérer les collaborateurs autour du projet d'amélioration continue de la production et des résultats.

Le maintien d'un **rythme de développement soutenable** est également une pratique agile qui peut s'appliquer dans différents contextes. Particulièrement dans notre d'activité récurrente et cyclique, il est important que les charges soient bien réparties et que le niveau d'activité de chacun permette de délivrer un service constant et de manière durable.

L'attention portée à **la qualité de la conception** est également très importante pour une maintenance informatique. En effet, pour une application ayant un vécu l'ajout de couches successives peut entraîner des dysfonctionnements si celles-ci ne sont pas correctement étudiées. De plus, en assurant en parallèle la maintenance et le support de l'application, l'équipe doit anticiper les impacts futurs des modifications qui sont apportées, que ce soit sur le plan de la maintenabilité ou de la stabilité des composants.

Nous voyons donc que l'ensemble des préconisations de la philosophie Agile peuvent s'appliquer, à des degrés divers, dans le cadre de la maintenance d'un ERP. Si nous nous penchons sur l'évaluation des méthodes agiles comme SCRUM et XP qui imposent plus de formalisme, nous pouvons penser qu'il est plus difficile de les implémenter dans notre contexte. Pourtant certaines similitudes sont à noter :

D'une part, les piliers de la méthode SCRUM : transparence, adaptation et inspection font partie de l'état d'esprit agile et peuvent être appliqués sans retenue dans le cadre de notre service. En effet, cette philosophie est de nature à faciliter l'interaction entre un prestataire et son client ce qui engendre nécessairement des effets positifs pour toutes les parties.

D'autre part, l'organisation présente certaines similitudes puisque nous retrouvons un utilisateur clef qui possède sensiblement les mêmes attributions qu'un responsable produit et un SPOC côté prestataire qui fait figure de SCRUM Master.

La différence principale avec la méthode SCRUM réside dans la durée des sprints. Pour Schneider-Electric les releases sont trimestrielles et sont fixées en accord avec le client. De ce fait, le volume de sujets traités à chaque release est plus important qu'avec une fréquence mensuelle. Les échelles de temps se trouvent élargies en conséquence.

Les points d'avancement ne sont donc pas faits quotidiennement mais hebdomadairement, sauf incident ou événement nécessitant une revue exceptionnelle des difficultés. Dans ce cas un dispositif est mis en place afin de suivre les sujets à risque de manière à les remettre sous contrôle.

Si nous inspectons les 5 piliers de la méthode XP nous trouvons également des bonnes pratiques applicables à notre contexte. Communication, simplicité, feedback, courage et respect sont autant de principes de bon sens qu'il convient de rappeler et de mettre en œuvre.

Dans les 12 pratiques formelles d'XP, on constate que certaines sont réutilisables dans notre contexte.

Les **séances de planification** (Planning Game) sont une pratique déjà utilisée pour Schneider-Electric. Sur la base de demandes client, une évaluation est faite du coût et des impacts de chaque sujet. La planification de la release est ensuite revue avec le client en fonction des priorités et de la capacité de l'équipe.

Les **conventions de codage** sont également déjà utilisées pour le développement SAP. On constate même que cette pratique est une norme globale commune à tous les développeurs du langage ABAP même si le niveau d'application de la norme peut varier selon les projets et les personnes.

De même, **l'appropriation collective du code** et **l'intégration continue** sont des notions natives dans la maintenance d'un ERP.

L'utilisation d'un **rythme soutenable**, principe de base de l'agilité est très important pour un contrat de maintenance. En effet sur des contrats à long terme, il est important de garantir la durabilité du service.

La **conception simple** est une priorité dans le cadre d'un contrat de maintenance. Notamment pour les équipes ALM qui gèrent également le support de l'application, la lisibilité, la maintenabilité et la simplicité des évolutions mises en place sont des préoccupations majeures.

Enfin la standardisation des **tests** est une pratique importante pour n'importe quel projet informatique. Si l'automatisation des tests est délicate dans le cadre du service BW, l'utilisation d'HPQC est un élément de normalisation des tests qui permet de garantir la validation du système après une évolution.

Au contraire, les pratiques XP ci-dessous sont plus difficilement applicables à la maintenance d'une solution SAP.

L'utilisation de **métaphores** semble peu appropriée puisque les contacts se font uniquement entre personnes suffisamment expérimentées et compétentes sur l'application pour maîtriser l'architecture du système. L'utilisation de métaphores à ce stade de la vie d'une application serait mal perçue par les clients. Toutefois, ce principe pourrait être envisagé dans le cadre de nouveaux arrivants, métier comme techniques.

La **programmation en binôme** est une pratique intéressante mais qui est limitée dans le cadre de développement BW par la nature des développements écrits. En effet, il s'agit le plus souvent de petites routines utilisées pour la transformation des données. Ces routines n'excèdent en générale pas plus de 30 lignes de code. Une autre partie des activités est liée au paramétrage des flux de données. Ces deux tâches majeures du développement BW n'induisent que très peu de problèmes de robustesse ou de propreté du code mis en place. En revanche le risque et la difficulté sont davantage portés sur la conception des flux de données qui peuvent entraîner des incohérences ou des résultats non satisfaisants pour l'utilisateur. C'est pourquoi nous pouvons dire que dans le cadre de BW il faut privilégier la conception en binôme pour les sujets les plus importants.

Pour la même raison, la notion de **révision continue** du code (refactoring) est peu intéressante pour le service, même si rien n'interdit d'optimiser un code si cela présente une utilité.

Comme évoqué dans le cadre général de l'agilité, la cadence des releases est fixe pour Schneider-Electric et donc l'application de **livraisons fréquentes**, préconisées également par XP est contrainte.

Pour finir, la **présence du client sur le site** de développement n'est pas possible. En effet, les responsables clients ne passent qu'une partie de leur temps pour gérer l'application. Il est néanmoins important de garder à l'esprit cette pratique d'XP pour arriver à réduire, au moins virtuellement, la distance avec le client. L'utilisation du téléphone plutôt que l'e-mail, la mise en place de points de synchronisation réguliers ou d'ateliers spécifiques sont autant de moyens pour se rapprocher de l'objectif.

1.3.5. Synthèse de l'agilité et adaptation au centre de service BW

Le croisement entre la difficulté de mise en œuvre d'un point et l'intérêt qu'il suscite nous a servi à dresser la liste des directions à suivre et a guidé les réflexions quant aux solutions que nous allons mettre en place.

Le tableau II résume les principes de l'agilité et leur application au centre de service.

Tableau II - Synthèse de l'agilité dans contexte du centre de service

Applicabilité des principes agiles au centre de service BW	Difficulté	Intérêt
Livraison rapide et fréquente de fonctionnalités à forte valeur ajoutée	Normale mais fréquence de livraison contrainte	Fort
Acceptation du changement	Forte	Fort
Proximité avec le client	Normale	Très fort
Motivation et autonomie des équipes	Forte	Très fort
Rythme de développement soutenable	Normale	Très fort
Qualité de la conception	Forte	Très fort
SCRUM - Acteurs	Faible	Fort
SCRUM - Cycle de vie	Faible hormis la durée des cycles	Fort
SCRUM - Artéfacts	Faible	Moyen
Pratique XP - Séance de planification	Faible	Fort
Pratique XP - Conventions de codage	Faible	Fort
Pratique XP - Appropriation collective du code	Forte	Fort
Pratique XP - Intégration continue	Faible	Fort
Pratique XP - Rythme soutenable	Normale	Très fort
Pratique XP - Conception simple	Forte	Très fort
Pratique XP - Standardisation des tests	Normale	Fort
Pratique XP - Utilisation de métaphore	Faible	Faible
Pratique XP - Programmation en binôme	Normale	Moyen
Pratique XP - Remaniement continu	Forte	Moyen
Pratique XP - Livraisons fréquentes	Forte car fréquence contrainte	Fort
Pratique XP - Présence du client sur site	Forte	Moyen
Piliers XP - Communication	Normale	Très fort
Piliers XP - Simplicité	Normale	Très fort
Piliers XP - Retours	Normale	Très fort
Piliers XP - Courage	Normale	Très fort
Piliers XP - Respect	Normale	Très fort

1.4.Lean

Même si le terme Lean n'est apparu que dans les années 80 (Wikipedia, 2012b), la méthodologie date des années 50 et tire ses origines du système de production de Toyota développé par Taiichi Ohno. En France, de nombreuses entreprises se sont déjà converties, en premier lieu dans le secteur manufacturier et tout particulièrement l'industrie automobile comme PSA et Renault.

Aujourd'hui le Lean s'est étendu à de plus vastes domaines dont notamment le Lean IT qui s'adresse à la gestion des systèmes d'information.

La méthodologie Lean traite des manières d'apporter le plus de valeur possible aux clients en réduisant les gaspillages engendrés par une organisation inadéquate ou inefficace.

Le Lean partage les mêmes objectifs que les méthodes agiles : améliorer la qualité et la productivité du développement de logiciel (Hibbs et al, 2009). Cette partie définit la philosophie Lean ainsi que ses outils dans le contexte de l'informatique et la compare avec les méthodes agiles.

1.4.1. Définition du Lean

Le Lean est bâti sur 5 principes :

1. Définir quelle est la valeur pour le client.
2. Créer de la valeur pour le client en éliminant complexité et gaspillages.
3. Lier la création de valeur à la demande client.
4. Donner aux employés des moyens de gérer les processus et de s'engager dans les améliorations.
5. Avoir en permanence à l'esprit l'amélioration continue.

Appliqués aux services informatiques, ces cinq principes visent à augmenter l'efficacité opérationnelle des équipes en s'attaquant aux causes des pertes de productivité qui induisent des gaspillages.

Le Lean est traditionnellement associé à sept gaspillages (ou muda en Japonais) dont l'identification est issu du système de production de Toyota défini par Ohno :

1. **L'attente**, d'un matériel, d'une décision, de la fin d'une étape.
2. La **surproduction**, ou toute production non directement en lien avec un besoin du client.
3. La **non-qualité**, générant des actions de correction, non financées par le client.
4. Les **mouvements**, ou tout déplacement inutile de personnes ou de matériel
5. Les **processus inutiles**, ou tous les processus allant au-delà de la qualité requise par le client.
6. Le **transport** inutile des pièces entre les processus
7. Le **stock**, qui n'est pas directement indispensable à assurer la fluidité de la chaîne de production

Ces sept gaspillages trouvent leurs équivalents dans le cadre de développement informatiques :

1. **L'attente**

Un projet de développement implique des prises de décisions ou des demandes d'information de manière constante. Toute attente dans l'une ou l'autre de ces situations peut engendrer un ralentissement du développement. Pire, un développeur peut même être tenté de prendre une initiative en dehors de son champ de compétence. En cas d'erreur l'impact peut être découvert tardivement et générer un gaspillage important.

2. La **surproduction**

En termes informatiques, la surproduction correspond à la création de fonctionnalités n'apportant pas de réelle valeur au client et qui deviendra une fonctionnalité non utilisée. Et tant qu'elle n'est pas complètement abandonnée, cette fonction génèrera un coût de support et de maintenance qui vient s'ajouter au coût relatif à sa création.

3. La **non-qualité**

Une des parts importantes de la philosophie Lean est la gestion de la qualité dans le développement de logiciels. Le constat est qu'un défaut dans la chaîne de développement logiciel entraîne inévitablement des dépenses inutiles pour le corriger et que plus le défaut est détecté tard, plus il est coûteux à résoudre. Dans le Lean, lorsqu'un défaut est trouvé, la procédure consiste à identifier sa cause et à la traiter de manière à garantir qu'il ne se reproduira plus.

4. Les **mouvements**

Les mouvements représentent le passage d'une tâche à une autre et les interruptions au sein d'une tâche. Dans le cadre d'un projet de maintenance informatique, les interruptions sont fréquentes car la diversité des tâches à mener est grande et les priorités sont multiples. Les collaborateurs gèrent à la fois le développement des évolutions, le support utilisateur, le support technique et la surveillance des applications. Le passage d'une tâche à l'autre se fait au détriment de la productivité globale.

Sur de courtes interruptions, le collaborateur doit faire un effort intellectuel pour reconfigurer sa pensée sur un autre sujet et lorsqu'il peut reprendre son activité initiale, il doit faire cette démarche dans l'autre sens. Ceci entraîne un effort et un temps de concentration qui représentent un gaspillage.

Dans le cas d'interruptions plus longues sur un processus, une partie des connaissances non correctement capitalisés sont perdues et nécessitent un réapprentissage lors du redémarrage du processus. Là encore cela représente un effort en pure perte.

5. Les **processus inutiles**

Les processus inutiles représentent évidemment un gaspillage au sens propre. Il s'agit par exemple d'un document inutile ou de tâches manuelles et répétitives qui devraient être automatisées.

6. Le **transport**

Dans le développement informatique, le transport est associé à la transmission d'informations entre les différentes parties prenantes, ce qui génère une déperdition de la connaissance et donc un risque d'inadéquation du produit final aux besoins. Ce gaspillage est typique du cycle en cascade qui présente des démarcations très nettes entre chaque équipe et donc des sources d'incompréhensions.

7. Le **stock**

Le stock informatique correspond à un encours de demandes n'ayant pas été complètement résolues. Cela peut être une demande de support non traitée ou un développement non terminé. L'approche Lean conseille de réduire le niveau d'encours et de traiter les demandes jusqu'à leur terme avant de commencer des nouvelles demandes.

Comme l'illustre la figure 17, l'origine de ces gaspillages provient en partie de la variabilité des demandes et de la rigidité des solutions (Cappgemini, 2012).



Figure 17 - Origine des gaspillages dans la gestion des systèmes d'information

En effet, contrairement à une chaîne de production, les tâches à réaliser dans une équipe de maintenance informatique sont souvent de natures très différentes. Deux demandes de même type (une évolution par exemple) exigent souvent des efforts d'analyse spécifiques et des compétences différentes.

Tout l'enjeu du management est donc de gérer des demandes variables dans un environnement limité tout en apportant un cadre et une méthode réutilisable dans un maximum de cas.

L'importance de la motivation

80% du délai d'un processus n'est pas consacré à créer de la valeur (Cappgemini, 2012). C'est-à-dire que pour une demande traitée en 100 jours, 80 jours sont passés à attendre le passage à l'étape suivante.

Un des aspects fondamental du Lean est de réduire cette proportion de perte en améliorant de manière continue les processus de production. Or, d'après la méthode Lean, cette amélioration doit venir en mode « bottom-up », c'est-à-dire des équipes opérationnelles vers le management. Le manager n'intervenant plus comme l'unique porteur des consignes, surveillant leur bonne application mais agissant comme un soutien de l'équipe en animant la collaboration entre les individus.

Pour réussir dans cette voie, il faut que les équipes adhèrent à la méthode et soient partie prenante de leur métier. Les collaborateurs doivent donc trouver leur compte dans cette nouvelle culture.

Parmi les points positifs pour les équipes, on peut citer :

1. L'augmentation de l'intérêt du travail par la suppression de tâches inutiles et inintéressantes.
2. L'utilisation à leur juste valeur de collaborateurs souvent diplômés dans des cursus de management ou d'ingénieur, mais présents depuis trop peu de temps dans l'entreprise pour occuper des postes de manager.
3. L'amélioration de la qualité de la production et donc la réduction de la quantité de temps passé à corriger des erreurs.

1.4.2. Les pratiques du développement de logiciel Lean

Même si le Lean est avant tout une philosophie, certains outils et pratiques y ont été associés pour appliquer les principes au développement de logiciel. Mary et Tom Poppendieck ont publié en 2003 le premier ouvrage sur le sujet : *Lean Software Development : An Agile Toolkit for Software Development Managers* (Addison-Wesley Professional).

Dans cet ouvrage, Mary et Tom Poppendieck présentent 7 pratiques décomposées en 22 outils. Voici ces pratiques :

Elimination des gaspillages

Le principal objectif du Lean est d'éliminer tous les gaspillages, et la première pratique du développement Lean est de repérer et d'éliminer ces gaspillages. Certains outils peuvent être mis en place pour faire apparaître les gains possibles comme la cartographie du flux de valeur.

Le VSM (Visual Stream Mapping) vise à cartographier les processus de manière à identifier les temps d'attente ou les étapes génératrices de non-qualité. Cette base une fois créée permet d'analyser l'existant de manière critique et d'établir un plan d'action visant à améliorer les résultats.

Favoriser l'apprentissage

Afin de faire progresser la maturité d'un projet ou d'une équipe, il est nécessaire de retenir les leçons apprises au fil du temps. Si les individus acquièrent de l'expérience et retiennent plus ou moins ce qu'ils apprennent, le projet doit mettre en place les outils nécessaires à la capitalisation des connaissances qui rendent le savoir pérenne et indépendant des personnes. La connaissance et son partage sont des éléments centraux d'un projet et doivent faire l'objet d'une attention permanente à divers stades de l'élaboration du logiciel.

Pour être utile, l'information doit être partagée. Il faut donc organiser et faciliter la collaboration entre les différents membres de l'équipe, clients ou fournisseurs. Le Lean matérialise cet aspect par la notion de « feedback » (retours) qui doivent permettre à chacun d'avoir une idée précise de l'évolution du produit.

- L'utilisation de prototypes plutôt que d'analyses détaillées permet d'offrir un retour tangible au client de la compréhension de son besoin.
- Les tests, effectués fréquemment et très tôt dans le processus de réalisation, permettent au développeur d'avoir un retour sur son travail.

A ce titre, l'utilisation d'itérations est une très bonne option pour aider l'équipe à augmenter sa connaissance sur son environnement technique, fonctionnel ou relationnel grâce aux retours apportés par chaque livraison.

La conception avec option (set-based design) est un autre outil proposé par le Lean qui vise à analyser simultanément plusieurs solutions pour laisser la plus pertinente découler de l'expression des contraintes de chacun (Web builder zone, 2012).

Un exemple de cette approche est l'utilisation de l'outil de planification « Doodle ». L'initiateur d'un événement énumère toutes ses disponibilités pour l'organisation d'un événement. Chaque participant indique ensuite quelles dates sont compatibles avec ses contraintes. La date choisie est celle qui réunit le plus de votes favorables.

Cette démarche permet de faire émerger plus rapidement et plus sûrement la meilleure solution plutôt que dans un schéma où l'organisateur propose uniquement sa date préférée et où chaque participant partage tour à tour ses contraintes.

Retarder la décision

Dans l'approche Lean, on conseille d'attendre le dernier moment possible pour prendre une décision quant à un choix de conception ou d'architecture. Cette pratique se rapproche de la conception avec option puisqu'elle conduit à traiter en parallèle plusieurs options lorsque cela est acceptable et évite une limitation abusive des solutions potentielles. On définit pour cela le dernier moment responsable qui représente la limite à laquelle un choix doit être fait pour un composant donné.

Livrer aussi vite que possible

Tous les clients aiment avoir des livraisons rapides, et plus celle-ci est proche de l'expression du besoin, plus il y a de chances qu'elle apporte de la valeur au client. Un cycle court permet également de limiter les gaspillages dus au stock de demandes non traitées.

Un des aspects essentiel du Lean est de travailler en flux tendu. C'est-à-dire d'une part, de ne travailler que sur les demandes apportant de la valeur au client (c'est-à-dire celles expressément réclamées par lui) et d'autre part d'avoir un en-cours de travail limité permettant de traiter les requêtes au fil de l'eau.

Cette approche vise ainsi à adapter la capacité d'une équipe au niveau récurrent des demandes. Ainsi, en permettant aux collaborateurs d'avoir un volume d'activité moyen acceptable, on favorise la qualité du travail fourni ainsi que la possibilité pour les équipes de s'impliquer dans l'amélioration du service.

Pour gérer le niveau d'en-cours, la méthode Lean est basée sur un management visuel permettant de rapidement partager les priorités, les problèmes et l'avancement des sujets. Ceci se matérialise par une réunion quotidienne (ou « daily stand-up meeting ») où sont affichés sur un tableau les sujets en cours (représentés par des post-it), les actions ou les points de blocage. Ces réunions d'équipes, rapides, vivantes et centrées sur l'action, favorisent l'esprit d'équipe, la créativité et la collaboration entre les différents collaborateurs.

Dans le Lean, on considère comme une priorité la réduction des files d'attente. En effet, une file d'attente génère un stock de produit non fini et donc selon la pensée Lean, des gaspillages. De plus, une file d'attente ralentit le cycle de développement et va donc à l'encontre d'une livraison rapide de produit complet.

L'outil le plus célèbre pour ajuster les files d'attente à leur niveau minimal est le tableau Kanban (du japonais « Kan » : fiche et « Ban » : signal). Dans le cas d'une production soutenue et avec un flux de demandes constantes, la mise en place du Kanban peut s'avérer utile. Le principe de base est l'utilisation de fiches ou tout autre symbole qui indique qu'une pièce est épuisée. Cette fiche est créée par un atelier aval et remontée à l'atelier amont pour lui demander de produire les pièces manquantes. Les produits sont ainsi tirés par la sortie du produit du cycle de production, matérialisée par sa distribution au client. La demande remonte automatiquement la chaîne jusqu'aux premières étapes de fabrication.

Dans le développement de logiciels informatiques, la chaîne de production peut se matérialiser par le déroulement des étapes de conception, d'implémentation, de test puis de déploiement. La figure 18 montre l'exemple d'un tableau Kanban :

Release 2012-4 📊					
	1 - Analyse CDC	2 - Redaction EID	3 - Developpement	4 - Recette	5 - Déploiement
1	AMFR04707 MAP-CCO Nouveaux ratio	X	AMFR04651 SSRM Spend modification	Start AMFR04650 SSRM SOC in supplier MD	X
2			AMFR04649 SSRM Ultimate Linked	AMFR04115 ? MAP-CCO Transco	
3			AMFR04664 SSRM SOC status	Start AMFR04547 ASARIS FSD8	
4			AMFR04648 SSRM Street Character	Start AMFR04328 ASARIS FSD49	
5				Start AMFR04688 ASARIS P2DC	
6					

Figure 18 - Tableau Kanban de gestion des demandes

Ici le tableau est matérialisé par un fichier graphique, chaque étiquette est remplacée par une entrée dans une cellule avec une indication sur l'état d'avancement de la tâche dans la phase en question.

Dans un flux continu, la fin du déploiement d'un sujet (étape 5) entraîne la requête d'une nouvelle demande à déployer (c'est-à-dire une demande dont les tests doivent être complétés – étape 4). La fin du test d'un sujet entraîne, lui, la requête d'une nouvelle demande à tester donc d'une demande pour laquelle le développement est terminé (étape 3). Ce flux remonte jusqu'à la demande initiale du client (étape 1).

Idéalement, chaque étape comporte le même nombre d'unités de travail et permettent de traiter toutes les tâches en parallèle dans le même laps de temps. Une fois le travail terminé à une étape, chaque demande passe à l'étape suivante et le cycle est ainsi continu sans créer de file d'attente.

Malheureusement, cette vision utopique fonctionne rarement dans la réalité, chaque étape ayant une charge et un délai qui lui est propre, par exemple il faut souvent plus de temps pour tester un sujet que pour le déployer.

Le nombre de demandes dans l'une ou l'autre des phases dépend de la capacité de production de l'équipe, du nombre de ressources allouées à chaque étape et de la durée relative d'une étape pour un sujet donné. Il est donc nécessaire d'instaurer des limites de tâches en cours par étape pour assurer un flux continu. Si nous reprenons notre exemple, ceci signifie que le nombre de sujets en cours de test soit plus faible que le nombre de sujets en cours de déploiement.

Ce système demande une vigilance constante pour éviter l'encombrement du flux dans les phases amont et la raréfaction des sujets à traiter dans les phases aval. Dans ce cas, la reconfiguration des ressources allouées et des limites définies à chaque étape permet de redonner de la fluidité au système.

Donner le pouvoir aux équipes

La motivation des équipes est un élément crucial dans la réussite d'un projet et permet d'obtenir un avantage compétitif dans la durée (Poppendieck, 2003). Partant de ce postulat, le Lean implique les collaborateurs dans les décisions d'organisation et les engage en tant qu'acteurs prioritaires de l'amélioration continue.

Le Lean pose le postulat que les personnes qui produisent sont les mieux placées pour améliorer le processus de production en éliminant ses défauts et ses gaspillages. Cette démarche réintègre les collaborateurs dans la réflexion globale sur leur activité et les incite ainsi à être des acteurs de leur métier.

Construire la qualité intrinsèque

La qualité est un des deux objectifs majeurs du Lean car en éliminant les défauts on supprime une source importante de gaspillage. La qualité est présente à tous les niveaux de la philosophie Lean.

Le Lean est avant tout un système qui tend à développer la qualité « intrinsèque » d'un produit, sous différentes formes, la première étant la qualité perçue.

La qualité perçue est l'impression immédiate, rationnelle et subjective qu'un client a vis-à-vis d'un produit (Giordano, 2006). Viser cette qualité dans le cadre de la production de logiciel c'est donner au client la confiance dans l'outil et dans l'équipe de développement. Cette prise en compte de la perception du client est l'assurance de progresser vers un logiciel qui sera utile et donc qui apportera la valeur attendue.

La démarche Lean vise également la qualité dès la conception. Certaines bonnes pratiques sont données pour répondre à cet objectif et mettent principalement l'accent sur la communication dans les phases de design. La priorité est donnée à des communications simples découpées par petites unités homogènes plutôt que par de grands échanges complexes. Comme dans l'agilité, on privilégie une discussion en face-à-face plutôt que via des échanges de documentations, ce qui permet de faciliter une communication bidirectionnelle, signe d'une vraie interaction entre le prestataire et le client.

Enfin, la validation de la qualité par les tests est un élément important de la philosophie Lean. Selon l'idée originale de Taiichi Ohno, la qualité d'un produit doit être contrôlée le plus tôt possible et tout au long de sa chaîne de fabrication. Dans le développement de logiciel Lean, il s'agit de tester dès le début des phases de réalisation et, si possible, de manière automatisée. L'automatisation porte sur l'exécution du test aussi bien que sur la détection et le recensement des erreurs. En plus de l'aspect qualité, cette pratique permet également d'éliminer des gaspillages (déroulement du test, détection anticipée de problèmes) et de favoriser la connaissance en mettant en évidence de manière pratique ce que l'application doit faire ou ne doit pas faire. C'est également un bon moyen d'obtenir des retours d'une grande valeur à moindre coût. Une application disposant d'une suite de tests automatisés permet de réduire la durée d'une étude préalable à un changement. En effet, le développeur peut implémenter la modification souhaitée et valider rapidement en exécutant l'ensemble du jeu de tests. Les erreurs potentielles sont ainsi immédiatement identifiées. Cette notion est également présente dans le monde Agile avec une méthode comme TDD (Test Driven Development) qui reprend l'approche Test-First (test en premier) d'XP.

Avoir une vision d'ensemble du système

Lorsque l'on cherche à optimiser un processus complexe, on le découpe généralement en sous-processus plus compréhensibles. L'optimisation d'un composant peut amener à déplacer le problème vers un autre composant et ainsi bloquer l'amélioration du système dans son ensemble. C'est pourquoi il faut toujours garder une vision globale des processus pour garantir l'amélioration globale de la valeur du produit.

Pour ce faire, il existe la méthode des 5P qui sert à guider la résolution de problèmes. Son nom provient de l'utilisation consécutive de 5 « pourquoi ? » servant à retrouver la cause d'un dysfonctionnement. Le principe de la méthode est de trouver la cause première d'un phénomène, seul moyen de le comprendre complètement et de pouvoir le traiter. Dans un système, cela permet de résoudre un trouble de manière globale plutôt que de traiter uniquement des symptômes locaux.

1.4.3. Comparaison avec les méthodes agile

Il existe souvent une confusion entre le Lean et le monde Agile. Pour preuve, le classement donné dans la partie 1.3.2 figure 14 cite le Lean comme une méthode Agile.

Les deux approches Lean et Agile partagent une philosophie commune et des objectifs similaires comme celui de livrer le plus rapidement possible un logiciel apportant de la valeur au client. Elles accordent également une grande importance à la communication et au partage d'information entre les prestataires et les utilisateurs de l'outil informatique.

Les deux méthodes encouragent l'efficacité en développant des organisations adaptatives, fondées sur un apprentissage et une amélioration continue que l'on retrouve dans la pensée de Deming, c'est-à-dire en instaurant des cycles de progression qui permettent de standardiser les progrès constatés et de corriger les dérives engendrées.

Tant dans l'agilité que dans le Lean, la collaboration et l'esprit d'équipe sont mis en avant pour favoriser l'aspect humain du travail. La motivation et l'implication des collaborateurs est d'ailleurs un élément indissociable de ces philosophies.

Les principes de base de l'agilité et du Lean sont donc semblables, pourtant il existe quelques différences.

Le domaine d'intervention du Lean est plus vaste que celui de l'agilité. En effet, le Lean s'applique à l'entreprise dans son ensemble en la considérant comme un système composé de multiples sous-systèmes. De son côté, l'agilité s'applique au niveau d'un projet ou d'un système.

Enfin, l'approche agile regroupe plusieurs méthodes formelles (XP, SCRUM) alors que le Lean n'en propose pas et se matérialise uniquement par des outils ou bonnes pratiques à suivre.

1.4.4. Critiques du Lean

La démarche Lean suscite une profonde méfiance de la part des organisations syndicales (La riposte, 2007), ce qui rend son application compliquée en France. En effet, par le biais des CHSCT (Comité Hygiène Sécurité et Conditions de Travail), les élus du personnel tentent d'encadrer la démarche (01 Net, 2012). Cette méfiance se base notamment sur des recherches du centre d'études de l'emploi (Etablissement public administratif placé sous la tutelle du ministère chargé du travail et de l'emploi, et du ministère chargé de la recherche). Selon cette étude (Centre Etude Emploi, 2006) la démarche Lean est celle qui produit le plus grand stress chez les salariés, devant les organisations Tayloriennes et les organisations apprenantes. Ainsi les troubles sur la santé s'avèrent systématiquement plus importants dans les organisations en Lean production que dans les organisations apprenantes.

Même si les résultats de l'étude du CEE (Centre Etude Emploi) semblent pertinents, celle-ci porte avant tout sur le Lean Production qui est utilisé dans l'industrie. Au niveau des sociétés de services ce sont plutôt les démarches Lean Management, Lean IT ou Lean Software Development qui sont implémentées. Même si ces méthodes partagent la même philosophie que le Lean Production, on peut imaginer que les problématiques physiques ne concernent pas le secteur des services informatiques, comme par exemple, l'exposition à des risques toxiques, les problèmes de posture ou les effets de tâches répétitives sur l'appareil musculo-squelettique.

Pourtant même dans les services, de nombreuses organisations syndicales dénoncent systématiquement la mise en place du Lean (CGT Atos, 2011), (CGT Thales, 2008), craignant la mise en place de cadences de travail plus importantes, de suppression des temps de repos ou une mise en place de chronométrage sur l'activité du salarié. Toutes ces mesures induisant de fait une hausse du stress ressenti par des collaborateurs soumis à cette pression.

Or, ces problèmes ne relèvent pas du Lean Management mais d'une mauvaise application ou d'une mauvaise compréhension de celui-ci (Les échos, 2010). En effet, la méthode ne vise pas à augmenter les cadences ou à supprimer les temps de pause mais à limiter les gaspillages dus à des processus mal configurés voire totalement inutiles.

De plus la méthode du Lean Management est tournée vers l'humain et vise à enrichir la mission des collaborateurs en tirant parti de leurs compétences et de leur retour sur leur perception de l'organisation. Dans une vraie démarche Lean ce sont donc les salariés qui proposent une modification du processus et non pas la Direction qui impose une méthode.

L'augmentation du stress est un facteur tendant à réduire les bienfaits du Lean, il doit donc être évité pour construire une amélioration durable (Oryx conseil, 2011).

Le Lean n'est donc pas le « monstre » décrit par les organisations syndicales, mais il doit être mis en œuvre avec beaucoup de précautions. Il s'agit là d'une profonde modification des habitudes de management et celles-ci doivent être accompagnées tout autant que les modifications sur l'organisation des équipes.

Pour illustrer notre propos, nous pouvons faire l'analogie entre un régime alimentaire et la mise en place du Lean. Si le régime est fait uniquement de privations brutales, il va conduire à un rejet et à une reprise de poids inéluctable. En revanche, s'il y a une vraie modification des habitudes alimentaires, la perte de poids sera durable.

1.4.5. Adaptation du Lean au contexte du centre de service

La philosophie Lean a déjà été adaptée à de nombreux cas d'utilisation, partant de la gestion de la production, elle a été étendue à la gestion du développement de logiciels informatiques, aux processus administratifs, à l'organisation des entreprises, etc....

De plus, le Lean a une origine industrielle, où l'organisation est assez proche d'un modèle de centre de service par la recherche de répétabilité dans les processus et les techniques et son objectif de production en masse.

De part ces deux aspects -polymorphisme intrinsèque et culture industrielle- le Lean est une méthode qui se décline bien dans n'importe quel environnement et donc à fortiori dans un centre de service informatique (Effitic, 2012).

Toutes les pratiques Lean sont applicables dans notre contexte mais à des degrés divers. Voici l'évaluation de ces pratiques en regard des activités de maintenance autour de SAP BW.

L'élimination des gaspillages trouve un écho dans le centre de service.

- **L'attente** dans le processus de décision peut être minimisée par une meilleure planification et une meilleure anticipation des besoins ainsi que par une gestion flexible de la production de manière à reprogrammer le dispositif au gré des aléas.
- La **surproduction** est un gaspillage moins fréquent pour notre centre de service SAP, mais elle peut exister dans des cas où la conception est trop riche par rapport à la demande, intégrant par exemple d'hypothétiques évolutions futures. Cet aspect, pour être évité, demande une promotion continue des principes de simplicité et d'efficacité du design.
- Cette attention portée sur la conception influe également sur la traque de la non-qualité puisqu'elle vise à limiter les erreurs dès le début du traitement d'une évolution. Plusieurs autres principes sont applicables pour réduire la **non-qualité** comme le travail itératif, ou la proximité avec le client.

Dans l'environnement SAP, l'automatisation des tests, autre aspect de la gestion de la qualité, est très difficilement réalisable car les outils à mettre en place sont très coûteux. Le développement de tests automatisés est d'autant plus cher qu'il n'a pas été prévu à la mise en place de SAP. Il serait donc difficile d'obtenir un retour sur investissement de la mise en place de cette pratique. Pour autant, il est possible d'étudier une standardisation des tests permettant de générer un gain intéressant en capitalisant sur la définition et l'exécution des scénarios de tests. Dans le cadre du centre de service BW l'utilisation de l'outil HPQC permet de mettre en œuvre cette standardisation et d'améliorer trois aspects du service : qualité, connaissance et réduction des gaspillages (répétition des scénarii de tests, impact de la non-qualité).

Enfin, le dernier pan de la gestion de la qualité est apporté par les processus ITIL, comme notamment la gestion des problèmes.

- Les **mouvements**, ou le passage d'une tâche à l'autre est également une des sources majeure de gaspillage dans un centre de service intégrant développement, support utilisateur et surveillance des systèmes. En effet une personne travaillant sur l'analyse d'une évolution peut être dérangée de multiples manières : par une demande utilisateur, un incident sur l'environnement de production (problème système nécessitant une reprise de donnée ou une coordination avec le client), des tâches administratives (délais à respecter pour une saisie des temps). Cet aspect peut être amélioré par une meilleure organisation de l'équipe : gestion des plages de support, répartition des tâches transverses, gestion des priorités.
- Autre source importante de gaspillage : le **transport** équivalent à la transmission d'information d'une équipe à une autre. Cet élément est particulièrement vrai dans le cadre d'un centre de service utilisant des ressources offshore car, outre la différence naturelle entre équipes fonctionnelles et équipes techniques, apparaît une différence culturelle liée à la nationalité et à la langue des personnes. L'implémentation du model « One team » est un axe d'amélioration sur ce point, de même que la complicité recherchée avec le client pour partager les difficultés et les questions le plus tôt possible dans le cycle de développement.
- Certains **processus inutiles** sont présents dans le cadre du centre de service. Ils concernent principalement toutes les tâches de surveillance du système qui sont faites de manière manuelle et qui peuvent être automatisées. Cette source d'élimination de gaspillage est intéressante dans le contexte de la maintenance SAP car une solution développée pour un client ou une application particulière est facilement reproductible dans un autre environnement.

- Enfin la gestion du **stock** est peut-être la source de gaspillage la plus compliquée à traiter dans notre modèle. En effet, la durée des releases est contrainte et il est rare qu'un sujet dépasse le cadre d'une release. De ce fait, il est difficile de compléter plus rapidement les demandes d'évolution car celles-ci sont forcément bloquées au stade de la mise en recette. D'un point de vue support cependant ou pour les clients n'adhérant pas encore au cycle de release, comme Rio-Tinto, il est possible d'utiliser des méthodes comme le Kanban pour contribuer à réduire le niveau d'en-cours.

Autre pratique Lean, la **favorisation de l'apprentissage** est évidemment applicable dans le cadre du centre de service par la capitalisation des connaissances tacites ou des leçons apprises dans le cadre de l'activité.

Retarder la décision ou le design avec option est une bonne pratique facilement applicable. Sa portée est toutefois limitée dans le contexte SAP car la part d'architecture est souvent limitée et contrainte par les standards du logiciel.

Comme évoqué précédemment, l'idée de **livrer aussi vite que possible** est limitée par les rythmes de release Schneider-Electric mais elle peut être appliquée dans tous les autres cas.

La **responsabilisation des équipes** est un aspect lié à l'humain qui n'a que peu d'adhérence avec le contexte d'application de la méthode. De plus cet élément est nécessaire à la motivation des collaborateurs et donc à la réussite globale de l'adoption de la démarche.

Enfin, la dernière pratique étudiée est **la vision d'ensemble du système**. La mise en place de cartographie de processus, de cartographie applicative ou fonctionnelle sur la base des modèles utilisés en urbanisme est une bonne méthode pour avoir une meilleure vue d'ensemble des systèmes (Longépé, 2009).

Les critiques vis-à-vis du Lean, ainsi que les craintes des syndicats freinent l'adoption du Lean dans les organisations. Dans notre contexte, la démarche n'est donc pas impulsée par le management. De ce fait, les méthodes mises en place dans le cadre de ce mémoire se font au fil de l'eau dans une approche ascendante. Il est donc d'autant plus critique d'obtenir l'adhésion des collaborateurs sur les changements effectués.

De plus, certaines applications tirées des principes de développement de logiciel Lean sont très difficilement applicables dans le cadre de SAP. Par exemple, l'automatisation des tests génère souvent un investissement important qui est difficile à rentabiliser. Nous avons donc privilégié dans ce cas l'industrialisation et la capitalisation des tests qui peut se faire au fur et à mesure et pour lesquels le retour sur investissement est quasi immédiat.

Enfin, certaines pratiques Lean sont natives dans un environnement SAP comme la gestion du code source ou l'intégration continue.

2. Solutions proposées

Cette partie décrit le projet d'organisation de l'équipe et la mise en place de bonnes pratiques inspirées des philosophies Agile et Lean. Comme il a déjà été expliqué, il n'a pas été possible ni même souhaitable d'appliquer formellement l'une ou l'autre des méthodes car le contexte ne s'y prêtait pas et parce qu'il n'y avait pas de volonté managériale de le faire. Il a plutôt été décidé de choisir les pratiques les plus adéquates dans le catalogue disponible et de les adapter à notre utilisation.

2.1. Amélioration continue

En tant que responsable de la production de l'équipe BW, ma mission s'est portée sur la mise en place d'une organisation d'équipe capable de résoudre les problèmes posés et de s'inscrire dans un cycle d'amélioration continue.

2.1.1. Définition et planning

Hormis l'activité liée à Rio-Tinto, les autres services étaient déjà en place au début du mémoire, bien que dans une forme différente. Une fois la mutualisation actée, et afin de ne pas déstabiliser l'ensemble, nous avons choisi de fonctionner par perfectionnements progressifs selon des principes d'amélioration continue, similaires à ceux proposés par W.E. Deming dans sa méthode de gestion de la qualité PDCA (Plan Do Check Act). Cette méthode se découpe en 4 étapes :

- Plan : Planifier ce que l'on va faire.
- Do : Mettre en œuvre les améliorations imaginées.
- Check : Vérifier les résultats, à l'aide d'indicateurs.
- Act : Ajuster et corriger l'organisation.

A l'issue de chaque cycle, la capitalisation permet de faire grandir l'organisation en retenant les leçons apprises dans l'exercice précédent. La méthode est illustrée en figure 19 où le cycle est représenté par une roue qui avance continuellement sur une pente ascendante et qui est maintenue par une cale représentant la capitalisation.



Figure 19 - Cycle d'amélioration continue

Etant donné que Schneider-Electric est encore le client le plus important du service et que l'activité de maintenance évolutive qui lui est dédiée est organisée en releases trimestrielles, il était opportun de caler ce cycle PDCA sur le cycle de mise en production Schneider-Electric.

La figure 20 illustre l'évolution parallèle entre production et amélioration continue.

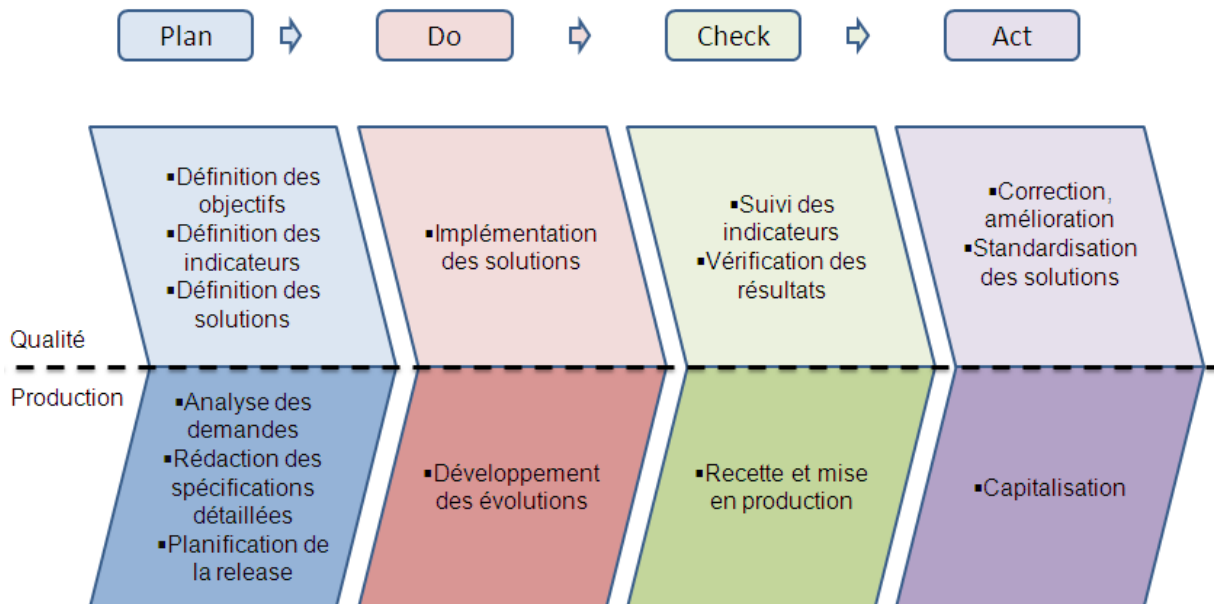


Figure 20 - Comparaison des aspects productifs et qualitatifs du cycle de développement

Le planning, illustré par la figure 21 a donc été construit de manière à aménager des périodes propices au questionnement sur l'organisation et les méthodes de travail (périodes bleues).

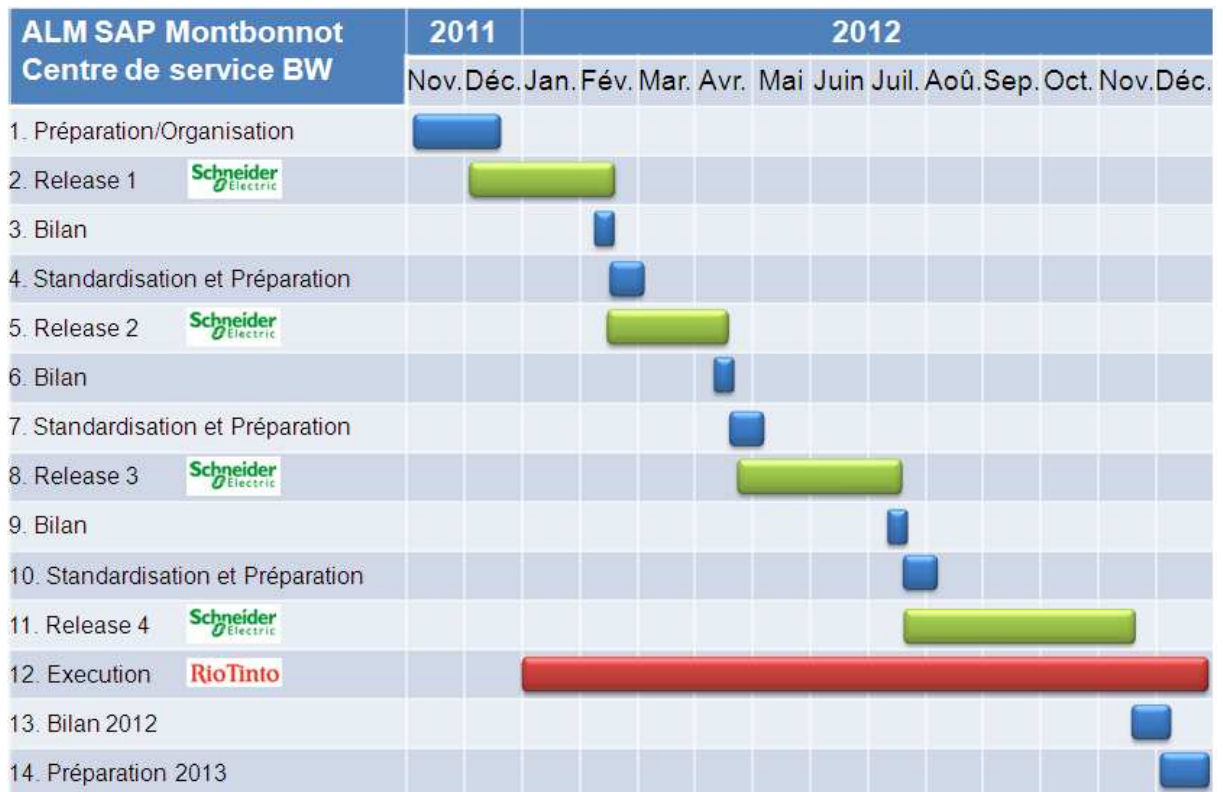


Figure 21 - Plan projet

La phase numéro 1 « Préparation/Organisation » visait à planifier l'année de manière globale et à définir les objectifs et les indicateurs associés.

Les phases 2, 5, 8 et 11 correspondent à la réalisation des développements pour Schneider-Electric (Release).

Les phases 3, 6, 9 et 13 sont les phases de bilan qui nous ont permis d'avoir un regard rétrospectif sur la période écoulée et d'identifier à la fois les réussites et les échecs de cette période. Le regard porté était à la fois subjectif pour essayer de capter le ressenti des collaborateurs (démarche Lean) et à la fois objectif en s'appuyant sur les résultats chiffrés.

Les phases 4, 7, 10 et 14 servaient à tirer des enseignements des bilans effectués et à ajuster les processus (capitalisation).

La phase 12 correspond à l'exécution du contrat Rio-Tinto. Celui-ci n'ayant pas de rythme de release il n'y avait pas de raisons de le découper artificiellement. Pour autant, les phases de bilan et de planification incluait bien les retours sur le déroulement des opérations liées à ce client.

Enfin la phase 14 qui n'est pas terminée à la fin de ce mémoire vise à préparer l'année 2013 sur la base du bilan de 2012.

2.1.2. Objectifs

La première étape du projet a été de formaliser les objectifs que nous allions vouloir remplir dans le cadre de l'organisation du centre de service BW. Ces objectifs ont permis de donner du sens aux actions à mener et ont servis de base à la construction des indicateurs permettant de mesurer les progrès accomplis.

Conformément aux attentes du management d'ALM, les objectifs définis ont été les suivants :

Production du service attendu par le client

Outre la fiabilité de l'application, la tâche essentielle de l'équipe a été de produire des évolutions amenant de la valeur à ses clients. Cette production est suivie par des budgets alloués en début d'année qu'il a fallu respecter tout en garantissant la tenue des jalons et la qualité nécessaire.

Baisse du support

Un des objectifs communs à toutes les équipes d'ALM a été de contribuer à la réduction des tâches de support et de maintenance corrective. Cet objectif est une des clefs pour permettre de proposer au client des tarifs dégressifs au fil du temps.

Augmentation de l'utilisation des ressources offshore

L'utilisation de ressources offshore est un des éléments importants des contrats de gestion de parc applicatif. Cet aspect permet d'une part de réduire les coûts et d'autre part de disposer d'une réserve de ressources extrêmement importante. L'objectif de cette année a été d'accroître la part de développement et de support faite en Inde.

Mutualisation des connaissances

Afin de garantir la capacité de l'équipe à intégrer de nouveaux contrats, il a été nécessaire de donner comme objectif global, une montée en compétences de chaque membre sur l'ensemble des domaines et de cultiver la mutualisation des connaissances.

Amélioration de la productivité de l'équipe

L'équipe a dû veiller à ce que les évolutions développées respectent les estimations vendues au client pour garantir les intérêts de Capgemini dans la relation.

2.1.3. Indicateurs

Afin de mesurer l'évolution des résultats relatifs aux objectifs fixés, il a été nécessaire de mettre en place des indicateurs de suivi.

Ces indicateurs ont été déterminés en fonction des objectifs mentionnés dans la partie 2.1.2. Le tableau III présente la correspondance entre ces objectifs et les indicateurs utilisés.

Tableau III - Liste des indicateurs

Objectif	Indicateurs
Production du service	Gabarit de release
	Nombre de retours sur recette
	Notes de comités
Amélioration de la productivité de l'équipe	Ratio entre vendu et consommé par évolution
Baisse du support	Nombre de jours de support imputés
	Nombre de tickets de correctifs
Mutualisation des connaissances	Nombres de fiches FAQ et A4 créées

Gabarit de release

En début d'année, les budgets prévisionnels ont été fournis par application. Cette information, couplée à la programmation des releases sur l'année, a permis de définir une taille pour chacune d'entre elles correspondant à un nombre de jours à produire. Le suivi de cet indicateur a servi à mesurer quantitativement l'état d'avancement du budget.

Par exemple pour l'application SSRM, le budget de 160 jours a été découpé comme suit :

- Release 1 (Février 2012) : 30 jours
- Release 2 (Avril 2012) : 30 jours
- Release 3 (Juillet 2012) : 50 jours
- Release 4 (Novembre 2012) : 50 jours

Ce découpage s'est fait selon la règle suivante : environ 20 jours de productions pour les mois pleins, et 10 jours pour les mois de release, ce qui donne :

- 20 jours en janvier et 10 jours en février pour la release 1
- 20 jours en mars et 10 jours en avril pour la release 2
- 20 jours en mai, 20 jours en juin et 10 jours en juillet pour la release 3
- 20 jours en septembre, 20 jours en octobre et 10 jours en novembre pour la release 4

Le mois d'août n'a pas été considéré comme productif à cause des congés d'été et le mois de décembre est réservé pour la préparation de l'année suivante.

La figure 22 est une illustration réelle du suivi de cet indicateur tel qu'il a été partagé avec le client. La courbe verte représente la production mensuelle nominale. Les colonnes bordeaux représentent les jours réellement livrés, c'est-à-dire mis en production alors que les colonnes bleues représentent les évolutions commandées. La courbe bleue intègre les prévisions de commande.

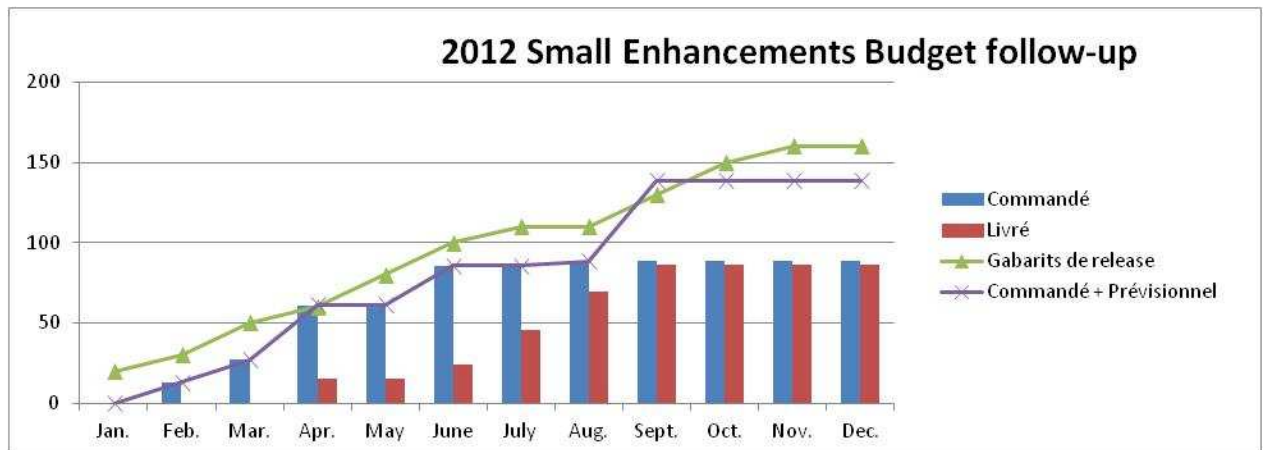


Figure 22 - Evolution du budget SSRM

Cet indicateur est suivi après chaque release mais il fait également partie des indicateurs suivis de façon mensuelle en comité de maintenance avec le client.

Notes de comités

Pour chaque application, un comité mensuel est organisé avec le client afin d'avoir un échange sur les opérations du mois passé. A l'issue de ce comité de maintenance, le client donne son évaluation sous la forme d'une couleur (Vert = Satisfaisant, Orange = Vigilance, Rouge = Problème avéré). Trois évaluations sont données, une sur la production, une sur le déroulement du comité lui-même (qualité de la présentation, respect des normes, collaboration), enfin la dernière évaluation est attribuée automatiquement en fonction du respect ou non du SLA.

Dans le cadre de ce projet, seule la notation de la production a été intégrée dans les indicateurs suivis. En effet, le déroulement des comités n'est pas lié à l'organisation de l'équipe et les niveaux de services dépendent de trop de facteurs pour être gérés uniquement au niveau du service.

Ratio entre vendu et consommé

Le suivi de ce ratio a été mis en place pour identifier les sujets en dérive, au cas où l'information n'a pas déjà été remontée par le biais des réunions d'équipe.

Nombre de jours de support imputés

Cet indicateur a été créé pour suivre les effets des actions visant à limiter le temps passé sur les problématiques de support.

Nombres de retours sur recette

Cet indicateur a été mis en place pour mesurer l'aspect qualitatif du développement et a été suivi dans l'outil HPQC présenté en partie 1.2.2. Seuls les retours ne découlant pas d'un changement d'avis du client ont été enregistrés.

Nombre de tickets de correctifs

Cet indicateur a été utilisé pour suivre la non-qualité générée suite à une évolution. Il s'apparente à l'indicateur sur le nombre de retours en recette, mais cette fois il concerne les évolutions montées en production pour lesquelles un dysfonctionnement ou un impact négatif sur un autre composant n'aurait pas été détecté en test.

Nombre de fiches FAQ et A4 créées

Les fiches FAQ et A4 sont des documents visant à capitaliser la connaissance acquise sur les différentes applications. Ces fiches sont décrites plus en détail dans le chapitre 2.5 traitant de la gestion de la connaissance. Le suivi de cet indicateur a permis de vérifier et de promouvoir l'implication de l'équipe dans l'effort de capitalisation.

2.2.Organisation

Comme mentionné dans la partie 2.1.1, nous avons mis en place une organisation mutualisée au début du projet pour gérer les applications BW au sein du service « Finance/BW/AM mutualisées ».

La figure 23 illustre la nouvelle organisation :

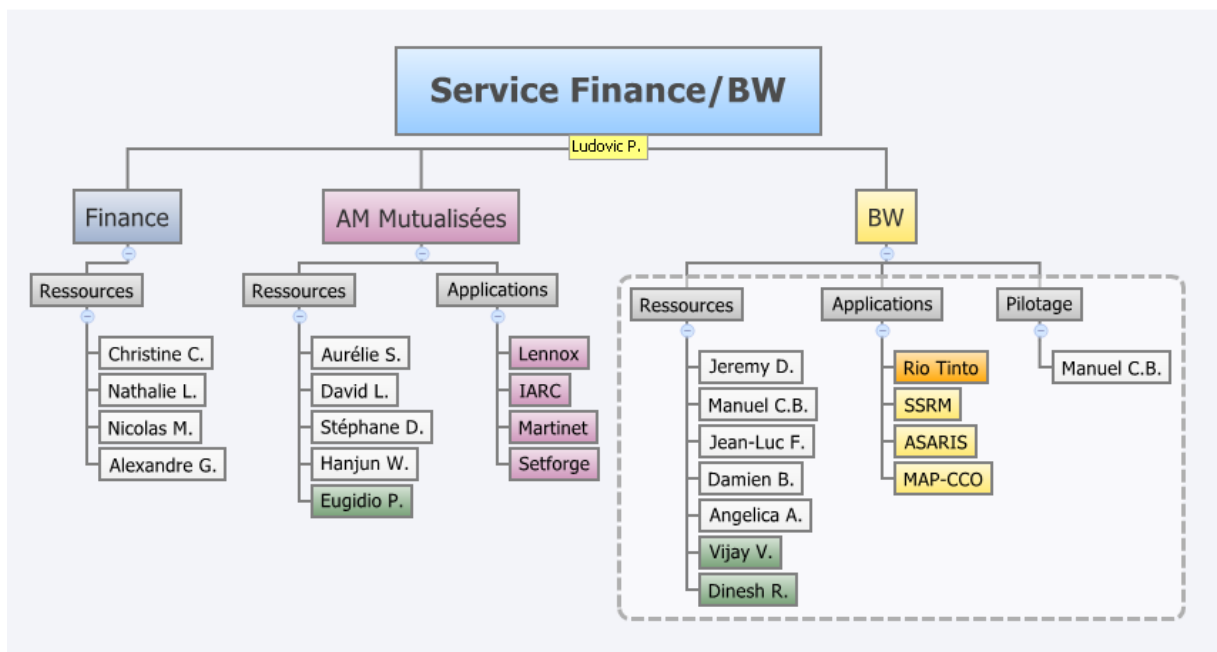


Figure 23 - Nouveau schéma d'organisation

Comme nous pouvons le voir sur cet organigramme, nous n'avons pas créé de niveau hiérarchique supplémentaire, le service étant toujours dirigé uniquement par le SDM.

Néanmoins, afin de mettre en œuvre l'organisation mutualisée évoquée, nous avons défini en collaboration avec le SDM le rôle de pilote du service BW. Ce rôle a été voulu comme un trait d'union entre le client, l'équipe de développement et l'organisation avec, par délégation du SDM, la responsabilité de coordonner les trois.

Pour le client, le pilote est le point d'entrée pour la gestion des demandes, et il est responsable de la prise en compte de ses besoins et de leurs planifications dans le cadre des releases. Pour ce faire, le pilote doit avoir une vision claire des évolutions et être capable de comprendre comment ces changements vont apporter de la valeur au client. Cela passe par une communication fréquente et efficace qui permet à l'information de naviguer de manière transparente vers le client ou vers l'équipe. Dans ce cadre, nous avons mis en place des instances hebdomadaires avec chaque responsable d'application client pour suivre l'avancement des demandes, proposer et valider un planning de réalisation et échanger sur le contenu des sujets en apportant des retours sur notre compréhension des besoins.

Pour l'équipe de production, le pilote a été présenté comme une aide, servant à prioriser les demandes, clarifier les objectifs, et réduire les obstacles qui nuisent à l'avancement. L'effet recherché a été de donner aux consultants les moyens de se concentrer uniquement sur leur métier et d'utiliser au maximum leurs compétences.

Dans ce cadre, la tâche de faire progresser les collaborateurs dans leur vision du besoin client et de leur faire adopter des bonnes pratiques de développement a également été imputée au pilote. Sa mission vis-à-vis des collaborateurs est de les aider à avoir une attitude communicante et transparente. Pour cela, il est décisif de laisser la place aux erreurs, pour encourager les collaborateurs à les partager rapidement et ainsi permettre de les corriger aussitôt (Stien, 2011).

Enfin, pour l'organisation, le rôle a été défini pour apporter les changements capables d'accroître l'efficacité de l'équipe et donc d'atteindre les objectifs fixés tout en standardisant les pratiques ayant portées leurs fruits.

2.3. Management

Cette partie présente le modèle de pilotage opérationnel qui a été mis en place pour répondre aux objectifs fixés.

Le premier objectif de l'équipe est de fournir un service au client en faisant évoluer ses applications tout en préservant les intérêts de Capgemini. Il s'agit là de notre cœur de métier et il est donc primordial de réussir cette mission.

2.3.1. Management visuel

Comme nous l'avons expliqué plus tôt, la mutualisation a engendré de nouveaux risques qui pouvaient venir perturber la bonne marche de la production. Il était donc important de mettre en place des moyens de réduire ces risques (perte de repères pour les membres de l'équipe, activité éclatée sur différents clients et dossiers, gestion des priorités plus complexe)

Afin de réduire l'impact et la survenance de ces risques, nous avons mis en place un management visuel des priorités via une réunion qui reprend le principe de la philosophie Lean du « daily stand-up meeting » (réunion quotidienne debout). Cette réunion permet également de partager la liste des tâches et de remonter les problèmes via un affichage des tâches, généralement matérialisé par des post-it.

Par rapport aux principes du Lean, nous avons fait trois adaptations :

1. La réunion a été tenue de manière hebdomadaire et non pas quotidienne. En effet, compte tenu des rythmes de release plus long et de la nature des tâches à effectuer, il ne nous a pas paru utile de multiplier les points de synchronisation. Nous avons quand même laissé ouvert la possibilité d'adapter la fréquence en cas de crise ou de revoir les tâches et les priorités de manière individuelle.
2. Les tâches n'ont pas été affichées sur un tableau à l'aide de post-it mais via un outil de carte heuristique nommé « Xmind ». La première raison à cette adaptation fut d'abord pratique : en effet, l'organisation des bureaux en plateaux ouverts n'a pas permis de disposer de l'affichage nécessaire. La seconde raison était liée à l'aspect international de l'équipe et au besoin de partager l'information avec les collaborateurs indiens sur un support dématérialisé.
3. Enfin, les réunions se sont déroulées assis, dans une salle. Toujours à cause du caractère ouvert des espaces de travail, il n'était pas possible d'installer une réunion au milieu des autres équipes sans les déranger. De plus, l'utilisation d'un fichier informatisé nécessitait de projeter sur un écran et cet équipement n'est disponible qu'en salle de réunion.

Malgré ces trois changements par rapport au Lean, la philosophie de la réunion a été préservée en essayant de respecter une durée maximale de 30 minutes. Dans un intervalle de temps aussi réduit, il est obligatoire de se consacrer uniquement à l'essentiel, c'est-à-dire au suivi de l'avancement et à la remontée des problèmes.

La figure 24 présente l'exemple d'un fichier hebdomadaire simplifié (toute l'équipe n'est pas représentée ici).

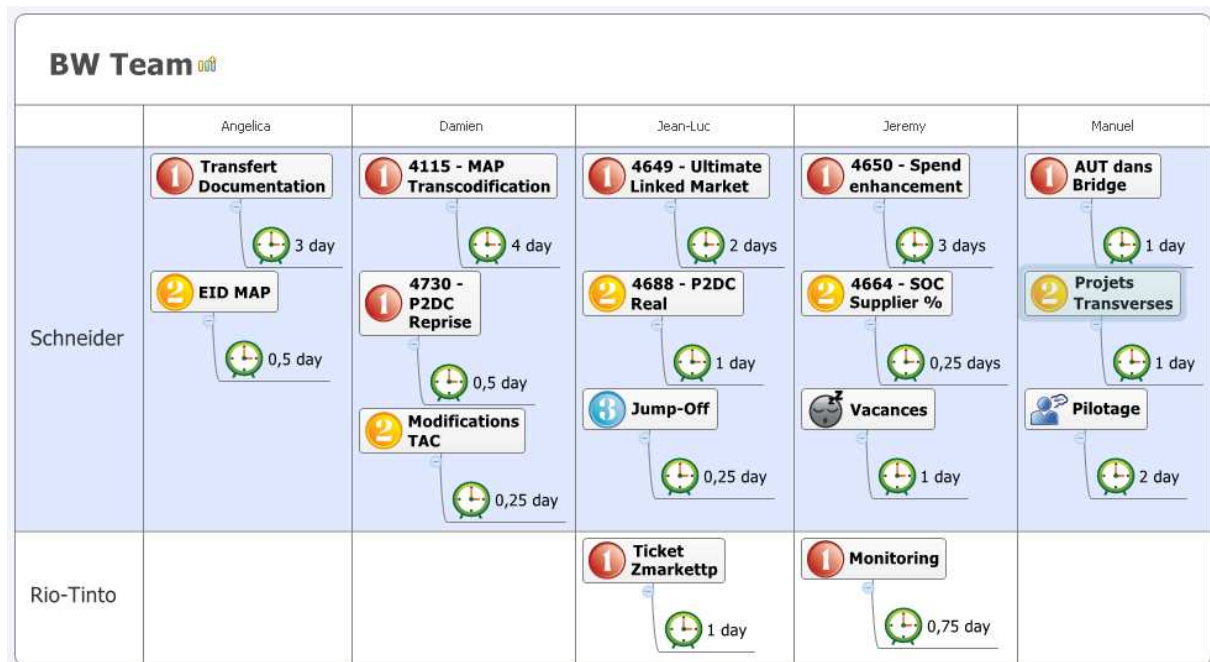


Figure 24 - Tableau visuel de gestion du plan de charge

Les collaborateurs sont présentés en colonnes et les clients en ligne. A l'intérieur de chaque cellule, les tâches sont identifiées par une étiquette à laquelle est associé un marqueur de priorité.

Ensuite une notion de charge est associée à chaque tâche. Seules les tâches de production sont listées, toutes les tâches récurrentes, comme la saisie d'activité, ne le sont pas. Pour cette raison, nous avons appliqué le principe qu'une charge hebdomadaire totale supérieure à 4,5 jours pour un collaborateur à temps plein présente un risque qu'il faut traiter soit par la réassignation d'un sujet, soit par une renégociation des priorités avec le client.

La notion de charge estimée pour la semaine a constitué le premier élément de remontée d'information et a permis d'anticiper les problèmes à venir.

Nous sommes repartis chaque semaine du fichier de la semaine précédente de manière à avoir un retour rapide sur la semaine écoulée :

- Le collaborateur a-t-il pu produire la charge estimée sur chaque tâche ?
- Sinon, quels ont été les événements qui l'ont empêché de le faire ?
- Le collaborateur a-t-il besoin d'une aide (managériale, ressources supplémentaires, communication client) pour résoudre le problème rencontré ?
- Ce décalage génère-t-il un risque de ne pas respecter les engagements pris sur la tâche en question ? Impacte-t-il d'autres tâches ?

Ce mode de fonctionnement a permis de s'assurer que l'équipe était focalisée sur les tâches à accomplir et qu'elle partageait les priorités. Il a également permis de vérifier que l'équipe disposait des moyens d'atteindre les objectifs fixés pour la semaine.

Enfin, cela a contribué à dynamiser l'équipe et à instaurer un partage régulier et ouvert. Nous avons constaté qu'il était fréquent que les collaborateurs ayant une sous-charge pour la semaine à venir proposent d'eux-mêmes leur aide aux collaborateurs les plus chargés.

D'un point de vue humain, ce mode de fonctionnement a renforcé l'écoute des collaborateurs, qui de fait se sont sentis plus soutenus dans leur quotidien. Même si tous les problèmes n'ont pas pu être réglés par cette méthode, le fait de les partager a réduit leur poids sur les personnes.

2.3.2. Carte de l'amélioration continue

Afin d'enregistrer et d'élaborer toutes les actions d'améliorations possibles, nous avons mis en place une carte d'amélioration qui sert à recenser les idées de perfectionnement du service ou les difficultés remontées par les collaborateurs (figure 25).

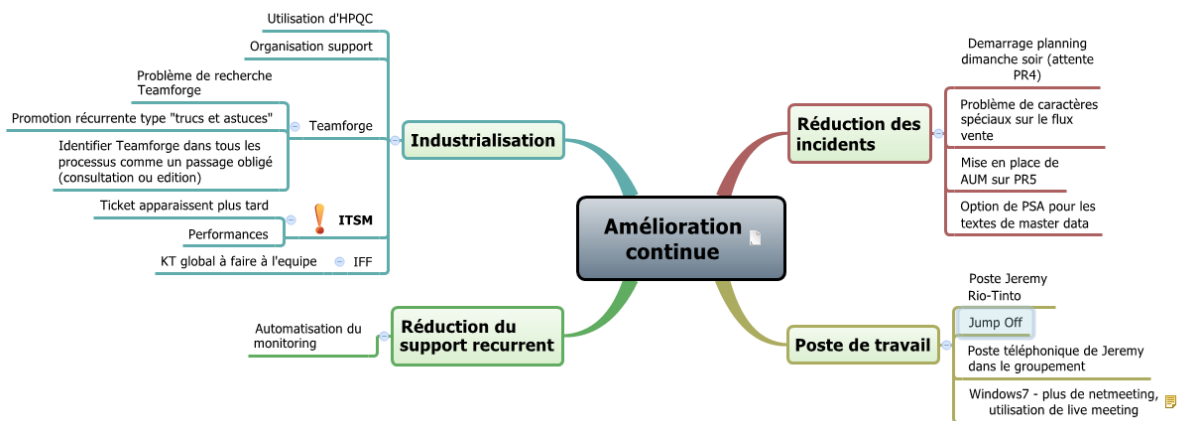


Figure 25 - Carte de l'amélioration continue

Les problèmes sont collectés aussi souvent que nécessaire et à minima en réunion hebdomadaire ce qui permet d'enregistrer toutes les idées possibles pour faire baisser les coûts de support, faire progresser la productivité mais aussi améliorer le confort des collaborateurs.

Les sujets sont traités ou affectés en fonction des priorités, de l'urgence et de la disponibilité de chacun et une revue est effectuée de manière trimestrielle pour revenir sur la période écoulée, mesurer les progrès effectués et standardiser ce qui peut l'être (nouvelles consignes et création de la fiche A4 associée, actions de transfert de connaissance).

Les résultats de cette revue comparés aux objectifs globaux permettent de définir la feuille de route pour le prochain trimestre. Toujours selon les principes du Lean qui met en avant le management visuel, ces objectifs sont déclinés sous forme de carte permettant à chaque collaborateur d'avoir en permanence une vision claire des objectifs à long terme du service.

La figure 26 illustre une carte d'objectifs pour le quatrième trimestre 2012.

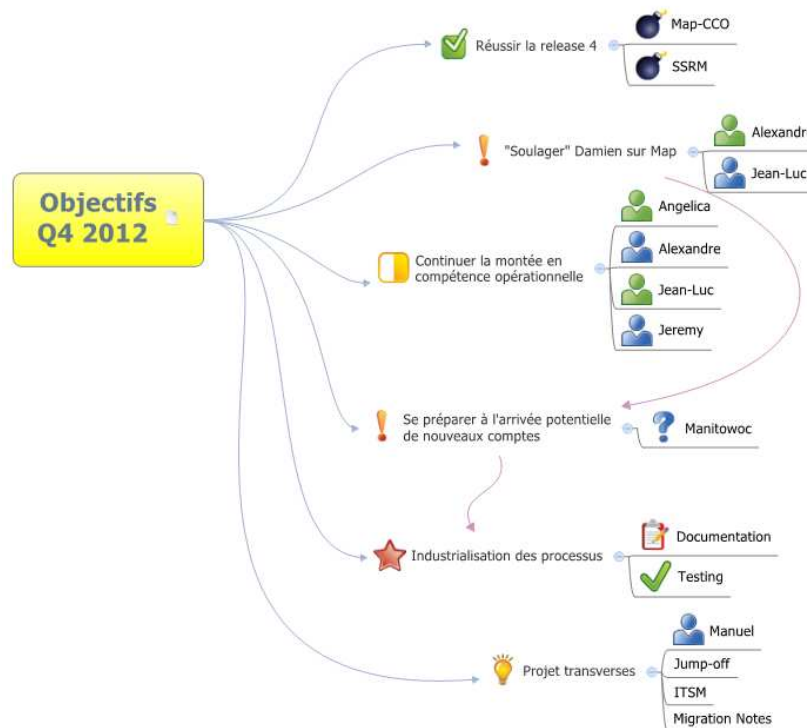


Figure 26 - Tableau des objectifs 2012 (Quatrième trimestre)

2.4. Cartographie des processus

Une étape importante de l'industrialisation a été la cartographie des processus. En effet cette étape nous a permis de remettre à plat les différentes manières de travailler et de les uniformiser. En effet, même si dans les différents services BW, un socle commun était présent et respecté, il existait des différences portant notamment sur les phases d'analyse et sur les tests.

L'uniformisation des processus ainsi que leur documentation a permis de standardiser la façon de traiter une requête mais également d'intégrer plus facilement une nouvelle ressource sur le domaine.

De plus, la cartographie des processus est une première étape vers la définition d'un document de type VSM tel que décrit dans la partie 1.4.2 sur les outils du Lean Management.

A ce jour, deux processus ont été documentés : il s'agit de la gestion des demandes d'évolution et de la gestion des demandes de support. Ils ont été construits pour être suffisamment génériques pour pouvoir s'appliquer à tous les clients, quel que soit l'outillage dont ils disposent. S'il est fréquent que chaque société possède son propre outil de gestion des demandes, la philosophie est sensiblement toujours la même.

2.4.1. Gestion des demandes d'évolution

Les travaux de formalisation du processus de gestion des demandes d'évolution nous ont conduits à nous questionner sur les principales sources de gaspillage présentes dans ce processus.

Le principal problème identifié a été la non-qualité issue d'une mauvaise expression du besoin ou d'une mauvaise compréhension de celui-ci et les allers-retours induits en phase de recette.

Pour limiter les risques de retour, il a été décidé d'ajouter un point de synchronisation, avant la phase d'analyse, permettant de s'assurer de la bonne compréhension de la requête. Cette étape débouche sur la création de deux livrables :

1. la production d'un document de synthèse très court (une page A4) présentant la reformulation du besoin par le consultant fonctionnel Capgemini en charge du dossier.
2. une fiche de tests décrivant les principaux cas (2 ou 3 pas plus) ainsi que leurs résultats attendus.

Le premier livrable est tiré du pilier de la méthode XP concernant les retours. En effet, la restitution au client de la compréhension du besoin par le consultant fonctionnel a permis de créer une boucle de retour supplémentaire, limitant ainsi les confusions.

Le second livrable reprend l'idée des histoires utilisateurs de la méthode XP en les transformant sous formes de scénario de tests. L'avantage de ce livrable est double : renforcer la prise en charge des tests et créer une autre forme de retours et d'échanges avec le client.

De plus, nous avons matérialisé dans ce processus la collaboration précoce avec le consultant technique. Cette approche est valable dans le cas de l'utilisation d'un développeur offshore (modèle One Team) mais également dans le cas de l'utilisation d'un développeur local. Les philosophies Agile et Lean préconisent toutes les deux d'avoir des équipes communicantes à toutes les phases du développement et d'éviter le phénomène de passage de relais des cycles en cascade.

La figure 27 illustre le processus de gestion de la demande d'évolution et met en évidence la collaboration entre les différents acteurs à divers stades de la demande.

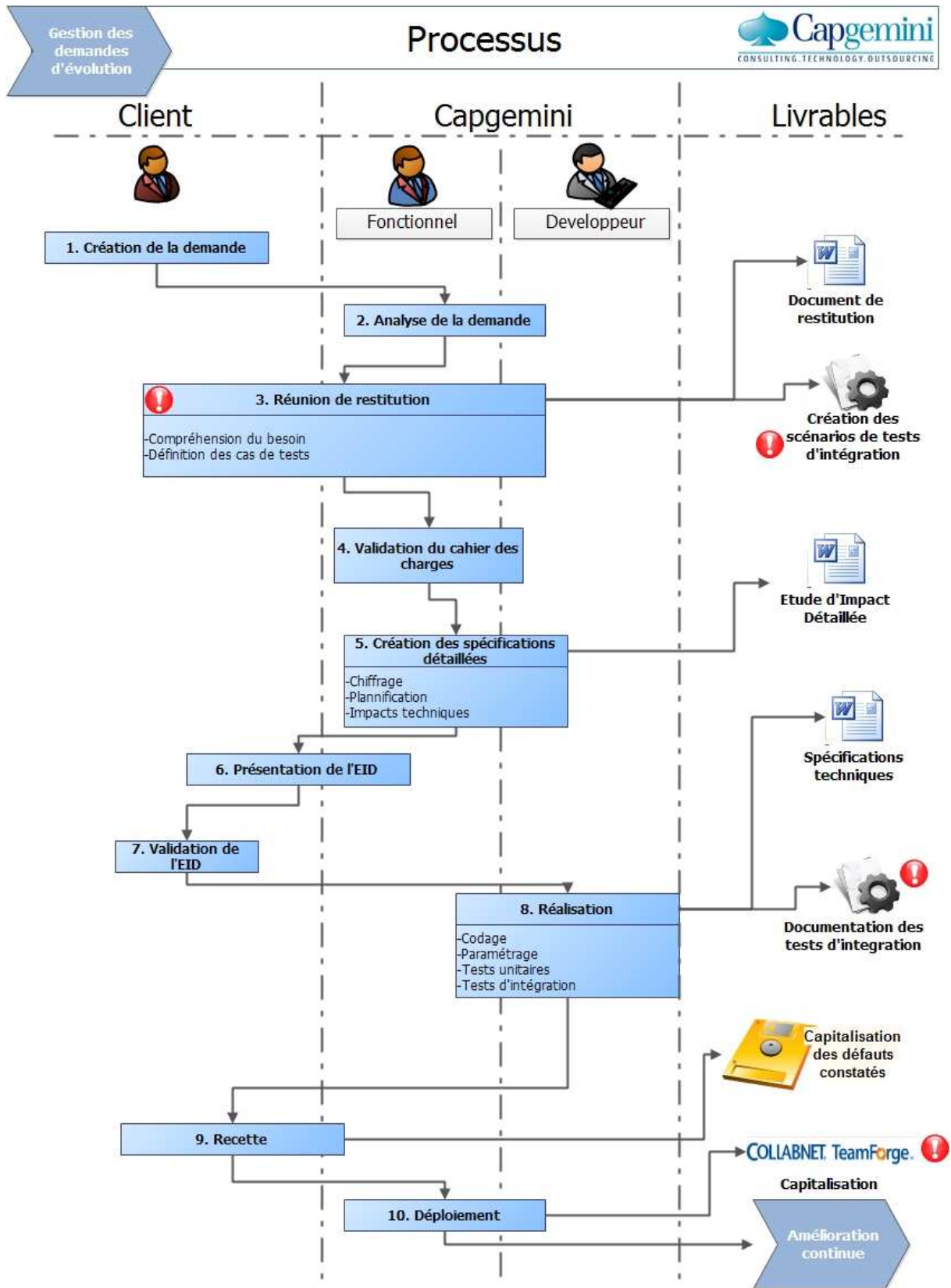


Figure 27 - Processus de gestion des demandes

Enfin, nous avons voulu que ce processus se termine par une ouverture sur l'amélioration continue pour matérialiser le fait que chaque exécution du processus peut amener des optimisations dans le fonctionnement.

2.4.2. Gestion de la demande de support

La gestion de la demande de support est plus simple dans la mesure où elle ne fait intervenir qu'un seul acteur Capgemini. Néanmoins, la cartographie de ce processus a permis de mettre en avant plusieurs aspects du traitement d'une demande qui étaient parfois négligés.

1. La nécessité d'enregistrer la demande avant toute action

Cette étape est nécessaire pour pouvoir garder une trace des demandes traitées et permettre de justifier la charge de support effectuée. C'est également une source importante de données pour identifier les principaux générateurs de support et ainsi pouvoir mettre en place des plans d'améliorations pour diminuer la charge associée.

2. L'importance de l'accusé de réception de la demande

Cette étape est celle qui est le plus souvent oubliée. Il s'agit pourtant d'une étape importante permettant de valider avec le client la criticité de sa demande. Cette information est fondamentale pour gérer les plans de charges et les affectations de chacun avec plus de précision et pour respecter au mieux les attentes des métiers. Cette phase permet également d'offrir un retour au demandeur sur le contenu et la compréhension de la requête et de lui donner le contact privilégié pour toute communication relative à celle-ci. Tout accusé de réception doit comporter les informations suivantes :

- Identifiant de la demande dans l'outil de gestion
- Nom de la personne affectée au traitement de la demande
- Criticité et date prévisionnelle de clôture
- Reformulation du besoin

3. La capitalisation

Comme pour les demandes d'évolutions, la capitalisation permet, d'une part de partager la connaissance acquise au cours du traitement d'une requête, et d'autre part d'enregistrer ce savoir dans le temps. Cet aspect est fondamental pour assurer la rotation des collaborateurs et leur compétence sur plusieurs applications.

La figure 28 illustre le processus de gestion de la demande de support, qui comme pour la demande d'évolution se conclut par l'ouverture sur l'amélioration continue.

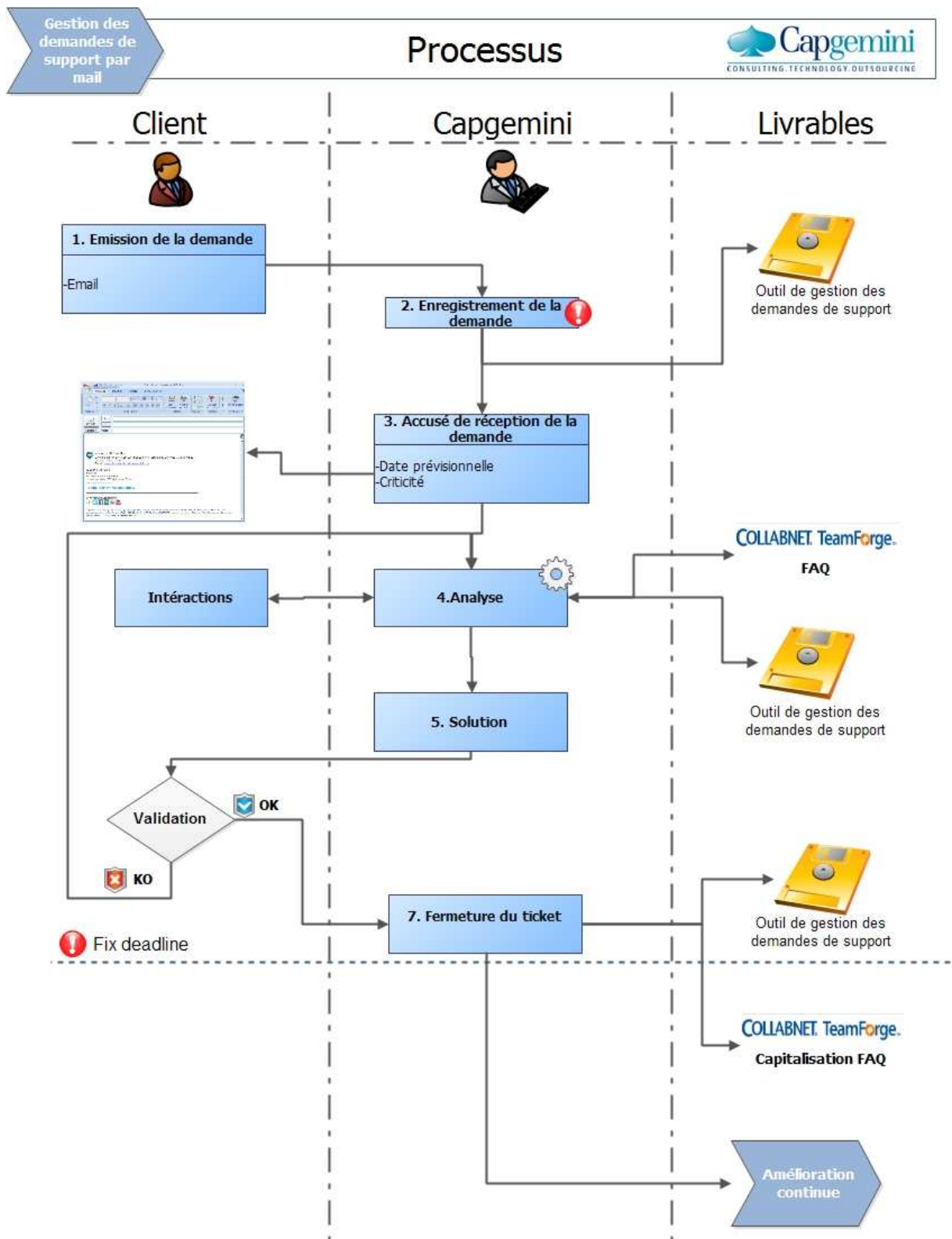


Figure 28 - Processus de gestion des demandes de support

2.5. Cartographie interculturelle

Au delà des problématiques Agile et Lean et afin de répondre aux enjeux de la multiculturalité dans le service, il nous a semblé important de prendre en considération les spécificités culturelles de chaque pays en dressant une cartographie des cultures en présence. Ces éléments, mis à la connaissance des membres de l'équipe, ont servi à donner une première approche des différences culturelles dans le but de réduire les jugements de valeurs et les préjugés. Néanmoins, il est difficile de donner une vision globale de la culture d'un pays sans tomber dans le piège des stéréotypes. Il a donc été important de rappeler dans cette cartographie l'aspect limité de l'exercice.

En effet, comme le montre la figure 29, la programmation mentale d'un individu repose sur une partie qui est acquise par le biais de son environnement social (Hofstede et al, 2010). C'est à ce niveau que s'adresse la cartographie décrite ici. Mais la personnalité des individus fondée à la fois par le patrimoine génétique de chacun et par l'influence des expériences personnelles est un caractère différenciateur unique qui ne doit pas être négligé.

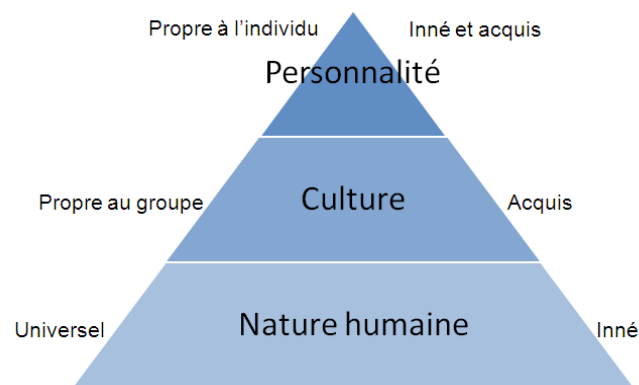


Figure 29 - Les trois niveaux de la programmation mentale d'un individu (Hofstede et al, 2010)

Afin de nous aider à dresser cette cartographie, nous avons étudié les travaux de Geert Hofstede qui ont permis de mettre en évidence un lien entre les cultures nationales et les comportements dans l'entreprise.

Les données d'Hofstede sont issues d'une vaste enquête menée auprès des employés d'IBM dès la fin des années 60 et qui a concerné plus de 60 000 personnes dans 71 pays. La particularité de cette étude est qu'elle a concerné des employés d'une même entreprise, ce qui a permis de fixer un certain nombre de paramètres pour ne laisser émerger que les différences nationales. Depuis sa première parution, l'ouvrage d'Hofstede a connu de nombreux remaniements et a été réactualisé pour la troisième fois en 2010. Les données qui ont été utilisées sont donc suffisamment récentes.

D'après Hofstede, les cultures sont définies selon 5 dimensions bipolaires indépendantes les unes des autres qui permettent de modéliser les styles de management ou les leviers de motivation des salariés.

Ces dimensions sont :

1. L'indice de distance hiérarchique
2. Le degré d'individualisme de la société
3. Le degré de masculinité/féminité
4. L'indice de contrôle de l'incertitude
5. L'indice d'orientation à long terme

Ces indices ont été mesurés à travers des questionnaires à choix multiples qui, même s'ils ne garantissent pas une fidélité absolue entre les réponses apportées et les comportements constatés, donnent quand même une tendance de la vision de chaque culture face à un aspect donné.

2.5.1. Indices culturels

Dans le contexte du centre de service SAP, l'équipe est répartie entre France et Inde, mais une des particularités du pôle BW est de travailler avec un client chinois sur l'application SSRM. La partie qui suit compare les cultures de ces deux pays avec la culture française selon les axes définis.

Mesure du degré d'inégalité dans la société : l'indice de distance hiérarchique

Cet indice donne une vision de la relation à l'autorité et de la gestion des inégalités selon chaque pays (voir tableau IV). Il est calculé selon une formule mathématique pour donner un chiffre entre 0 (faible distance hiérarchique) et 100 (forte distance hiérarchique) et il est composé de trois aspects:

- La crainte ou non pour un subordonné d'exprimer son désaccord à un supérieur hiérarchique
- La perception par un employé de la prise en compte, ou non, de son opinion par son supérieur
- La préférence des employés pour des styles de management autoritaires ou au contraire consultatifs.

Tableau IV - Indice de distance hiérarchique (Hofstede et al, 2010)

Pays	Indice de distance hiérarchique
Chine	80
France	68
Inde	77

Même si les résultats montrent que l'Inde a un modèle plus inégalitaire que la France, nous voyons que la différence n'est pas significative. Les deux pays se classent dans une catégorie où la distance hiérarchique est forte et où les employés attendent plutôt un management directif et paternaliste. Les cultures à forte distance hiérarchique montrent les caractéristiques suivantes :

- Les salariés ont une faible tendance à faire preuve d'initiative
- Les individus les moins puissants développent des liens affectifs bipolaires (affection/rejet) avec leurs supérieurs
- Les salariés respectent leurs supérieurs et considèrent que le statut amène également des obligations

Ces éléments ont été pris en compte dans les relations avec l'équipe indienne en adoptant un management de style paternaliste caractérisé par des directives claires et précises, laissant assez peu d'autonomie, mais en préservant la dimension affective de la relation.

De plus, la gestion des équipes indiennes qui possèdent un supérieur hiérarchique local et un supérieur fonctionnel en France a pu conduire à des situations incohérentes. Ce risque d'inconfort pour les salariés indiens a été identifié et a été traité en renforçant les points de partage transverses, incluant le management.

Mesure du degré d'individualisme d'une société.

Cet indice donne une vision des relations entre l'individu et le groupe dans une société. Cette dimension a un impact fort sur les aspirations professionnelles des personnes.

Un employé d'une culture individualiste (indice fort) attend avant tout de pouvoir organiser son travail comme il l'entend et de disposer d'une grande liberté. Pour s'épanouir, cet employé a besoin de relever des défis et de disposer de suffisamment de temps pour avoir une vie privée riche et des loisirs satisfaisants. Dans ce genre de société, les relations de travail sont essentiellement basées sur des intérêts réciproques évalués généralement selon des critères économiques.

Au contraire, un employé d'une culture collectiviste (indice faible) privilégie un travail qui lui permet de progresser et d'apprendre de nouvelles connaissances. Il est attaché à son espace de travail et aux conditions matérielles qui en découle. Enfin, il espère être reconnu pour ses mérites et ses accomplissements. Dans une société collectiviste, la relation de travail est une relation de confiance et de loyauté et se rapproche des liens familiaux.

Pour cet indice, la France montre de grandes disparités avec l'Inde et encore plus avec la Chine (voir tableau V). Cela a plusieurs impacts sur les relations de collaboration avec l'équipe offshore et sur les relations commerciales avec nos clients Chinois.

Tableau V - Indice d'individualisme d'une société (Hofstede et al, 2010)

Pays	Indice d'individualisme
Chine	20
France	71
Inde	48

Les sociétés collectivistes sont attachées à des relations harmonieuses entre les différents protagonistes. C'est le cas notamment dans des situations de négociations où une issue équitable est toujours privilégiée.

Cet élément essentiel nous a permis d'adapter notre attitude vis-à-vis de nos clients chinois en recherchant des compromis acceptables par tous en cas de désaccord.

De la même manière, la gestion des erreurs ou des fautes de l'équipe indienne a du être modérée pour éviter une mise en cause directe, sous peine de faire perdre la face à la personne visée. Ce point diffère des sociétés individualistes, et notamment de la France qui considère un affrontement comme salvateur car pouvant éliminer les non-dits ou les problèmes sous-jacents et donc faire naître une vision nouvelle.

Enfin les sociétés collectivistes privilégient plus le contact que la tâche. Il a donc été nécessaire de ne pas nous limiter à l'aspect professionnel de la relation avec nos collègues indiens ou nos clients chinois. Nous avons pour cela tenté de tisser des liens en réservant toujours une place à un échange culturel, même bref, lors de nos réunions.

Mesure de la masculinité/féminité de la société

On associe au caractère masculin des notions de compétition, d'agressivité ou d'assurance alors que l'on évoque plutôt la douceur, la modestie, l'altruisme ou l'empathie pour la féminité. Cet indice donne une vision de la dimension masculine ou féminine d'une société (voir tableau VII). Ce rattachement a été associé, dans le cadre de l'étude IBM, à l'importance donnée par la société à un certain nombre d'objectifs professionnels :

Pour le pôle masculin :

- Le niveau de rémunération
- La reconnaissance apportée par ses réussites professionnelles
- L'avancement et les promotions obtenues
- Les défis à relever permettant de stimuler le salarié

Pour le pôle féminin :

- L'harmonie avec sa hiérarchie
- La collaboration avec ses collègues
- Le cadre de vie personnel
- La sécurité de l'emploi

La répartition de ces notions entre le pôle masculin et le pôle féminin est relative et statistique. Par exemple, en moyenne, les hommes sont plus attentifs que les femmes au niveau de rémunération mais rien n'interdit que ce soit l'objectif prioritaire d'une femme.

Le défi est une notion associée également à l'individualisme et plus généralement, nous pouvons confondre ces deux notions. Néanmoins, l'étude de la corrélation entre les deux dimensions montre qu'il n'y a pas de liens. L'individualisme différencie le « je » du « nous » alors que la masculinité/féminité mesure la représentation que l'on a de soi-même.

Les données pour nos trois pays sont présentées dans le tableau VI, 100 étant une société très masculine et 0 une société très féminine. :

Tableau VI - Indice de masculinité d'une société (Hofstede et al, 2010)

Pays	Indice de masculinité/féminité
Chine	66
France	43
Inde	56

On note que les sociétés chinoises et indiennes sont plus masculines que la société française mais les résultats sont relativement groupés. Aux extrêmes pour exemple, le Japon obtient un indice de 95 quand la Suède a un indice de 5. Il a donc été difficile d'en tirer des conclusions sur des bonnes pratiques de communication ou de gestion à adopter.

Nous avons néanmoins pu intégrer dans notre gestion de la relation le fait qu'un salarié chinois aurait plus tendance à vivre pour son travail qu'un salarié français et qu'il fallait donc être encore plus attentif à ses demandes.

Mesure du degré de tolérance de l'équivoque dans une société : l'indice de contrôle de l'incertitude.

Cet indice donne une vision de la gestion de l'incertitude et de l'ambiguïté. Afin de déterminer l'indice, l'enquête IBM se base sur des questions relatives : au stress au travail, au respect des règles et des règlements et à la volonté des salariés de rester dans leur entreprise pour une longue période. Le degré de contrôle de l'incertitude est corrélé avec l'anxiété d'une population, et dans les cultures anxieuses, il est plus facilement acceptable d'extérioriser ses sentiments ou de hausser le ton dans une discussion.

Il ne faut pas confondre risque et incertitude. Le risque est lié à un objet et concret alors que l'incertitude est une notion plus vague qui n'a pas forcément de prise tangible. Dès qu'un risque est exprimé, il cesse d'être une source d'anxiété puisqu'il est connu, partagé et traité.

Dans le domaine du travail et de l'organisation, cet indice a plusieurs impacts :

Les sociétés dont l'indice de contrôle de l'incertitude est plus fort changeront très peu d'employeur au cours de leur carrière et privilégieront des emplois stables. Nous voyons dans le tableau VII que, au contraire, les cultures chinoises et indiennes auront moins de problèmes à changer souvent de société. Ce point est souvent caractéristique des équipes offshore qui présentent un taux de rotation assez élevé. Il a donc été nécessaire d'imaginer des stratégies pour essayer d'en limiter l'impact, notamment grâce à une gestion active des connaissances.

Le deuxième aspect de cette dimension est la relation aux règles. Une culture anxieuse (indice fort) a besoin d'être rassurée par un ensemble de règles formelles et précises, même si celles-ci ne sont pas appliquées. Au contraire, une société avec un faible indice de contrôle de l'incertitude peut s'auto-discipliner sans recourir à des lois explicites. Notre client chinois tout comme l'équipe offshore est donc à priori peu réceptif à des procédures strictes. Nous avons donc plutôt essayé de mettre en avant les processus existants et leur esprit plutôt que de s'attacher au strict respect des règles.

Enfin, on peut classer les sociétés présentant un faible indice de contrôle de l'incertitude comme ayant un penchant pour l'innovation au détriment de la routine. Dans l'affectation des demandes, nous avons donc essayé de privilégier l'équipe offshore pour des sujets de recherche plutôt que pour des sujets routiniers.

Cette stratégie était pourtant en contradiction avec la tendance habituelle de confier aux équipes indiennes les tâches les plus courantes et les plus familières et non celles liées à un processus créatif. En effet, le besoin de limiter les coûts et le manque de proximité des ressources offshore avec le client peuvent entraîner une incompréhension totale des besoins dans le cas de la création d'un nouvel outil et donc des risques importants. Cependant, nous avons essayé de garder cet élément à l'esprit quand cela était possible pour affecter plus équitablement les demandes et permettre aux salariés indiens de s'épanouir.

Tableau VII - Indice de contrôle de l'incertitude (Hofstede et al, 2010)

Pays	Indice de contrôle de l'incertitude
Chine	30
France	86
Inde	40

Indice d'orientation à long terme

Cette dimension n'est pas issue de l'étude IBM originale mais d'une étude nommée CVS (Chinese Value Survey) menée par Michael Bond dans le milieu des années 1980, appliqué à 23 pays et étendue par la suite à l'ensemble des pays concernés par l'étude IBM dans le cadre d'un programme nommé WVS (World Value Survey, 2012).

L'orientation à long terme est associée à des qualités comme la persévérance, le goût de l'effort, la patience et le pragmatisme alors que l'orientation à court terme fait plus de place à la chance, au mysticisme, aux traditions et à l'épanouissement personnel.

Dans ce domaine, la Chine comme plusieurs pays asiatiques (Japon, Corée du Sud, Taiwan) présente une très forte orientation à long terme au contraire des pays africains. L'Inde et la France ont des positions assez similaires et médianes dans ce domaine qui n'entraînent pas de différences culturelles majeures.

Tableau VIII - Indice d'orientation à long terme (Hofstede et al, 2010)

Pays	Indice d'orientation à long terme
Chine	87
France	63
Inde	51

En ce qui concerne notre client chinois, l'impact de son orientation à long terme est une propension à épargner très forte.

Nous avons pris cet élément en compte dans la gestion du budget annuel d'évolution pour lequel notre client avait une prédisposition culturelle à retarder ses décisions afin de conserver une réserve budgétaire conséquente.

Ceci générerait un risque de décalage de la production du budget sur la fin d'année et donc une possible incapacité à produire toutes les évolutions sur les dernières releases. Pour réduire ce risque, nous avons découpé le budget annuel sur les quatre releases afin d'avoir un budget trimestriel. Le client a ainsi eu plus de facilité à « dépenser » son budget de manière lissée sur l'année.

Perspectives

Nous voyons sur la figure 30 reprenant les indices des cinq dimensions que les cultures des pays nous intéressant (Chine, France, Inde) présentent de nombreuses disparités.

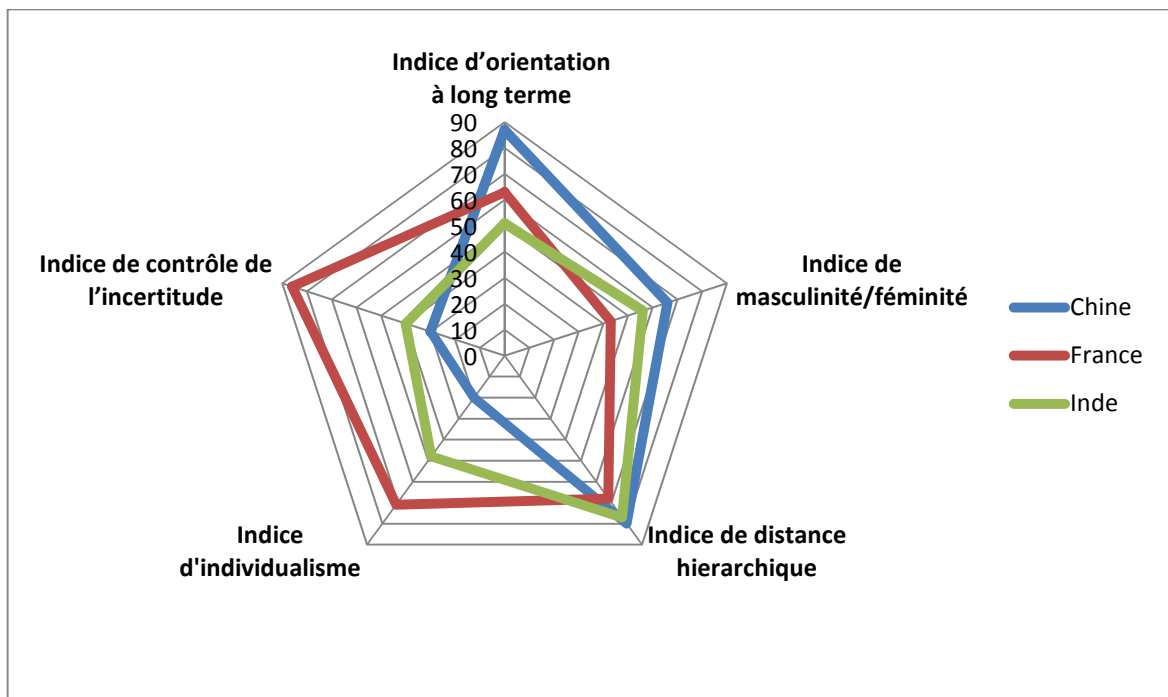


Figure 30 - Résumé des indices culturels pour la Chine, la France et l'Inde

Comme expliqué en introduction de cette partie, ces indices doivent être relativisés par les différences culturelles à l'intérieur d'une nation ou d'une entreprise et par la personnalité de chaque individu. Pour éviter de créer de nouveaux stéréotypes, nous avons donc utilisé ces informations comme des clefs ou des éléments de connaissance supplémentaires mais en les confrontant toujours à notre perception du réel.

L'intérêt de cette démarche est difficile à quantifier, mais la bonne entente avec nos différents partenaires tout au long de cette année et l'esprit de collaboration qui a prédominé sont autant d'éléments positifs à mettre à son crédit.

D'autres approches ont également vu le jour pour étudier le lien entre culture nationale et l'environnement de travail (Meier, 2010) mais ces études n'ont pas été prises en compte dans le cadre de ce mémoire. Elles pourront être consultées pour une amélioration de la cartographie.

2.5.2. Cartes d'identités

Afin d'améliorer la connaissance de chacune des différentes cultures, nous avons décidé de mettre à disposition des collaborateurs une cartographie des cultures sur la base des observations de l'étude de Hofstede.

Cette cartographie a été découpée en deux niveaux : une cartographie nationale (figure 35) qui a eu pour but de présenter la culture d'un pays d'une manière générale et des cartes d'identité qui ont eu pour objet de présenter les individus.

Les cartes nationales présentent d'abord quelques éléments de contexte du pays comme sa superficie, sa population, etc. Ensuite elles présentent les résultats de l'étude d'Hofstede sur ce pays en explicitant trois points clefs qui ressortent des résultats.

Afin d'éviter d'éventuelles incompréhensions, cette cartographie a été présentée aux membres de l'équipe offshore pour relecture et commentaires. Il en a été fait de même avec la carte pour la France. Enfin, une notice explicative des différents indices et des objectifs de la démarche a été fournie aux membres de l'équipe.

Voici l'exemple de la carte nationale pour l'Inde :

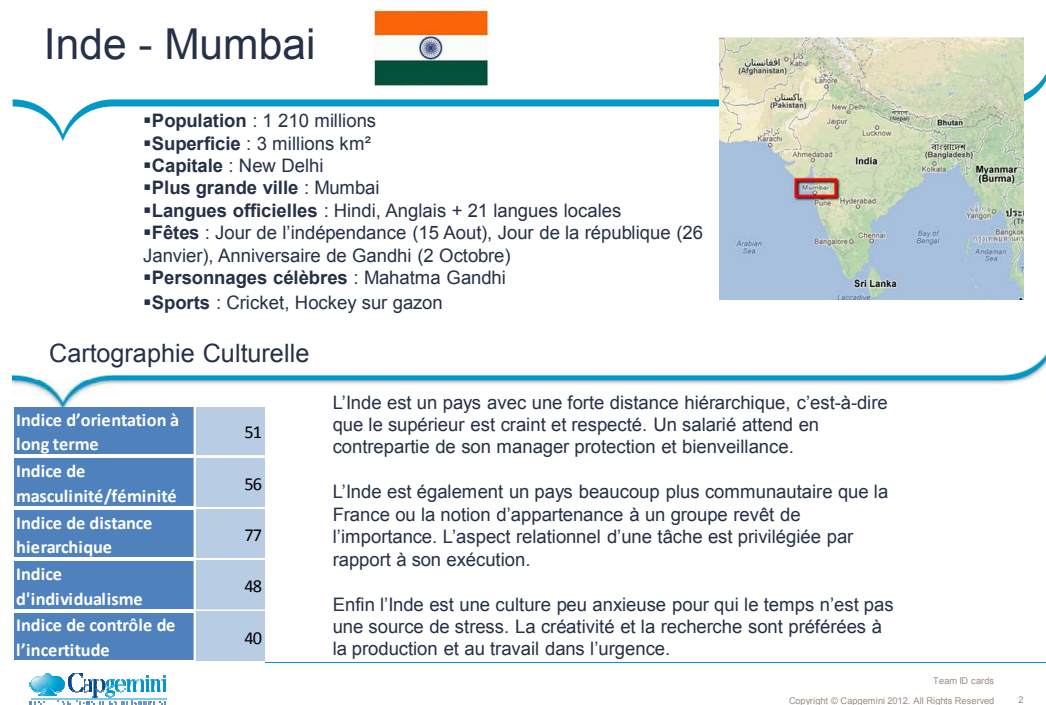


Figure 31 - Carte culturelle de l'Inde

En second lieu, la carte d'identité (figure 32) a pour but de créer du lien entre les équipes françaises et indiennes afin d'améliorer la collaboration.

Nous avons vu que l'affectif était un élément important dans la culture indienne et le fait de mieux connaître les salariés français permet de renforcer la perception de l'équipe en tant que groupe.

Manuel COLLET-BEILLON



- **Birth Date** : November 17th 1978
- **Marital status** : Civil union
- **Childrens** : Maëlle (Girl, 2006), Thomas (Boy, 2009)
- **In Capgemini since** : 2006
- **Free-time**
 - Football (soccer) in the “Dynamo of Grenoble” Team
 - Trail running (Ecotrail of Paris 2008 : 80km, TGV 2011 : 72km)
 - Ski, snowboarding
 - Favorite music : Metallica, Serj Tankian, Manu Chao
 - Favorite movies: Lord of the rings
 - Favorite series : Dexter, Game of Thrones
 - Favorite sport team : OM (Football), Bruleurs de loups (Ice hockey)
 - Favorite meal : Seared foie gras



Team ID cards
Copyright © Capgemini 2012. All Rights Reserved 1

Figure 32 - Carte d'identité

Il est difficile de mesurer les résultats de ces initiatives, cependant un sondage effectué auprès de nos collègues indiens a fait ressortir leur satisfaction quant à cette cartographie.

2.6.Modèle de gestion de la connaissance

Comme nous l'avons vu dans la partie 1.1.2, la gestion de la connaissance est un thème crucial pour la qualité d'un service informatique mutualisé. En effet, il faut d'une part minimiser les risques liés à la nouvelle organisation et d'autre part être en capacité d'intégrer facilement de nouveaux comptes ou de nouveaux collaborateurs pour optimiser l'intérêt du service. Cette partie décrit spécifiquement les actions mises en place autour de cet aspect.

2.6.1. Gestion de la documentation

L'ALM dispose d'un outil de gestion de la connaissance développé par la société Collabnet et nommé Teamforge. Cet outil présente plusieurs avantages :

- Accès partagé entre les équipes France et Inde.
- Client léger disponible sur le web depuis n'importe quel réseau pour un consultant Capgemini disposant des droits d'accès.
- Gestion des versions de documentation avec notes associées.
- Surveillance de la modification d'un document.
- Verrouillage d'un document.

La figure 33 illustre les métadonnées d'un document :


Document Details (Active Version)																															
Document Name:	 DT-EN-SSRM-BW - Guidelines - Sorting Rules & Navigation Block.xls		Status: Final																												
Document ID:	doc3893463		Current Version: 6																												
Description:	Document content:																														
	. List of the SSRM BW Queries, Views and Web Templates, and the sorting rule of the Characteristics by Key Figure (example: descending Spends amount).																														
	. List of Characteristics in Navigation Block, by Query, and sorting rule of Characteristics in the Navigation Block.																														
	. List of Selection Variables by Query, and sorting rule of Selection Variables.																														
Version Comment:																															
Version Created By:	Jeremy Duffrene - 08/20/2012 2:12 PM CEST																														
<div style="display: flex; border-bottom: 1px solid #ccc;"> <div style="margin-right: 10px;">Versions</div> <div style="margin-right: 10px;">Associations</div> <div>Review</div> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 5%;">Active</th> <th style="width: 15%;">Version</th> <th style="width: 75%;">Version Comment</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="radio"/></td> <td><input checked="" type="checkbox"/></td> <td>Version 6</td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/></td> <td>Version 5</td> <td>Updated by Angelica Aycard AMFR04235</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/></td> <td>Version 4</td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/></td> <td>Version 3</td> <td>.</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/></td> <td>Version 2</td> <td>. In tab "SSRM sorting rules by report", there were some mistakes in the sorting rules of Entity Country. The sorting rules of Entity Region have been aligned report. ? Excel file version 3.1.</td> </tr> <tr> <td><input type="radio"/></td> <td><input type="checkbox"/></td> <td>Version 1</td> <td>Excel file version: 3.1.</td> </tr> </tbody> </table>					Active	Version	Version Comment	<input checked="" type="radio"/>	<input checked="" type="checkbox"/>	Version 6		<input type="radio"/>	<input type="checkbox"/>	Version 5	Updated by Angelica Aycard AMFR04235	<input type="radio"/>	<input type="checkbox"/>	Version 4		<input type="radio"/>	<input type="checkbox"/>	Version 3	.	<input type="radio"/>	<input type="checkbox"/>	Version 2	. In tab "SSRM sorting rules by report", there were some mistakes in the sorting rules of Entity Country. The sorting rules of Entity Region have been aligned report. ? Excel file version 3.1.	<input type="radio"/>	<input type="checkbox"/>	Version 1	Excel file version: 3.1.
	Active	Version	Version Comment																												
<input checked="" type="radio"/>	<input checked="" type="checkbox"/>	Version 6																													
<input type="radio"/>	<input type="checkbox"/>	Version 5	Updated by Angelica Aycard AMFR04235																												
<input type="radio"/>	<input type="checkbox"/>	Version 4																													
<input type="radio"/>	<input type="checkbox"/>	Version 3	.																												
<input type="radio"/>	<input type="checkbox"/>	Version 2	. In tab "SSRM sorting rules by report", there were some mistakes in the sorting rules of Entity Country. The sorting rules of Entity Region have been aligned report. ? Excel file version 3.1.																												
<input type="radio"/>	<input type="checkbox"/>	Version 1	Excel file version: 3.1.																												

Figure 33 - Outil de gestion documentaire

L'utilisation de cet outil est assez récente dans le cadre de l'ALM pour Schneider puisqu'il a été mis en place en 2010.

Plusieurs difficultés ont été rencontrées suite à cette mise en place. La première difficulté fut la migration de la documentation qui provenait de sources différentes : d'une part d'un disque partagé et d'autre part de bases Lotus Notes.

Afin de migrer les documents, il a été nécessaire de mener un projet de rationalisation de la documentation qui s'était accumulée au fil des années. De plus, dans de nombreux cas, de multiples versions d'un même document coexistaient et l'implication des consultants a été nécessaire pour identifier les versions applicables.

La bascule sur le nouvel outil a été assez mal perçue par les équipes qui jugeaient l'outil peu performant et qui perdaient leurs repères avec cette nouvelle organisation de la documentation.

Notre projet de mutualisation des équipes BW ayant rendu encore plus fondamentale l'utilisation de cet outil, nous avons dû mettre en place des actions visant à favoriser l'adoption de Teamforge par les consultants.

Ces actions se sont essentiellement portées sur deux axes :

1. Une promotion de l'outil via un certain nombre de communications ciblées aidant les collaborateurs à se familiariser avec les différentes fonctions de Teamforge.
2. Une revue des processus impliquant l'accès à la documentation pour identifier les passages obligés par l'outil, que ce soit en consultation ou en mise à jour de document.

Ces actions de sensibilisation ont été importantes pour la prise en main de l'outil mais doivent encore être répétées tout au long de la vie du projet et notamment dans le cas d'arrivée de nouveaux collaborateurs ou dans le cas d'intégration de nouveaux comptes. En effet ce type d'outil est une source de progrès pour l'organisation mais nécessite beaucoup de rigueur pour rester optimisé.

2.6.2. Mise en place du kit du nouvel arrivant

Dès l'année 2011, il avait été fait le constat que l'intégration de nouveaux collaborateurs dans le service SAP était une tâche difficile et pour laquelle nous manquions de méthodologie.

Par le passé, les transferts de compétences s'appuyaient sur la documentation technique et fonctionnelle de l'application. Cette documentation, bien que très utile, présentait divers inconvénients :

- Il n'existait pas de schéma directeur présentant un parcours de formation dans son ensemble. De ce fait, les transferts de compétences vers un nouvel arrivant étaient souvent tributaires de la sensibilité du formateur et la qualité dispensée pouvait donc varier d'un collaborateur à l'autre. De plus, il n'y avait pas de moyen de capitaliser d'un transfert à l'autre.
- Le caractère détaillé des documentations existantes rendait son appréhension très difficile pour une personne non-initiée. En effet, les documentations applicatives sont profitables pour approfondir des points précis quand le collaborateur a déjà une vision d'ensemble de l'application.
- Il n'existait pas de documentation sur le contexte du service. Ce transfert de compétence étant fait, là aussi, en fonction du formateur ou de manière informelle.

La mutualisation des équipes BW, l'arrivée de nouvelles ressources et d'un nouveau client n'ont fait qu'accentuer ce besoin de disposer d'un kit de documentation permettant de faciliter l'arrivée de nouveaux consultants.

Afin de limiter le coût de création de ce kit, il a été décidé de le mettre en place lors de la formation d'un jeune embauché en le réalisant au fur et à mesure du transfert et en l'améliorant au fil du temps et des nouvelles entrées.

La première étape a consisté à construire le squelette du plan de formation servant à décrire les rubriques à aborder. Ce squelette a été découpé en une partie générique, commune à tous les domaines du service, et une partie spécialisée par application.

La figure 34 représente le squelette de la documentation.

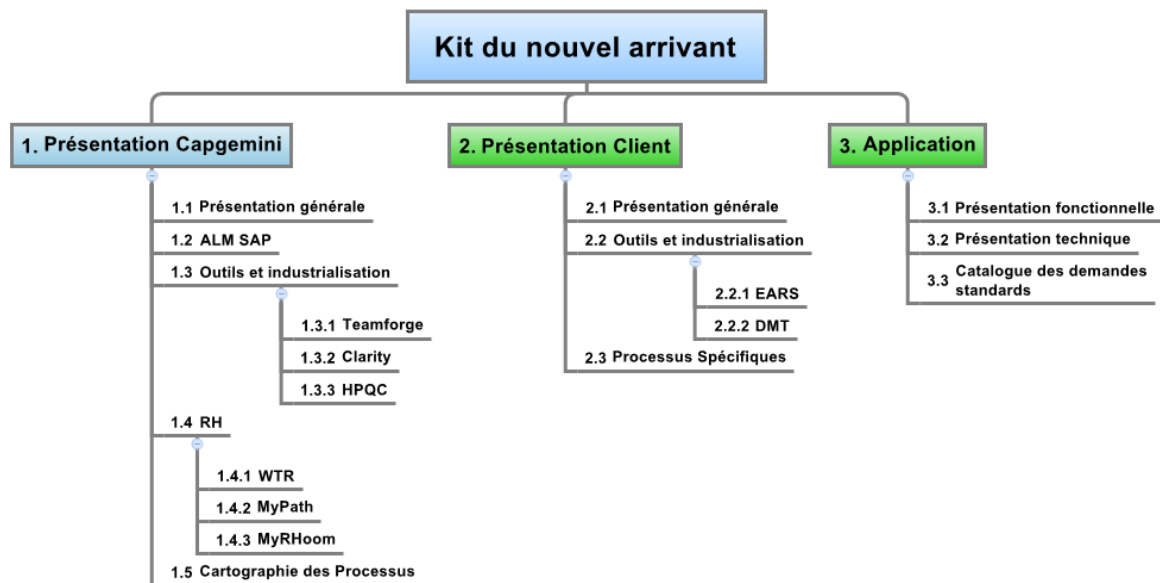


Figure 34 - Modèle de kit de nouvel arrivant

Chaque point du squelette a été rempli par un lien sur la documentation appropriée. Ce lien portait, soit sur la documentation applicative existante si elle était suffisamment synthétique, soit vers une nouvelle documentation, comme les fiches A4 décrites dans la partie suivante. L'utilisation de liens vers des documents externes a permis à ce kit de vivre au même rythme que la documentation de l'application et de garantir une cohérence de l'information quel que soit le périmètre par la réutilisation de documents transverses plutôt que par la duplication dans chacun des domaines.

Les nouveaux documents ont été rédigés par la personne formée pour, d'une part valider sa bonne compréhension du sujet et d'autre part l'impliquer d'avantage dans sa propre formation. Ce principe a été retenu pour chaque nouvel arrivant afin d'enrichir le kit sans générer de surcoût ou de contraintes sur les équipes en place.

2.6.3. Capitalisation

Pour répondre à l'objectif de baisse du support mais également à celui de productivité de l'équipe, il a été décidé d'utiliser l'outil documentaire *Teamforge* pour stocker trois nouveaux types de documents.

- Les fiches A4

Ces documents sont des procédures qui tirent leur nom du format de papier A4, car une des règles liée à ces procédures est qu'elles ne doivent pas dépasser une page dans un document Word.

Les bénéfiques de cette règle sont multiples :

- Les documents sont rapides à construire.
- La limitation de taille impose une approche synthétique.
- La recherche et la lecture de l'information est efficace.
- La mise à jour d'une procédure est simplifiée.

Le principe de fonctionnement lié aux fiches A4 est qu'à chaque recherche d'information sur un sujet, on crée la fiche correspondante. Par exemple, si un collaborateur cherche comment réserver une salle de réunion via la messagerie, on lui demande de créer la fiche correspondante une fois la solution trouvée.

D'autre part, les consignes formelles diffusées à l'équipe se font sous ce format. Par ce biais, toutes les consignes sont en permanence accessibles sur la base documentaire.

- Les fiches FAQ

Les fiches de FAQ (Foire Aux Questions) sont, elles, orientées sur une solution particulière mise en place pour un problème donné. La taille n'est pas limitée même si la consigne est quand même de veiller à être synthétique et efficace.

Ce type de document est également relu par un autre collaborateur pour vérifier qu'il est exploitable puis stocké dans la base documentaire. Il doit permettre à toute personne formée à la technologie de résoudre un cas de support ou d'évolution déjà identifié et documenté.

L'objectif d'une FAQ est de permettre à l'équipe d'apprendre collectivement du travail individuel et de ne pas perdre les connaissances acquises au fil du temps, celles-ci sont donc mutualisées et partagées et peuvent servir de base à la formation d'un nouvel arrivant.

- Le fichier FAQ d'astreinte

Selon le même principe, un document Excel a été mis en place pour capitaliser toutes les interventions faites dans le cadre des astreintes de nuit.

Le format doit répondre à des besoins spécifiques comme notamment un accès très rapide à l'information. L'utilisation d'un fichier Excel donne la possibilité de créer de multiples indexes de recherche (message d'erreur, nom de la chaîne impactée). Ensuite le document doit contenir toute l'information à un seul endroit. Enfin, l'orientation du document est beaucoup plus opérationnelle que celle des fiches FAQ qui ont un rôle plus pédagogique. On suppose donc dans ce fichier que le lecteur possède déjà une expérience solide du système.

3. Résultats

Cette partie présente une description des enseignements tirés de l'exécution du service sur la période allant de Janvier à Novembre 2012 au cours de laquelle ont été produites les 4 releases pour Schneider-Electric et plus de 50 demandes pour Rio-Tinto.

3.1. Indicateurs

Cette partie présente l'évolution des indicateurs décrits au chapitre 2.1.3 au fil de l'année.

Gabarits de release

La figure 35 représente l'évolution de l'indicateur de production sur l'année pour le client Schneider-Electric. Nous constatons que malgré un départ difficile sur le mois de janvier, la production a été lissée et stabilisée tout au long de l'année.

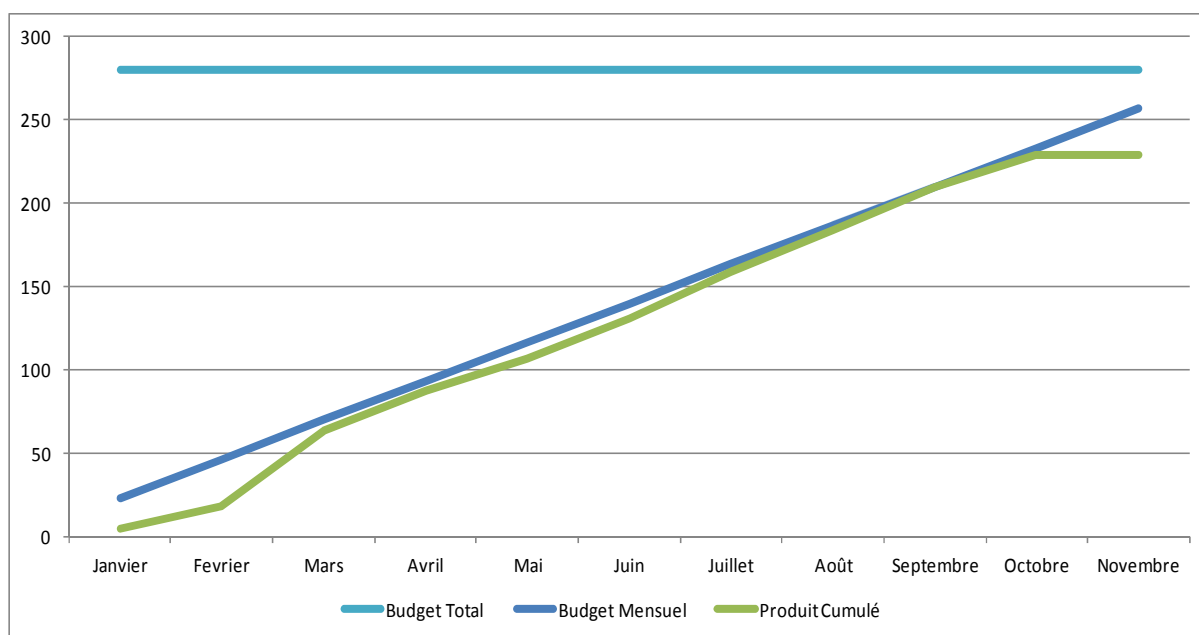


Figure 35 - Graphique de production pour le client Schneider-Electric

Nombres de retours sur recette

Le suivi de l'indicateur du nombre de retours sur recette nous a permis de suivre au cours de l'année l'amélioration de la qualité produite. La figure 36 illustre l'évolution de cet indicateur sur les 4 releases. Le nombre de retours est relativement faible. De ce fait, malgré une évolution positive sur l'année, un seul sujet totalisant 4 retours sur la dernière release a suffi pour dégrader les statistiques.

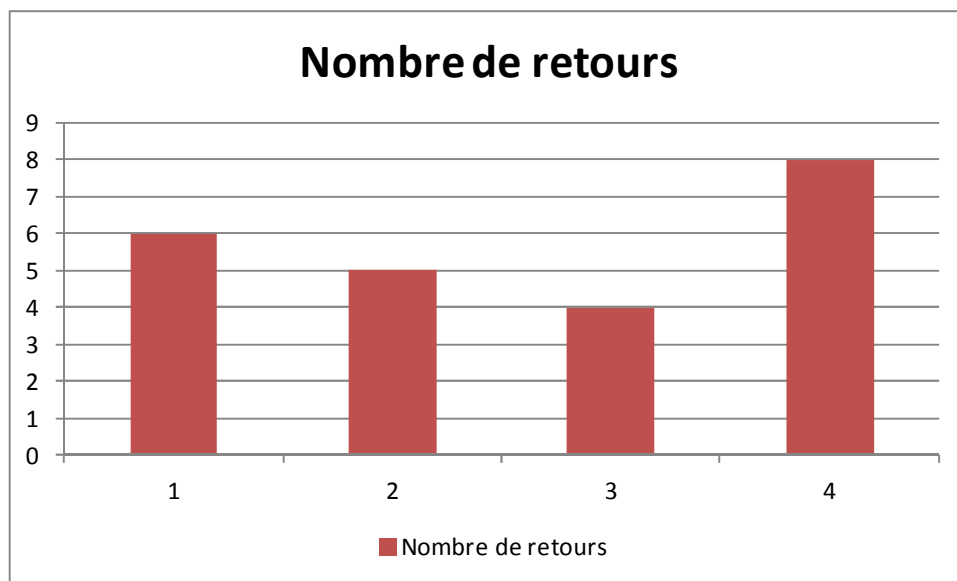


Figure 36 - Nombre de retours sur recette

Le bilan de la dernière release nous a permis d'identifier les causes de ces défauts :

- Solution applicative existante complexe.
- Difficulté de l'analyse.
- Hypothèses prises dans la conception qui se sont révélées fausses.
- Faiblesse dans la définition des scénarii de test au moment de la conception.

Les pistes mises en place pour limiter le risque de survenue de retours en recette ont été les suivants :

- Analyse en binôme et revue de solution pour les sujets complexes.
- Renforcement de la définition des scénarii de test.
- Renforcement de l'analyse de risque.

Notes de comités

L'évaluation de la production mensuelle a révélé une bonne satisfaction du client malgré quelques points de vigilance. Chaque point a été traité de manière spécifique :

- Problème sur une demande d'évolution non livrée en 2011
 - Une structure spéciale a été mise en place avec un binôme Capgemini et des instances de suivi avec le client afin de lever les incompréhensions.
- Problème sur le planning d'exploitation lié à un changement non maîtrisé
 - Il s'agit d'un incident très ponctuel lié à une erreur humaine. Les consignes ont été rappelées aux équipes et le point a été noté comme une piste d'amélioration potentielle afin de sécuriser le processus.

- Problème opérationnel pendant une période de congés
 - Ce point a été traité dans le cadre de la gestion de la connaissance en renforçant la documentation et en mettant en place des sessions de formations.

Ces différentes améliorations ont permis de résoudre les problèmes et de restaurer la confiance du client. La figure 37 montre que tous les comités ont été au vert à partir d'avril.

Comités de maintenance : évaluation de la production 🚩									
	01 - Janvier	02 - Février	03 - Mars	04 - Avril	05 - Mai	06 - Juin	07/08 - Juillet/Août	09 - Septembre	10 - Octobre
ASARIS	Orange	Vert	Orange	Vert	Vert	Vert	Vert	Vert	Vert
SSRM	Vert	Vert	Orange	Vert	Vert	Vert	Vert	Vert	Vert
MAP-CCO	Vert	Vert	Vert	Vert	Vert	Vert	Vert	Vert	Vert

Figure 37 - Evaluation mensuelle de la production

Ratio entre vendu et consommé

Ce ratio nous a permis de suivre les sujets en dérive pour lesquels le budget vendu était dépassé ou en voie de l'être. En règle générale, cet indicateur a montré que les évolutions respectaient les estimations initiales sauf pour quelques sujets dont la problématique avait été mal cernée et qui avaient généré de forts dépassements.

Les points d'amélioration mis en place pour limiter le risque sur ce genre de demandes ont été les mêmes que pour la réduction des retours sur recette (analyse en binôme, renforcement de la définition des scénarii de test et renforcement de l'analyse de risque).

Nombre de jours de support imputés

Le suivi du nombre de jours de support imputés par mois sur le domaine indique une tendance globale baissière sur l'année qui est le fruit de diverses actions d'amélioration ou d'automatisation mises en place :

- Actions de rechargements automatiques de données (SSRM).
- Correction automatiques de caractères interdits (ASARIS).
- Mise en place des fiches FAQ.

La figure 38 présente l'évolution du nombre de jours de support imputés par mois sur l'année.

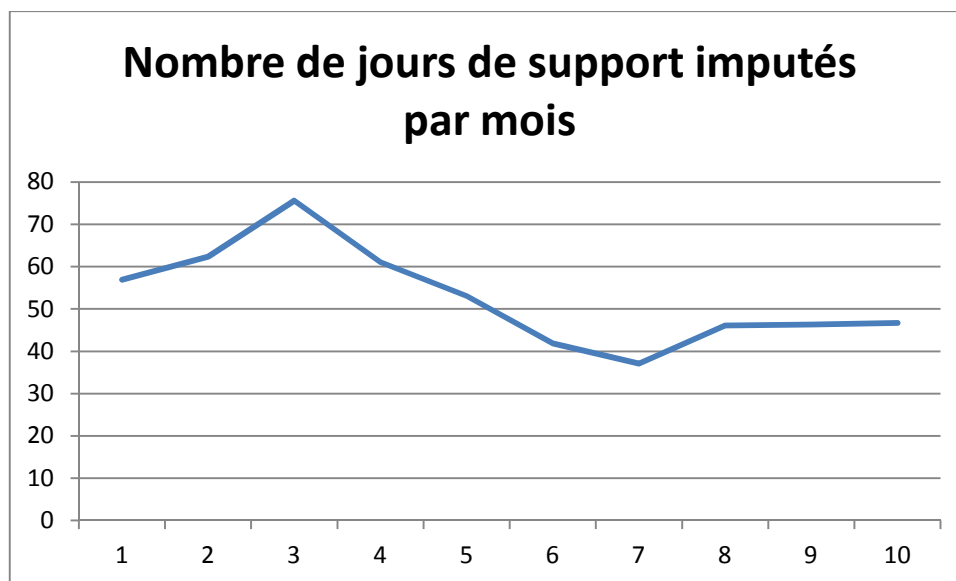


Figure 38 - Progression des imputations de support sur l'année

Nombre de tickets de correctifs

Le nombre de tickets correctifs, c'est-à-dire des demandes support liées à un changement, est resté nul tout au long de l'année alors qu'il y avait 7 demandes à fin 2011. Ce résultat très positif est à mettre au crédit de meilleures phases de tests et de recette et d'une plus grande collaboration avec le client.

Nombre de fiches FAQ et A4 créées

A la fin du mois d'octobre, le domaine comptait 34 fiches A4 et 19 fiches FAQ ce qui fait une moyenne de 5 documents créés par mois pour la capitalisation (voir tableau IX).

Tableau IX - Suivi des documents de capitalisation

Document	Mois dernier	Ce mois	Progression
A4	33	34	+1
FAQ	18	20	+2

Il est difficile de fixer des objectifs chiffrés pour la création de ces documents mais le suivi mensuel et la publication de cet indicateur permet de mettre en valeur l'implication de l'équipe dans l'effort de capitalisation.

3.2. Production Opérationnelle

La période sur laquelle s'étend ce mémoire (Janvier à Novembre 2012) a vu la production de 890 jours de maintenance sur le périmètre Schneider-Electric. Sur ces 890 jours, près de 600 étaient des jours de support utilisateur. Les 290 autres jours étaient répartis sur des demandes de maintenance évolutives ou correctives.

Sur le périmètre Rio-Tinto, le volume d'activité global s'élève à près de 80 jours, confondu entre surveillance des traitements, résolution des problèmes d'exploitation et demandes d'évolutions.

3.2.1. SAP BW

D'un point de vue opérationnel, l'équipe intervient sur le produit BW de SAP. Cette partie présente les concepts principaux du logiciel.

Ce produit est un logiciel décisionnel du marché qui était leader en 2010 devant Oracle, SAS et IBM Cognos (tableau X). La version utilisée est la version 3.1 qui est une ancienne version. Depuis, l'éditeur allemand a intégré Business Object, société française rachetée en 2007 (Le Monde Informatique, 2007).

Tableau X - Parts de marché pour les éditeurs de logiciels décisionnels en 2010 (Gartner, 2010)

Company	2010 Revenue	2010 Market Share (%)
SAP	2,413.1	22.9
Oracle	1,645.8	15.6
SAS Institute	1,386.5	13.2
IBM	1,222.0	11.6
Microsoft	913.7	8.7
Other Vendors	2,940.6	27.9
Total	10,521.8	100.0

On peut expliquer l'avance de SAP sur ce marché par son intégration entre son progiciel de gestion (également connu sous le nom ERP pour Enterprise Resource Planning) et sa solution décisionnelle. Cette intégration permet de tirer parti de la multitude de données présentes sur le progiciel et disponibles facilement grâce à des extracteurs livrés avec le produit.

L'architecture BW est répartie en 3 blocs :

Source Systems

Il s'agit de l'application qui génère ou stocke les données en entrée du décisionnel. Ces données sont ensuite transmises à BW par le biais d'extracteurs nommés ETL (Extraction, Transformation, Loading). BW peut être connecté à différents types de systèmes, le plus naturel étant le progiciel de gestion SAP R/3. C'est le cas dans l'application ASARIS qui est connectée avec le module logistique de SAP. C'est également le cas de l'application MAP-CCO qui est connectée avec les modules comptabilité et contrôle de gestion.

SAP édite également des solutions de gestion de la relation client ou de la relation fournisseur qui sont séparées de l'ERP. C'est le cas de l'application SSRM qui est connectée avec l'application de gestion des fournisseurs implémentée chez Schneider-Electric.

Enfin, SAP autorise des connexions à des systèmes non-SAP par le biais d'API (Application Programming Interface) ou de fichiers plats.

Administrator Workbench

Ce bloc est le cœur de la solution décisionnelle, c'est ici que sont définis les flux et les différents objets de stockage qui permettent d'implémenter les états suivant les axes d'analyses demandés par le métier.

Un flux dans BW est composé de trois objets de stockage ayant chacun des rôles différents.

La PSA (Persistent Staging Area) est une zone de stockage permettant sur le BW de conserver et visualiser les données brutes issues du système source avant de les charger dans les objets suivants. Cette étape permet notamment de corriger certaines erreurs, comme des caractères non reconnus qui seraient problématiques pour la suite du flux.

L'ODS (Object Data Store) correspond à une table de base de données stockant en colonnes toutes les informations extraites depuis la PSA. La définition de la structure de l'ODS est importante puisqu'elle va déterminer quelles sont les caractéristiques clefs du flux, quels sont les attributs et les ratios ainsi que leurs options d'agrégation :

ODS Object	Techn. name / value	O.	Dat...	L	Key f...	C.	N.	A...	E...	Unit
SSRM Spend Analysis ODS 1	O_SPEND1									
Status Info										
Version	Active									
Save	Saved									
Status	Active, executable									
Settings										
Key fields										
Calendar Year/Month	0CALMONTH		NUMC	06						
Time stamp	C_TIMESTP		NUMC	14						
Sender (SMS File)	C_SENDER		CHAR	03						
EPC Code (SMS File)	C_EPCCODE		CHAR	12						
DUNS Number	C_DUNSNUM		CHAR	09						
Family Code (SMS File)	C_FAMCODE		CHAR	03						
Currency (SMS File)	C_CURCY3		CHAR	03						
Data Fields										
Business Partner	C_PARTNER		CHAR	10						
P. Org. SSRM Organizational Unit	C_ORGSRM		NUMC	08						
STARS Product Category	C_PRODSRM		CHAR	32						
Currency key	0CURRENCY		CUKY	05						
Amount	0AMOUNT		CURR	09	Amount	<input checked="" type="checkbox"/>	<input type="checkbox"/>	SUM	SUM	0CURREN...

Figure 39 - Structure d'un objet de stockage de donnée BW

Le cube (ou Infocube) est l'objet final de stockage de données. Contrairement à l'ODS, il est construit autour d'un schéma en étoile (figure 40). L'élément central de ce schéma est une « table de fait » qui stocke les ratios associés à des pointeurs (ID) renvoyant sur les caractéristiques. Les attributs de ces caractéristiques sont stockés dans des tables appelées dimensions.

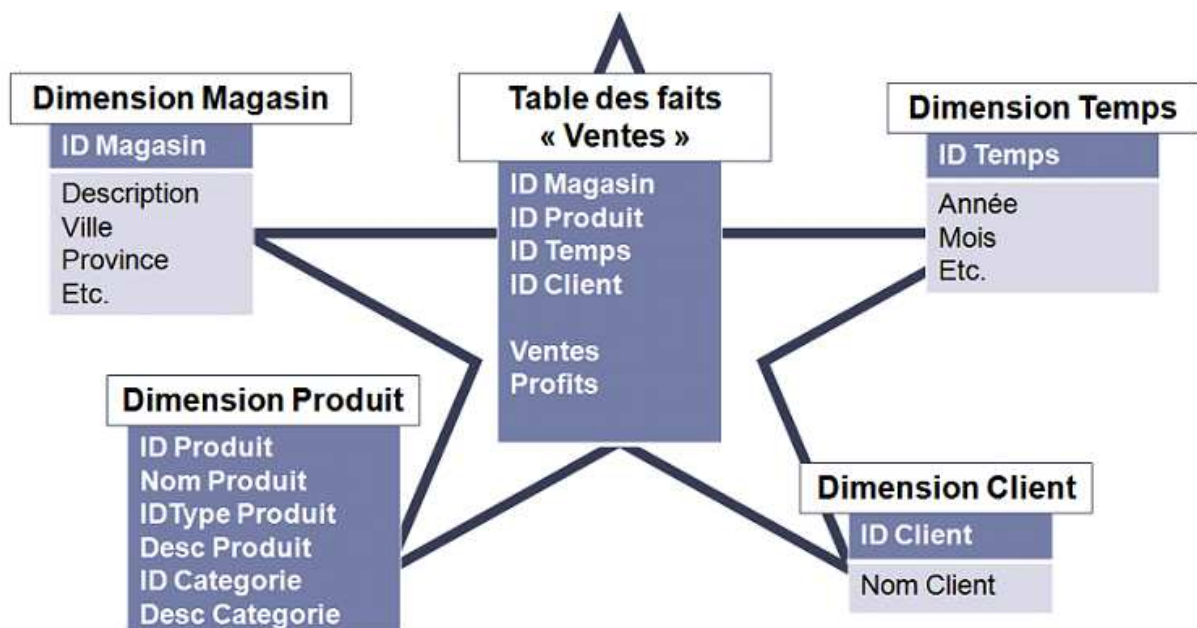


Figure 40 - Schéma en étoile d'un cube BW

Ces trois objets (PSA, ODS, CUBE) sont tous composés « d'info-objets » qui représentent la plus petite unité d'information dans BW. Elles peuvent être de cinq types :

1. Caractéristique : par exemple un client, un fournisseur ou un produit. Les caractéristiques sont en suite utilisées dans les états pour être des clefs de tris ou d'analyse.
2. Ratios : ce type d'info-objet fournit les valeurs qui sont affichées dans le reporting, par exemple des quantités, des volumes, des montants.
3. Unité : ces info-objets sont associés aux ratios pour préciser leur sens. Pour les ratios de type montant, il s'agit de la devise de la transaction.
4. Caractéristique temporelle: ces info-objets stockent des valeurs comme la date, la période fiscale ou l'année de la transaction.
5. Les caractéristiques techniques : elles stockent des informations importantes pour BW comme l'identifiant d'un chargement dans un cube.

Les info-objets de type caractéristique sont définis par un type et une longueur. Elles peuvent également porter :

- des attributs stockés comme données de base,
- des textes de différentes tailles et de différentes langues,
- une hiérarchie.

L'utilisation d'une info-objet dans un cube rend accessible dans ce cube tous les attributs de données de base de cette caractéristique.

Dans SSRM, l'info-objet principal est le fournisseur : il porte toutes les informations du fournisseur qui sont ensuite utilisées dans le reporting pour afficher les ratios suivant les axes d'analyse voulus (pays, région, responsable achats, ...)

Business Explorer

C'est la partie édition de BW. Elle permet de créer des rapports visuels en agrégeant les ratios suivant les axes d'analyses choisis. Ces axes pouvant être n'importe quelles caractéristiques contenues dans les dimensions. La figure 41 représente un exemple d'état pour lequel on affiche un volume d'achat suivant la région et le marché concerné. L'utilisateur peut, à loisir, ajouter ou retirer des dimensions, filtrer sa sélection ou modifier l'affichage et l'ordonnement des données.

Navigation block:			
Level 1		Level 2	Level 3
Level 4		Level 5 – Entity	Entity Region
Entity Country		DUNS Ultimate	DUNS Number
Supplier Country		Linked Market	Linked Commodity
Classification		Supplier Status	SOC Supplier Status
Commodity Manager		Commodity Delegate	Cdty Delegate Org.
Supplier Leader		Supplier Leader Org.	STARS Market
STARS Commodity		STARS Tech. group	STARS Family
SAM			

5 Spend Analysis			
Entity Region	STARS Market		Spend in Euro
APAC	ELECTRONICS AND ELEC	EE	369.632.488 EUR
APAC	FABRICATED COMPONENT	FC	273.322.509 EUR
APAC	PCB&A	PCBA	107.455.969 EUR
APAC	RAW MATERIALS AND P	RM	99.581.004 EUR
APAC	NON PRODUCTION MARKE	SE	108.924.959 EUR
APAC	Result		970.890.279 EUR

Figure 41 - Exemple d'état édité par BW

3.2.2. Catalogue de demandes

Voici la liste des demandes d'évolutions traitées dans la période sur les différents clients :

Schneider-Electric

La plupart des changements concernent des modifications d'état avec des ajouts ou des suppressions de données (caractéristiques) impliquant des modifications dans toute la chaîne (Extracteurs dans le système source, Objets de stockage sur BW et Etats de reporting).

Chaque demande représente une évolution ou une action de gestion de données non standard et donc n'entrant pas dans le cadre du support utilisateur.

Les demandes d'évolution consistent en majorité à améliorer les flux de données existants pour ajouter des nouveaux ratios ou de nouveaux attributs dans les cubes. Elles peuvent également correspondre à la modification d'une règle de gestion sur un attribut ou un ratio particulier.

La majorité des demandes traitées impliquent de bien connaître l'outil mais également de maîtriser l'environnement fonctionnel et les impacts de la modification d'une règle existante sur le fonctionnement global de l'application. La gestion des connaissances est donc un élément fondamental de la réussite de ces demandes.

Le tableau XI présente la liste des évolutions traitées pour Schneider-Electric cette année.

Tableau XI - Liste des demandes traitées dans l'année pour Schneider-Electric

Id	Titre	Application	Chiffrage	Description
AMFR03763	Flux transport adresse de livraison: resoudre le probleme des caracteres non reconnus	ASARIS	6	Correction automatique d'erreurs dans un flux de chargement
AMFR03942	FSD71: Passage des ratios stock dédié d'une vue actuelle à une vue historisée + corrections	ASARIS	11	Modification de caractéristiques BW et de requêtes
AMFR04099	Requêtes Transport - Prise en Compte des Colis Supprimés	ASARIS	6	Modification de caractéristiques BW et de requêtes
AMFR04144	Fiabilisation de la gestion du stock sécurité dans les flux	ASARIS	16	Modification de caractéristiques BW et de requêtes
AMFR04145	RFC fonction	ASARIS	3	Developpement d'un module d'extraction
AMFR04328	FSD49B: modification de l'affichage std pour mise en conformité avec la macro	ASARIS	2,5	Modification de caractéristiques BW et de requêtes
AMFR04547	FSD8: SIR Modification de la macro	ASARIS	3	Modification de caractéristiques BW et de requêtes
AMFR04688	P2DC: Cube TRANSPORT FSD 49 Ajout du pt d'expedition	ASARIS	4,25	Modification de caractéristiques BW et de requêtes
AMFR04730	P2DC:Cube TRANSPORT Ajout du pt d'expedition: reprise de l'historique	ASARIS	5	Reprise de données
AMFR03904	Currency rate variation	MAP-CCO	1,5	Modification de caractéristiques BW et de requêtes
AMFR03973	MAPCCO 2012 - correction sur le traitement mensuel integration des fichiers des CME lorsque la charge est superieure au run mensuel prévu sur la solution MAPCCO -	MAP-CCO	9	Modification de données
AMFR04031	Nouveau ratio - Sales GF au TP GF	MAP-CCO	5	Création de caractéristiques BW et de requêtes
AMFR04115	Transcodification Ref commerciale / logistique	MAP-CCO	33	Création/Modification de caractéristiques BW et de requêtes
AMFR04116	Maintien du "0" du material code	MAP-CCO	11,75	Modification de données
AMFR04118	Recalcul cube - Janvier 2012	MAP-CCO	2,5	Modification de données
AMFR04254	Recalcul cube Jan/Fev - Pelco(TAC)	MAP-CCO	0,5	Modification de données
AMFR03819	Developpement Small pour implementation des preferred	SSRM	24	Modification d'un format d'echange de données
AMFR03946	Additional SPM revise action by 20th Jan 2012	SSRM	1	Modification de données
AMFR04101	Regular SPM revise campaign - 20th Jan 2011 - 4th revise of 2011	SSRM	3	Modification de données
AMFR04102	2012-Mar Regular SPM revise campaign - 1st revise of 2012	SSRM	3	Modification de données
AMFR04103	2012-Jun Regular SPM revise campaign - 2nd revise of 2012	SSRM	3	Modification de données
AMFR04104	2012-Sept Regular SPM revise campaign - 3rd revise of 2012	SSRM	3	Modification de données
AMFR04126	Removing of DP3 calculation - 2nd Release Y2012	SSRM	4,5	Modification de caractéristiques BW et de requêtes
AMFR04129	Specific attribute reengineering - 2nd Release 2012	SSRM	9,5	Modification d'un processus de chargement de données
AMFR04234	SOC Supplier Status - 2nd Release 2012	SSRM	17,5	Modification de caractéristiques BW et de requêtes
AMFR04235	OTHER DUNS & STARS KPI new reporting function - 2nd Release 2012	SSRM	6,5	Création de caractéristiques BW et de requêtes
AMFR04236	Month to Month identification of missing spend (long term solution) - 3rd Release 2012	SSRM	10	Création de caractéristiques BW et de requêtes
AMFR04395	Design DMTs for Release 3 & 4 - Y2012	SSRM	10	Prestation d'assistance à maitrise d'ouvrage
AMFR04420	Enhance Purchasing KPI Performances - 3rd Release Y2012	SSRM	2,25	Modification de requêtes
AMFR04421	Fill a vacancy of linked market & commodity - 3rd Release Y2012	SSRM	3	Modification de requêtes
AMFR04422	Remove D&B Hierarchy - 3rd Release Y2012	SSRM	5,5	Modification d'un processus de chargement de données
AMFR04435	Modification of the IDOC between VENUS & SSRM	SSRM	3,75	Modification d'un format d'echange de données
AMFR04580	Y2012 Release 4 - Purchasing KPI spend calculation correction for monthly missing spend	SSRM	1	Modification de requêtes
AMFR04648	Release 4 - Y2012 Master Data Characters	SSRM	5	Modification d'un processus de chargement de données
AMFR04649	Release 4 - Y2012 Reporting improvement - add ultimate supplier linked market and commodity	SSRM	21	Modification de caractéristiques BW et de requêtes
AMFR04650	Release 4 - Y2012 Add SOC in supplier master data query	SSRM	2	Modification de caractéristiques BW et de requêtes
AMFR04651	Release 4 - Y2012 Reporting data slight modification (spend and spend+payment terms)	SSRM	17	Modification d'un processus de chargement de données
AMFR04664	Release 4 - Y2012 Optimize SOC status % reporting	SSRM	5	Modification de requêtes
AMFR04751	SSRM decomissioning - first wave	SSRM	5	Prestation d'assistance à maitrise d'ouvrage
AMFR04814	Y2012 Release 4 - Add Ultimate Linked Commodity and Ultimate Linked Market into SOC Status % report	SSRM	2	Création de caractéristiques BW et modification de requêtes

Rio-Tinto

Concernant le client Rio-Tinto, leurs flux de données étant beaucoup moins matures que pour Schneider-Electric, les principales demandes d'évolutions ont concerné la fiabilisation de ces flux. Le tableau XII présente les demandes gérées pour Rio-Tinto en dehors des incidents d'exploitation.

Tableau XII - Liste des demandes traitées dans l'année pour Rio-Tinto

Titre	Description
RT-B-Problème dans les structures clients	Modification d'un processus de chargement de données
RT-B-Charger version P01 à partir de Syscom et de SAP	Reprise de données
RT-B-Données de ZS PIC non transférées dans ZS_SICV1	Modification d'un processus de chargement de données
RT-B-Charger version B00 année 2012 usine AU	Reprise de données
RT-B-BW - charger le fichier GCBW0111_DONNEES	Reprise de données
RT-B-Mise en place des purges des logs BW sur PEB	Modification d'un processus de chargement de données
RT-B-Chargement des taux BUDG pour USD et relance process chain data load_2	Reprise de données
RT-B-Charger dans BW le fichier PICDATA	Reprise de données
RT-B-BW Charger fichier extrait de Syscom	Reprise de données
RT-B- ZMAKETTP is empty in our queries results	Modification d'un processus de chargement de données
RT-B-Vérifier la documentation technique de l'application BW EHS sur PGB	Documentation
RT-B-Modifier enchaînements chaînes BW	Modification d'un processus de chargement de données
RT-B-Réaliser une documentation technique pour l'application BW EHS sur PGB	Documentation
RT-B-BW: bloquer accès quand process chains tournent	Modification d'un processus de chargement de données
RT-B-Test du programme de surveillance de process chain sur SICA3	Modification d'un processus de chargement de données
RT-B-BW charger fichier extrait de SAP PIC	Reprise de données
RT-B-Compléter la documentation technique pour l'application BW EHS sur PGB	Documentation
RT-B-Problèmes suite à création nouveau marché	Modification d'un processus de chargement de données
RT-B- modification job de la process chain Transactional Data load3 for Sica3	Modification d'un processus de chargement de données
RT-B-Charger dans BW fichier à partir de Syscom	Reprise de données
RT-B-Problème avec la table ZMARCHE	Modification d'un processus de chargement de données
RT-B-Suite au ticket 3400081 : demande de modification des P.C. EHS/PDM sur PGB	Modification d'un processus de chargement de données
RT-B-BW charger table des types de marchés	Reprise de données

3.2.3. Exemple de demandes de changement : Projet « Preferred »

Parmi les requêtes traitées cette année, la plus volumineuse est celle dénommée « Preferred » pour l'application SSRM.

Contexte de la demande

L'application SSRM comprend deux parties :

1. Une partie nommée EBP (Enterprise Business Partner) pour la gestion des fiches fournisseurs et des données statiques,
2. Une partie BW pour la restitution des données transactionnelles (volumes achats et qualité)

Cette application fait partie du système d'information Schneider dit « Legacy » par opposition au « Core system », qui est le système unique cible mondialisé développé dans le cadre du projet BRIDGE (projet débuté il y a plus de dix ans).

Dans le cadre de ce projet, Schneider a émis la volonté de remplacer l'application SSRM par une application nommée VENUS.

Les principaux avantages de ce remplacement sont :

1. Montée de version du module CRM (« Customer Relationship Management » ou gestion des relations clients) 4.0 en CRM 5.0
2. Intégration de VENUS dans le core system BRIDGE : VENUS devenant la base fournisseur maître des SAP en version Bridge

VENUS est déjà utilisée par des régions telles que l'Inde ou l'Autriche-Hongrie.

L'application VENUS ne disposant pas de la partie BW de SSRM, le scénario retenu a été de développer une interface entre VENUS et SSRM EBP pour la gestion des fiches fournisseur et de garder ainsi la partie BW de SSRM intacte. L'ancienne partie SSRM EBP a donc été gardée en « boîte noire ».

La figure 42 présente le schéma d'architecture actuel de l'application. La partie verte est gérée par Schneider-Electric qui a externalisé cette prestation chez une société de service indienne nommée WIPRO. La partie bleue est gérée par Capgemini dans le cadre du centre de service SAP BW.

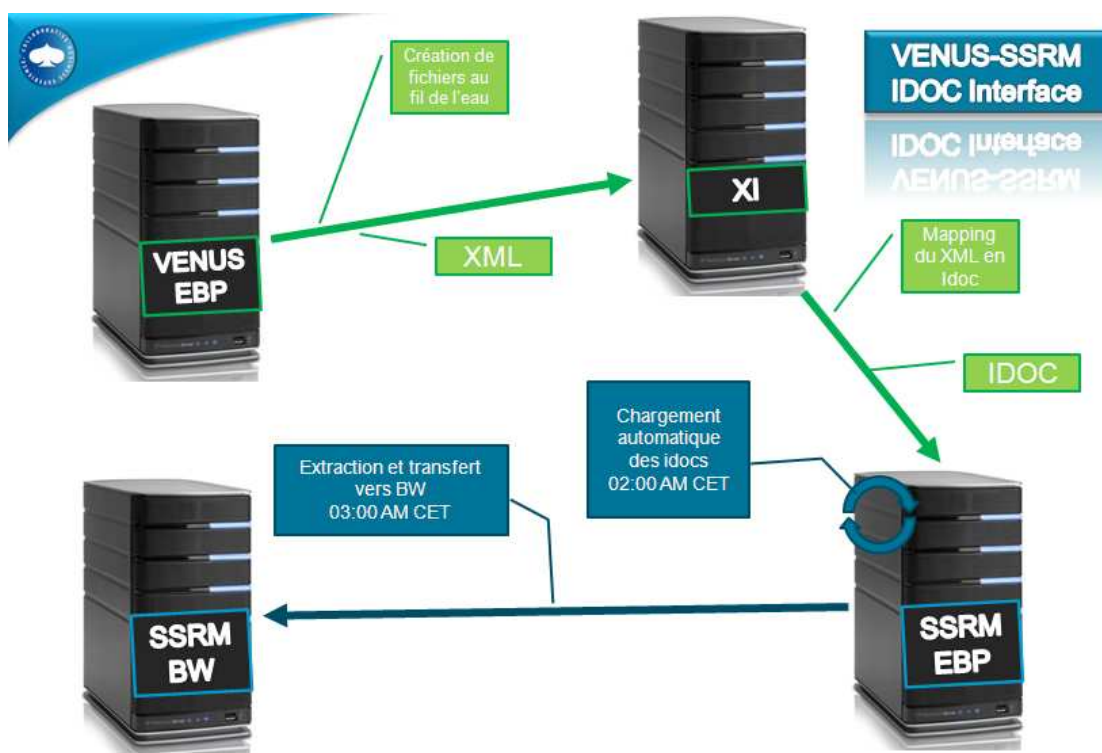


Figure 42 - Architecture des interfaces de l'application SSRM

L'interface entre l'application VENUS et l'application SSRM se fait par l'intermédiaire d'iDOC (intermediate DOCument) créés par le middleware SAP nommé XI (eXchange Infrastructure).

Un iDOC est constitué d'un en-tête, de plusieurs segments de données et d'enregistrements de statut.

Chaque élément d'un iDOC a une fonction particulière :

- Le contenu, la structure, l'émetteur, le récepteur et le statut actuel de l'iDOC sont définis dans l'en-tête de l'iDOC.
- Chaque segment de données contient un en-tête standard constitué d'un numéro de segment séquentiel, d'une description du type de segment et d'une zone de chaîne de 1000 caractères contenant les données réelles du segment.
- Les enregistrements de statut retracent les étapes du traitement de l'iDOC jusqu'à la date indiquée.

Les en-têtes et les éléments de statut ont un format fixe pour tous les types d'iDOC alors que les segments de données ont une taille, une séquence, un nombre et une signification différents selon le type d'information porté par l'iDOC.

Par exemple, un iDOC de type « pièce comptable » portera dans ses segments de données : un segment d'en-tête de pièces et deux, ou plus, segments de postes de pièces comptables.

Dans notre cas, l'iDOC CREMAS (pour « Creditor Master data » ou « donnée de base fournisseur ») porte les informations des fournisseurs gérés dans l'application VENUS. Il a le format suivant :

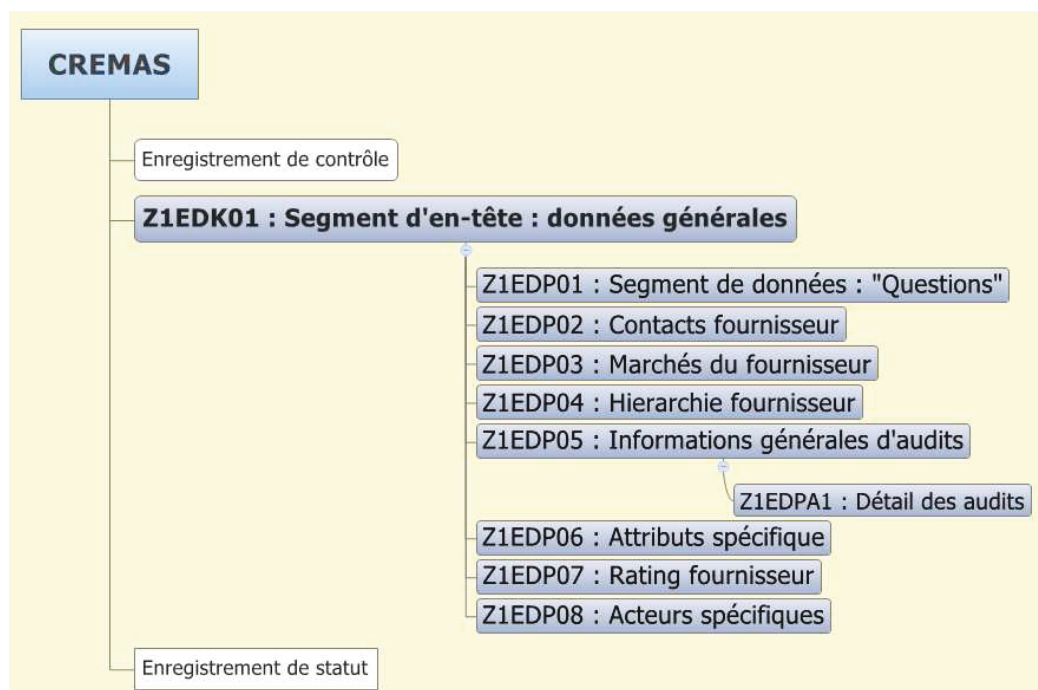


Figure 43 - Modèle d'un iDOC de type fournisseur

Objet de la demande

Le management des achats de Schneider-Electric a décidé de modifier la classification de ses fournisseurs en leur attribuant un statut « Préféré » ou non en fonction de leur positionnement. Par ailleurs, les acheteurs ont entrepris de renseigner 5 nouvelles informations supplémentaires et de les faire transiter via SSRM pour que ces caractéristiques ajoutées soient disponibles comme axes d'analyses dans le reporting.

Cette demande par sa nature impliquait l'intégralité du flux de données de bases fournisseurs de l'application mère (VENUS) jusqu'aux états d'analyse.

La figure 44 représente l'ensemble des impacts techniques de cette demande :

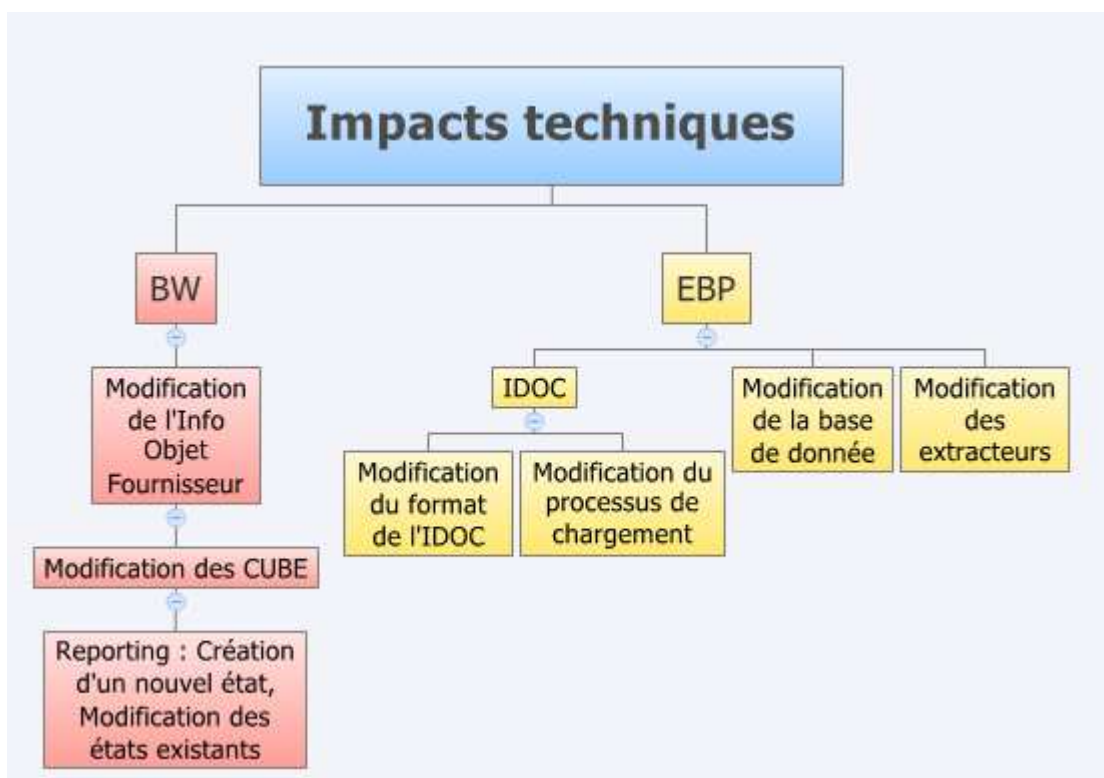


Figure 44 - Impacts techniques de la demande

Sur l'application BW, il fallait modifier la caractéristique « Fournisseur » pour ajouter les nouveaux attributs à stocker mais également modifier les Cubes pour leur permettre d'utiliser ces nouveaux attributs comme axes d'analyse. Enfin, il fallait ajouter ces nouvelles caractéristiques dans les états de reporting.

Côté EBP, il était nécessaire de modifier le format de l'iDOC pour recevoir ces données. Un nouveau segment de données générales a été ajouté (voir figure 45). Il fallait aussi ajouter de nouvelles colonnes dans les tables de données existantes pour stocker les nouveaux champs et enfin modifier le programme de chargement des iDOCs dans l'application.

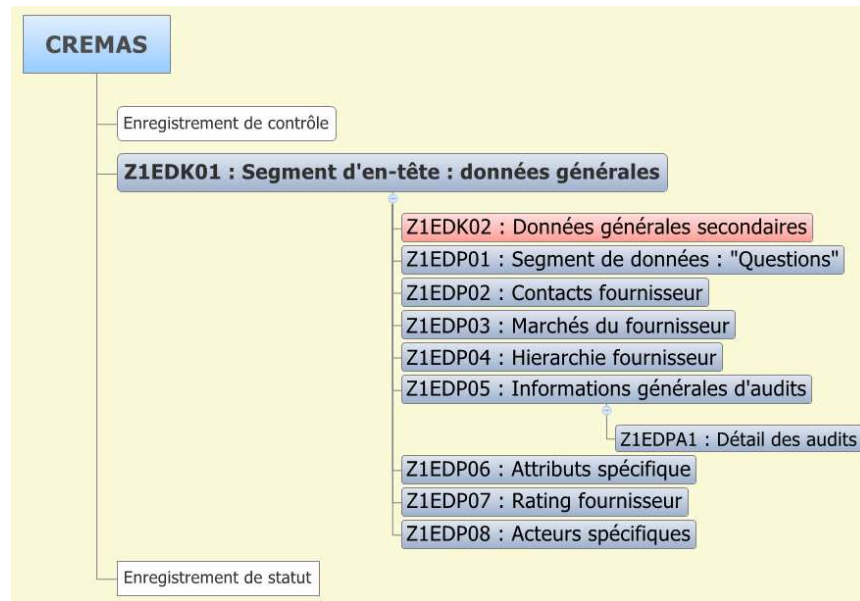


Figure 45 - Structure de l'iDOC modifiée

L'extracteur utilisé pour transférer les données de l'application EBP vers BW a également été modifié pour permettre le transfert des nouveaux champs d'un système à l'autre.

Traitement de la demande

Les demandes de maintenance évolutive dont la charge estimée est supérieure à 20 jours/hommes sont traitées sous forme de projet. C'est le cas de ce changement qui a été estimé à environ 60 jours de charges.

Dans le cadre du contrat de gestion d'application opéré pour Schneider-Electric, le traitement sous forme de projet implique la mise en place de dispositions particulières :

- Revue d'engagement interne Capgemini présentant le projet, ses aspects financiers et la gestion des risques.
- Rédaction d'un ordre de travail (document contractuel entre le client et Capgemini) qui comprend notamment les parties suivantes :
 - Description du projet.
 - Liste des livrables et des critères d'acceptation de ces livrables.
 - Planning d'implémentation.
 - Gestion des risques.
 - Informations financières concernant la charge et la facturation.
- Reporting hebdomadaire de l'avancement du projet à destination du management
- Revue de clôture externe, avec Schneider-Electric, pour résumer les points forts et les points faibles du projet et recueillir l'appréciation du client sur le déroulement du projet.
- Revue de clôture interne pour évaluer la réussite du projet et capitaliser sur les problèmes rencontrés et les solutions trouvées.

Ces tâches sont menées par un chef de projet Capgemini, nommé pour l'occasion. Suivant la configuration et la taille des chantiers, les personnes nommées chefs de projets font soit partie de l'équipe, soit ils sont désignés parmi un pool de collaborateurs formant la cellule gestion de projet.

Dans le cas présent, j'ai été désigné chef de projet.

Organisation projet

Pour mener à bien ce projet, nous avons mis en place certains des principes de l'agilité vus plus tôt dans ce mémoire.

Un des risques majeurs identifiés sur le projet était la complexité des modifications engendrées sur les différents objets, notamment côté EBP. Il était donc primordial de lever ce risque au plus tôt dans le développement, de manière à avoir suffisamment de marge pour réagir en cas de problème.

Le second risque important était l'intégration avec l'application amont dont les modifications étaient prises en charge par une équipe d'une société différente, sous pilotage de Schneider-Electric.

Pour appliquer le principe de livraisons plus fréquentes sur des volumes plus petits, le découpage du planning projet s'est donc fait autour de deux itérations :

- La mise en place du cœur de l'évolution, c'est-à-dire la modification du format de l'iDOC, des programmes de chargement et des objets de stockages (base de donnée EBP ou objets BW).

- L'évolution des états de reporting

Second aspect issu des principes d'agilité, le client a été intégré très tôt dans le développement de l'évolution. En particulier, il a pu avoir une vision des évolutions dès les tests unitaires.

Concernant l'organisation de l'équipe en interne nous avons mis en place une réunion quotidienne de 10 minutes avec les développeurs pour partager sur l'avancement, les problèmes rencontrés et planifier les tâches de la journée (principe du daily stand-up meeting).

3.2.4. Deuxième exemple de demande : Modification des volumes achat

Cette partie met en lumière une autre évolution faisant partie du catalogue des changements et qui a appliqué certaines recommandations des méthodes Agile.

Contexte de la demande

La restitution des volumes d'achat fournisseur est une part importante de l'application SSRM.

Les données sont collectées sur 19 ERP différents correspondant à des zones géographiques et utilisant chacun des systèmes locaux de technologies hétérogènes (SAP, BAAN et Oracle Applications). Les chiffres sont extraits sous forme de fichiers plats qui sont consolidés manuellement par le métier et envoyés de manière mensuelle à notre équipe pour chargement dans l'application. Les 19 fichiers sont d'abord concaténés en un fichier global qui contient en moyenne 100.000 lignes correspondant au volume d'achat effectué chez un fournisseur pour un marché et un mois donné (figure 46).

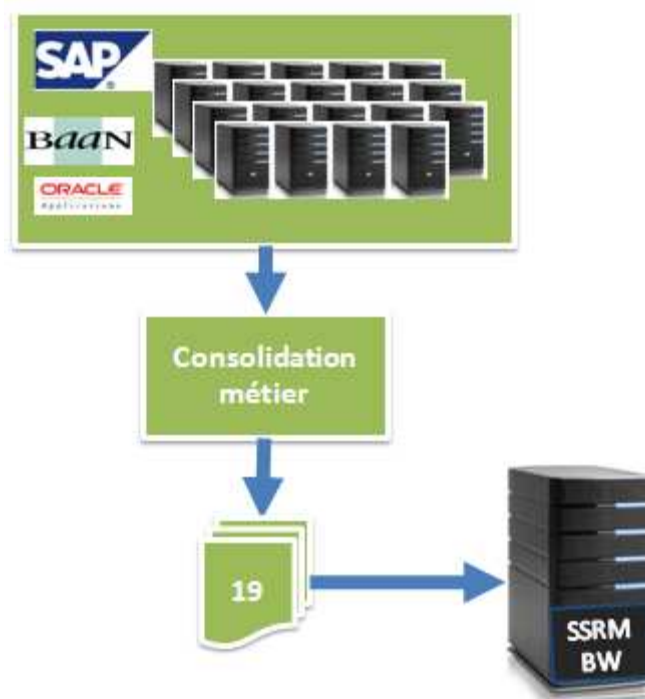


Figure 46 - Flux de donnée pour les volumes d'achats

Objet de la demande

Ce processus de chargement manuel génère un nombre important d'erreurs de données et nécessite en moyenne un ou deux rechargements mensuels pour corriger des données ayant été mal extraites ou mal consolidées par le métier. Afin de garantir l'intégrité des données dans l'application SSRM, il est nécessaire de recharger le fichier concaténé dans son ensemble. Pour chaque correction à effectuer, le métier doit donc mettre à jour une ou plusieurs lignes dans un fichier plat volumineux. Cette action est très consommatrice de temps pour le métier par rapport à la faible quantité d'informations à modifier. Le client nous a donc demandé d'imaginer une solution permettant de mettre à jour les données plus facilement sans avoir à manipuler tout le fichier et garantissant la qualité des données.

Traitement de la demande

Cette demande avait déjà été formulée plusieurs fois auparavant par le client mais sans jamais aboutir car les solutions envisagées auparavant prévoyaient toutes de modifier les données directement dans les objets BW ce qui est fortement déconseillé par l'éditeur. La demande avait donc jusqu'alors toujours été refusée car non réalisable sans mettre en péril la stabilité de l'application.

Comme le prévoit le modèle « One team », l'équipe de développement, composée de deux développeurs indiens, a été intégrée dès le début de l'analyse et des spécifications, par le biais d'un atelier de design réunissant le responsable client et le responsable de solution Capgemini en plus des développeurs. C'est dans cet atelier que les développeurs ont proposé une solution technique dont le principe était de créer une table intermédiaire sur SAP BW ainsi qu'un programme d'accès en modification à cette table.

Cette table avait pour but de stocker les données originales et de permettre un accès ergonomique aux enregistrements, grâce notamment à un écran de sélection par mois et par numéro de fournisseur (figure 47).

Figure 47 - Ecran de sélection de l'interface

La modification des données était ensuite possible via un accès en modification à la table qui permet de changer le montant des achats fait pour un fournisseur sur une période et un marché (figure 48).

RECORD_...	CALMONTH	SENDER	RECTYPE	EPCCODE	DUNSNUM	FAMCODE	AMOUNT	CURRENCY	PAYTERM	K_DELAIST
6	08.2012	AOD	S3	DIG	654525187	60D	000000000000000002	EUR		0
7	08.2012	AOD	S3	DIG	654525187	63K	000000000000000088	EUR		0
8	08.2012	AOD	S3	DIG	654525187	63N	000000000000000034	EUR		0
9	08.2012	AOD	S3	DIG	654525187	63R	000000000000000786	EUR		0
10	08.2012	AOD	S3	DIG	654525187	64A	0000000000109806	EUR		0

Figure 48 - Modification des données

Il fallait ensuite modifier la partie de chargement vers les objets BW à partir de cette nouvelle table et non plus du fichier. Pour ce faire, l'équipe de développement a également proposé un processus automatique permettant de réduire la durée des actions nécessaires à un rechargement.

Cette solution respectait donc les principes du Lean d'élimination des gaspillages en automatisant une tâche dont l'exécution manuelle n'apporte pas de valeur au client.

Même si le client a accepté le principe de la solution, il n'était pas familier avec l'interface utilisateur BW. De plus, ce type de développement étant relativement nouveau pour nos développeurs, il a été décidé, toujours selon les principes Agile, de construire un prototype fonctionnel en environnement de développement afin de valider la solution avec le client.

Le prototype a donc été développé pour matérialiser le cœur de la solution, c'est-à-dire l'accès en modification à la table contenant les données d'achat. Des accès à l'environnement de développement ont été fournis au client pour qu'il puisse tester et approuver le prototype et formuler quelques retours sur l'ergonomie de la transaction.

Sur cette base, le reste de la solution moins critique et moins visible pour le client a été chiffré et planifié dans le cadre de la release numéro 4 (Novembre 2012).

Afin de suivre l'avancement des travaux, des réunions régulières (trois fois par semaine) ont été mises en place avec l'équipe de développement. Ces réunions ont repris le principe des réunions Lean en étant très courtes et focalisées sur les progrès et les problèmes rencontrés depuis la dernière instance.

Les trois éléments mis en place (validation du prototype par le client, implication de l'équipe de développement dès la phase de design, suivi fréquent de l'avancement) nous ont permis de réduire au minimum les risques liés à cette demande de changement et de livrer une évolution qui apporte de la valeur au client tout en respectant le triptyque : délais, coûts et qualité.

3.3.Résultats méthodologiques

A fin Septembre, les différents indicateurs nous ont montré que le modèle organisationnel mis en place avait rempli ses objectifs sur le périmètre. Dès le mois d'Octobre 2012, le modèle a été déployé sur le second domaine du service : le domaine Finance avec l'objectif d'en faire de même sur la dernière partie « AM Mutualisées » début Janvier 2013.

3.3.1. Modèle de pilotage de service

L'objectif principal de ce mémoire était de pouvoir modéliser le pilotage d'un service mutualisé en formalisant les pratiques existantes et en incorporant d'autres pratiques issues de la littérature. Cette partie présente le modèle tel qu'il est défini à la fin du mémoire. L'intention étant de garder un modèle vivant capable d'évoluer et d'être adapté en fonction des nouvelles idées, des retours d'expériences futures ou des avancées technologiques (nouveaux outils).

Le modèle est découpé en 5 parties : les instances, les outils, le reporting, la gestion culturelle et la philosophie.

Les instances

Les instances retenues sont au nombre de quatre.

1. Le comité technique

Cette réunion hebdomadaire permet de faire le point avec le client sur l'avancement des tâches en cours et de se synchroniser sur les priorités. Ce comité se situe à un fort niveau de détail permettant de partager des problématiques très opérationnelles. Il n'existe pas vraiment d'instances équivalentes dans les méthodes Agile ou le Lean mais par contre il correspond à la volonté de ces méthodes d'être en proximité du client et d'échanger fréquemment sur les priorités et les points en cours.

2. Le stand-up meeting

Il s'agit de la réunion hebdomadaire définie dans la partie 3.3.1. qui correspond à la mêlée quotidienne de SCRUM ou au daily stand-up meeting du Lean. Cette réunion est la propriété de l'équipe qui doit s'en servir pour partager le déroulement de la période écoulée, prioriser la période à venir et faire remonter ses problèmes.

3. Le comité de maintenance

Le comité de maintenance mensuel est décrit en partie 2.2.2 dans le pilotage des comptes. Cette instance est nécessaire pour vérifier la progression de la période écoulée et planifier la stratégie pour la période à venir. Il permet également de balayer les problèmes ou points de blocages nécessitant des décisions managériales ou commerciales.

4. La rétrospective de release

Cette instance est l'équivalent de la rétrospective de sprint dans SCRUM. Le but de cette réunion est d'analyser le déroulement de la dernière release et de capitaliser sur les erreurs et les réussites afin de déterminer les cibles d'améliorations potentielles.

Au contraire de SCRUM, nous avons mis en place une telle réunion sans la participation du client afin de laisser l'équipe de développement s'exprimer pleinement et sans retenue. La consolidation avec les retours clients a été faite dans le cadre des comités de maintenance.

Les outils

1. Gestion des tests

Nous avons vu dans le Lean (partie 2.4.2 sur la validation de la qualité) la nécessité d'industrialiser les tests. Dans le cadre de l'ALM, nous avons pu nous appuyer sur l'utilitaire HPQC. Mais même dans un contexte ne bénéficiant pas de l'apport d'un tel outil, il est important de mettre en place ce processus. Avec un peu d'imagination, un simple fichier Excel peut servir de base à la capitalisation et apporter des gains sur les campagnes de tests futures.

2. Gestion de la connaissance

Ici l'apport d'un outil comme Teamforge est primordial pour garantir la cohérence et la traçabilité de la documentation. Pour autant, l'outil doit être utilisé avec une philosophie partagée par tous les membres de l'équipe.

3. Cartographie des processus

La cartographie des processus est un élément essentiel d'une équipe mutualisée pour assurer l'uniformité des modes de fonctionnement à travers tous les domaines. Cette uniformité est nécessaire pour éviter des gaspillages et permettre au groupe d'apprendre plus facilement.

Le reporting

1. Suivi des budgets

Les états budgétaires permettent de suivre l'aspect financier du domaine et de vérifier l'adéquation entre les moyens déployés et les besoins du client.

2. Suivi des imputations

Ce suivi donne une vision du temps réellement passé par les collaborateurs sur leurs diverses tâches. C'est un élément de contrôle ainsi qu'une base fiable pour la planification des ressources.

3. Suivi des indicateurs

Ce reporting est une aide pour l'équipe afin de mesurer qualitativement et quantitativement ses résultats. De plus, le suivi des indicateurs permet d'identifier les points sensibles nécessitant une attention particulière.

La philosophie

Le modèle n'est efficace que par la philosophie qui l'anime. Il est donc essentiel d'intégrer cette philosophie comme un élément à part entière, indissociable de l'ensemble. Elle est issue des bonnes pratiques de l'Agile ou du Lean et de l'expérience de l'équipe accumulée au fil du projet.

Gestion culturelle

Afin de prendre en compte les différences culturelles de l'équipe, deux artéfacts ont été créés.

- La carte nationale
- La carte d'identité

La figure 49 illustre la modélisation de l'organisation mise en place pour l'équipe BW.

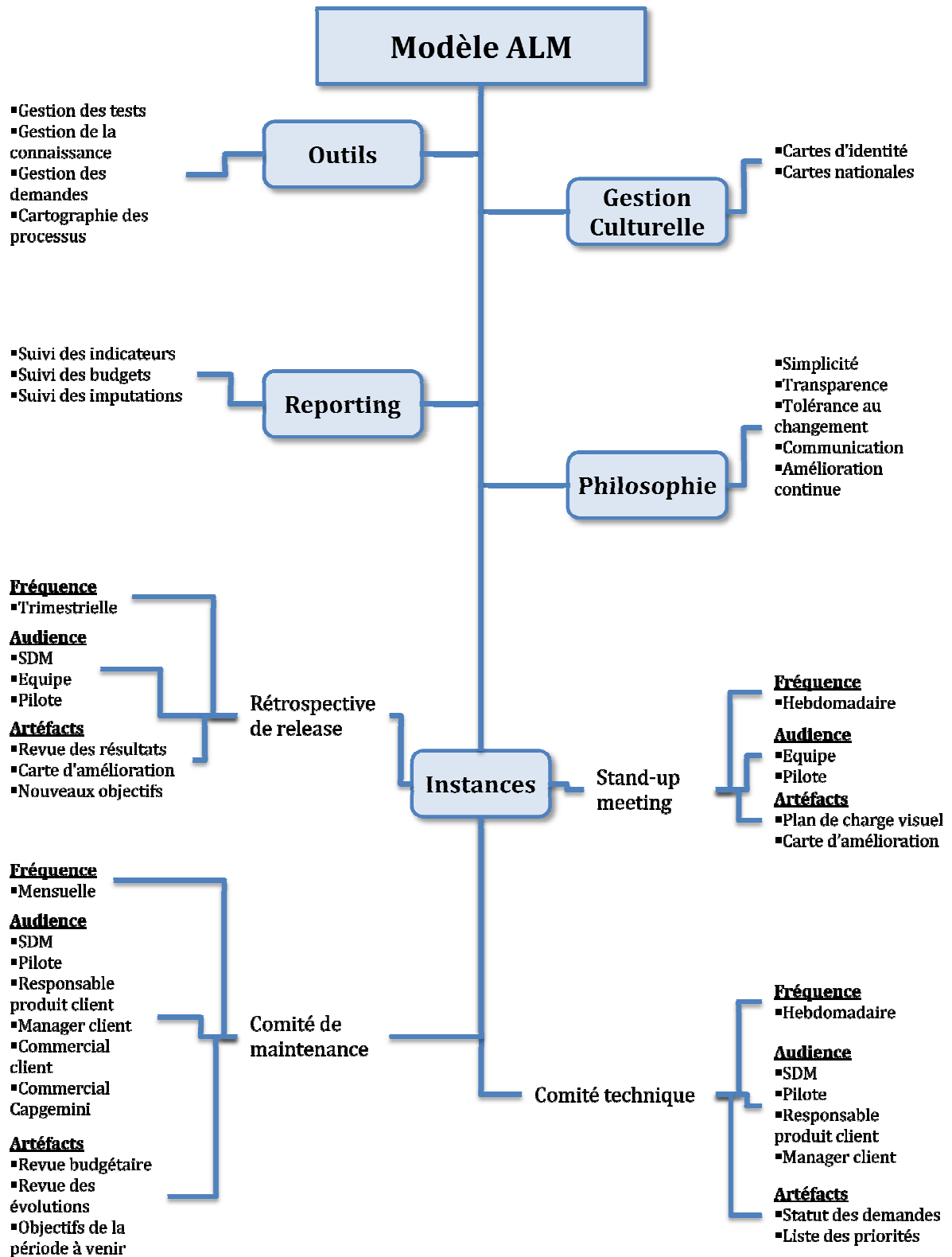


Figure 49 - Modèle d'organisation d'une équipe mutualisée

Conclusion

A ce jour, les implémentations existantes des méthodes Agiles ne couvrent que des activités de développement et, même si SAP a introduit la méthode SCRUM pour des projets de déploiement depuis 2010, il n'existe pas de méthode spécifiquement définie pour la maintenance de solution ERP.

De même, il existe très peu de publications en France liées à l'application du Lean dans ce domaine, et sa progression dans le pays est freinée par la mauvaise presse dont il fait l'objet.

Enfin, il est toujours compliqué de mettre en place un nouveau modèle de pensée, car si un échec survient, il sera en premier lieu attribué à la nouvelle méthode et celle-ci aura toutes les chances d'être abandonnée. Il est donc particulièrement délicat à l'heure actuelle d'imposer ce modèle et il semble qu'il faudra encore attendre plusieurs années pour que les schémas classiques s'effacent.

Pourtant, les clients, les fournisseurs de services et leurs employés auraient tout autant intérêt à mettre en place ces démarches. En effet, il est évident que le monde de l'informatique a, par nature, besoin de flexibilité, de communication et de collaboration entre les clients et les fournisseurs. Dans un domaine où l'on peut facilement « déplacer l'immeuble d'un cm » à chaque instant, la raideur du modèle en cascade et ses nombreux passages de relais finira par être abandonnée, au terme d'une prise de conscience générale.

Etant moi-même convaincu de l'intérêt d'ajouter de l'agilité dans la maintenance applicative, ce mémoire m'a permis de mettre en pratique cette idée et de démontrer qu'elle pouvait être utile dans le cadre du centre de service BW.

L'extension du modèle créé pour l'équipe BW à un périmètre plus grand est une preuve de sa réussite. Celui-ci doit maintenant montrer son efficacité sur un domaine plus vaste. Pour y parvenir, le modèle devra sans doute être adapté et perfectionné, en conservant la stratégie des petits pas, plutôt que celle de la révolution, afin de faire rentrer progressivement la notion d'agilité dans les services de maintenance informatique. Il sera peut-être utile d'aller piocher dans d'autres philosophies, comme par exemple les organisations apprenantes.

L'objectif futur pour moi est de pouvoir disposer d'un modèle complet, configurable, et évolutif pouvant être déployé sur n'importe quelle gestion de parc applicative.

Bibliographie

Ouvrages imprimés

- BARRAND J., 2012. *Le manager agile*. 2^{ème} édition. Dunod, Paris, 185 p.
- CHAMFRAULT T., DURAND C., 2011. *Les services agiles et les processus*. Dunod, Paris, 363 p.
- GIORDANO J.L., 2006. *L'approche qualité perçue*. Editions d'Organisation, Paris, 320 p.
- HIBBS C., JEWETT S., SULLIVAN M., 2010. *L'art du Lean Software Development*. Dunod, Paris, 175 p.
- LONGEPE C., 2009. *Le projet d'urbanisation du SI : Cas concret d'architecture d'entreprise*. 4^{ème} édition. Dunod, Paris, 297 p.
- PILLOU J.F., 2006. *Tout sur les Systèmes d'Information*. Dunod, Paris, 179 p.
- POPPENDIECK M., POPPENDIECK T., 2003. *Lean Software Development: An Agile Toolkit*. Addison Wesley, 240 p.
- STIEN C., 2011. *Manager sans gaspillage*. Afnor Editions, Paris, 112 p.
- MEIER O., 2010. *Management Interculturel*. 4^{ème} édition. Dunod, Paris, 320 p.
- HOFSTEDE G., HOFSTEDE G.J., MINKOV M., 2010. *Cultures et organisations*. 3^{ème} édition. Pearson, Paris, 634 p.

Sites web

- 01 NET, 2012. 01net Entreprises : emploi, formation et conseils IT pro, [en ligne]. Disponible sur <http://pro.01net.com/editorial/562616/la-justice-reconnait-linfluence-du-Lean-management-sur-les-conditions-de-travail/> (consulté le 19/08/2012)
- AGILE ALLIANCE, 2012. Agile Alliance :: Home, [en ligne]. Disponible sur <http://www.agilealliance.org/the-alliance/the-agile-manifesto/the-twelve-principles-of-agile-software/> (consulté le 23/08/2012).
- AGILE SOFTWARE DEVELOPMENT, 2007. Agile Software Development | Better ways of developing software, [en ligne]. Disponible sur <http://agilesoftwaredevelopment.com/xp/values/respect> (consulté le 23/08/2012).
- AMBYSOFT, 2006. Ambysoft Home Page, [en ligne]. Disponible sur <http://www.ambysoft.com/unifiedprocess/agileUP.html> (consulté le 27/09/2012).

- BLUEFIN SOLUTIONS, 2012. Bluefin solutions, [en ligne]. Disponible sur http://www.bluefinsolutions.com/insights/blog/what_happens_when_SAP_meets_agile/ (consulté le 09/09/2012).
- CAPGEMINI, 2011. Collaborative Business Experience | Capgemini Worldwide, [en ligne]. Disponible sur <http://www.capgemini.com/insights-and-resources/by-publication/application-landscape-report-2011-edition/> (consulté le 23/08/2012).
- CAPGEMINI, 2012. Collaborative Business Experience | Capgemini Worldwide, [en ligne]. Disponible sur <http://www.fr.capgemini.com/ressources/publications/Lean-it/> (consulté le 23/09/2012).
- CENTRE ETUDE EMPLOI, 2006. Centre d'études de l'emploi, les politiques d'emploi et le marché du travail, [en ligne]. Disponible sur http://www.cee-recherche.fr/fr/doctrav/travail_conditions_sante_europe_73.pdf (consulté le 26/08/2012)
- CGT ATOS, 2011. CGT Atos, [en ligne]. Disponible sur <http://www.cgt-atos.org/spip.php?article118> (consulté le 02/09/2012).
- CGT THALES, 2008. CGT Thales, [en ligne]. Disponible sur <http://www.cgt-thalospace.fr/public/tracts/tract-02oct2008.pdf> (consulté le 02/09/2012).
- EFFITIC, 2012. effitic | make IT simple, [en ligne]. Disponible sur <http://www.effitic.com/Lean-centre-service-ministere/> (consulté le 07/09/2012).
- GARTNER, 2010. Technology Research | Gartner Inc. , [en ligne]. Disponible sur <http://www.gartner.com/it/page.jsp?id=1642714> (consulté le 07/08/2012).
- GARTNER, 2011a. Technology Research | Gartner Inc. , [en ligne]. Disponible sur <http://www.gartner.com/id=1823015> (payant, consulté le 17/10/2012).
- GARTNER, 2011b. Technology Research | Gartner Inc. , [en ligne]. Disponible sur <http://www.gartner.com/id=1867414> (payant, consulté le 17/10/2012).
- LA RIPOSTE, 2007. Journal communiste : La Riposte, [en ligne]. Disponible sur <http://www.lariposte.com/le-Lean-manufacturing-ou-l,756.html> (consulté le 23/09/2012)
- LE MONDE INFORMATIQUE, 2007. Actualité informatique, actualité des entreprises, information économique, [en ligne]. Disponible sur <http://www.lemondeinformatique.fr/actualites/lire-sap-rachete-business-objects-pour-4-8-milliards-d-euros-24203.html> (consulté le 17/09/2012).
- LES ECHOS, 2010. Bienvenue | Le Cercle Les Echos, [en ligne]. Disponible sur <http://lecercle.lesechos.fr/entreprises-marches/management/221131110/le-Lean-facteur-de-progres-social>, (consulté le 02/09/2012).

MCNEXT, 2010a. SSII spécialisée SharePoint, DotNET, Silverlight, Biztalk et décisionnel Microsoft, [en ligne]. Disponible sur http://www.mcnnext.com/pole_dotnet/projet-agile/Pages/SCRUM.aspx (consulté le 03/09/2012).

MCNEXT, 2010b. SSII spécialisée SharePoint, DotNET, Silverlight, Biztalk et décisionnel Microsoft, [en ligne]. Disponible sur http://www.mcnnext.com/pole_dotnet/projet-agile/Pages/XP-les-bonnes-pratiques.aspx (consulté le 03/09/2012).

ORYX CONSEIL, 2011. La performance par le plaisir de réaliser ensemble, [en ligne]. Disponible sur <http://www.oryxconseil.com/actualites-oryx-conseil/idees-clefs/entry/Lean-management--facteur-ou-reducteur-de-stress.html> (consulté 21/09/2012).

SCRUM, 2012. SCRUM.org > Home, [en ligne]. Disponible sur <http://www.SCRUM.org/SCRUM-Guides> (consulté le 15/09/2012).

STANDISH GROUP, 1994. Project Smart, [en ligne]. Disponible sur <http://www.projectsart.co.uk/docs/chaos-report.pdf> (consulté le 29/08/2012).

VERSIONONE, 2011. VersionOne, [en ligne]. Disponible sur http://www.versionone.com/state_of_agile_development_survey/11/ (consulté le 27/08/2012).

WEB BUILDER ZONE, 2012. Web builder zone, [en ligne]. Disponible sur <http://css.dzone.com/articles/Lean-tools-set-based> (consulté le 23/09/2012).

WIKIPEDIA, 2012a. Wikipédia, l'encyclopédie libre, [en ligne]. Disponible sur <http://fr.wikipedia.org/wiki/Inde> (consulté le 11/08/2012)

WIKIPEDIA, 2012b. Wikipédia, l'encyclopédie libre, [en ligne]. Disponible sur http://fr.wikipedia.org/wiki/Lean_IT (consulté le 16/09/2012).

WORLD VALUE SURVEY, 2012. World Values Survey, [en ligne]. Disponible sur <http://www.worldvaluessurvey.org> (consulté le 03/11/2012).

Illustrations

Figure 1 - Modèle conceptuel du mémoire	12
Figure 2 - Fusion des services.....	14
Figure 3 - Organigramme général Capgemini.....	20
Figure 4 - Modèle traditionnel.....	22
Figure 5 - Modèle d'équipes mutualisées.....	22
Figure 6 - Organigramme ALM SAP Montbonnot.....	23
Figure 7 - Exemple de tableaux de bord SLA	24
Figure 8 - Processus de gestion des changements.....	27
Figure 9 - Processus de gestion des incidents	28
Figure 10 - Processus de gestion des problèmes	28
Figure 11 - Modèle traditionnel d'offshore.....	30
Figure 12 - Modèle "One Team"	31
Figure 13 - Cycle en cascade.....	32
Figure 14 - Utilisation des méthodes Agiles (Versionone, 2011)	34
Figure 15 - Cycle de vie SCRUM (MCNext, 2010a)	36
Figure 16 - Cycle de vie XP (MCNext, 2010b)	39
Figure 17 - Origine des gaspillages dans la gestion des systèmes d'information.....	49
Figure 18 - Tableau Kanban de gestion des demandes.....	53
Figure 19 - Cycle d'amélioration continue.....	62
Figure 20 - Comparaison des aspects productifs et qualitatifs du cycle de développement	62
Figure 21 - Plan projet	63
Figure 22 - Evolution du budget SSRM.....	66
Figure 23 - Nouveau schéma d'organisation.....	67
Figure 24 - Tableau visuel de gestion du plan de charge	70
Figure 25 - Carte de l'amélioration continue	71
Figure 26 - Tableau des objectifs 2012 (Quatrième trimestre).....	72
Figure 27 - Processus de gestion des demandes	74
Figure 28 - Processus de gestion des demandes de support.....	76
Figure 29 - Les trois niveaux de la programmation mentale d'un individu (Hofstede et al, 2010)	77
Figure 30 - Résumé des indices culturels pour la Chine, la France et l'Inde	84
Figure 31 - Carte culturelle de l'Inde.....	85
Figure 32 - Carte d'identité.....	86
Figure 33 - Outil de gestion documentaire	87
Figure 34 - Modèle de kit de nouvel arrivant.....	89
Figure 35 - Graphique de production pour le client Schneider-Electric	91
Figure 36 - Nombre de retours sur recette.....	92
Figure 37 - Evaluation mensuelle de la production	93

Figure 38 - Progression des imputations de support sur l'année.....	94
Figure 39 - Structure d'un objet de stockage de donnée BW.....	97
Figure 40 - Schéma en étoile d'un cube BW	97
Figure 41 - Exemple d'état édité par BW.....	99
Figure 42 - Architecture des interfaces de l'application SSRM	102
Figure 43 - Modèle d'un iDOC de type fournisseur.....	103
Figure 44 - Impacts techniques de la demande	104
Figure 45 - Structure de l'iDOC modifiée.....	105
Figure 46 - Flux de donnée pour les volumes d'achats	107
Figure 47 - Ecran de sélection de l'interface	108
Figure 48 - Modification des données.....	108
Figure 49 - Modèle d'organisation d'une équipe mutualisée	112

Tableaux

Tableau I - Synthèse de l'application du modèle ALM par client	31
Tableau II - Synthèse de l'agilité dans contexte du centre de service	46
Tableau III - Liste des indicateurs	65
Tableau IV - Indice de distance hiérarchique (Hofstede et al, 2010)	78
Tableau V - Indice d'individualisme d'une société (Hofstede et al, 2010)	79
Tableau VI - Indice de masculinité d'une société (Hofstede et al, 2010)	81
Tableau VII - Indice de contrôle de l'incertitude (Hofstede et al, 2010)	82
Tableau VIII - Indice d'orientation à long terme (Hofstede et al, 2010)	83
Tableau IX - Suivi des documents de capitalisation	94
Tableau X - Parts de marché pour les éditeurs de logiciels décisionnels en 2010 (Gartner, 2010)	95
Tableau XI - Liste des demandes traitées dans l'année pour Schneider-Electric	100
Tableau XII - Liste des demandes traitées dans l'année pour Rio-Tinto.....	101

Résumé

Implémentation du modèle ALM Capgemini dans un service SAP-BW

Mémoire d'Ingénieur C.N.A.M., Lyon 2012

RESUME

Afin de mieux gérer notre service de maintenance applicative opérant sur plusieurs applications SAP-BW, il a été décidé de regrouper les consultants dans une équipe mutualisée chargée de gérer l'ensemble des applications et de se mettre en capacité d'intégrer de nouveaux périmètres.

La difficulté de cette mission était d'industrialiser la production de tous les domaines pour partager à la fois les connaissances et les charges de travail de chaque application tout en maintenant le niveau de service.

Pour ce faire, un modèle d'organisation a été défini en s'inspirant des méthodes Agile et Lean tout en capitalisant sur les outils et pratiques présentes chez Capgemini. Aujourd'hui, ce modèle est inscrit dans une dynamique d'amélioration continue afin de le perfectionner et de le garder à jour.

Mots clés: SAP, BW, Maintenance applicative, Agile, Lean, amélioration continue.

SUMMARY

To enhance the management of our application maintenance team, operating on several SAP-BW applications, it was decided to merge the consultants in a mutualised team responsible for managing the entire domain and being able to integrate new perimeters.

The difficulty of this mission was to industrialize the production of all fields to share both knowledge of each application and workloads while maintaining service levels and optimizing costs.

To do this, an organization model was defined, based on the existing tools and practices used in Capgemini, and enhanced with parts of Lean and Agile methods. In addition, this model was set-up in a continuous improvement process in order to develop it and keep it up to date.

Key words: SAP, BW, Software maintenance, Agile, Lean, continuous improvement.