

PROJET D'ANALYSE DU NIVEAU DE MATURITÉ
D'UN PROCESSUS DE MAINTENANCE LOGICIELLE
EN ENTREPRISE

par

ERIC TREMBLAY

RAPPORT DE PROJET
PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE
LA MAÎTRISE DE GÉNIE LOGICIEL

MONTRÉAL, LE 2017/08/24

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC



Eric Tremblay, 2017



Cette licence [Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/) signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette œuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'œuvre n'ait pas été modifié.

PRÉSENTATION DU JURY

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Alain Abran, directeur de projet,
Département de génie logiciel et des TI à l'École de technologie supérieure

Mme Sylvie Trudel, codirectrice de projet,
Département d'informatique à l'Université du Québec à Montréal

M. Abdelouahab Gherbi , président du jury
Département de génie logiciel et des à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 20017/08/25

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je tiens à remercier mes proches et mes amis dans le support qu'ils m'ont apporté au cours de ce périple universitaire. Je tiens à remercier les professeurs et les chargés de cours d'avoir partagé leur grand savoir ainsi que leurs expériences. Je tiens à remercier ceux qui ont bien voulu me recommander lors de mes soumissions de candidature à la maîtrise de génie logiciel. Je tiens aussi à remercier M. Alain Abran d'avoir dirigé ce projet et Mme Sylvie Trudel. Je tiens aussi à remercier particulièrement mes parents et mon frère (Judes Tremblay, Lise Simard et Jean-Daniel Tremblay), la nouvelle famille Therrien (composé de ma sœur, son conjoint et leurs deux beaux enfants) ainsi que Mme Bendjina Charles pour son sourire, son rire et ses paroles positives qui me permirent de ne pas abandonner dans les moments de doutes que j'ai pu avoir au cours de cette aventure. Ce ne fut pas de tout repos que de concilier études et travail : ça s'est avéré une expérience toute aussi enrichissante qu'éprouvante. Avant d'entamer mes études universitaires, il m'arrivait d'avoir des discussions sur l'université avec des proches qui avaient fait des études universitaires. Lors de ces discussions, j'en arrivais souvent à essayer de mettre en perspective le défi que l'université représente avec des expériences d'études que j'avais faites dans le passé. La plupart de ses discussions se terminaient par : "Pour savoir ce que c'est, tu dois aller à l'université!". Il s'est avéré que vous aviez tous raison : je tiens à vous remercier de m'avoir rabroué aussi souvent, ces quelques gifles à mon orgueil m'ont motivé et maintenant je comprends de quoi vous vouliez me parler.

Merci !

PROJET D'ANALYSE DU NIVEAU DE MATURITÉ D'UN PROCESSUS DE MAINTENANCE LOGICIELLE EN ENTREPRISE

ERIC TREMBLAY

RÉSUMÉ

Ce projet de 15 crédits sur le processus de maintenance logicielle en entreprise a été réalisé suite à une période d'observation du processus s'étalant sur une année et ayant comme objectif de trouver une solution à une problématique identifiée et pour laquelle l'entreprise n'était pas encore parvenue à trouver de solution au moment de débiter ce projet., Suite à une période d'observation en entreprise, l'information recueillie a été utilisée pour analyser le processus dans son ensemble pour parvenir à comprendre et à expliquer pourquoi les solutions proposées pour résoudre la problématique avortaient et, par la suite, proposer une solution pour parvenir à résoudre la problématique.

Mots-clés : Maintenance du logiciel, Processus, Observation en entreprise

TABLE DES MATIÈRES

Table des matières

INTRODUCTION.....	1
CHAPITRE 1 : ÉTAT DES LIEUX.....	3
1.1. Mise en contexte de l'entreprise.....	3
1.2. Culture d'entreprise.....	5
1.3. Cycle de vie de l'équipe de développement WEB.....	6
1.4. Mise en contexte du sujet à l'étude.....	9
1.5. Problématique inhérente à l'équipe.....	11
1.6. La courbe des cas de maintenance tend vers la hausse.....	14
1.7. Problématique : Pourquoi la courbe de maintenance continue de croître?.....	15
1.8. Modèle de maintenance.....	15
1.9. Maintenance.....	16
1.10. Les rôles de la maintenance logicielle chez ACME.....	18
1.11. Architecture technologique de l'environnement maintenance.....	23
1.12. Proposition d'une approche méthodologique.....	24
CHAPITRE 2	25
REVUE DE LA LITTÉRATURE.....	25
2.1. Introduction.....	25
2.2. La maintenance logicielle.....	25
2.3. La maintenance logicielle : ses définitions et son vocabulaire, selon la norme ISO 14764.....	26
2.4. Le processus de maintenance.....	27
2.5. Les types de maintenance.....	29
2.6. Le gaspillage dans un processus de maintenance.....	30
2.7. Éliminer le gaspillage.....	32
2.8. "Mapper" le processus.....	33
2.9. Le "benchmarking" et l'estimation.....	33
CHAPITRE 3	35
MÉTHODOLOGIE DE TRAVAIL.....	35
3.1. Mise en contexte du choix de la méthodologie de travail.....	35
3.2. La méthodologie de la maintenance logicielle.....	37
3.3. Le «Mapping» du processus de maintenance logicielle.....	38
3.4. Les sources de gaspillage en maintenance logicielle.....	40
3.5. Les mesures de la maintenance logicielle.....	44
CHAPITRE 4	47

LISTE DES TABLEAUX

	Page
Tableau 1.1 - Définition d'un cas de maintenance	10
Tableau 1.2. - Définition d'un projet technique.....	10
Tableau 1.3 - Définition d'une planification	10
Tableau 1.4 - Définition de l'amélioration continue	11
Tableau 2.1 - Terminologie employée en maintenance selon ISO 14764	26
Tableau 2-2 - Pyramide des standards [Abran-April, 2008].....	27
Tableau 2-3 - Type de maintenance selon ISO 14764.....	29
Tableau 2-4 - Les sept principes LEAN	32
Tableau 2-5 - Limiter le gaspillage	32
Tableau 2-6 - mesures collectées et utiles à la maintenance [April-Abran, 2008].....	33
Tableau 3.1- Questionnement soulevé au chapitre 1	35
Tableau 3.2 - Les fonctionnalités en Extra (Extra features)	41
Tableau 3.3 - Changement de tâches (Task switching)	41
Tableau 3.4 - L'attente (Waiting).....	42
Tableau 3.5 - La manutention (Motion)	43
Tableau 3.6 - Les activités de gestion (Management Activities)	44
Tableau 3.7 - Quelques exemples de mesures applicables en maintenance logicielle	45
Tableau 5.1 - Proposition d'un programme de mesure	99

LISTE DES FIGURES

	Page
Figure 1 - Structure de la direction d'ACME	4
Figure 2 - Structure organisationnelle du développement WEB	5
Figure 3 - Processus de développement de l'équipe WEB.....	7
Figure 4 - Observation du décompte des cas de maintenance observables avec le système TFS.	12
Figure 5 - Décompte du total de cas de maintenance par version et du cycle moyen (avec les données brutes).....	14
Figure 6 - Étapes principales du processus actuel de maintenance.....	16
Figure 7 - Processus de maintenance détaillée.....	17
Figure 8 - Méthodologie de travail.....	37
Figure 9 - Exemple de « Mapping » actuel chez ACME.....	39
Figure 10 - Exemple de « Mapping » du processus de maintenance chez ACME	40
Figure 11 - Mesure d'un élément du processus de maintenances logicielle.....	45
Figure 12 - Niveau 0 – résultats de la gestion du processus	51
Figure 13 - Niveau 0 - Résultats de la Gestion des évènements et des requêtes	52
Figure 14 - Niveau 0 - Résultats de l'ingénierie des livraisons	53
Figure 15 - Niveau 0 -Résultats du support à l'ingénierie des livraisons.....	55
Figure 16 - Résultats du niveau 1 - Process Management.....	58
Figure 17 - Résultat du Niveau 1 - Event/Request Management.....	60
Figure 18 - Résultat du Niveau 1 - Evolution Engineering	61
Figure 19 - Résultat du Niveau - Support to Evolution Engineering	63
Figure 20 - Résultat du Niveau 2 - Process Management.....	72

Figure 21 - Résultat du Niveau 2 - Event/Request Management.....	81
Figure 22 - Résultat du Niveau 2 - Evolution Engineering	87
Figure 23 - Résultat du Niveau 2 - Support to Evolution Engineering	93

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ACME	Nom d'emprunt anonyme de l'entreprise sujette à l'étude.
BD	Base de données
BIG-BANG	Changement majeur dans une entreprise, sans transition.
C2	Système de traitement des requêtes de support et utilisé avant l'intégration de TFS.
CDO	Consultation des données d'OPÉRA
CESAD	Type de compagnie dans le système d'ACME
CRM	Système de gestion des relations avec les clients « <i>Customer relationship management</i> »
CSP	Centre de service de paie
GCQ	Groupe de contrôle de la qualité
IP	Internet protocol
MEP	Mise en production
OPÉRA	L'un des systèmes patrimoniaux d'ACME, communément appelé « La calculatrice » au niveau technique.
PC	Ordinateur
SAAS	Software As A Service.
SAO	Soutien aux opérations
SAV	Soutien aux ventes
SQL	Type de langage transactionnel de base de données
SWAT	Équipe de support et maintenance composée de développeur sur une rotation fixe.
TF	Tests fonctionnels
TFS	Team Foundation Server
TI	Technologie de l'information
TI	Tests d'intégration
TU	Tests unitaire
WEB	Le « <i>World Wide Web</i> »: internet.

INTRODUCTION

Les TI font désormais partie de notre société depuis suffisamment longtemps pour qu'il puisse sembler impensable que notre société actuelle puisse s'en passer et le rythme de leurs évolutions creuse l'écart entre les décennies tellement rapidement qu'une façon de faire qui nous semblait incontestablement la meilleure il y a moins de 10 ans puisse nous paraître aujourd'hui archaïque. Pourtant, lorsque comparé aux autres domaines de l'ingénierie, le génie logiciel fait office de cadet de la famille. Le génie logiciel a émergé et évolué à un rythme qui tente de suivre l'évolution technologique du domaine.

Face à cette évolution rapide du domaine et à l'émergence d'entreprises qui peuvent passer du statut de « Start-up » à celui de multinationale de plusieurs dizaines de milliers d'employés en quelques années le génie logiciel se retrouve sous pression de devoir faire évoluer les processus afin qu'ils atteignent un niveau de maturité permettant de continuer à gérer avec efficacité les activités de développement, de maintenance, d'assurance qualité et de support du logiciel en fonction du contexte actuel de l'entreprise.

Suite à cette pression sur les processus du domaine, il arrive que le niveau de maturité d'un processus puisse ne plus répondre au contexte actuel de l'entreprise. Ce déphasage entre les façons de faire et les besoins actuels provoque parfois des situations où l'évolution du processus n'est plus sous contrôle et, dans le cas où des problèmes sont détectés, il devient alors difficile d'identifier et de réaliser les ajustements requis pour parvenir à redresser la situation.

Dans le cadre de ce projet de maîtrise, une perte de contrôle du processus de maintenance logicielle fût soumise à une observation et une analyse afin de comparer la situation aux façons de faire reconnues du domaine du génie logiciel dans l'intention de pouvoir cibler la source

ayant causée cette problématique et, par la suite, élaborer une proposition de solution ainsi que des recommandations pour parvenir à reprendre le contrôle du processus.

Ce projet se divise en quatre étapes soit :

1. L'observation du contexte d'entreprise et de la problématique identifiée,
2. L'analyse de la littérature traitant du domaine,
3. L'élaboration d'une méthodologie de travail, et
4. L'analyse de la problématique en fonction des observations de la littérature, puis d'une proposition de solution et de recommandations.

CHAPITRE 1 : ÉTAT DES LIEUX

1.1. Mise en contexte de l'entreprise

L'entreprise ACME est un fournisseur de services TI de grande envergure qui possède un large catalogue de services en technologie de l'information (TI) en mode *Software As A Service* (SAAS) qu'elle offre à plusieurs centaines de clients à travers le monde :

- Conseils stratégiques en technologie
- Intégration de systèmes
- Développement et gestion des applications
- Infrastructures
- Gestion des processus d'affaires et d'autres solutions d'affaires exclusives.

Les secteurs industriels couverts par les services de l'entreprise sont vastes, tels que:

- Les banques et les marchés de capitaux
- Les compagnies d'assurances
- Les commerces de détail
- Le domaine des communications
- Les gouvernements
- Les entreprises pétrolières et gazières
- Le domaine de la santé
- Le secteur manufacturier
- Les services publics
- Les services de transports et services postaux et logistiques.

La taille de cette entreprise a contribué à l'émergence d'une structure administrative dans laquelle chaque secteur mondial est sous la direction d'un président des opérations.

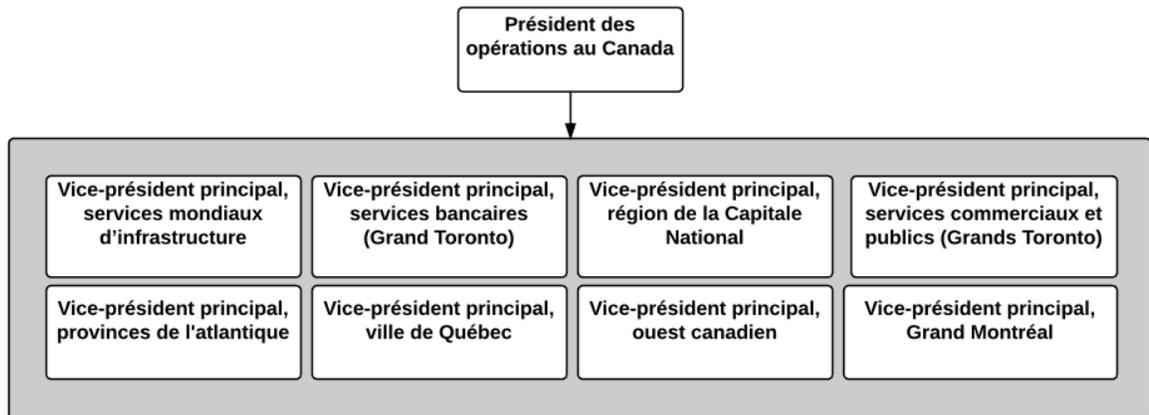


Figure 1 - Structure de la direction d'ACME

Le projet de maîtrise consiste en une période d'observation dans l'équipe de développement WEB du Centre de services de paie de la Vice-présidence principale de la région du Grand Montréal qui sera suivie d'une analyse afin de proposer une solution à la problématique.

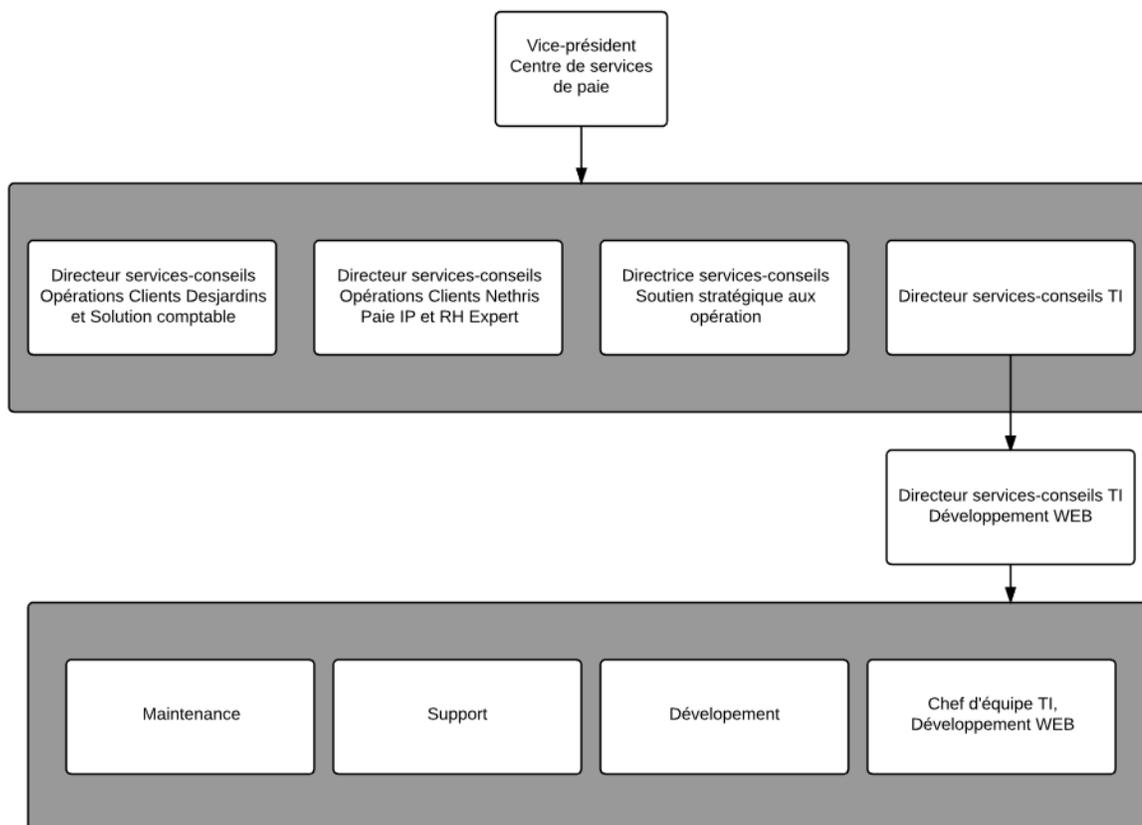


Figure 2 - Structure organisationnelle du développement WEB

1.2. Culture d'entreprise

L'entreprise ACME a une quarantaine d'années de présence sur le marché et sa culture d'entreprise s'est forgée au cours de cette période. La structure organisationnelle donne une bonne vue d'ensemble du niveau de hiérarchisation que l'on retrouve dans cette firme. Au fil des ans, l'entreprise ACME a fait l'acquisition de plusieurs nouveaux produits TI suite à l'achat d'autres entreprises. Cette stratégie de croissance est répandue dans l'industrie et semble devenir l'option la plus rapide pour une firme de cette envergure lorsqu'elle décide de croître. Considérant la taille qu'un projet doit avoir afin d'avoir un impact significatif sur le chiffre d'affaires de l'entreprise, il est effectivement plus rapide et plus fiable au niveau du retour sur investissement d'investir dans un projet dont la rentabilité peut être démontrée plutôt que de

s'aventurer dans des activités de développement de nouveaux projets qui s'échelonnaient sur plusieurs années avant de pouvoir espérer le rentabiliser.

Les unités d'affaires possèdent leur propre budget et évoluent en silo. Les grandes lignes des stratégies de l'entreprise sont communiquées du haut vers le bas et il est de la responsabilité des directeurs concernés de s'assurer qu'elles soient appliquées dans leurs unités d'affaires respectives.

Il existe beaucoup de départements, d'équipes et de projets dans cette entreprise : il serait trop long de tenter de tous les lister. La somme des projets de l'entreprise est composée de produits développés à l'interne et d'autres qui proviennent d'acquisitions. Dans le cadre de ce projet de maîtrise, le système maintenu par l'équipe en place provient d'une acquisition faite il y a un peu plus d'une dizaine d'années. L'entreprise de type PME qui avait conçu ce système a été rachetée par ACME et les employés furent intégrés dans l'entreprise ACME. Cette fusion provoqua sa part de choc culturel entre l'entreprise ACME et les employés de l'entreprise acquise.

Aux cours des dernières années, le mot d'ordre du passage vers les méthodologies Agile fût dicté par la haute direction de l'entreprise, ce qui confirma la transition vers des méthodologies de travail Agile.

1.3. Cycle de vie de l'équipe de développement WEB

Le cycle de vie qui a émergé au cours de la longue durée de vie de cette entreprise est de type cascade et celui de l'équipe obtenue suite à l'acquisition était de type cascade aussi.

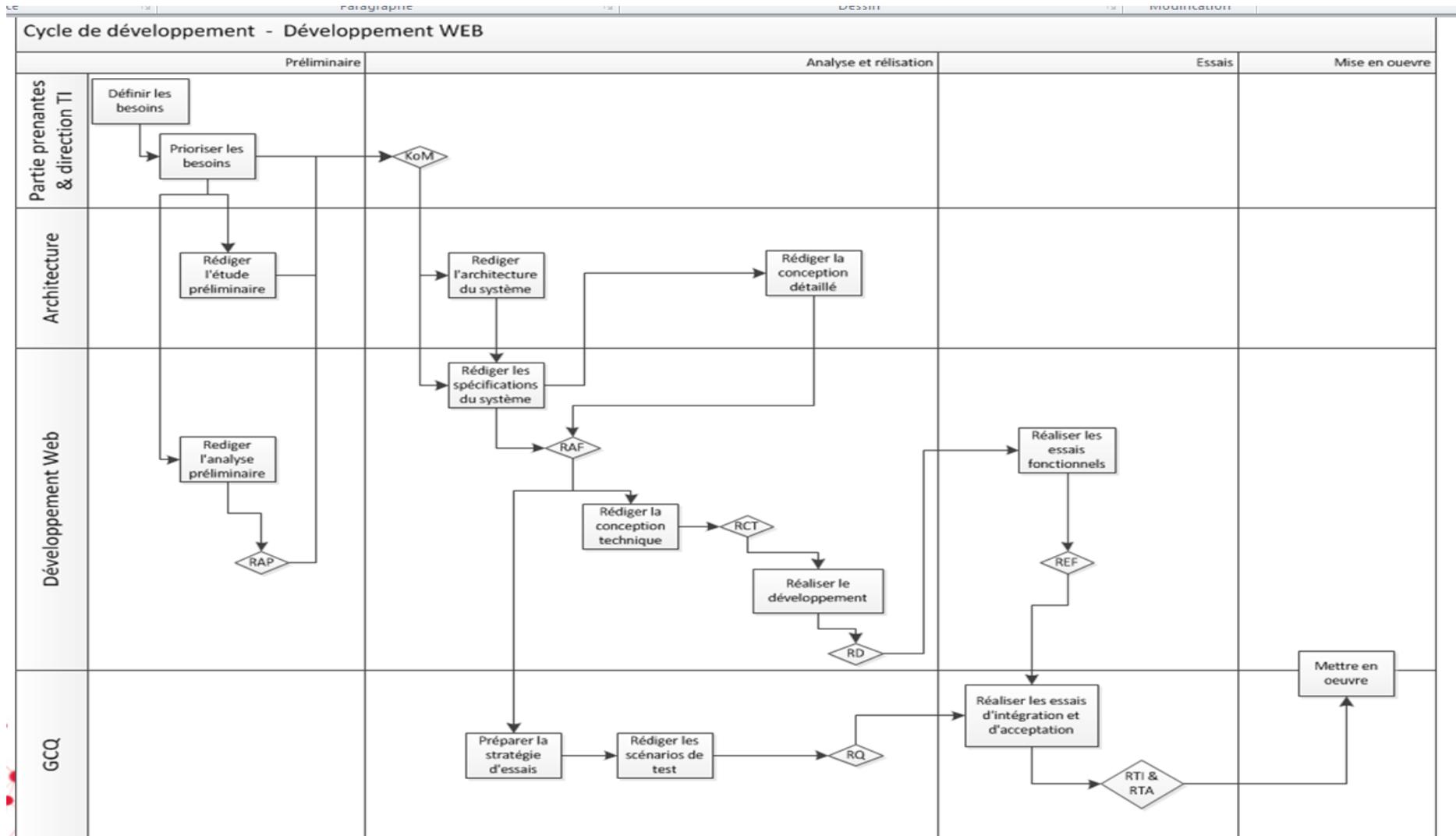


Figure 3 - Processus de développement de l'équipe WEB

Suite au démarrage de la transition vers l'Agilité la communication entre les différents acteurs du processus est fortement encouragée et de plus en plus incluse dans la réalisation des activités de développement, mais dans la pratique le processus est encore en cascade bien qu'il inclue aujourd'hui quelques bonnes pratiques Agile.

Chez ACME, le niveau d'analyse est encore perçu comme un niveau supérieur à celui de la réalisation. Que l'analyse soit simplement une étape qui en précède une autre dans le cycle de vie du système ne fait pas encore partie de la culture de l'équipe et les acteurs du niveau d'analyse sont encore perçus comme étant sur un piédestal par les autres membres des équipes.

Les acteurs de réalisation sont vus dans bien des cas comme de simples exécutants des consignes données par les acteurs du niveau d'analyse et leur rétroaction sur le niveau de qualité d'analyse nécessaire pour pouvoir parvenir à une réalisation efficace ne dépassent pas le stade de discussion entre programmeurs et est perçu comme de la perte de temps ou de manque de volonté lorsqu'il monte au niveau d'analyse.

Les activités d'assurance qualité sont encore principalement concentrées en fin de cycle de vie et elles n'ont pas encore une forte présence dans le processus lors de la planification et des analyses préliminaires. La collaboration entre l'équipe de réalisation/maintenance et d'assurance qualité se résume principalement à un échange de type « Partie de volley-ball » : les évolutions/correctifs réalisées par l'équipe de développement/maintenance sont livrées et les bogues provoqués par les modifications sont retournés par l'équipe d'assurance qualité à l'équipe de développement/maintenance. Ce cycle se poursuit jusqu'à ce qu'il n'y ait plus de bogue détecté ou jusqu'à ce que la modification soit reportée à une version ultérieure ou abandonnée.

1.4. Mise en contexte du sujet à l'étude

Ce projet porte sur le processus de maintenance d'un service de paye offert à plusieurs dizaines de clients sur le territoire Québécois. Les deux plus gros clients de l'entreprise ont accès à des versions personnalisées du système de paye et les autres clients ont accès à une version générique. L'équipe de développement de ce système de paye est sous la direction du directeur du développement WEB : elle a comme responsabilité d'assurer le développement, le support et la maintenance de la solution WEB du système de paye.

Actuellement l'équipe est composée d'une vingtaine de personnes : Directeur du développement WEB, Chef d'équipe, Analystes seniors, Analystes et Analystes programmeurs. Les différents membres de l'équipe sont assignés soit aux équipes de développement, de support ou de maintenance du système.

Le cycle de développement s'étend actuellement sur des itérations de 6 mois pour les déploiements des versions principales avec de nouvelles fonctionnalités et il y a aussi des cycles mensuels pour la livraison de correctifs.

Au cours des mois précédant le début de ce projet, des changements au niveau du processus de développement furent progressivement intégrés avec comme objectif de réaliser une transition vers une méthodologie de développement Agile au cours des prochaines années. Voici une liste non exhaustive des changements qui ont eu lieu au cours de cette transition pendant les derniers mois:

- Formalisation de la revue de code
- Tests fonctionnels automatisés (projet en cours)
- Intégration d'une forge logicielle qui inclut :
 - Gestionnaire de source
 - Gestionnaire de « *build* » (multi versions)
 - Suivi des éléments de travail
 - « *Backlog* »

- Revue de code
- Cas de maintenance

Tableau 1.1 - Définition d'un cas de maintenance

Un cas de maintenance correspond aux éléments qui ont traversé les trois niveaux sont aboutir à une solution permanente et qui nécessite une correction ou une amélioration afin de ne plus se reproduire.

- Projets techniques

Tableau 1.2. - Définition d'un projet technique

Un projet technique est requis lorsqu'il s'agit d'une amélioration au système qui n'est pas forcément reliée à un cas de maintenance et qui doit être livrée entre deux versions d'évolution (aussitôt que possible).

Ex. : Amélioration de performance (exigences non fonctionnelles) sur des transactions qui diminue les performances de l'application.

- Planification du carnet des tâches

Tableau 1.3 - Définition d'une planification

La planification inclut la préanalyse des cas à réaliser afin de déterminer s'ils doivent être réalisés en maintenance. Le travail à réaliser est priorisé et l'estimation de l'effort est faite autour d'une table rassemblant des employés avec une expérience pertinente dans le domaine. Par la suite des « Ressources » sont assignées aux différentes étapes : analyse, réalisation, tests, livraison.

- Concept de livraison itérative
 - Annuellement auparavant
 - Aux 6 mois (versions avec de nouvelles fonctionnalités)
 - Mensuel (correctifs de maintenance).

- Équipe SWAT
 - Rotation de membres du développement dans l'équipe de maintenance (période de 6 mois).
 - Rotation des membres de l'équipe de maintenance au support (période d'une semaine).
- Carnet de tâches
 - Découpage des éléments du carnet des tâches en éléments de la taille réalisable dans une itération.
 - Estimation et ordonnancement des éléments du carnet des tâches en fonction de leurs priorités lors des réunions du comité de maintenance.
- Intégration de projets d'amélioration continue.

Tableau 1.4 - Définition de l'amélioration continue

Après son arrivée, le directeur du développement a initié l'équipe à l'approche de l'amélioration continue : l'amélioration des technologies et des façons de faire de l'équipe au travers de projets réalisables sur de courtes périodes de temps. L'intention de cette approche est de privilégier les changements progressifs et d'éviter les changements de type « BIG-BANG ». Les micro-projets d'amélioration continue nécessitent moins d'effort et provoquent moins de résistance dans l'équipe en plus d'avoir un impact négatif moindre dans le cas où ceux-ci avortent ou demandent plus d'efforts que prévus initialement.

(0.1)

1.5. Problématique inhérente à l'équipe

Suite à son arrivée à la direction de l'équipe de développement WEB, il y a près de trois ans, des travaux furent entamés afin de modifier les mesures et indicateurs de la productivité de l'équipe. Antérieurement, les rapports mensuels faisaient état du nombre de cas de maintenance résolus lors de la période comptabilisée; maintenant l'équipe observe aussi le nombre total de cas, en plus du nombre de nouveaux cas et du nombre de cas fermés. Bien qu'à première vue ce changement soit significatif dans la perception des résultats présentés à

l'équipe, cette façon de mesurer la productivité reste à haut niveau et ne permet pas de comprendre ou sont dépensés les efforts.

Ce changement au niveau du décompte des cas de maintenance totaux a mis en évidence que le nombre de cas de maintenance non traités par le processus de maintenance était près de 400 et que le nombre de nouveau cas ajoutés suite à une livraison dépassait le nombre de cas réglés lors de l'itération précédente.

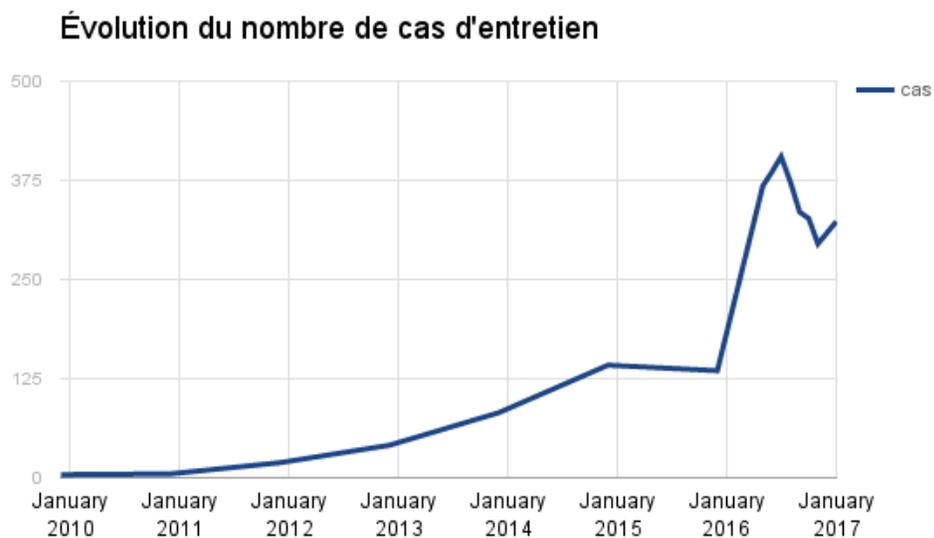


Figure 4 - Observation du décompte des cas de maintenance observables avec le système TFS.

Le passage de l'ancien outil de gestion du processus de la maintenance C2 vers un plus récent appelé TFS rend difficile la cueillette d'informations sur les cas de maintenance qui sont antérieurs à l'année d'intégration de TFS, ce qui donne peu d'utilité à la courbe tracée avec les données en provenance d'anciennes présentations d'équipe et celles dans TFS.

Suite à ce constat, une attention particulière fût accordée à cet aspect du processus de maintenance afin de reprendre le contrôle sur l'évolution de cette courbe et l'objectif « 0 cas

de maintenance» fût fixé : le directeur a fixé comme objectif de fermer tous les cas de maintenance et d'atteindre un total de 0 cas de maintenance, ce qui semble à première vue être un objectif irréaliste dans un système de cette envergure.

Les données dans TFS permettent de faire le décompte du nombre de cas de maintenance et de calculer un ratio de cas réglés vs cas créés : cette information est utile pour faire un constat à la fin d'une itération. Par contre ce type d'information ne permet pas d'identifier les goulots d'étranglement du processus.

Pour obtenir une vue plus détaillée des cas de maintenance, j'ai fait le décompte et j'ai regroupé par version les cas de maintenance qui ont été fermés lors des versions antérieures du système et j'ai aussi mesuré la période de temps en jours du moment de la création d'un cas de maintenance au moment où il est fermé. Le Tableau 1.5 présente les variables de la formule du calcul du cycle moyen des cas de maintenance. Sur l'axe des Y :

- Les colonnes bleues représentent le total de cas de maintenance.
- La ligne rouge représente le nombre de jours moyen du total des cas de la colonne bleue de leur date de création à leur date de fermeture.

Tableau 1.5 - Variables de la formule du calcul du cycle moyen des cas de maintenance.

Variables	Symboles
Cas 'premier cas de la suite (cas 1)'	i
Total du nombre de cas	n
Cas de maintenance de i à n	C _i , C _(i+1) , C _(i+2) , ..., C _(n-1) , C _n
Date de création	dtCr
Date de fermeture	dtFr

Tableau 1.6 - Équation du calcul du cycle moyen des cas de maintenance.

$\frac{(([\text{dtFr de } C_i]) - ([\text{dtcr de } C_i])) + (([\text{dtFr de } C_{(i+1)}]) - ([\text{dtcr de } C_{(i+1)}])) + \dots + (([\text{dtFr de } C_{(n-1)}]) - ([\text{dtcr de } C_{(n-1)}])) + (([\text{dtFr de } C_{(n)}]) - ([\text{dtcr de } C_{(n)}]))}{n}$

Sur l'axe des X

- La version dans laquelle les cas ont été livrés et fermés

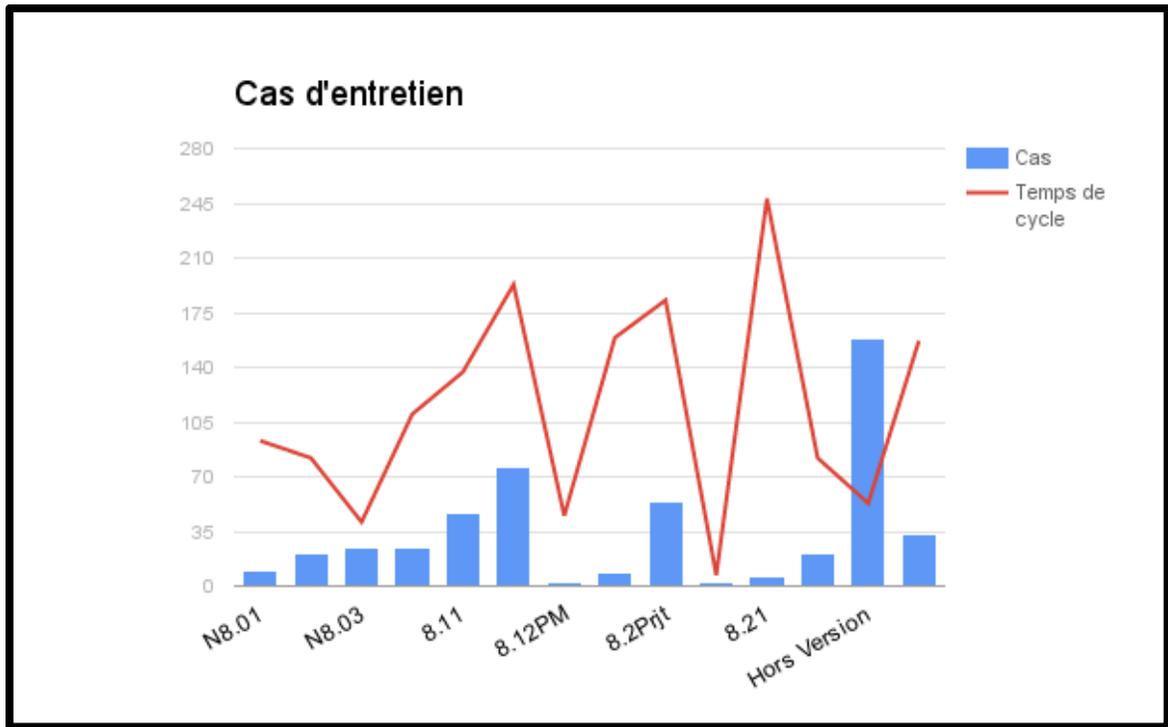


Figure 5 - Décompte du total de cas de maintenance par version et du cycle moyen (avec les données brutes)

La figure 5 indique que le temps de cycle moyen des cas de maintenance varie énormément d'une version à l'autre. Suite à cette observation, une analyse des données a révélé qu'il y a des valeurs aberrantes: par exemple, certains cas de maintenance ont requis plus d'une année complète avant d'être fermés.

1.6. La courbe des cas de maintenance tend vers la hausse

Comme plusieurs entreprises, ACME fût influencée dernièrement par les méthodologies Agile : la haute direction a ordonné la transition aux méthodologies Agile. Suite au démarrage de cette transition vers l'Agilité, plusieurs projets d'amélioration continue ont vu le jour et ont

été réalisés ou sont en cours de réalisation dans le but d'augmenter l'efficacité du processus de développement et de maintenance du logiciel.

Malgré les efforts déployés depuis 2014, la courbe des cas de maintenance continue de croître:

- Les directeurs précédents présentaient à l'équipe le nombre de cas de maintenance résolus,
- La direction en place pendant ce projet présente l'évolution des cas de maintenance ouverts.

Bien que les efforts d'amélioration semblent avoir permis à l'équipe d'intégrer de nouvelles technologies et d'ajouter de nouvelles pratiques, l'impact sur l'évolution de la courbe des cas de maintenance ne se dirige pas vers l'atteinte de l'objectif 0.

1.7. Problématique : Pourquoi la courbe de maintenance continue de croître?

Q1 : Pourquoi les efforts d'amélioration continue ne permettent pas de diminuer le nombre de cas de maintenance ouverts, c'est-à-dire de corriger plus de cas de maintenance qu'il ne s'en crée afin de permettre d'atteindre l'objectif 0?

Q2 : Quelles améliorations peuvent être faites au processus de maintenance logiciel pour le rendre plus efficace?

1.8. Modèle de maintenance

Il y a deux équipes de maintenance dirigées par deux directeurs différents : l'équipe de développement et de maintenance du logiciel WEB et l'équipe de l'assurance qualité, appelée GCQ. La stratégie de réalisation des tests unitaires de l'équipe de développement et de maintenance ne demandait pas de formalisation des activités de TU : le processus reposait sur une confiance implicite envers le professionnalisme des développeurs et les autres employés de la maintenance. La direction supposait que chacun faisait les TU qui le concernait.

Une stratégie de réalisation des tests unitaires (TU) a été élaborée et mise en pratique. Chaque équipe de développement se doit de posséder une batterie de tests unitaires manuels qui sont rédigés dans l'outil TFS et, lors d'une livraison, les TU doivent répondre aux exigences, c'est-à-dire qu'il ne doit pas y avoir de scénario de TU en échec pour que les évolutions et les corrections du système puisse être livrées.

La figure 6 présente les activités du groupe d'assurance qualité, activités qui débutent à l'étape de l'analyse et de la réalisation. L'équipe de GCQ n'est pas impliquée dans la phase d'analyse préliminaire du cycle de développement.

1.9. Maintenance

Le processus actuel de maintenance contient 5 étapes principales :

1. Création, documentation et réception du cas de maintenance
2. Suivi et gestion du cas de maintenance
3. Réalisation du cas de maintenance
4. Fermeture du cas de maintenance
5. Test et mise en production (MEP) du cas de maintenance

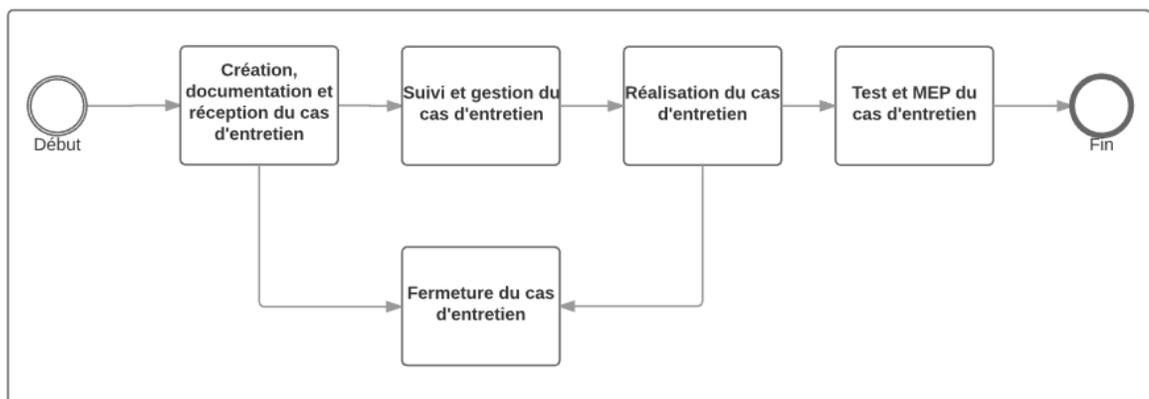


Figure 6 - Étapes principales du processus actuel de maintenance.

Au cours de ces étapes, il existe plusieurs états possibles pour un cas de maintenance dans le processus de maintenance détaillé.

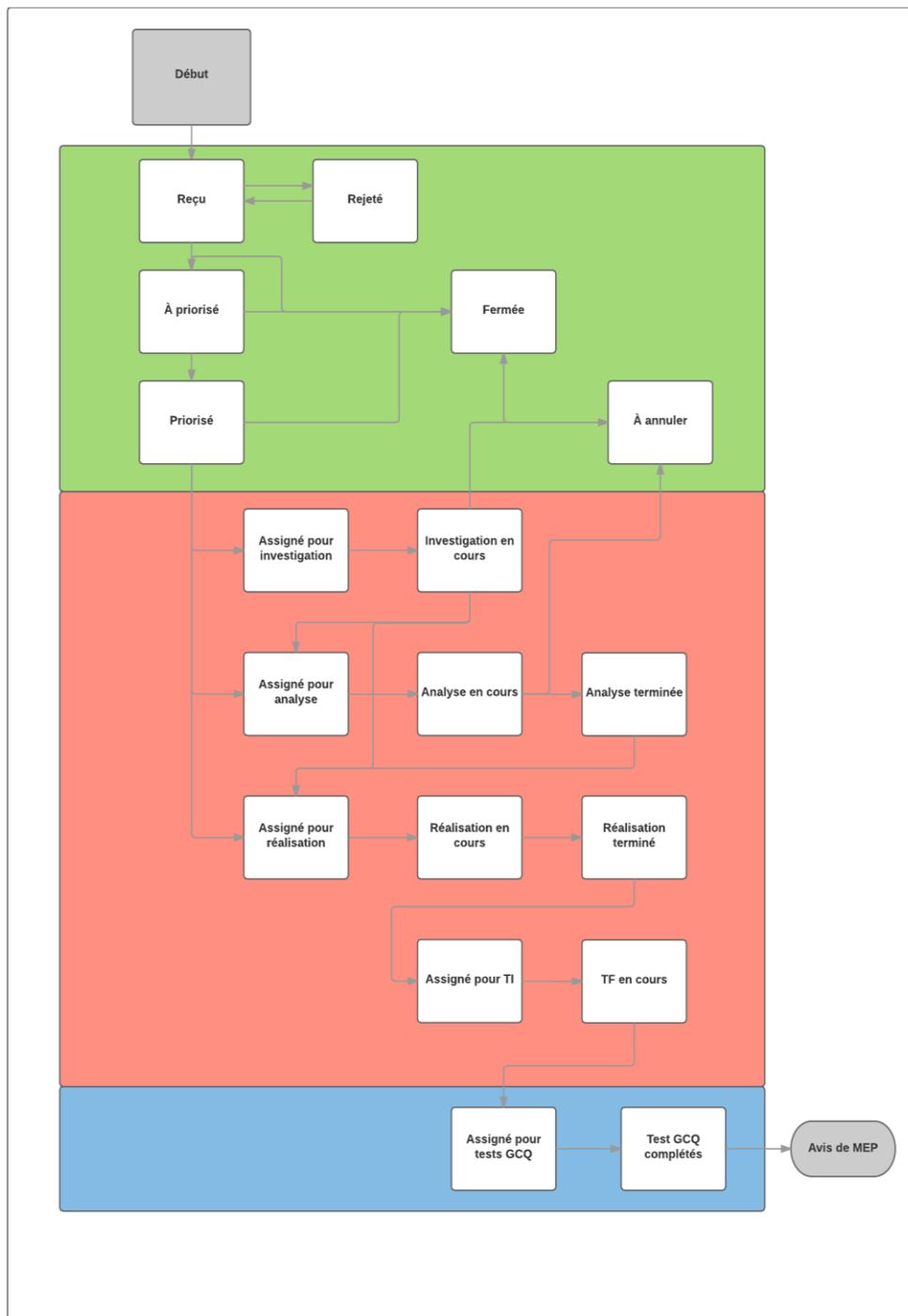


Figure 7 - Processus de maintenance détaillée.

Le membre de l'équipe de maintenance agit sur les cas de maintenance en fonction de son rôle et de l'état du cas de maintenance.

1.10. Les rôles de la maintenance logicielle chez ACME

Les rôles tenus par les employés d'ACME dans le processus de maintenance ainsi que les tâches qui leur sont associées chez ACME ont été documentés.

Tableau 1.7 - Les rôles de la maintenance logicielle chez ACME

Analyste de la maintenance
<ul style="list-style-type: none"> ● Créer les cas de maintenance ● Corriger les cas de maintenance incomplets ● Documenter les cas de maintenance ● Investiguer les cas de maintenance ● Analyser les cas de maintenance ● Participer aux synergies entre les analyses ● Réaliser des tests fonctionnels
Analyste GCQ
<ul style="list-style-type: none"> ● Rédiger les scripts de test ● Tester les cas de maintenance ● Préparation de la documentation ● Faire la mise en production des cas de maintenance ● Envoyer l'avis de MEP
Analyste programmeur à la maintenance
<ul style="list-style-type: none"> ● Réaliser le cas de maintenance ● MEP du cas de maintenance
Membre du comité de maintenance
<ul style="list-style-type: none"> ● Contribuer aux activités de maintenance ● Participer aux rencontres du comité de maintenance ● Confirmer la fermeture des cas de maintenance
<p>Lors des rencontres du comité de maintenance, ses membres ont comme responsabilités:</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p><i>Le comité de maintenance doit au minimum être composé d'un représentant de :</i></p> <ul style="list-style-type: none"> ● <i>De l'équipe de maintenance</i> ● <i>De l'équipe de soutien</i> ● <i>De l'équipe de GCQ</i> ● <i>Du produit</i> </div>
<ul style="list-style-type: none"> ● Clarifier les cas de maintenance soumis

- Prioriser les cas de maintenance
- Représenter les intérêts de leur département respectif
- Informer les parties prenantes qu'ils représentent des décisions prises par le comité

Lors des rencontres du comité de maintenance, ses membres ont comme responsabilités:

Le comité de maintenance doit au minimum être composé d'un représentant de :

- *De l'équipe de maintenance*
- *De l'équipe de soutien*
- *De l'équipe de GCQ*
- *Du produit*

- Clarifier les cas de maintenance soumis
- Prioriser les cas de maintenance
- Représenter les intérêts de leur département respectif
- Informer les parties prenantes qu'ils représentent des décisions prises par le comité

Responsable de maintenance

- Fermer les cas de maintenance
- Assigner les ressources
- Confirmer la fermeture du cas de maintenance
- Mettre à jour l'information
- Vérifier la documentation de la requête
- Vérifier les requêtes de maintenance reçues

Responsable de maintenance

- Fermer les cas de maintenance
- Assigner les ressources
- Confirmer la fermeture du cas de maintenance
- Mettre à jour l'information
- Vérifier la documentation de la requête
- Vérifier les requêtes de maintenance d'entretien reçues

Responsable GCQ

- Assigner les ressources GCQ
- Participer aux rencontres du comité
- Vérifier les activités de test

Analyste principal GCQ

- Préparer l'avis de MEP
- Assigner les ressources GCQ
- Investiguer les cas de maintenance
- Vérifier les activités de test

Les tâches réalisées par les différents employés qui occupent les rôles mentionnés précédemment produisent des artefacts livrables.

Tableau 1.8 - Les artéfacts du processus de maintenance chez ACME

Les cas de maintenance
Liste de cas de maintenance
Vérification des requêtes de maintenance reçues
Mise à jour des informations selon les recommandations du comité
Avis de MEP
Avis au groupe de soutien via C2
Notification de rejet
Avis d'assignation
Cas de test
Test fonctionnel

Le formulaire de cas de maintenance est intégré à l'outil TFS et est mis à jour en fonction de l'évolution du cas de maintenance dans le processus.

Tableau 1.9 - Éléments du formulaire d'un cas de maintenance chez ACME

Numéro du cas de maintenance
Identifiant généré par TFS lors de l'ajout du cas de maintenance à partir du C2.
Numéro du C2
Numéro du C2 à partir duquel le cas de maintenance de TFS est créé.
Créé par :
Nom de la personne qui a créé la requête
Sommaire
Description sommaire du problème
Date de création
Date de création du cas de maintenance dans TFS
Type d'intervention
<ul style="list-style-type: none"> ● Correctif applicatif ● Correctif technique ● « <i>Project life saver</i> » ● Changement fonctionnel ● Amélioration ● Demande de changement
Environnement
Environnement dans lequel le problème a été détecté: <ul style="list-style-type: none"> ● Démo ● Production
Clientèle

Clientèle impactée par le problème :
<ul style="list-style-type: none"> ● Administration et contrôle ● CESAD ● Informatique CSP ● Paie IP ● Paie téléphone ● ACME ● Banques
Produit
Produit impacté:
<ul style="list-style-type: none"> ● Suite internet ● Suite internet mobile ● Suite internet - outil intégration clients ● Paie PC ● Paie IP ● SAO - soutien aux opérations ● SAV - soutien aux ventes ● CRM ● Convexcel ● CDO - consultation donnée Opéra
Module
Identifie le module concerné
Étape
<ul style="list-style-type: none"> ● Investigation requise ● Analyse à faire ● Correction du code ● Test de transparence
Area
Identifie la fonction impactée (optionnel)
Itération
Version dans laquelle le cas d'entretien sera traité
Complexité
Ordre de grandeur du correctif :
<ul style="list-style-type: none"> ● Élevé ● Majeur ● Moyen ● Mineur
Efforts de développement (en jours)
Estimation du temps de développement et des tests en nombre de jours
Efforts GCQ (en jours)
Estimation du tempo requis pour les tests intégrés
Impact avec autres systèmes

<p>Si d'autres systèmes sont impactés par le cas d'utilisation les équipes doivent se synchroniser pour la réalisation des travaux:</p> <ul style="list-style-type: none"> ● Opéra ● PaiePc ● SAV ● InfoPath
<p>Priorité</p>
<p>Valeur attribuée par le comité de maintenance:</p> <ul style="list-style-type: none"> ● Critique ● Haute ● Moyenne ● Faible
<p>Version</p>
<p>Permet d'identifier la branche ainsi que la BD dans laquelle les modifications doivent être faites:</p>
<p>Responsable de maintenance</p>
<p>Identifie la ressource responsable de la requête de maintenance</p>
<p>Responsable de GCQ</p>
<p>Identifie la ressource GCQ responsable lorsque le cas de maintenance a été priorisé pour une prochaine version en entretien.</p>
<p>État/Raison</p>
<p>Évolution de la requête au cours des différentes transitions dans le processus de maintenance</p> <ul style="list-style-type: none"> ● Reçu ● Rejeté ● À prioriser ● À annuler ● Priorisé ● Assigné pour investigation ● Investigation en cours ● Assigné pour analyse ● Analyse en cours ● Analyse terminée ● Assigné pour réalisation ● Réalisation en cours ● Réalisation terminée ● Assigné pour TF ● TF en cours ● Assigné pour GCQ ● Tests GCQ en cours ● Prêt pour MEP ● Fermé
<p>En suspens et raison de la suspension</p>
<p>Champs attribués à "Oui" et avec un commentaire lorsque le cas d'utilisation est suspendu et avec une brève description de la raison.</p>
<p>Bloc d'assignation</p>

Bloc divisé en 5 types d'assignation : investigation, analyse, développement, test fonctionnel et assurance qualité (GCQ). À chaque étape deux ressources du groupe concerné sont assignées et les opérations sont bornées par une date de début et une date de fin.
Onglets problèmes et solution.
Description du problème, de la solution et de la raison du non-traitement de la requête lorsqu'il y a lieu de ne pas traiter la requête.
Onglet liste des modifications
Liste de tous les documents, fonctions et objets modifiés: <ul style="list-style-type: none"> ● Fonctions et traitement modifié ● Modification faite à la BD ● Conversion de données requises? ● Impacts sur les tests intégrés
Onglet impact
Identification des impacts sur: <ol style="list-style-type: none"> 1. Les données <ol style="list-style-type: none"> a. Impact le résultat de la paie (O/N) b. Corruption de données dans la BD (O/N) c. Commentaires 2. Les opérations <ol style="list-style-type: none"> a. Impact la clientèle (O/N) b. Impact les opérations (O/N) c. Modification requise à la doc (O/N) d. Commentaires
Onglet commentaires
Utilisée au besoin lors des annulations de requête

1.11. Architecture technologique de l'environnement maintenance

L'environnement technologique de l'équipe de maintenance logiciel chez ACME est principalement composé des outils suivants.

Tableau 1.10 - Environnement technologique

C2
Outil de gestion des cas de maintenance utilisé au niveau du support à la clientèle. Les cas reportés à ce niveau qui devront être traités par l'équipe de maintenance seront rédigés en cas de maintenance dans TFS.
TFS
Suite tout-en-un de Microsoft destinée au développement de logiciel : gestionnaire de source, Visual Studio, Gestion des backlog et des revues de code, gestion de configuration

Visual Studio
IDE de Microsoft qui intègre l'utilisation des cas de maintenance de TFS ainsi que les multiples versions des projets contenus dans le gestionnaire de source.
SQL Server
Outils d'accès aux contenues des différentes bases de données des projets

1.12. Proposition d'une approche méthodologique

Le processus utilisé dans la réalisation des activités de maintenance ne permet pas d'atteindre l'objectif de 0 cas de maintenance.

Pour fins d'analyse et pour trouver une solution, le processus de maintenance sera évalué et comparé à des processus connus afin de parvenir à trouver des solutions qui pourraient aider à résoudre la problématique initiale.

CHAPITRE 2 REVUE DE LA LITTÉRATURE

2.1. Introduction

Avant de pouvoir analyser et faire des recommandations sur le processus de maintenance de l'entreprise ACME afin de pouvoir résoudre la problématique, il faut d'abord avoir une bonne idée de ce qu'est un processus de maintenance logicielle selon les façons de faire de l'industrie de la maintenance logiciel ou de la littérature.

Selon la documentation d'ACME et ses sites internes décrivant leurs processus à haut niveau, leur processus de maintenance a été construit sur mesure en se basant sur diverses normes et façons de faire de l'industrie. Plusieurs de ces normes datent d'il y a plusieurs années, voire même de plus d'une décennie :

Cette revue de littérature porte sur quatre thèmes principaux que j'ai identifiés comme pertinents dans l'élaboration de la méthodologie qui me permettra de proposer une solution à la problématique :

1. Le processus de maintenance logicielle
2. Le « *Mapping* » du processus de maintenance logicielle
3. Les sources de gaspillage en maintenance logicielle
4. Les mesures de la maintenance logicielle

2.2. La maintenance logicielle

Il existe plusieurs pratiques dans le domaine de la maintenance ainsi qu'une très vaste gamme d'outils pour l'environnement de travail d'une équipe de maintenance logicielle.

« Il faut des outils afin de former un environnement de développement logiciel qui permettra de réaliser le développement ainsi que la modification du système ainsi qu'un environnement

de test logiciel afin de pouvoir tester le système à l'extérieur de l'environnement de production. Les efforts reliés à la mise sur pied de ces suites d'outils doivent être planifiés méticuleusement. La norme ISO/IEC 14471 encadre ce type d'efforts et la norme ISO/IEC 19759 fournit d'avantage d'information sur les outils de maintenance. » ISO/IEC 14764 section 6.4

Il revient aux gestionnaires et à l'équipe de maintenance logicielle de faire ces choix en fonction du contexte d'entreprise.

2.3. La maintenance logicielle : ses définitions et son vocabulaire, selon la norme ISO 14764.

En ingénierie, il est important d'utiliser un langage commun afin de bien se faire comprendre et diminuer les risques d'erreur d'interprétation lors des communications entre les différents employés d'un processus. La maintenance logicielle ne fait pas exception à cette pratique et elle possède une terminologie propre.

Certains de ses termes utilisés en maintenance sont listés et définis par la norme ISO 14764, le tableau suivant liste une traduction d'une partie de la terminologie employée en maintenance logicielle selon la norme ISO 14764.

Tableau 2.1 - Terminologie employée en maintenance selon ISO 14764

Maintenance adaptative : Réalisée après la livraison pour garder le logiciel utilisable dans un environnement dynamique (qui change).
Maintenance corrective : Réalisée après la livraison pour corriger les problèmes (pour satisfaire aux exigences)
Maintenance d'urgence : Maintenance non planifiée, réalisée pour garder un système fonctionnel en attendant la maintenance corrective (la maintenance d'urgence fait aussi partie de la maintenance corrective).
“Maintenabilité” : La capacité qu'un système a de pouvoir être modifié (corrections, adaptations et améliorations à son environnement, aux exigences fonctionnelles et non fonctionnelles).

Maintenance d'amélioration : Pour satisfaire à une nouvelle exigence (adaptative ou perfective). Ce n'est pas un correctif.
Requête de modification (ou requête de changement) : Une modification proposée à un système maintenu (qui doit être par la suite catégorisé: corrective, préventive, adaptative ou perfective).
Maintenance préventive : Modification post livraison qui vise à détecter ou à corriger des défauts avant qu'ils ne se manifestent.
Rapport de problème : Identification et détection d'un problème dans un produit logiciel (soumis lorsqu'une faute est détectée ou après l'analyse d'impact d'une requête de modification dans laquelle une faute est découverte).
Maintenance logicielle : Toutes les activités requises pour assurer un support à coûts efficients d'un système logiciel (pré livraison: planification, « supportabilité » et détermination logistique ; post livraison : modifications logicielles, formation et aide/support technique).
Transition logicielle : Séquence contrôlée et organisée d'actions pour qu'une organisation effectuer le passage de l'étape du développement logiciel vers la maintenance du logiciel.

Le vocabulaire employé dans un processus de maintenance est important afin de permettre une bonne communication entre les acteurs du processus : des utilisations inappropriées de langage peuvent mener à de la confusion et nuire au bon déroulement des opérations. L'utilisation de termes reconnus dans l'industrie aide à assimiler l'information disponible sur le processus de maintenance et ses activités. En normalisant le vocabulaire employé dans le processus, on facilite aussi l'intégration de nouveaux acteurs, ce qui n'est pas négligeable dans un domaine avec roulement élevé de personnel.

2.4. Le processus de maintenance

Le niveau de maturité d'un processus de maintenance peut être évalué en observant l'application des normes, tel que mentionné dans [Abran-April, 2008]. Les différents niveaux de maturité du processus de maintenance ne seront pas forcément atteints ni requis dans toutes les entreprises. Selon [Abran-April, 2008], il existe 6 niveaux de maturité.

Tableau 2-2 - Pyramide des standards [Abran-April, 2008]

0. Standards Locaux

1. ISO 12207
2. ISO 14764 et SWEBOK
3. ISO 9001, 9294, 15504, 25000 et IEEE 1028, 828, 829, 1012, 1062, 9294
4. ISO 15939, 19761 et IEEE 730, 830, 1008, 1044, 1061
5. Tous les standards applicables en génie logiciel

Dans le cadre de ce projet j'ai porté d'avantage d'attention à la norme ISO 14764 et à [Abran-April 2008] qui reprennent l'essentiel du niveau 2 de cette pyramide.

La norme ISO 14764 définit le processus de maintenance comme suit : « Le processus de maintenance débute lorsqu'il y a une exigence qui spécifie que le système doit être maintenu. Suite au démarrage du processus de maintenance des ressources doivent être assignées au processus de maintenance afin de pouvoir effectuer les modifications répondant aux demandes de changement et aux problèmes rapportés. L'objectif du processus de maintenance est de modifier le produit existant en préservant son intégrité. Le processus se termine lorsque le produit est retiré de la production. »

Toujours selon la norme ISO 14764, la maintenance logicielle est décrite comme suit : « La maintenance commence lors de sa planification et se termine lorsque le système est retiré de la production. Son objectif ultime est de préserver l'intégrité du système : coder et documenter les corrections et les modifications. Elle est requise à cause des erreurs et à cause des nouvelles spécifications et des modifications imprévues devant être faites aux spécifications. Tout comme le développement, la maintenance nécessite l'utilisation d'outils qui la facilite sans pour autant éliminer sa nécessité. La maintenance peut nécessiter du code et des modifications aux données, au fonctionnel, aux interfaces ainsi que le retrait de spécifications qui ne sont plus requises. La maintenance rapporte, trace, analyse, résout, catégorise, approuve ou rejette les rapports de problèmes et les demandes de changement. Elle s'assure que le système répond aux exigences du client et elle s'applique sur tous les systèmes et ce peu importe la méthodologie de développement qui l'encadre. L'environnement de production d'un système impose des contraintes aux processus de maintenance: disponibilité 24/7 exige

une maintenance 24/7. La maintenance accapare une grande partie des ressources consacrées au cycle de vie d'un système et le type de maintenance requis est directement relié à l'affectation des ressources (monétaire, technique, humaine, temporelle, etc.). »

2.5. Les types de maintenance

Selon la norme ISO 14764 il existe trois types de maintenance.

Tableau 2-3 - Type de maintenance selon ISO 14764

Corrective
<ul style="list-style-type: none"> • Requis à cause des erreurs • Lorsqu'un système ne répond pas aux exigences • Une maintenance d'urgence (modification non planifiée) pour garder le système opérationnel temporairement en attendant la maintenance corrective.
Préventive
<ul style="list-style-type: none"> • Suite à la détection d'erreurs potentielles.
Adaptative et perfective
<ul style="list-style-type: none"> • Modification qui améliore le produit qui ne figure pas dans les spécifications d'origines. • Accommodement à l'environnement changeant de la production. • Nouvelles interfaces. • Nouvelles spécifications (ou améliorations) fonctionnelles ou non. • Rehaussement de la performance ou de la maintenabilité. • Rétro-ingénierie pour documenter un bout sans documentation (ou pour améliorer ou mettre à jour la documentation existante). • La maintenance perfective requiert de travailler dans un design et une architecture d'un système existant et nécessite de les modifier et/ou de les améliorer. • La maintenance perfective est une activité très coûteuse en temps et en efforts. • La maintenance perfective peut rapidement consommer une très grande partie des dépenses consacrées au processus de maintenance.

2.6. Le gaspillage dans un processus de maintenance

Un processus de maintenance logicielle conforme aux normes de l'industrie ne garantit pas qu'il soit optimal ; les nombreuses activités présentes dans le processus peuvent devenir une source de gaspillage non négligeable.

« Le plus gros problème de la maintenance de logiciel n'est pas la technique, mais la gestion. »

[Abran-April,2008]

Les transitions entre les différentes étapes du processus ainsi que les mouvements qui ont lieu pendant les activités peuvent venir alourdir le processus au point d'en augmenter le coût sans ajouter de valeurs aux activités réalisées.

« Il a été reconnu depuis longtemps qu'un processus immature de développement crée bon nombre de problèmes dans le produit livré au client et à la maintenance. »

[Abran-April,2008]

Dans cet énoncé, les auteurs expriment l'un des problèmes rencontrés par les équipes de maintenance en entreprise : elles risquent se retrouver en queue du processus de développement sans y être impliquées ce qui ne leur permet pas d'amener les améliorations pouvant rehausser la qualité du système livré et d'en diminuer le gaspillage en maintenance. Leur expérience de la maintenabilité d'un système n'étant pas formellement mis en pratique lors du développement, elles se retrouvent à hériter de problèmes qui auraient pu être évités lors du développement, qui lui a, dans certains cas, eu lieu à une époque différente avec des façons de faire et des technologies différentes.

« ... une grande part du cout du cycle de vie d'un système-50 à 90%- provient de la maintenance logicielle ... »

[Abran-April,2008]

« L'âge d'un système a un impact sur le coût de la maintenance, lorsqu'un logiciel a été construit en utilisant d'anciennes techniques qui ne sont plus tenues en compte dans les architectures modernes, avec des structures complexes, du code non standardisé et une pauvre documentation, le travail de la maintenance est plus difficile. »

[Abran-April,2008]

Maintenir un système vieillissant demande de plus en plus de dépenses, améliorer le processus en soit peut aider à rehausser l'efficacité du processus, mais ne règlera pas les problèmes qui proviennent de l'âge du système. [Abran-April, 2008] cite une liste de [Dekleva 1992] qui ordonnance des problèmes identifiés par des acteurs de la maintenance. Parmi ceux-ci j'en ai identifié quelques-uns qui peuvent devenir source de gaspillage dans un système vieillissant tel que :

- Les techniques de tests inadéquates
- Une documentation logicielle absente ou incomplète
- Difficulté d'adaptation aux changements rapides en provenance des utilisateurs
- Une très grande liste de demandes de changements
- Une grande difficulté à trouver des experts du domaine
- Peu de méthodologie, de standard, de procédures et d'outils spécifiques à la maintenance.
- Code source complexe et peu structuré.
- Problèmes, d'intégration et de compatibilité avec les systèmes existants.
- Peu de formations disponibles aux ingénieurs de la maintenance.
- Le logiciel s'exécute sur des systèmes et des technologies obsolètes.
- Peu de support dans la réingénierie des systèmes existants.
- Perte d'expertises suite aux départs des employés de l'équipe ou de la compagnie.

Dans leur livre [Mary et Tom Poppendieck,2003] basent leur ouvrage sur sept principes LEAN:

Tableau 2-4 - Les sept principes LEAN

<ol style="list-style-type: none"> 1. Éliminer le gaspillage 2. Amplifier l'apprentissage 3. Décider au dernier moment possible 4. Livrer le plus vite possible 5. Impliquer l'équipe 6. Bâtir l'intégrité 7. Avoir la vue d'ensemble <p>[Mary Poppendieck-Thomas David Poppendieck,2003]</p>
--

Dans le cadre de ce projet, je retiens le premier élément de cette liste soit l'élimination du gaspillage.

2.7. Éliminer le gaspillage

Selon le chapitre 1 de [Poppendieck- Poppendieck, 2003] le gaspillage peut être éliminé de différentes façons - voir tableau 15

Tableau 2-5 - Limiter le gaspillage

<p>Les fonctionnalités en Extra (<i>Extra features</i>) : Ne pas inclure de caractéristiques qui ne sont pas destinées à répondre aux exigences du client, car ça oblige à maintenir du code qui n'est pas requis.</p>
<p>Changement de tâches (<i>Task switching</i>) : assigner les équipes sur un projet à la fois.</p>
<p>L'attente (<i>Waiting</i>) : Les délais entre les exécutions des tâches et entre les différentes étapes du processus.</p>
<p>La manutention (<i>Motion</i>) : le nombre de mouvements qu'il faut avant qu'une personne obtienne ce dont elle a besoin pour accomplir son travail : à combien de gens elle doit poser une question avant d'avoir la réponse, quels sont les déplacements qu'elle doit faire au cours d'une journée de travail, etc. Chaque manutention demande du temps et est aussi une source de distraction qui fait perdre le focus sur le travail à réaliser.</p>
<p>Les défauts (<i>Defects</i>) : Un défaut qui est rapidement détecté engendrera beaucoup de gaspillage qu'un défaut qui existe dans le système sur une longue période de temps, autant dans les problèmes que ce défaut peut provoquer sur la convivialité du système que la complexité du correctif qu'il requiert lorsqu'il est découvert tardivement.</p>

Les activités de gestion (*Management Activities*) : Les activités de gestion autant humaines que les systèmes qui les gèrent demandent du temps et elles n’apportent pas de valeurs ajoutées au logiciel. Dans un processus LEAN, les activités de gestions sont minimales, car le travail à faire n’existe pas suffisamment longtemps dans le processus pour qu’il requière un grand effort d’activités de gestions.

2.8. “Mapper” le processus

Dans [Poppendieck- Poppendieck, 2003] il est mentionné qu’il est important de “Mapper” le processus afin de pouvoir identifier plus aisément les sources de gaspillage du processus. Le « *Mapping* » de processus est une activité souvent rencontrée en entreprise, dont chez ACME, mais le modèle proposé dans le livre “*Lean software development : an agile toolkit*” possède des périodes de temps associées aux différentes étapes du processus. Cette façon de faire permet d’établir un ordre de grandeur de l’impact temporel d’une étape sur l’ensemble du processus. On peut aussi imaginer que ce « *Mapping* » peut être bonifié d’autres informations pertinentes, telles que le nombre d’acteurs impliqués par étape du processus ainsi que leurs différentes tâches et le temps qu’ils y ont consacré. Des coûts monétaires pourraient aussi être présents afin de permettre de cibler plus aisément quelles étapes du processus ont un plus grand impact sur les dépenses. Le « mapping » du processus permet de faire des relations entre les différentes informations recueillies sur celui-ci et ainsi être un outil d’identification des sources de gaspillage.

2.9. Le “*benchmarking*” et l’estimation

Selon [April-Abran, 2008] il existe trois types de « *benchmarking* » : interne, compétitif et fonctionnel. La maintenance peut profiter du « *benchmarking* » interne afin de tirer des conclusions suite à des améliorations. Les auteurs font aussi une liste non exhaustive de mesures qui peuvent être collectées et servir à la maintenance, en voici quelques-unes :

Tableau 2-6 - mesures collectées et utiles à la maintenance [April-Abran, 2008]

Nombre de points de fonction supportés par personne
Coût par point de fonction

Âge moyen des applications supportées
Nombre de langages de programmation supportés
% des types de programmation (maintenance, amélioration, nouvelle application)
Diversité technique du support (nombre d'applications, langage de programmation, technologie d'archivage, outils, système d'exploitation)
% de la rotation du personnel
Nombre d'années travaillées dans l'équipe
Salaires
Effort de formation (nombre de jours de formation par employés par année)
Ratio de défauts en production par 1000 points de fonction supportés (par criticité : critique, majeur, mineur ou cosmétique)
Ratio de défauts en production par 1000 points de fonction supportés (par catégorie : design, programmation, environnement ou autre)

Les auteurs citent aussi [Deming, 1986] « En mesurant, nous pouvons en apprendre plus sur le comportement des ressources, des processus et des produits logiciels avec leur relation de causes à effet. »

« Les organisations matures ont établi des plans de mesure de leur maintenance aussi tôt qu'en 1987. »[April-Abran, 2008]

En combinant les activités de mesures du processus de maintenance logiciel à son « *Mapping* » il est possible d'établir des relations entre les données recueillies et le processus de maintenance logicielle et par la suite permettre d'identifier les sources de gaspillage plus aisément.

CHAPITRE 3 MÉTHODOLOGIE DE TRAVAIL

3.1. Mise en contexte du choix de la méthodologie de travail.

Dans le chapitre 1 deux questions avaient été posées suite à l'identification de la problématique :

Tableau 3.1- Questionnement soulevé au chapitre 1

<p>Q1 : Pourquoi les efforts d'améliorations continues ne permettent pas de diminuer le nombre de cas de maintenance ouverts, c'est-à-dire de corriger plus de cas de maintenance qu'il ne s'en crée afin de permettre d'atteindre l'objectif 0?</p> <p>Q2 : Quelles améliorations peuvent être faites au processus de maintenance logiciel pour le rendre plus efficace?</p>

Suite aux lectures et aux observations faites du processus de maintenance logiciel chez ACME, il s'est avéré difficile de proposer une réponse à la question 1, car les informations obtenues sur le processus de maintenance logiciel chez ACME ne m'ont pas permis de cibler la ou les sources qui causent la problématique.

En début de projet, l'une des premières approches avait consisté à proposer des activités identifiées comme étant de bonnes pratiques en génie logiciel et de les appliquer au processus de maintenance en espérant que ça résoudrait la problématique. Cette approche fût laissée de côté car le choix des activités en question ainsi que les résultats espérés étaient hypothétiques et il ne m'était pas possible de garantir que les efforts dépensés pour intégrer ces activités permettraient de résoudre la problématique. Le manque d'informations sur le déroulement du processus de maintenance logicielle ne permet pas de pouvoir en faire une analyse sur laquelle tirer des conclusions et de proposer une solution à la problématique.

Suites aux lectures faites en cours de projet et tel qu'expliqué lors de la revue de littérature, pour pouvoir proposer une solution à la problématique identifiée au premier chapitre il faut d'abord pouvoir répondre à la question 2 et identifier les zones à problèmes du processus de maintenance logiciel chez ACME.

Afin d'y parvenir, la méthodologie a été divisée en quatre étapes telles qu'illustrées par la figure 8. Le processus de maintenance logicielle d'ACME doit pouvoir être observé, analysé, ses sources de gaspillage doivent être identifiées et les effets des solutions mises en place pour diminuer le gaspillage doivent pouvoir être comparées dans le temps afin de pouvoir en tirer des conclusions. Pour y parvenir, il faut détenir de l'information sur les efforts déployés et sur les résultats obtenus suite à ses efforts. Cette information doit provenir de mesures appliquées aux différentes étapes du processus et aux activités réalisées.

Mesurer le processus de maintenance logicielle n'est pas une solution en soi. En mesurant l'ensemble d'un processus de maintenance logicielle pour ensuite tirer des conclusions sur lesquelles fonder une proposition de solution à une problématique ne garantit pas la résolution de la problématique.

Je vais commencer par cibler des zones du processus susceptibles d'être source de gaspillage, les zones de gaspillage potentiel peuvent ensuite être analysées pour parvenir à déterminer ce qu'il faut mesurer pour pouvoir faire des comparaisons, puis faire une rétrospection pour déterminer quelles actions peuvent permettre de diminuer le gaspillage et ensuite observer les effets et déterminer si les modifications faites améliorent ou dégradent l'efficacité du processus de maintenance logicielle.

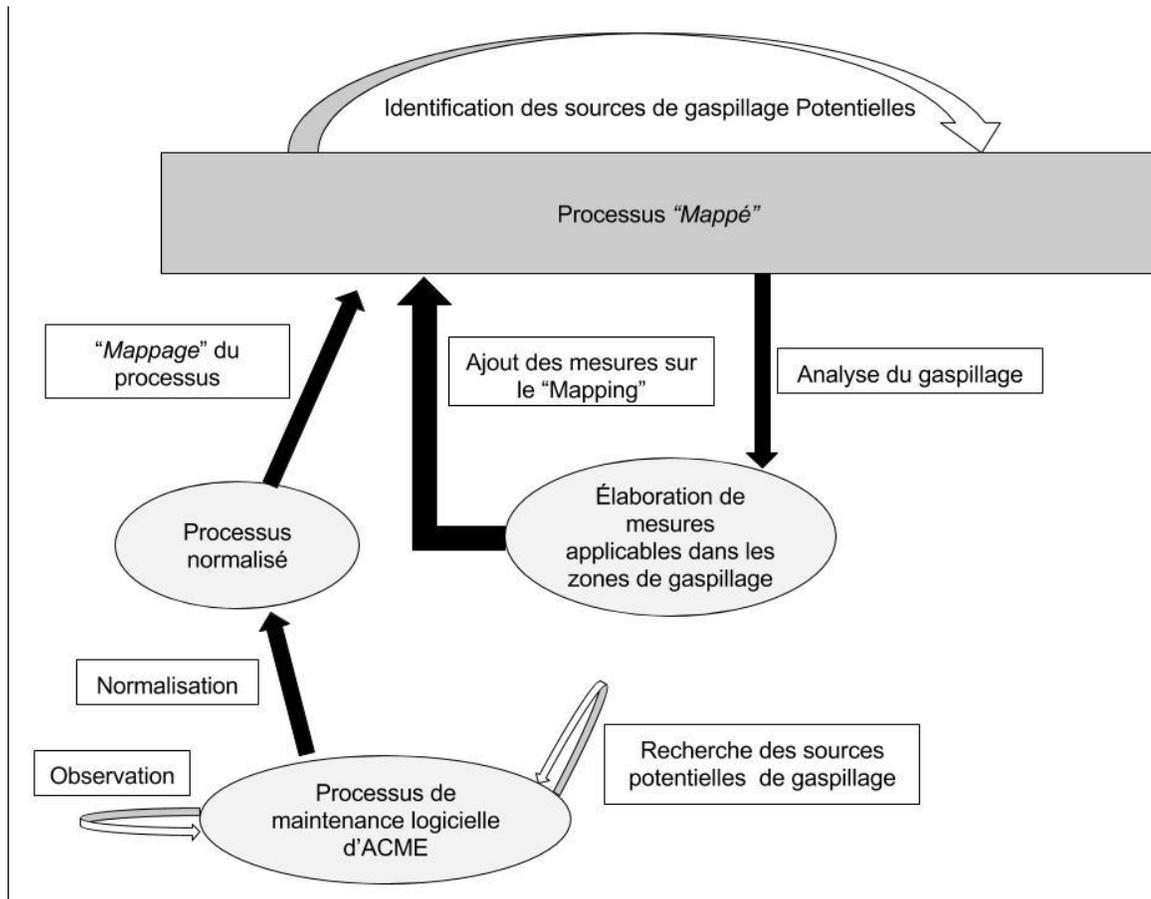


Figure 8 - Méthodologie de travail

3.2. La méthodologie de la maintenance logicielle

La première étape de la méthodologie de travail consiste à comparer le processus de maintenance logicielle actuelle avec les standards contenus dans la norme ISO 14764 ainsi que ceux de [April-Abran, 2008]. L'objectif de cet exercice est de s'assurer que le processus de maintenance logicielle d'ACME est normalisé. Si des écarts entre le processus de maintenance logicielle d'ACME et les standards de l'industrie sont identifiés, ils seront mentionnés.

Pour y parvenir, la méthodologie d'évaluation du niveau de maturité de la maintenance logicielle publié par [D-A Paquette-A April-A Abran,2006] sera utilisée en guise de modèle afin de définir à quel niveau de maturité se situe le processus de maintenance chez ACME. Les

questionnaires des 3 niveaux [annexe 1] seront utilisés en fonction de l'information recueillie lors des observations chez ACME.

3.3. Le «*Mapping*» du processus de maintenance logicielle

ACME possède un « *Mapping* » de son processus de maintenance, par contre les informations de mesures qui peuvent permettre d'aider à cibler les activités sensibles de contenir du gaspillage ne sont pas reliées au « *mapping* », ce qui ne permet pas d'avoir une vue d'ensemble du déroulement des opérations ainsi que de réaliser des comparaisons au fil du temps. L'efficacité du « *Mapping* » du processus de maintenance logicielle ne correspond pas à celui montré en exemple dans [Poppendieck- Poppendieck, 2003] et son efficacité est diminuée.

Le « *Mapping* » actuel, tel que représenté par les figures 9 et 10, du processus n'est pas complet, il représente les étapes du processus de maintenance logicielle d'ACME, mais il n'inclut pas de mesures permettant d'analyser l'évaluation des activités au cours du cycle de maintenance. Le « *Mapping* » du processus permet d'avoir une vue d'ensemble du processus de maintenance logicielle et des mesures qui lui sont associées.

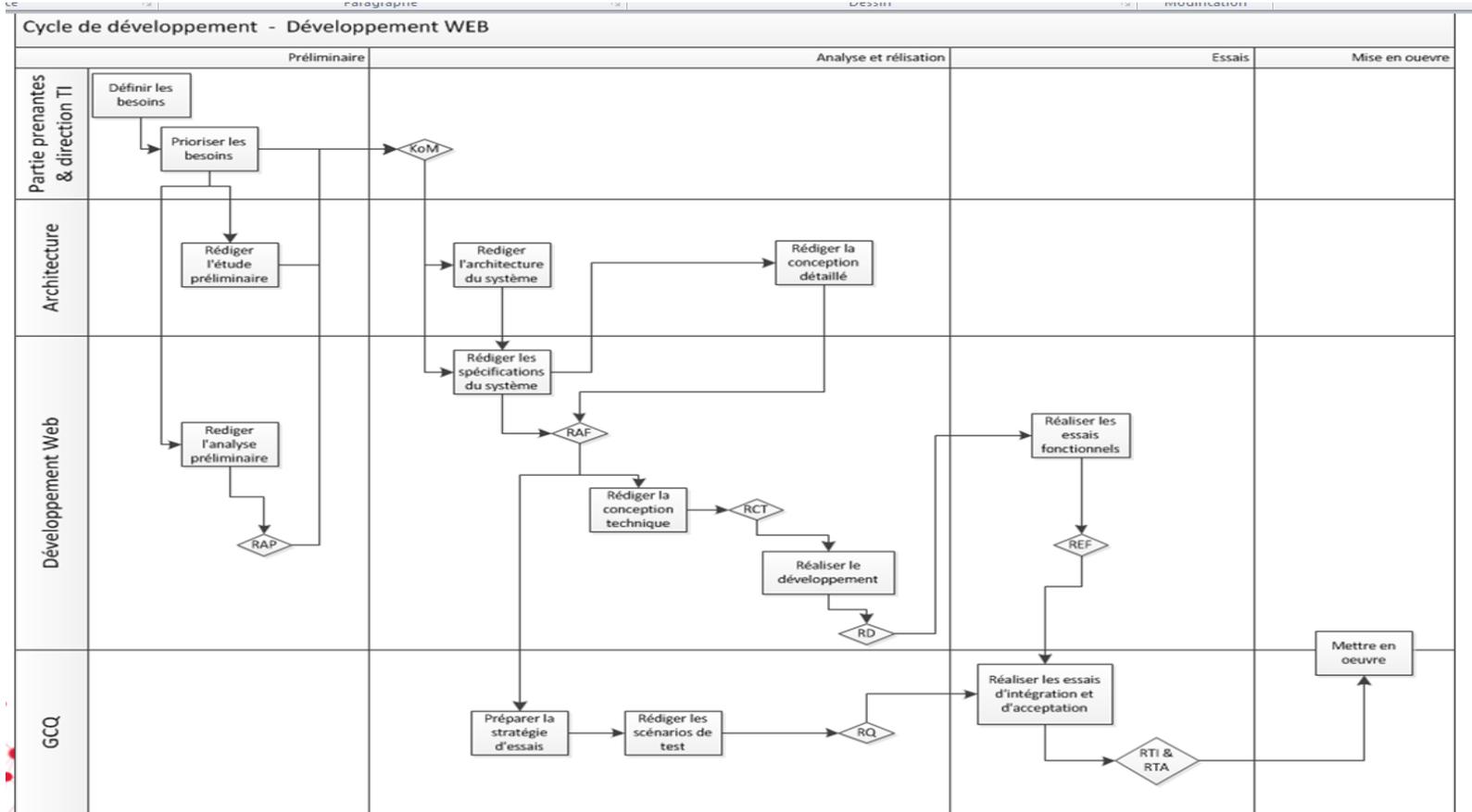


Figure 9 - Exemple de « Mapping » actuel chez ACME.

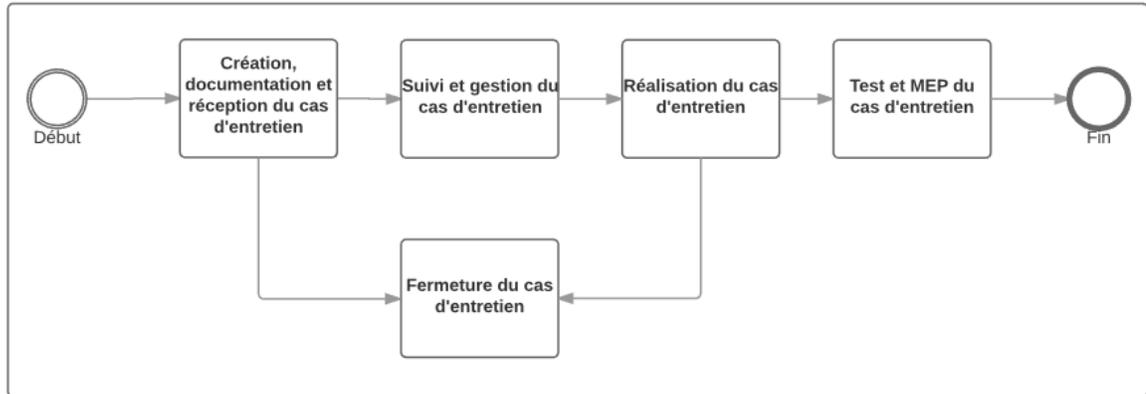


Figure 10 - Exemple de « Mapping » du processus de maintenance chez ACME

3.4. Les sources de gaspillage en maintenance logicielle

Pour parvenir à identifier les sources de gaspillage dans le processus et par la suite les éliminer, il faut d’abord observer le processus en soi et connaître ses différentes étapes et activités qui y sont réalisées. Il est plausible de penser qu’il puisse exister des défauts dans le processus ainsi que dans le système qui augmentent les efforts de la maintenance logicielle

Lors de cette étape, j’utiliserai les six problèmes qui peuvent provoquer du gaspillage tel qu’énoncés dans [[Mary Poppendieck](#), [Thomas David Poppendieck- 2003](#)] afin d’identifier de quelle façon ils peuvent se manifester lors des différentes étapes du processus de maintenance logicielle chez ACME. L’objectif de cette étape est de munir les acteurs de la maintenance d’une sorte de « Checklist » leur permettant de laisser une trace de leurs ouvrages dans laquelle il serait possible d’identifier les embuches au déroulement des opérations, embuches qui s’avèrent être des sources de gaspillage. Idéalement les traces de cette activité devraient être conservées dans un outil de l’environnement technologique de la maintenance.

Tableau 3.2 - Les fonctionnalités en Extra (Extra features)

Ne pas inclure de caractéristiques qui ne sont pas destinées à répondre aux exigences du client, car ça oblige à maintenir du code qui n'est pas requis. [[Mary Poppendieck](#), [Thomas David Poppendieck- 2003](#)]

En maintenance logicielle, cette source de gaspillage nuit aux processus à cause du code et de la documentation inutile qu'elle génère. Lors de l'analyse d'un système, il est important d'en venir à l'essentiel efficacement et de se limiter à la solution qui peut corriger le défaut sans en causer de nouveaux. La présence de code mort, de commentaires inutiles et de documentation désuète peut confondre les acteurs de la maintenance vers de fausses pistes et augmenter l'effort d'analyse à déployer pour parvenir à comprendre le problème, à en cerner la cause et par la suite à proposer et implémenter la solution. Lorsque des traces de ce type de gaspillage sont détectées, il faut les identifier de sorte à pouvoir les éliminer ou au moins : ce sera un des objectifs visés lors de la réalisation de cette étape de la méthodologie de travail tel que représenté dans la figure 8.

Tableau 3.3 - Changement de tâches (Task switching)

Assigner les équipes sur un projet à la fois. [[Mary Poppendieck](#), [Thomas David Poppendieck- 2003](#)]

Lors de l'assignation d'une tâche à un acteur de la maintenance il est important de connaître la charge de travail qu'il a déjà ainsi que le nombre de contextes différents auxquels il doit faire face lors du déroulement de cette tâche, contexte qui peut dépendre de période d'attente ou de manutention due au processus ou à des tâches qui requiert des interventions en provenance d'acteurs. Dans un environnement de maintenance logicielle le changement de tâches peut devenir une grande source de gaspillage en termes de temps et d'effort : par exemple, un environnement technologique inefficace peut venir alourdir le passage d'un projet X vers le contexte particulier d'exécution du cas de maintenance Y. Le temps de chargement, de la compilation et de l'exécution du projet en mémoire peut alourdir le processus s'il doit se

produire à une fréquence déraisonnable ainsi qu'un cas peut demander un ensemble de données spécifiques avant de pouvoir être exécuté, ce qui peut ajouter à l'effort devant être déployé pour réaliser la tâche en question. On peut imaginer qu'un acteur possédant une machine désuète et sous-performante puisse perdre de nombreuses heures par semaine si son emploi du temps exige qu'il change régulièrement de contexte pour des raisons hors de son contrôle. Il existe d'autres conséquences qui peuvent découler de l'excès de gaspillage relié à un nombre de changements de tâches tels que la fatigue et la démotivation de l'acteur concerné, qui peut par la suite mener à une diminution de l'engagement de l'acteur envers la tâche qu'il a à réaliser. Bien que difficile à mesurer cette conséquence ne doit pas être négligée.

Tableau 3.4 - L'attente (Waiting)

Les délais entre les exécutions des tâches et entre les différentes étapes du processus.

[[Mary Poppendieck](#), [Thomas David Poppendieck- 2003](#)]

Le temps d'attente entre les activités à réaliser pour compléter un cycle de maintenances a un impact significatif sur le temps de cycle requis pour régler un problème assigné à la maintenance. Lors de l'analyse de la réalisation des cas de maintenance, il faut connaître le temps de cycle des cas de maintenance et par la suite calculer leur temps de cycle moyen. Avec cette information il est possible de détecter les cas de maintenance qui ont un temps de cycle aberrant et de faire une analyse de sa réalisation afin de détecter les temps d'attentes hors de l'ordinaire. L'observation du flot d'un cas de maintenance dans le processus de maintenance représenté par un « *Mapping* » auquel des mesures ont été jointes permettra de cibler les délais suspectés d'allonger le temps de cycle d'un cas de maintenance.

L'attente peut dépendre de plusieurs facteurs, un processus LEAN visera à éliminer ces temps d'attente dans l'objectif de fluidifier le flot des cas de maintenance. La disponibilité des ressources, les priorités, la charge de travail, le matériel, le roulement des employés, les absences, la présence sur les lieux du travail, etc. L'analyse du traitement d'un cas de maintenance dans le processus de maintenance devrait permettre de cibler les raisons qui

provoquent les temps d'attente et par la suite permettre de mettre sur pied un plan d'intervention.

Tableau 3.5 - La manutention (Motion)

Le nombre de mouvements qu'il faut avant qu'une personne obtienne ce dont elle a besoin pour accomplir son travail : à combien de gens elle doit poser une question avant d'avoir la réponse, quels sont les déplacements qu'elle doit faire au cours d'une journée de travail, etc. Chaque manutention demande du temps et est aussi une source de distraction qui fait perdre le focus sur le travail à réaliser.

[[Mary Poppendieck](#), [Thomas David Poppendieck- 2003](#)]

Plusieurs acteurs participent au processus de maintenance, il peut arriver que le flot de la réalisation d'un cas de maintenance ne soit pas aussi linéaire que lorsqu'il est observé via le « *Mapping* » du processus. Un cas de maintenance en cours de réalisation peut être retourné à l'étape précédente pour diverses raisons ou bien sa progression dans le flot de maintenance peut être bloquée pour différentes raisons. Il est important de pouvoir décortiquer le temps de cycle d'un cas de maintenance afin de pouvoir comprendre les raisons pour lesquelles un cas peut avoir pris davantage de temps que prévu. Les différentes manutentions et temps d'attente ne sont pas toujours sous contrôle de l'acteur responsable de l'étape à laquelle les cas de maintenance sont rendus dans le processus. Afin de pouvoir faire une rétrospective de l'évolution des cas de maintenance dans le processus de maintenance logicielle, il est important de posséder suffisamment d'information sur l'état de l'évolution du cas.

Cette source de gaspillage est intimement liée au temps d'attente, aux changements de tâches, à la manutention et aux activités de gestions qui peuvent ralentir le processus lorsqu'elles occupent trop de temps. Les acteurs d'une équipe doivent pouvoir rapporter rapidement les défauts qui sont détectés dans le système, ses rapports de défauts doivent être enregistrés et priorisés par la suite.

Tableau 3.6 - Les activités de gestion (Management Activities)

Les activités de gestion autant humaines que les systèmes qui les gèrent demandent du temps et elles n'apportent pas de valeurs ajoutées au logiciel. Dans un processus LEAN, les activités de gestions sont minimales, car le travail à faire n'existe pas suffisamment longtemps dans le processus pour qu'il requière un grand effort d'activités de gestion.

[[Mary Poppendieck](#), [Thomas David Poppendieck- 2003](#)]

Le nombre d'activités de gestion ainsi que le temps qu'elles peuvent occuper dans un processus de maintenance peuvent devenir une source de gaspillage non négligeable. Les temps d'attente occasionnés par ce type d'activités doivent être justifiés. Le nombre d'intervenants impliqués ainsi que l'ordre du jour des activités doivent être inclus dans le cycle du processus et visibles afin de pouvoir détecter le gaspillage qui en résulte. Un cas de maintenance avec davantage d'activités de gestion que la moyenne des cas de maintenance peut être repéré rapidement pour vue d'analyse.

3.5. Les mesures de la maintenance logicielle

Il existe plusieurs mesures qui peuvent être appliquées aux activités d'un processus de maintenances logicielles. Suite à l'identification des sources de gaspillage potentiel, il faut réaliser un plan de mesures à appliquer lors des activités des zones contenant du gaspillage. Le plan contiendra les mesures les plus susceptibles de permettre de vérifier les effets de changements sur le processus de maintenance.

Lorsqu'une mesure est choisie, il faut définir l'information nécessaire à cette mesure :

- La mesure dérivée
- La formule de cette mesure
- Le moment où il faut mesurer
- Le mesureur
- Le stockage

- Les données
- L'effort
- La taille
- L'indicateur

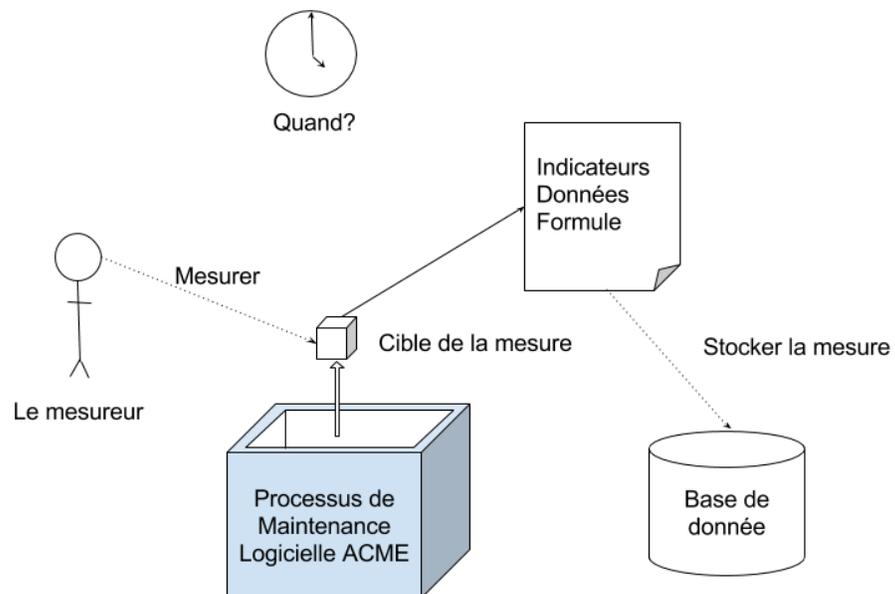


Figure 11 - Mesure d'un élément du processus de maintenances logicielle

L'application d'un plan de mesures permettra de récolter davantage d'informations et aussi d'observer si des changements au processus ont les effets espérés. Il existe plusieurs éléments qui peuvent être mesurés dans un processus de maintenance logicielle.

Tableau 3.7 - Quelques exemples de mesures applicables en maintenance logicielle

La taille d'un cas de maintenance

Le temps de cycle d'un cas de maintenance
Le temps de cycle moyen des cas de maintenance
Ratio de défauts par point de fonction

CHAPITRE 4 OBSERVATIONS ET ANALYSE

4.1. Le processus de maintenance logicielle

[D-A Paquette-A April-A Abran,2006] présentent un questionnaire permettant d'évaluer le niveau de maturité d'un processus de maintenance logicielle en entreprise.

Dans ce chapitre je vais répondre aux questions de chaque étape du questionnaire et fournir une justification des réponses données afin d'analyser la méthodologie du processus de maintenance logicielle. Cette évaluation a comme objectif d'évaluer le processus de maintenance chez ACME et les conclusions tirées seront utilisées pour déterminer les actions pouvant être réalisées afin d'améliorer le processus. Les questions sont telles quelles dans [D-A Paquette-A April-A Abran,2006].

Le modèle employé est constitué d'une série de questionnements sur le processus utilisé dans une entreprise donnée afin de parvenir à déterminer le niveau de maturité du processus.

- Le niveau 0 consiste en un niveau de maturité bas, celui d'un processus informel qui n'implique pas d'activités de maintenance et de support logiciel.
- Le niveau de maturité 1 est celui d'un processus qui réalise la maintenance et le support logiciel, mais qui n'a pas encore standardisé ses pratiques.
- Le niveau 2 est un niveau de maturité dans lequel la maintenance et le support sont réalisés formellement selon des processus standards.

4.2. Évaluation du niveau 0 chez ACME

Ce premier niveau permet d'estimer s'il existe une forme d'activité pour les différents domaines clés du processus. Un processus de maintenance logicielle ne dépassant pas ce

niveau correspond à une organisation ou l'organisation du processus de la maintenance logicielle est marginale et non structurée.

Gestion du processus

PM0 - Q1.0.1 - L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités d'améliorations structurées des processus menant à des améliorations de processus persistantes et contrôlées.

Au cours des dernières années, il y a eu de nombreuses activités d'amélioration du processus de maintenance logicielle chez ACME, en voici quelques-unes :

- Amélioration de l'environnement technologique des outils de la maintenance logicielle ex : intégration de TFS
- Amélioration du processus de maintenance logicielle ex : mise à jour du gabarit des cas de maintenance, projets étudiants de maîtrise en génie logiciel sur l'automatisation des tests et l'analyse du processus de maintenance.

PM 0 - Q2.0.1 - L'unité organisationnelle de la maintenance du logiciel ne fait pas d'activités de définition des processus et des services offerts avec l'objectif d'établir et de maintenir un répertoire courant et utilisable par sa clientèle.

Il existe trois niveaux de support offert aux clients du logiciel maintenu par ACME. Ces trois niveaux du support sont documentés et connus des utilisateurs et les cas ne pouvant pas être réglés par le support aboutissent à la maintenance logicielle après qu'ils aient été soumis au comité de maintenance logicielle dont font partie des utilisateurs qui représentent les clients afin de documenter, estimer et prioriser les cas qui devront être soumis au niveau de la maintenance logicielle.

PM 0 – Q3.0.1 - L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités structurées de formation sur les projets en transition vers la maintenance, les logiciels opérationnels existants, les processus ou la motivation du personnel de la maintenance logiciel.

La formation n'est pas structurée : elle a lieu au besoin sur demande au fur et à mesure en fonction des disponibilités des professionnels en mesure de fournir la formation. Il n'y a pas de processus de motivation du personnel de la maintenance logicielle.

PM 0 – Q4.0.1 - L'unité organisationnelle de la maintenance du logiciel ne fait pas d'activités de mesure de la performance de ses processus.

Le nombre de cas de maintenance est compté; par contre il n'y a pas d'activités de mesures de la performance de ses processus autre que le nombre de cas ouvert versus le nombre de cas fermés. Le niveau de précision de cette mesure ne permet pas d'évaluer correctement la performance dans le processus de maintenance. Les cas sont parfois très gros ou très petits et leur priorité varie en fonction de plusieurs critères non connus des acteurs de la maintenance. L'estimation de la taille des cas et leur priorisation sont approximatives. Il n'est pas possible de tirer des conclusions suite à l'observation du nombre des cas ouverts versus celui des cas fermés autres que de dire qu'il y a davantage de cas ouverts que de cas fermés. Le caractère approximatif des estimations de taille des cas de maintenance ainsi que de leur priorisation en plus de l'inconsistance de leur répertorisation dans le système ne permet pas non plus de tirer des conclusions en fonction de l'observation des données.

PM 0 – Q5.0.1 - L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de collecte des propositions d'amélioration auprès des entités externes (clientèles, autres groupes, interface de la maintenance du logiciel, étalonnage externe, conseillers spécialisés, etc.).

Cette information est recueillie lors de comités qui impliquent des utilisateurs qui représentent les clients.

PM 0 – Q5.0.2 - L'unité organisationnelle de la maintenance du logiciel ne fait pas d'activités d'analyse préalable, d'établissement de constats ni de pilotage des propositions majeures en amélioration des processus ou d'innovations technologiques.

Il y a des activités d'analyse officielle et non officielle sur les possibles améliorations pouvant être faites à la maintenance logicielle. Des projets d'amélioration continue sont délégués aux acteurs du processus dans leurs objectifs devant être réalisés au courant de l'année de travail en cours. Ces activités sont identifiées et planifiées par les acteurs de la maintenance et leurs gestionnaires. Les propositions d'amélioration émergent de l'équipe suite à des observations et des analyses.

PM 0 – Q5.0.3 - L'unité organisationnelle de la maintenance du logiciel ne fait pas de contrôle ni de vérification des bénéfices de ses projets d'amélioration ou d'innovation.

Les bénéfices apportés sont démontrés avant les débuts de la réalisation d'un projet d'amélioration. Ex : au cours de la période d'observation dans l'équipe de maintenance, des problèmes de performances des outils de l'environnement de maintenance ont été détectés. Suite à des benchmarks il a été démontré que les acteurs possédant un ordinateur sans disque SSD souffraient des problèmes de performance nuisibles à leur productivité. Suite à cette activité de « *Benchmarking* », une demande d'amélioration du matériel a été faite.

Maintenance process focus	1.0.1	Non	100
Maintenance process/service definition	2.0.1	Non	100
Maintenance training	3.0.1	Oui	0
Maintenance process performance	4.0.1	Oui	0
Maintenance innovation and deployment	5.0.1	Non	100
	5.0.2	Non	100
	5.0.3	Non	100
			71,4

Figure 12 - Niveau 0 – résultats de la gestion du processus

Gestion des évènements et des requêtes

ERM0 – Q1.0.1 - L'unité organisationnelle de la maintenance du logiciel ne fait pas d'activités de gestion des requêtes et des évènements sur les différents logiciels en maintenance.

Il y a des activités de gestion des requêtes telles que représentées par les figures 1.6 et 1.7.

ERM0- Q2.0.1 - L'unité organisationnelle de la maintenance du logiciel ne fait pas d'activités de planification.

Il y a un comité de maintenance composé d'experts du domaine qui sélectionne les cas devant être réglés prochainement et qui a comme mandat de les prioriser, les estimer et les assigner.

ERM0- Q3.0.1 - L'unité organisationnelle de la maintenance du logiciel ne fait pas d'activités de suivi ni de supervision des engagements, des plans et des logiciels opérationnels.

Une analyste principale est responsable du suivi des cas de maintenance assignés au fil de son exécution. L'analyste principal au dossier s'occupe de gérer les tâches de réalisations et de tests du processus de maintenance et du suivi avec l'analyste principale de la maintenance. Le plan de réalisation est normalisé tel que représenté dans la figure 1.7.

ERM0- Q4.0.1 - L'unité organisationnelle de la maintenance du logiciel n'a pas encore reconnu le besoin d'une entente de service et de contrats formels.

Il y a des ententes de services et des contrats formels.

Event/request management	Event/request management	1.0.1	Non	100
	Maintenance planning	2.0.1	Non	100
	Requests/software monitoring and control	3.0.1	Non	100
	SLA and supplier agreements management	4.0.1	Non	100
				100

Figure 13 - Niveau 0 - Résultats de la Gestion des évènements et des requêtes

Ingénierie des livraisons

EE0- Q1.0.1 - L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de transition du logiciel vers la maintenance.

Il y a techniquement des activités de transition, mais les acteurs de la maintenance ne sont pas informés de ce qui est ajouté à leur charge de responsabilités de maintenance lors des livraisons des nouvelles fonctionnalités ou des modifications aux fonctionnalités existantes.

EE0- Q2.0.1- L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de support opérationnel.

Il existe 3 niveaux de support et la maintenance logicielle est responsable du troisième niveau de support qui s'occupe des problèmes qui n'ont pas pu être réglés par les deux premiers niveaux.

EE0- Q3.0.1- L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de correction et d'évolution du logiciel.

Il y a des activités de maintenance corrective et perfective du logicielle qui ont lieu, telles que des améliorations de performance du système. Suite à la mise en production, les phases de développement se sont poursuivies : il s'agit de la phase d'évolution du système et elle est sous la responsabilité de l'équipe de développement.

EE0- Q4.0.1- L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de vérification et de validation du logiciel.

Il y a des activités de vérification et de validation du logiciel avant de livrer les cas à l'équipe d'assurance qualité. Cette étape est partagée entre le responsable de la réalisation du cas et l'analyste au dossier.

Evolution Engineering	Pre-delivery and transition services	1.0.1	Oui	0
	Operational support services	2.0.1	Non	100
	Software evolution and correction services	3.0.1	Non	100
	Verification and validation	4.0.1	Non	100
				75

Figure 14 - Niveau 0 - Résultats de l'ingénierie des livraisons

Support à l'ingénierie des livraisons

SEE0- Q1.0.1- L'unité organisationnelle de maintenance du logiciel ne fait pas d'activité de management de la configuration du logiciel.

Il y a un outil de gestion de configuration du système qui est utilisé ainsi qu'un « versionage ».

SEE0- Q2.0.1- L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités d'assurance qualité du logiciel.

Il y a une équipe d'assurance qualité et il y a aussi des activités de tests unitaires manuels dans l'équipe de maintenance logicielle.

SEE0- Q3.0.1- L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de mesure ni d'analyse de la maintenance du logiciel.

Il y a une présence d'activités de mesure dont la priorisation et une estimation de l'effort requis pour résoudre un cas de maintenance, par contre la réalisation de ses activités n'a pas lieu dans tous les cas. Il y a des activités d'analyse de la maintenance logicielle et ce projet en fait partie.

SEE0- Q4.0.1- L'unité organisationnelle de maintenance du logiciel ne fait pas de résolution de problème ni d'analyse causale.

Il y a de la résolution de problèmes, mais les analyses causales ne sont pas toujours réalisées formellement. Lorsqu'il y en a, elles le sont au niveau de la réalisation et il n'y a pas forcément de suivi.

SEE0- Q5.0.1- L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de rajeunissement du logiciel.

Il y a des activités de rajeunissement du logiciel, dont un projet de développement de fonctionnalités dans un langage choisi en fonction de sa popularité dans l'industrie ainsi que son efficacité. Le développement est réalisé en mode Agile et les fonctionnalités développées permettent de transférer des utilisateurs vers cette nouvelle architecture technologique.

Support to Evolution Engineering	Configuration and version management	1.0.1	Non	100
	Process, service and software quality assurance	2.0.1	Non	100
	Maintenance measurement and analysis	3.0.1	Non	100
	Causal analysis and problem resolution	4.0.1	Oui	0
	Software rejuvenation, migration and	5.0.1	Non	100
				80

Figure 15 - Niveau 0 -Résultats du support à l'ingénierie des livraisons.

4.3. Évaluation du niveau 1 chez ACME

Le niveau 1 du modèle permet de « Mapper » le travail réalisé par les acteurs de la maintenance dans l'organisation et de déceler les zones où il y a place à amélioration.

Process Management

PM1- Q1.1.1- Dans l'unité organisationnelle de la maintenance du logiciel, l'amélioration s'effectue de manière informelle.

Il y a des traces d'initiatives personnelles visant à améliorer la maintenance logicielle telle que des « *brain-storming* » sur les façons de faire actuelle et les améliorations possibles. Par contre, les améliorations informelles ne sont pas encouragées.

PM1- Q1.1.2- Des initiatives individuelles visent principalement des améliorations des aspects techniques internes de la maintenance du logiciel.

Il y a des traces d'initiatives personnelles visant à améliorer la maintenance logicielle telle que des correctifs de bogue spontané. Par contre, cette pratique n'est pas encouragée.

PM1- Q2.1.1- Les processus et services dans l'unité organisationnelle de la maintenance du logiciel sont informels et basés sur l'expérience des individus.

Les services offerts le sont en fonction d'ententes de service et de façons de faire établies dans l'organisation.

PM1- Q2.1.2- Des initiatives individuelles de documentation des processus et des services de la maintenance visent principalement des aspects techniques particuliers à certains logiciels ou décrivent, dans un format local, les activités d'une unité organisationnelle spécifique de la maintenance du logiciel.

Il y a des initiatives individuelles de documentation des processus de la maintenance qui ont été faites dans l'équipe de maintenance logicielle et sur certains aspects techniques du système, mais cette documentation est destinée à l'interne. Par exemple, il y a eu de la documentation produite sur le cycle de vie du système ainsi que sur les activités de maintenances logicielles réalisées par l'équipe.

PM1- Q3.1.1- La formation dans l'unité organisationnelle de maintenance du logiciel est dispensée quand le besoin se fait sentir.

Il y a des formations proposées à certains acteurs du processus par l'entremise de présentations mises sur pied par des experts du domaine, mais ces activités ont lieu si l'emploi du temps des experts impliqués le permet. Un projet pilote d'activités de formation en ligne est en cours, mais ce n'est pas encore tous les acteurs de la maintenance qui y ont accès.

PM1- Q3.1.2- Des initiatives individuelles de formation visent principalement les aspects techniques internes de la maintenance du logiciel. Une personne sénior est assignée à une personne junior et répond à ses questions.

Lors de l'arrivée de nouveaux acteurs dans l'équipe, un sénior lui est affecté comme parrain et répond à ses questions en plus de lui fournir la formation individuelle nécessaire à son intégration.

PM1- Q3.1.3- Les plans de formation du nouveau personnel traitent de sujets généraux en gestion et en maintenance du logiciel.

Il n'y a pas de plan de formation du nouveau personnel.

PM1- Q4.1.1- Des initiatives individuelles de mesure de processus ou du produit sont effectuées par des individus personnellement intéressés par ce domaine.

Il n'y a pas de mesure de processus et du produit.

PM1- Q4.1.2- Quelques mesures qualitatives des processus et des produits sont collectées.

Il n'y a pas de mesures qualitatives des processus, mais il y a quelques mesures de performance du produit, tel que du temps de traitement de requêtes et d'exécution de fonctionnalités.

PM1- Q5.1.1- La sélection, le déploiement et la vérification des bénéfices de ses projets d'amélioration ou d'innovations dans l'unité organisationnelle de la maintenance du logiciel sont élaborés de manière informelle.

Ces activités doivent être préalablement approuvées par la direction.

PM1- Q5.1.2- Des initiatives individuelles d'amélioration et d'innovation, visant principalement des aspects techniques internes de la maintenance du logiciel, sont accomplies.

Il y a des initiatives individuelles d'amélioration des outils de maintenance développés à l'interne ainsi que des mises à jour de ceux-ci qui sont accomplies. Ces améliorations sont habituellement réalisées par la personne responsable de l'outil concerné.

PM1- Q5.1.3- Les évaluations des nouvelles technologies, méthodologies et nouveaux outils de la maintenance sont réalisées informellement.

Il arrive que des acteurs de la maintenance explorent de nouvelles avenues technologiques, méthodologiques et au niveau des outils utilisés aussi, telles que de nouvelles activités Agiles ou l'utilisation de nouveau Framework tel qu' « *entity framework* » par exemple. Lorsque ce type de recherches est jugé pertinent, il arrive que les résultats soient présentés lors de réunions nommées « *TechTalk* ».

Process management	Maintenance process focus	1.1.1	P: Partially Achieved	33
		1.1.2	P: Partially Achieved	33
	Maintenance process/service definition	2.1.1	N: Not Achieved	0
		2.1.2	P: Partially Achieved	33
	Maintenance training	3.1.1	P: Partially Achieved	33
		3.1.2	F: Fully Achieved	93
		3.1.3	N: Not Achieved	0
	Maintenance process performance	4.1.1	N: Not Achieved	0
		4.1.2	P: Partially Achieved	33
	Maintenance innovation and deployment	5.1.1	N: Not Achieved	0
		5.1.2	P: Partially Achieved	33
		5.1.3	P: Partially Achieved	33
				27

Figure 16 - Résultats du niveau 1 - Process Management

Event/Request Management

ERM1- Q1.1.1- La gestion des requêtes et des évènements, dans l'organisation de maintenance du logiciel, se fait de manière informelle.

Le processus de gestion des requêtes et des évènements est formel.

ERM1- Q1.1.2- Une approche individuelle de gestion des requêtes et des évènements est principalement basée sur des relations personnelles entre le programmeur de la maintenance et une ressource de l'unité organisationnelle du client.

Les programmeurs de la maintenance ne sont pas en contact avec le client. Les contacts avec les clients sont à une étape précédente à la maintenance logicielle.

ERM1- Q2.1.1- La planification dans l'organisation de maintenance du logiciel s'effectue de manière informelle.

La planification de la maintenance logicielle est réalisée formellement.

ERM1- Q2.1.2- Des initiatives individuelles de planifications visent principalement à informer la clientèle verbalement, de la possibilité de traiter une requête spécifique et ponctuelle.

Cette étape a lieu lors des rencontres de planification de la maintenance logicielle qui impliquent les utilisateurs.

ERM1- Q2.1.3- Les requêtes de la clientèle et des transactions de logiciel de l'unité organisationnelle de la maintenance sont traités d'une manière réactive plutôt que d'une manière planifiée.

La planification des requêtes est gérée par l'analyste principal et les analystes responsables des requêtes.

ERM1- Q3.1.1- Le suivi et la supervision, dans l'unité organisationnelle de maintenance du logiciel, sont effectués de manière informelle.

C'est effectué formellement au travers des activités du processus.

ERM1- Q4.1.1- Les ententes avec les clients, les sous-traitants et l'impartiteur sont élaborées à partir de gabarits, de documents et de contrats du fournisseur et des sous-traitants.

ACME possède ses propres modèles d'ententes avec les clients.

Event/request management	Event/request management	1.1.1	N: Not Achieved	0
		1.1.2	N: Not Achieved	0
	Maintenance planning	2.1.1	N: Not Achieved	0
		2.1.2	N: Not Achieved	0
		2.1.3	N: Not Achieved	0
	Requests/software monitoring and control	3.1.1	N: Not Achieved	0
	SLA and supplier agreements management	4.1.1	N: Not Achieved	0

Figure 17 - Résultat du Niveau 1 - Event/Request Management

Evolution Engineering

EE1- Q1.1.1- La transition dans l'unité organisationnelle de maintenance du logiciel est opérée de façon informelle, quand le besoin s'en fait sentir.

La transition est formelle et gérée en fonction de processus établis dans l'entreprise.

EE1- Q2.1.1- Le support opérationnel, dans l'unité organisationnelle de maintenance du logiciel, est fait de façon informelle.

C'est réalisé formellement et divisé en niveau de support 1, 2 et 3.

EE1- Q3.1.1- L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de correction et d'évolution du logiciel d'une manière structurée et coordonnée.

Le processus est structuré et coordonné en suivant des méthodologies en cascades de l'industrie.

EE1- Q4.1.1- L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de vérification et de validation du logiciel de manière structurée et coordonnée.

Ces activités sont réalisées et sont sous la responsabilité de l'équipe d'assurance qualité qui est sous une autre direction que celle de la maintenance logicielle.

Evolution Engineering	Pre-delivery and transition services	1.1.1	N: Not Achieved	0
	Operational support services	2.1.1	N: Not Achieved	0
	Software evolution and correction services	3.1.1	N: Not Achieved	0
	Verification and validation	4.1.1	L: Largely Achieved	68
				17

Figure 18 - Résultat du Niveau 1 - Evolution Engineering

Support to Evolution Engineering

SEE1- 1.1.1- L'unité organisationnelle de maintenance du logiciel n'accomplit pas d'activités de management de la configuration d'une manière structurée et coordonnée.

Les activités de gestion de la configuration sont sous la responsabilité d'experts du domaine nommés par la direction et les activités reliées sont étroitement supervisées.

SEE1- 2.1.1- L'assurance qualité, dans l'unité organisationnelle de maintenance du logiciel, est effectuée de façon informelle.

Ces activités sont sous la responsabilité de l'équipe d'assurance qualité nommée GCQ, cette équipe est sous une autre direction que celle de la maintenance logicielle.

SEE1- 3.1.1- La mesure et l'analyse, dans l'unité organisationnelle de maintenance du logiciel, sont effectuées de façon informelle par chaque gestionnaire.

Les mesures et les analyses sont réalisées par les gestionnaires en fonction de leurs besoins et de ce qu'ils cherchent à démontrer afin de parvenir à diriger de sorte que les objectifs fixés soient compris et puissent être atteints.

SEE1- 4.1.1- L'unité organisationnelle de maintenance du logiciel reconnaît le besoin de détenir un processus de résolution de problème et d'analyse causale.

Il y a un processus de résolution de problème; par contre l'analyse causale ne fait pas partie des activités de maintenance.

SEE1- 5.1.1- Le rajeunissement, dans l'unité organisationnelle de maintenance du logiciel, est effectué de façon informelle, durant un changement spécifique préalable.

Le rajeunissement doit passer par un processus formel et doit être autorisé par la direction.

SEE1- 5.1.2- La retraite d'une fonctionnalité ou d'un logiciel est effectué en délaissant toute l'infrastructure des composants du logiciel et en empêchant son exécution.

Cette activité est réalisée par transition qui peut s'étaler sur plusieurs années en fonction de l'utilisation qui est faite du logiciel. Ce fût le cas lors d'une perte de contrat en 2016 : un client choisit de cesser d'utiliser un système et de garder le code source qui lui appartenait et de ne plus en déléguer la responsabilité d'évolution à ACME. Par contre, la maintenance et le support du système se doivent d'être poursuivis par ACME sur une période de 2 ans.

Support to Evolution Engineering	Configuration and version management	1.1.1	N: Not Achieved	0
	Process, service and software quality assurance	2.1.1	N: Not Achieved	0
	Maintenance measurement and analysis	3.1.1	P: Partially Achieved	33
	Causal analysis and problem resolution	4.1.1	N: Not Achieved	0
	Software rejuvenation, migration and retirement	5.1.1	N: Not Achieved	0
		5.1.2	N: Not Achieved	0
				5,5

Figure 19 - Résultat du Niveau - Support to Evolution Engineering

4.4.

4.5. Évaluation du niveau 2 chez ACME

Process Management

PM2- Q1.2.1- Un programme d'amélioration des processus de la qualité a été initié à l'échelle de toute l'organisation, et les gestionnaires de la maintenance du logiciel sont sensibilisés et ont eu une formation initiale sur ce programme.

Les gestionnaires ont été informés des améliorations qui doivent avoir lieu lors de la transition vers l'Agilité. Ils sont qualifiés et possèdent des connaissances sur le sujet.

PM2- Q1.2.2- Un représentant de la maintenance est assigné pour planifier et coordonner les activités d'améliorations.

Ces activités sont gérées par le directeur qui distribue les tâches à réaliser lors des activités d'améliorations.

PM2- Q1.2.3- Un sondage, des services et produits de la maintenance sont utilisés pour recueillir des données qui orientent les améliorations possibles des processus de la maintenance.

Je n'ai pas eu connaissance qu'il existe ce genre de sondage. Il y a des données sur la satisfaction des clients qui sont présentées à l'équipe lors des réunions mensuelles, mais rien qui oriente sur les améliorations possibles des processus de maintenance.

PM2- Q1.2.4- Les observations, commentaires et plaintes de la clientèle et des autres groupes interfaces internes (développeurs, infrastructure et opérations, bureau d'aide, sous-traitant, etc.) sont utilisés pour recueillir des données qui orientent les améliorations possibles des processus de la maintenance.

Il existe ce genre de retour d'information qui motivent des améliorations faites au système et réalisées parfois lors de maintenance perfective. Ex : améliorations de performance.

PM2- Q1.2.5- Les comparaisons des profils des logiciels applicatifs opérationnels et des processus d'étalonnage internes pertinents sont utilisées pour recueillir des données qui orientent les améliorations possibles des processus de la maintenance.

Il existe ce genre de comparaison qui est utilisé pour parvenir à développer ou acquérir de nouveaux outils ou de nouveaux matériels pour améliorer les processus de maintenance. Ex : achat de nouvelles machines.

PM2- Q1.2.6- Les données des défaillances de logiciels applicatifs sont utilisées afin de recueillir des données pour orienter les améliorations possibles des processus de la maintenance et de ses groupes interfaces internes (développeurs, infrastructure et opérations, bureau d'aide, sous-traitant, etc.).

Des données de performances et de données erronées lors des utilisations en production et aussi lors des tests sont collectées et analysées : j'ai observé une utilisation de ces données pour l'amélioration du système, mais pas de trace officielle d'utilisation de cette information pour améliorer le processus de maintenance lui-même autre que le total de cas de maintenance qui en résulte.

PM2- Q1.2.7- Le groupe de la maintenance fait l'objet d'audits internes (d'auditeur interne, de l'ISO 9001:2000 ou d'audit de qualité) réguliers et les résultats sont utilisés pour orienter les améliorations possibles des processus de la maintenance.

Il y a des projets étudiants, mais pas d'audit qui sont réalisés régulièrement.

PM2- Q1.2.8- Une évaluation de certains processus, en tout ou en partie, a été effectuée. Une unité organisationnelle de la maintenance a fait l'objet d'évaluations de la capacité des processus et les résultats sont utilisés pour orienter les améliorations possibles des processus de la maintenance.

Ce projet faisait partie de l'évaluation du processus de maintenance logicielle; par contre ACME n'y participe plus et a décidé de l'abandonner. Il y a un mot d'ordre de transition vers l'agilité; par contre la majorité des acteurs qui y ont été impliqués et qui y étaient favorables ont quitté l'organisation.

PM2- Q1.2.9- Une liste des améliorations possibles, par ordre de priorités, est développée et sert de guide pour des améliorations possibles.

Il semble exister ce type de liste; par contre elle n'est pas partagée avec les acteurs de la maintenance, ce qui est indiqué comme devant être réalisé prochainement semble être la priorité.

PM2- 1.2.10- La liste des améliorations possibles fait l'objet de discussions et d'actions auprès des dirigeants intermédiaires et séniors. Les activités et projets d'améliorations sont planifiés dans le cadre des budgets et des opérations courantes.

Les activités qui sont réalisées sont planifiées d'avance et incluses dans le budget.

PM2- Q1.2.11- Des améliorations de certains processus ont été initiées. Les planifications annuelles de chaque unité organisationnelle de la maintenance reflètent les activités d'améliorations planifiées et effectuées durant une année.

Les améliorations ont lieu au niveau des unités organisationnelles elles-mêmes. La stratégie globale est communiquée dans les communications de l'organisation, mais la réalisation de cette stratégie est laissée aux choix des dirigeants des unités concernées.

PM2- Q2.2.1- Il existe au moins une description des processus et des services de la maintenance qui sont utilisés par la clientèle et l'unité organisationnelle de la maintenance du logiciel de l'organisation.

Cette description existe dans l'équipe étudiée lors de ce projet et est décrite au chapitre 1.

PM2- Q2.2.2- Certaines parties des processus et des services de la maintenance sont documentées et utilisées par les unités organisationnelles de la maintenance du logiciel de l'organisation.

Cette documentation existe et est dispersée dans le répertoire de travail de l'équipe et les sites de leur intranet : elle détaille les différentes activités réalisées lors du processus de maintenance.

PM2- Q2.2.3- Il existe des activités visant à élaborer des processus et des services normalisés au sein de l'organisation de la maintenance du logiciel.

Les grandes lignes sont communiquées par l'organisation. Il en revient aux directeurs de s'assurer qu'elles soient réalisées.

PM2- Q2.2.4- L'organisation de la maintenance reconnaît et encourage l'utilisation de processus / services normalisés et l'utilisation de normes pertinentes à la maintenance du logiciel (c'est-à-dire l'ISO 12207, l'ISO 14764 et l'IEEE 1219).

Le processus officiel de l'organisation fut normalisé et a été accrédité de certaines certifications telles que CMMI niveau 3, mais il n'est pas possible de déterminer si c'est réalisé. Dans l'équipe étudiée lors de ce projet, il semble que le processus suive effectivement certaines normes, mais je ne peux pas m'avancer à dire si c'est volontaire ou si ce modèle a émergé avec l'entreprise qui a initialement développé et maintenu le projet qui lui a été racheté par ACME.

PM2- Q3.2.1- Le personnel de la maintenance se familiarise et refamiliarise avec le logiciel qu'il maintient sur une base régulière.

Il y a une rotation des développeurs sur 6 mois entre la maintenance et le développement; par contre il n'existe pas de familiarisation officielle. Les développeurs viennent à connaître les bouts du système sur lesquels ils ont à travailler, mais l'information sur les autres parties développées provient des échanges d'informations qui ont lieu en cours de projets. Si un développeur n'accorde pas d'attention à ce qui est développé et maintenu en dehors de ses tâches et qu'il ne prend pas l'initiative d'acquérir l'information et de se familiariser sur les parties du logiciel sur lesquels il ne travaille pas, il ne les connaîtra pas.

PM2- Q3.2.2- Le personnel de la maintenance est formé et motivé afin de donner une bonne performance dans le processus et dans son rôle.

La formation est donnée au cas par cas par du transfert de connaissance entre les développeurs lors de l'arrivée d'un nouveau membre ou sur demande. Les indicateurs de performance et de

rentabilité sont communiqués régulièrement lors des rencontres mensuelles pour motiver le personnel. Il y a un programme de participation au profit (bonus annuel), mais il n'y en a que très rarement qui soit distribuée et quand il y en a, elle est tellement faible qu'elle devient un facteur démotivant (inférieur à 1% du salaire annuel en moyenne). Les augmentations de salaire sont au minimum de ce qui est rencontré dans l'industrie, voire inexistantes et demandent des performances qui dépassent les attentes pour être obtenues (maximum de 3% par année). Le résultat de performance de l'équipe, qui s'est pour la plupart affichée en vert (excellent) lors des rencontres annuelles, ne concorde pas avec les bonus (1% du salaire brut et moins) et augmentations qui sont distribués (3% dans le meilleur cas et 2% et moins dans les cas moyens) : ceci s'avère être un autre facteur démotivant. Le haut roulement des employés est composé principalement d'employés avec moins de deux années d'ancienneté, il n'y a pas d'effort de fait pour garder ceux que l'équipe voudrait voir rester : ils sont remplacés par des consultants en provenance de l'étranger. Les problèmes sont connus, mais vu la lourdeur administrative de l'organisation les améliorations prennent beaucoup de temps avant de se concrétiser ce qui a aussi un effet démotivant sur l'équipe.

PM2- Q3.2.3- De la formation en communication avec la clientèle est offerte au personnel de la maintenance du logiciel.

Je n'ai pas eu connaissance de formations du genre.

PM2- Q3.2.4- Les comparaisons de formation des ressources du logiciel sous forme d'étalons internes pertinents sont utilisées afin de recueillir des données qui orientent les efforts de formation des ressources de la maintenance.

Ça émerge suite aux commentaires des développeurs formés ou qui donnent la formation, mais ce n'est pas formel.

PM2- Q3.2.5- Les ressources financières sont disponibles au niveau de l'unité organisationnelle pour l'éducation et la formation de chaque ressource.

Il y a un budget alloué pour les nouveaux arrivants, mais pas pour chaque ressource autre que les derniers arrivés. Il y a un projet pilote de formation en ligne auquel quelques ressources ont participé.

PM2- Q3.2.6- Il existe un plan d'éducation et de formation pour l'organisation et pour chaque logiciel. Ce plan énonce les besoins de formation, les cours offerts, les crédits, les ressources disponibles et le calendrier des activités.

Il existe un plan de formation destiné aux clients, mais il n'y a pas ce type de formation planifiée pour les acteurs de la maintenance.

PM2- Q3.2.7- Du temps pour la formation personnelle est planifié.

Ce temps est disponible sur demande et accordé puis planifié au besoin en fonction des priorités.

PM2- Q3.2.8- La personne sénior fait l'introduction des nouveaux employés et de la relève pour la maintenance des logiciels applicatifs sous sa responsabilité.

Il y a un système de parrainage et les autres développeurs sont disponibles au besoin sur demande.

PM2- Q3.2.9- Les besoins de formation technique et de gestion sont précisés pour chaque projet en transition (c'est-à-dire nature de la formation, personnes visées, moment où la formation est offerte, etc.).

Si c'est le cas, ce n'est pas communiqué aux développeurs de la maintenance.

PM2- Q3.2.10- Les personnes qui travaillent à la transition d'un projet reçoivent la formation jugée appropriée par le développeur.

Si c'est le cas, ce n'est pas communiqué aux développeurs de la maintenance.

PM2- Q3.2.11- L'élaboration du matériel de formation interne est effectuée conformément à une procédure locale et documentée.

Non. Il existe un wiki, mais il n'est pas maintenu et l'information est pour la plupart inutilisable, car incomplet ou n'est plus à jour.

PM2- Q3.2.12- L'utilisateur (et certains autres intervenants) reçoit suffisamment d'éducation et de formation afin de permettre l'utilisation autonome du logiciel applicatif.

Il y a des activités de formations formelles qui sont offertes aux utilisateurs et aux vendeurs, mais il n'existe rien de formel d'offert aux nouveaux intervenants de la maintenance.

PM2- Q3.2.13- L'utilisateur reçoit de l'information et de la formation sur les processus d'interface avec le bureau d'aide et la maintenance du logiciel.

Il y a un processus de support qui est communiqué aux utilisateurs du logiciel.

PM2- Q4.2.1- Le personnel de la maintenance se familiarise et refamiliarise avec le logiciel qu'il maintient sur une base régulière.

Il n'y a rien de formel et il va connaître les parties sur lesquelles il va avoir eu à travailler en fonction de ses compétences et de la formation qu'il aura eu.

PM2- Q4.2.2- Le personnel de la maintenance est formé et motivé afin de donner une bonne performance dans le processus et dans son rôle.

La formation est principalement octroyée aux nouveaux et octroyée au besoin en général. Les performances sont évaluées annuellement et les améliorations potentielles formulées lors de ses évaluations. Les motivations sont discutées à la question **PM2- Q3.2.2** et les facteurs démotivants semblent être plus grands.

PM2- Q4.2.3- De la formation en communication avec la clientèle est offerte au personnel de la maintenance du logiciel.

Non.

PM2- Q5.2.1- On évalue les technologies, les méthodologies et les nouveaux outils ayant du potentiel pour les appliquer dans l'unité organisationnelle de la maintenance du logiciel.

C'est discuté à l'occasion dans l'équipe et lors de rencontres dites « *Tech-talk* ». La plupart des analyses et discussions sur le sujet sont identifiées comme du « chialage » si elles ne proviennent pas d'employés séniors (plusieurs années d'anciennetés dans l'équipe, les années externes ne sont pas prises en compte dans le calcul du séniorat) et seront ignorées.

PM2- Q5.2.2- Les nouvelles technologies sont évaluées et introduites au niveau d'une requête.

Les nouvelles technologies émergent de demandes des développeurs, mais le processus n'est pas formalisé.

PM2- Q5.2.3- Des améliorations de certains processus de la maintenance ont été initiées d'une manière contrôlée au niveau de l'unité organisationnelle de la maintenance.

Les améliorations sont analysées et testées par l'analyste principal de la maintenance de concert avec le directeur et par la suite elles sont intégrées au processus de maintenance.

Process management	Maintenance process focus	1.2.1	F: Fully Achieved	93
		1.2.2	P: Partially Achieved	33
		1.2.3	N: Not Achieved	0
		1.2.4	L: Largely Achieved	68
		1.2.5	L: Largely Achieved	68
		1.2.6	P: Partially Achieved	33
		1.2.7	N: Not Achieved	0
		1.2.8	P: Partially Achieved	33
		1.2.9	P: Partially Achieved	33
		1.2.10	L: Largely Achieved	68
		1.2.11	P: Partially Achieved	33
	Maintenance process/service definition	2.2.1	F: Fully Achieved	93
		2.2.2	L: Largely Achieved	68
		2.2.3	P: Partially Achieved	33
		2.2.4	P: Partially Achieved	33
	Maintenance training	3.2.1	P: Partially Achieved	33
		3.2.2	P: Partially Achieved	33
		3.2.3	N: Not Achieved	0
		3.2.4	P: Partially Achieved	33
		3.2.5	P: Partially Achieved	33
		3.2.6	L: Largely Achieved	68
		3.2.7	P: Partially Achieved	33
		3.2.8	F: Fully Achieved	93
		3.2.9	N: Not Achieved	0
		3.2.10	N: Not Achieved	0
		3.2.11	N: Not Achieved	0
		3.2.12	L: Largely Achieved	68
		3.2.13	L: Largely Achieved	68
	Maintenance process performance	3.2.1	P: Partially Achieved	33
		3.2.2	P: Partially Achieved	33
		3.2.3	N: Not Achieved	0
	Maintenance innovation and deployment	5.2.1	P: Partially Achieved	33
		5.2.2	P: Partially Achieved	33
5.2.3		L: Largely Achieved	68	
				39,735

Figure 20 - Résultat du Niveau 2 - Process Management

Event/Request Management

ERM2- Q1.2.1- Il y a un point de contact unique pour fournir une aide directe aux clients.

Il y a 3 niveaux de support offert aux clients.

ERM2- Q1.2.2- Chaque requête et évènement font l'objet de création d'une requête de modification ou d'un rapport de problème du logiciel qui sert de billet de travail de la maintenance.

Oui, ces billets sont enregistrés dans le système nommé C2 et un cas de maintenance est créé dans TFS si la requête demande une intervention de la maintenance logicielle.

ERM2- Q1.2.3- Chaque requête et évènement font l'objet de catégorisation, de choix, de priorités et d'une estimation préliminaire de leur taille et de leur ampleur.

L'estimation et l'ampleur ne sont pas toujours réalisées.

ERM2- Q1.2.4- Les requêtes de modifications acceptées sont assignées. D'une manière préliminaire, à une version future du logiciel.

Oui, c'est évalué et décidé lors de réunions du comité de maintenance.

ERM2- Q2.2.1- Au niveau de l'organisation, il existe une politique relative aux plans des unités organisationnelles. Ces plans comprennent l'objet, la portée, les buts, les objectifs, les biens livrables et certains autres éléments importants.

Oui, c'est géré à plus haut niveau par les gestionnaires.

ERM2- Q2.2.2- Le plan de la maintenance est élaboré et mis à jour annuellement, conformément à une procédure documentée.

Non.

ERM2- Q2.2.3- Les activités de planification de l'unité organisationnelle de la maintenance suivent les normes de l'organisation et sont coordonnées avec le développeur, les infrastructures et les opérations.

Les activités sont planifiées et semblent respecter des normes d'entreprise, mais j'ignore si la façon de faire est un standard au niveau de toute l'organisation.

ERM2- Q2.2.4- La planification annuelle introduit les possibilités de rajeunissement pour les logiciels de la maintenance.

Oui et cette planification est mise à jour en cours d'année : il y a des objectifs à court, moyen et long terme pour le rajeunissement. L'intention est d'adopter une approche d'amélioration continue.

ERM2- Q2.2.5- La planification des infrastructures et les outils de soutien d'ingénierie d'évolution du logiciel sont préparés annuellement.

Dans la même optique que la question **ERM2- Q2.2.4.**

ERM2- Q2.2.6- Les activités de planification des différents aspects du projet (partie logicielle et partie administrative, par exemple) sont amorcées dès les premières étapes des activités de planification globale du projet et effectuées en parallèle avec celle-ci.

La planification est sous la responsabilité des chefs de projets et requiert les approbations de la haute direction pour l'allocation des budgets. Les activités de planifications sont découpées au début de chaque phase de projets et mises à jour, lorsque requis.

ERM2- Q2.2.7- Tous les groupes visés par la planification globale du projet collaborent ensemble à la planification globale de la transition.

Cette planification est gérée par les directeurs, les chefs de projets et les chefs d'équipes impliquées.

ERM2- Q2.2.8- L'échéancier de la transition est établi à partir de la planification du projet et de développement.

Oui, les échéanciers de transition sont planifiés d'avance et les dates sont ajustées lorsque requis.

ERM2- Q2.2.9- Le plan de transition décrit la relation avec les autres plans du projet de développement.

Je n'ai pas fait d'observations là-dessus.

ERM2- Q2.2.10- Des activités de soutien aux méthodes, aux techniques et aux outils d'essais et de la maintenance sont définies au plan de transition.

Il y a des batteries de tests devant être exécutées par l'équipe d'assurance qualité pour s'assurer que la transition a été correctement réalisée. Il y a aussi une stratégie de tests unitaires manuels qui a été implantée dans l'équipe de maintenance et de développement au cours des derniers mois.

ERM2- Q2.2.11- Les risques relatifs aux aspects techniques, aux coûts, aux bénéfices, aux ressources et à l'échéancier sont identifiés, évalués et documentés.

Les risques ne sont pas identifiés lors des activités de maintenance, les coûts ne sont pas non plus estimés avant le début des travaux; les bénéfices sont discutés, mais il n'y a pas de documentation formelle.

ERM2- Q2.2.12- Il existe une fonction de relève : les rôles sont définis et des personnes y sont assignées.

Il y a une possibilité de faire des « *Rollback* » lors des cas d'urgences, mais pas de version de relèves.

ERM2- Q2.2.13- Les données, les programmes, les langages de contrôle et la documentation critique sont copiés et emmagasinés à l'extérieur du site de production, et ce, sur une base régulière.

Les serveurs de production sont physiquement ailleurs qu'au site de production.

ERM2- Q2.2.14- Un plan de relève détaillé (incluant le centre de traitement et le réseau de communication) est testé et révisé périodiquement, notamment après tout changement important à l'infrastructure.

J'ai n'ai pas recueilli d'information là-dessus.

ERM2- Q2.2.15- Les modifications et les mises à niveau des logiciels sont appliquées par le mainteneur dès qu'elles sont nécessaires.

Les modifications et les mises à niveaux peuvent parfois prendre beaucoup de temps. La lourdeur administrative de l'organisation fait qu'il peut s'écouler beaucoup de temps avant

qu'une modification ou une mise à niveau soient réalisées. Les impacts immédiats sur la production doivent pouvoir être clairement démontrés pour qu'une mise à niveau ou une modification aient lieu immédiatement. Les modifications et mises à niveau qui n'atteignent pas les productions vont être mises dans la liste et il est difficile de savoir quand elles auront lieu.

ERM2- Q2.2.16- Un responsable de la maintenance est désigné comme responsable des activités de planification sur les requêtes visant son logiciel opérationnel.

C'est la responsabilité de l'analyste principal et des acteurs au support.

ERM2- Q2.2.17- Le plan de maintenance est utilisé pour faire le suivi des activités et pour communiquer l'état de situation des requêtes.

C'est réalisé et répertorié dans l'outil TFS, mais il y a quelques activités qui ne sont pas réalisées dans tous les cas tels que l'estimation, l'analyse, les détails de la réalisation et les tests unitaires.

ERM2- Q2.2.18- L'assignation des priorités aux différentes requêtes s'effectue en étroite collaboration avec le client, conformément à une procédure documentée.

Les priorités sont discutées et établies lors du comité de maintenance.

ERM2- Q2.2.19- Toutes les questions relatives au fonctionnement intergroupe sont documentées et négociées. Les problèmes non résolus sont acheminés aux gestionnaires concernés.

Les problèmes jugés non résolus sont rapidement acheminés aux gestionnaires; par contre leur documentation est défailante.

ERM2- Q2.2.20- L'unité organisationnelle de la maintenance met en œuvre un processus de gestion de la capacité fondé sur l'ensemble des planifications et des ententes de services.

J'ai n'ai pas observé ce genre de processus formel, mais le personnel assigné au support et à la maintenance est ajusté au besoin en fonctions du contexte, tel que les périodes de pointes, les transitions majeures.

ERM2- Q3.2.1- Les ressources de la maintenance assurent la surveillance et la coordination des activités de résolution des problèmes techniques des logiciels en production.

La surveillance est réalisée par le support et la coordination de mises en productions implique les acteurs de la maintenance, dont l'analyste principal.

ERM2- Q3.2.2- Les engagements des différents intervenants, tel que convenu dans les planifications de la maintenance du logiciel, font l'objet d'un suivi périodique.

Le suivi est sous la responsabilité de l'analyste principal de la maintenance et des analystes responsables des dossiers.

ERM2- Q3.2.3- Le suivi et la supervision des niveaux de service des logiciels en production, dans l'unité organisationnelle de maintenance du logiciel, sont effectués durant la réunion interne hebdomadaire du progrès du travail.

Il y a effectivement des rencontres qui impliquent les acteurs de la maintenance et du support, par contre il s'agit de deux rencontres séparées et elles n'ont pas lieu chaque semaine.

ERM2- Q3.2.4- Des revues techniques et des échanges intergroupes ont lieu périodiquement avec les responsables des logiciels de maintenance.

Ce genre de rencontre a lieu au développement, mais pas en maintenance.

ERM2- Q3.2.5- L'échéancier de la transition fait l'objet d'un suivi et des mesures correctives sont apportées, le cas échéant.

Les transitions sont supervisées par les analystes, les gestionnaires et les intégrateurs. Il y a des mesures correctives, lorsque nécessaires. Il y a des échéanciers qui peuvent pousser à remettre les transitions à une date ultérieure, lorsque non respectés.

ERM2- Q3.2.6- Les engagements de la maintenance font l'objet d'un suivi et des mesures correctives sont apportées, le cas échéant.

Les cas de maintenance devant être livrés doivent passer les tests de l'assurance qualité ou sinon être remis à plus tard.

ERM2- Q4.2.1- Des discussions préparatoires ont lieu sur la mise en œuvre d'une entente de services plus formelle et de contrats.

Il y a des représentants du marketing qui sont responsables de la vente des différents niveaux de produits disponibles et les contrats sont réalisés en fonctions des demandes du client et de sa capacité de payer. Les ententes sont des contrats formels, autant avec les clients qu'avec les sous-traitants.

ERM2- Q4.2.2- Un représentant de la maintenance est assigné pour planifier et coordonner l'activité des comptes clients.

Je n'ai rien observé là-dessus.

ERM2- Q4.2.3- Les deux parties (Client et fournisseur) s'entendent sur la nécessité d'une entente de services plus formelle et de contrats s'appliquant à la maintenance du logiciel.

Voir la question **ERM2- Q4.2.1**

ERM2- Q4.2.4- Le mainteneur choisit et réévalue ses fournisseurs. Ce choix est basé sur une évaluation de leurs capacités à rencontrer les conditions spécifiées et les critères établis.

La réévaluation n'est pas une activité régulière, mais elle semble se dérouler telle que décrite dans la question lorsque c'est le cas.

ERM2- Q4.2.5- Une entente sur le niveau de services et des contrats de sous-traitance de la maintenance du logiciel est définie, documentée et approuvée.

Voir la question **ERM2- Q4.2.1**

ERM2- Q4.2.6- L'entente conclue entre le fournisseur et le client établit le fondement de la gestion de l'entente.

Voir la question **ERM2- Q4.2.1**

ERM2- Q4.2.7- Les modifications à la portée du travail confié à la sous-traitance, aux modalités du contrat de sous-traitance et aux ententes de services sont effectuées conformément à une procédure documentée de revue des engagements concernant tous les intervenants.

Voir la question **ERM2- Q4.2.1**

ERM2- Q4.2.8- Des revues formelles portant sur les réalisations et les résultats des services fournis sont effectuées à des jalons choisis, ainsi qu'à la fin de certaines étapes du développement, de la maintenance ou de l'exploitation, conformément à une procédure documentée.

Il semble que oui vu le niveau de formalisation observé dans l'entreprise et la grande quantité d'informations et de réunions sur les avancements des différents projets, mais ça semble être laissé à la discrétion des gestionnaires.

ERM2- Q4.2.9- L'organisation de la maintenance établit la politique de facturation des services.

Ça semble être inclus dans les modalités des contrats vu que l'équipe est aussi responsable de l'évolution du produit dont ACME est propriétaire et vend les droits d'utilisation.

ERM2- Q4.2.10- L'organisation de la maintenance introduit une facturation pour ses services.

Voir la question **ERM2- Q4.2.9**

Event/request management	Event/request management	1.2.1	F: Fully Achieved	93
		1.2.2	F: Fully Achieved	93
		1.2.3	P: Partially Achieved	33
		1.2.4	F: Fully Achieved	93
	Maintenance planning	2.2.1	F: Fully Achieved	93
		2.2.2	N: Not Achieved	0
		2.2.3	L: Largely Achieved	68
		2.2.4	F: Fully Achieved	93
		2.2.5	F: Fully Achieved	93
		2.2.6	F: Fully Achieved	93
		2.2.7	F: Fully Achieved	93
		2.2.8	F: Fully Achieved	93
		2.2.9	N: Not Achieved	0
		2.2.10	F: Fully Achieved	93
		2.2.11	N: Not Achieved	0
		2.2.12	P: Partially Achieved	33
		2.2.13	F: Fully Achieved	93
		2.2.14	N: Not Achieved	0
		2.2.15	L: Largely Achieved	68
		2.2.16	F: Fully Achieved	93
	2.2.17	L: Largely Achieved	68	
	2.2.18	F: Fully Achieved	93	
	2.2.19	P: Partially Achieved	33	
	2.2.20	P: Partially Achieved	33	
	Requests/software monitoring and control	3.2.1	F: Fully Achieved	93
		3.2.2	F: Fully Achieved	93
		3.2.3	P: Partially Achieved	33
		3.2.4	N: Not Achieved	0
		3.2.5	L: Largely Achieved	68
		3.2.6	F: Fully Achieved	93
	SLA and supplier agreements management	4.2.1	F: Fully Achieved	93
		4.2.2	F: Fully Achieved	93
		4.2.3	F: Fully Achieved	93
		4.2.4	L: Largely Achieved	68
4.2.5		F: Fully Achieved	93	
4.2.6		F: Fully Achieved	93	
4.2.7		F: Fully Achieved	93	
4.2.8		L: Largely Achieved	68	
4.2.9		L: Largely Achieved	68	
4.2.10		L: Largely Achieved	68	
				64,07

Figure 21 - Résultat du Niveau 2 - Event/Request Management

Evolution Engineering

EE2- Q1.2.1- L'unité organisationnelle de la maintenance communique avec le développeur sur la transition du logiciel.

Dans le cas où il y a des problèmes lors de la transition, le développeur est responsable pour les régler lorsque ça concerne un bout sur lequel il a travaillé. S'il n'a pas le temps, il doit tout de même donner des disponibilités pour aider à résoudre le problème.

EE2- Q1.2.2- L'unité organisationnelle de la maintenance s'assure de gérer la relation avec le développeur et le client pour qu'il n'y ait pas de conflits, de résistance ou de plaintes sur la transition du logiciel.

Les développeurs ne sont pas en contact avec les clients.

EE2- Q1.2.3- L'unité organisationnelle de la maintenance communique avec l'unité organisationnelle des achats (ou directement avec le développeur) pour s'assurer que les considérations de la transition de la maintenance du logiciel figurent au contrat, au plan et à l'horaire du projet de développement.

Le logiciel est développé et maintenu par la même équipe et le développement est géré par le chef de projets. Le plan de projet est soumis à la direction en cours de projet afin d'être approuvé.

EE2- Q1.2.4- L'unité organisationnelle de la maintenance communique avec les clients pour s'assurer que les étapes et les activités de la transition sont claires. De plus, elle introduit les notions de base de l'entente de service.

Cette étape-là n'est pas gérée par la maintenance, mais par d'autres équipes.

EE2- Q1.2.5- L'unité organisationnelle de la maintenance développe, adapte et utilise une liste de vérification sur les sujets et activités clés de la transition logicielle.

C'est réalisé par l'équipe d'assurance qualité.

EE2- Q1.2.6- Les ressources de la maintenance du logiciel qui prendront en charge le logiciel connaissent les activités de la transition spécifique du projet de développement, en discutent et s'en informent.

Non, cette étape est opaque à ceux qui ne travaillent pas dessus.

EE2- Q1.2.7- L'entité organisationnelle de la maintenance s'assure de L'efficacité de la formation de maintenance fournie par le développeur (infrastructure, architecture, logiciel, documentation) et suivie par le personnel de maintenance, avant la transition du logiciel.

Non

EE2- Q1.2.8- L'entité organisationnelle de la maintenance évalue l'efficacité de la formation utilisateur suivie par la clientèle (accès. Fonctionnalité, fonction d'aide et documentation utilisateur) et prescrite par le développeur, avant la transition finale du logiciel, et en discute.

Non

EE2- Q1.2.9- Les ressources de la maintenance du logiciel qui prendront en charge le logiciel de transition s'assurent d'obtenir la documentation qui servira à l'équipe de la maintenance pendant la transition du logiciel.

La documentation est sous la responsabilité des analystes et il ne semble pas y avoir ce type de transition formelle du développement vers la maintenance.

EE2- Q1.2.10- Les ressources de la maintenance du logiciel qui prendront en charge le logiciel s'assurent d'obtenir la configuration finale du code source qui servira à l'équipe de la maintenance du logiciel à la suite de la transition.

Le code source est dans un gestionnaire de source et versionné de sorte à ce qu'il soit disponible pour la maintenance lorsqu'il est monté en production.

EE2- Q1.2.11- Les ressources de la maintenance du logiciel qui prendront en charge le logiciel s'assurent d'obtenir la liste officielle des incidents des essais d'acceptation, leurs catégories et leur statut afin d'identifier et de rendre visible les requêtes initiales qui sont transférées à l'équipe de la maintenance à la suite de la transition.

Il y a des cas de maintenance de créés lorsque des essais rencontrent des problèmes.

EE2- Q2.2.1- Le calendrier d'opération existe et est élaboré sur la base de la capacité des ressources et de la charge de travail pour l'ensemble des logiciels.

Oui et il est ajusté au besoin.

EE2- Q2.2.2- Le mainteneur et l'unité organisationnelle de la maintenance connaissent leurs responsabilités conjointes de surveillance des logiciels opérationnels. Le mainteneur surveille de façon proactive la capacité mémoire des liens de télécommunications et de l'espace disque des logiciels.

Oui et le support va créer des cas de maintenance au besoin.

EE2- Q2.2.3- Le mainteneur produit des rapports de surveillance des caractéristiques clés de chaque service et de chaque logiciel opérationnel.

Au besoin en fonction de la période de l'année (la demande).

EE2- Q2.2.4- L'horaire de support est créé et publié, et on s'assure de la couverture du service selon les termes des ententes de services.

Oui

EE2- Q2.2.5- Le mainteneur possède l'expertise de la fonctionnalité et des règles d'affaires contenues dans les logiciels.

Le taux de roulement élevé dans l'équipe en plus de la documentation désuète tant au niveau du contenu que technologique en plus de l'absence de formation des développeurs sur le domaine d'affaire empêche de pouvoir développer une expertise à ce niveau. Dans l'équipe une personne qui a déjà travaillé une fois sur quelque chose est identifiée comme l'expert. L'expertise s'acquiert sur le tas.

EE2- Q2.2.6- Le mainteneur produit des rapports ad hoc sur requête.

La documentation est réalisée si le temps est disponible et sur demande dans la plupart des cas. L'information est dispersée dans TFS et C2 la plupart du temps.

EE2- Q3.2.1- La conception détaillée élabore les solutions aux exigences de l'utilisateur et est développée conformément à une procédure documentée.

Oui et la conception est vérifiée par les analystes.

EE2- Q3.2.2- Les activités d'investigation et de conception d'une défaillance (identifiées sur un rapport de problèmes) sont effectuées selon une procédure documentée.

Pas dans tous les cas, ça va avoir lieu si l'analyste a le temps de le faire et les développeurs s'aventurent rarement à le faire, car ils n'en obtiennent pas le crédit et ce n'est pas une initiative appréciée par les analystes.

EE2- Q3.2.3- Les activités de mise en œuvre (programmation) sont effectuées conformément à une procédure documentée.

Lorsque la procédure est documentée et les erreurs seront détectées seulement à l'étape de l'assurance qualité dans la plupart des cas.

EE2- Q3.2.4- Le partage des rôles et des responsabilités en matière de données est formellement défini pour le concepteur du logiciel, le gestionnaire des données et l'administrateur des bases de données.

Ces rôles ne sont pas définis dans l'équipe.

EE2- Q3.2.5- Les langages de programmation originaux du logiciel sont utilisés pour les logiciels maintenus.

Oui.

EE2- Q3.2.6- Les interrelations (traçabilité) entre les procédures administratives, les modules, les processus et les données sont documentées.

C'est dans la documentation fonctionnelle, par contre elle n'est pas à jour dans bien des cas et le supporte technologique désuet diminue sa convivialité.

EE2- Q3.2.7- Les tests unitaires doivent être exécutés pour chaque composant qui fait l'objet d'une modification.

Non, les tests unitaires manuels commencent à être écrits lors du développement, mais ce n'est pas encore rendu au niveau de la maintenance.

EE2- Q3.2.8- Les essais d'intégration doivent être effectués pour chaque composant qui fait l'objet d'une modification.

Oui

EE2- Q3.2.9- La documentation de l'utilisateur est soumise à l'essai par le mainteneur pendant les essais unitaires et les essais d'intégration.

Cette étape est réalisée par l'équipe d'assurance qualité.

EE2- Q3.2.10- La documentation interne du logiciel est élaborée conformément à une procédure documentée.

Il y a des gabarits utilisés qui sont ajustés en fonction des besoins par les analystes.

EE2- Q3.2.11- La documentation externe du logiciel est élaborée conformément à une procédure documentée.

Voir la question **EE2- Q3.2.10.**

EE2- Q4.2.1- Les activités de vérification et de validation sont documentées, connues et planifiées. Les plans de vérification et de validation sont documentés, conformément à une procédure documentée.

Oui, c'est sous la responsabilité de l'équipe d'assurance qualité.

EE2- Q4.2.2- Les révisions par les pairs sont effectuées à la suite de la production de certains produits intermédiaires clés.

Il y a un processus de révision par les pairs lors du développement, mais j'ignore si cette pratique est réalisée lors de l'analyse.

EE2- Q4.2.3- On procède à des révisions des activités des fournisseurs et des sous-traitants à des étapes déterminées.

Je n'ai pas d'information là-dessus.

EE2- Q4.2.4- L'essai des systèmes est effectué conformément à une procédure documentée pour démontrer que le produit est conforme aux exigences et qu'il satisfait les exigences du client.

Oui, c'est validé par les analystes et par l'assurance qualité.

EE2- Q4.2.5- Des essais de régression sont systématiquement et formellement effectués sur toutes les parties du logiciel touchées par des modifications incluant des corrections.

Il y a des tests de régressions de réalisés par l'assurance qualité, mais ils ne semblent pas tous être automatisés. Cette étape est opaque aux développeurs et les problèmes surgissent en fin du cycle de vie et aboutissent en cas de maintenance dans tous les cas, car ils ne peuvent pas être réglés à la source.

EE2- Q4.2.6- La coordination de l'installation, de l'essai sur place et de la maintenance du logiciel avec l'utilisateur est effectuée conformément à une procédure documentée.

Oui

EE2- Q4.2.7- Les pratiques relatives à l'installation des modifications de logiciel (y compris la documentation d'installation) sont conçues pour réduire l'interruption du fonctionnement quotidien du système.

Oui, les mises en production ont lieu en dehors des heures d'utilisation du système.

EE2- Q4.2.8- Le mainteneur effectue des essais d'acceptation à la livraison du produit du sous-traitant et du fournisseur, conformément à une procédure documentée.

Non

Evolution Engineering	Pre-delivery and transition services	1.2.1	L: Largely Achieved	68	
		1.2.2	N: Not Achieved	0	
		1.2.3	L: Largely Achieved	68	
		1.2.4	F: Fully Achieved	93	
		1.2.5	F: Fully Achieved	93	
		1.2.6	N: Not Achieved	0	
		1.2.7	N: Not Achieved	0	
		1.2.8	N: Not Achieved	0	
		1.2.9	N: Not Achieved	0	
		1.2.10	F: Fully Achieved	93	
		1.2.11	L: Largely Achieved	68	
	Operational support services	2.2.1	F: Fully Achieved	93	
		2.2.2	F: Fully Achieved	93	
		2.2.3	L: Largely Achieved	68	
		2.2.4	F: Fully Achieved	93	
		2.2.5	P: Partially Achieved	33	
		2.2.6	P: Partially Achieved	33	
	Software evolution and correction services	3.2.1	F: Fully Achieved	93	
		3.2.2	P: Partially Achieved	33	
		3.2.3	P: Partially Achieved	33	
		3.2.4	N: Not Achieved	0	
		3.2.5	F: Fully Achieved	93	
		3.2.6	P: Partially Achieved	33	
		3.2.7	N: Not Achieved	0	
		3.2.8	F: Fully Achieved	93	
		3.2.9	N: Not Achieved	0	
		3.2.10	L: Largely Achieved	68	
		3.2.11	L: Largely Achieved	68	
	Verification and validation	4.2.1	L: Largely Achieved	68	
		4.2.2	L: Largely Achieved	68	
		4.2.3	N: Not Achieved	0	
		4.2.4	F: Fully Achieved	93	
		4.2.5	P: Partially Achieved	33	
		4.2.6	F: Fully Achieved	93	
		4.2.7	F: Fully Achieved	93	
		4.2.8	N: Not Achieved	0	
					51,611

Figure 22 - Résultat du Niveau 2 - Evolution Engineering

4.6.

Support to Evolution Engineering

SEE2- Q1.2.1- Les requêtes de modifications font l'objet de documentation, de classement des priorités et d'autorisation auprès de la clientèle avant de procéder et d'implanter un changement au logiciel opérationnel.

Oui, par contre la documentation n'est pas toujours au point.

SEE2- Q1.2.2- Le plan de management de la configuration, documenté et approuvé par le développeur, est utilisé comme base pour démarrer les activités de management de la configuration du mainteneur.

Il n'y a pas de plan de gestion de la configuration cette étape est en mutation et amène beaucoup de confusion.

SEE2- Q1.2.3- Des systèmes de management de la configuration sont utilisés comme référentiel pour les références produites.

Ce système est utilisé pour le versionnage du système.

SEE2- Q1.2.4- Le code source relève du contrôle du management de configuration.

Oui, au travers de TFS et c'est sous la responsabilité d'un expert du domaine.

SEE2- Q1.2.5- Le management de la documentation est en place pour l'organisation de la maintenance et pour chaque requête de modifications (création, modification, version, archivage, diffusion, destruction, validation).

Le support technologique et les formats de la documentation sont désuets et rendent la documentation presque inutilisable pour les développeurs. Les accès à la documentation ne sont pas accordés à tous les développeurs. Il y a de gros problèmes à ce niveau.

SEE2- Q1.2.6- Les éléments de configuration et le référentiel de configuration sont modifiés formellement, conformément à une procédure documentée.

Il ne semble pas y avoir de procédure documentée sur le sujet, si c'est le cas elle n'est pas partagée.

SEE2- Q1.2.7- Il y a traçabilité des requêtes de changements, des exigences, des conceptions détaillées et du code source d'une part, et des jeux d'essais d'intégration, d'autre part.

Au travers de TFS, par contre TFS n'est pas encore utilisé pour partager les jeux d'essais d'intégration. Les détails du niveau de l'assurance qualité sont inconnus des développeurs. La conception détaillée est faite sur le tas et documentée par la suite si le temps le permet ou si le directeur met de la pression pour que ça se fasse.

SEE2- Q1.2.8- Des rapports normalisés, documentant les activités de management de la configuration et le contenu des références produites sont développés et mis à la disposition des groupes et des personnes concernées.

Il y a des présentations sur le sujet, mais pas de documentation formelle.

SEE2- Q2.2.1- La responsabilité, l'autorité et les interrelations entre les personnes chargées du management, de l'exécution et de la vérification de travaux ayant une incidence sur la qualité sont définies.

Oui, ces personnes se rencontrent lors des réunions du comité de maintenance.

SEE2- Q2.2.2- Un programme d'assurance de la qualité est établi, documenté, approuvé et tenu à jour, conformément à une procédure documentée.

C'est sous la responsabilité de l'équipe d'assurance qualité.

SEE2- Q2.2.3- Des révisions indépendantes de conformité sont effectuées régulièrement.

Non.

SEE2- Q2.2.4- Les activités d'assurance de la qualité sont effectuées conformément au plan d'assurance de la qualité.

Supposément, les détails de ces activités ne sont pas connus de la maintenance.

SEE2- Q2.2.5- Les activités de développement, de maintenance et d'exploitation des systèmes sont passées en révision et auditées afin de s'assurer de leur conformité au processus.

C'est sous la responsabilité des gestionnaires, mais je n'ai pas collecté d'information concernant des audits.

SEE2- Q2.2.6- Des audits du management de la configuration des produits sont effectués conformément à une procédure documentée.

Je n'ai pas collecté d'information là-dessus.

SEE2- Q2.2.7- Les produits sont fabriqués, livrés, installés, acceptés et maintenus, conformément à une procédure documentée, couverte par le système de qualité.

Oui.

SEE2- Q2.2.8- La fonction d'assurance de la qualité du maître d'œuvre surveille les activités d'assurance de la qualité de l'importateur conformément à une procédure documentée.

Je n'ai pas collecté d'informations là-dessus.

SEE2- Q2.2.9- Des échantillons représentatifs des biens livrables intermédiaires et finaux sont passés en revue afin de s'assurer de leur conformité aux exigences du processus concerné.

Oui.

SEE2- Q2.2.10- Les écarts constatés dans les activités de développement, de maintenance et d'exploitation sont documentés et traités, conformément à une procédure documentée.

Oui, c'est réalisé par l'équipe d'assurance qualité au travers de TFS.

SEE2- Q2.2.11- Les résultats des activités d'assurance de la qualité font l'objet de rapports périodiques présentés à l'équipe de développement, de maintenance ou d'exploitation.

Non.

SEE2- Q2.2.12- Les résultats des révisions indépendantes de conformité sont passés en revue périodiquement avec la haute direction.

C'est sous la responsabilité de l'équipe d'assurance qualité.

SEE2- Q2.2.13- Le personnel qui gère, exécute et vérifie les travaux ayant une incidence sur la qualité dispose de la liberté et de l'autorité pour :

a) Prendre des mesures visant à prévenir la non-conformité de produit

Oui

b) Identifier et enregistrer tout problème de qualité relatif au produit

Oui

c) Susciter, recommander ou fournir des solutions par l'intermédiaire de canaux de communication préétablis

Non, l'équipe de l'assurance qualité ne possède pas cette expertise.

d) Vérifier la mise en œuvre des solutions

Oui, lors des tests qu'ils feront par la suite.

e) Maîtriser le traitement, la livraison ou l'installation d'un produit jugé non conforme jusqu'à ce que la déficience ou les éléments insatisfaisants aient été corrigés

Non

SEE2- Q3.2.1- Le mainteneur a défini des mesures opérationnelles de base s'alignant avec les objectifs de la mesure de la maintenance du logiciel.

Il n'y a pas de plan de mesure.

SEE2- Q3.2.2- Le mainteneur procède à des analyses et produits des rapports, de base, pour répondre à des requêtes spécifiques.

C'est sous la responsabilité des analystes.

SEE2- Q4.2.1- L'unité organisationnelle de maintenance du logiciel documente son processus de résolution de problème et d'analyse causale.

Lorsque le temps le permet, c'est réalisé bien souvent au travers de TFS, mais il n'y a pas de rapport causal formel.

SEE2- Q4.2.2- Les cadres supérieurs de l'organisation analysent régulièrement les données sur la qualité, les clients et la performance.

Oui, il y a des rapports de présentés à l'équipe lors des réunions mensuelles.

SEE2- Q4.2.3- Les cadres supérieurs analysent les résultats des sondages effectués auprès du personnel et, sur la base de cette analyse, élaborent et mettent en œuvre un plan d'action. Ces résultats et le plan d'action sont communiqués à tout le personnel.

Il n'y a pas de sondage au niveau du personnel autre que les sondages à très haut niveau qui sont envoyés par l'organisation une fois par an et qui ne sont pas reliés à la maintenance.

SEE2- Q4.2.4- Des rapports d'analyse des processus sont produits et distribués aux groupes concernés.

Non

SEE2- Q5.2.1- La re-documentation du code source est effectuée uniquement pour les composants faisant l'objet d'un changement ou d'une correction.

Si le temps le permet.

SEE2- Q5.2.2- La restructuration du code source est effectuée uniquement pour les composants faisant l'objet d'un changement ou d'une correction.

Il n'y a pas beaucoup de restructuration du code source et vu l'absence de tests automatisés cette pratique n'est pas encouragée ni envisageable dans bien des cas.

SEE2- Q5.2.3- La rétro-ingénierie du code source est réalisée uniquement pour les composants faisant l'objet d'un changement, d'une correction ou d'une requête de support opérationnel.

Oui

SEE2- Q5.2.4- La réingénierie du code source est réalisée uniquement pour les composants faisant l'objet d'un changement, d'une correction ou d'une requête de support opérationnel.

Voir la question **SEE2- Q5.2.2.**

SEE2- Q5.2.5- L'unité organisationnelle de la maintenance développe un plan de migration et le communique à la clientèle.

Je n'ai pas collecté d'information dans ce sens.

SEE2- Q5.2.6- Des outils logiciels sont utilisés pour supporter la migration d'une plateforme technologique vers une autre.

Ce n'est pas réalisé par la maintenance.

SEE2- Q5.2.7- L'unité organisationnelle de la maintenance développe un plan de retraite des logiciels et le communique à sa clientèle.

Ce n'est pas réalisé par la maintenance.

Support to Evolution Engineering	Configuration and version management	1.2.1	L: Largely Achieved	68
		1.2.2	N: Not Achieved	0
		1.2.3	L: Largely Achieved	68
		1.2.4	F: Fully Achieved	93
		1.2.5	P: Partially Achieved	33
		1.2.6	N: Not Achieved	0
		1.2.7	P: Partially Achieved	33
		1.2.8	P: Partially Achieved	33
Process, service and software quality assurance		2.2.1	F: Fully Achieved	93
		2.2.2	F: Fully Achieved	93
		2.2.3	N: Not Achieved	0
		2.2.4	L: Largely Achieved	68
		2.2.5	L: Largely Achieved	68
		2.2.6	N: Not Achieved	0
		2.2.7	F: Fully Achieved	93
		2.2.8	N: Not Achieved	0
		2.2.9	F: Fully Achieved	93
		2.2.10	L: Largely Achieved	68
		2.2.11	N: Not Achieved	0
		2.2.12	L: Largely Achieved	68
		2.2.13	P: Partially Achieved	33
Maintenance measurement and analysis		3.2.1	N: Not Achieved	0
		3.2.2	L: Largely Achieved	68
Causal analysis and problem resolution		4.2.1	P: Partially Achieved	33
		4.2.2	L: Largely Achieved	68
		4.2.3	N: Not Achieved	0
		4.2.4	N: Not Achieved	0
Software rejuvenation, migration and retirement		5.2.1	P: Partially Achieved	33
		5.2.2	N: Not Achieved	0
		5.2.3	F: Fully Achieved	93
		5.2.4	N: Not Achieved	0
		5.2.5	N: Not Achieved	0
		5.2.6	N: Not Achieved	0
		5.2.7	N: Not Achieved	0
				38,235

Figure 23 - Résultat du Niveau 2 - Support to Evolution Engineering

4.7. Conclusion sur l'évaluation chez ACME

Le résultat global obtenu de l'évaluation du niveau 0 chez ACME et détaillé par les figures [12-13-14-15] est de 81,6%. Selon le modèle développé par [D-A Paquette-A April-A Abran,2006], j'en conclus que ACME complète avec succès la grille de critères de ce niveau et en conséquence l'activité de maintenance n'est pas une activité marginale chez ACME.

Le résultat obtenu lors de l'évaluation du niveau 1 ACME et détaillé par les figures [16-17-18-19] est de 12,38% chez ACME. Ce résultat catégorie de « *Partially Achieved* » selon le modèle de [D-A Paquette-A April-A Abran,2006] est explicable par la forte présence de processus formels dans la réalisation des activités de maintenance logicielle. L'initiative et les activités informelles individuelles sont basses et ne sont pas encouragées si les impacts risquent de dépasser l'environnement individuel de l'acteur concerné et si les impacts risquent d'affecter d'autres acteurs, systèmes ou processus déjà établis dans l'équipe.

Les résultats obtenus lors de l'évaluation du niveau 2 chez ACME indiquent 3 tendances que j'ai eu l'occasion d'observer. Le résultat de 64,07 de la figure 21 obtenu dans la section « *Event/Request management* » indique que l'équipe de maintenance chez ACME concentre la majeure partie de ses efforts en réaction aux cas de maintenance qui émergent du niveau de support 3. Les aspects du processus permettant à l'équipe de réagir aux requêtes récoltent la grande majorité des résultats « *Fully Achieved* ». La section « *Evolution Engineering* » récolte le second plus haut résultat: cette section récolte la plupart des points lors des étapes de support à l'opération, des efforts correctifs et de la vérification et la validation des correctifs qui sont assurés lors de communication entre les analystes et développeurs de la maintenance avec les analystes de l'équipe d'assurance qualité.

Les sections « *Process management* » et « *Support to Evolution Engineering* » récoltent les deux pires résultats, soit de 39% et 38% tels que représentés par les figures 20 et 23. Ces deux scores catégorisés de partiellement satisfaisants par le modèle de [D-A Paquette-A April-A Abran,2006] révèlent des carences que j'identifie comme sources potentielles de l'émergence

de la problématique identifiée en début de projet. Les faiblesses quant au suivi des activités de la maintenance, ainsi que l'absence de données permettant d'observer, d'analyser et de remettre en question le processus de maintenance ainsi que la formation sur le tas du personnel de maintenance sur le processus et les façons de faire hypothèquent une bonne partie du résultat obtenu suite à l'évaluation de la section « Process management ».

La formalisation du processus de maintenance a été réalisée et documentée au travers de TFS, mais cette transition a été réalisée sans étude préalable du processus en fonction de mesures utilisables et utilisées lors du processus. Les résultats obtenus dans la section « *Support to Evolution Engineering* » souffrent de la sur-spécialisation du département : les différentes tâches sont dispersées entre 3 groupes, soit la maintenance, l'évolution et l'assurance qualité. L'équipe spécialisée à détecter les erreurs ne possède pas la spécialisation pour faire les correctifs, l'équipe spécialisée à faire les correctifs ne possède pas la spécialisation pour faire les tests leur permettant de valider les résultats. En cumulant ces deux aspects de cette étape du processus à un processus de transition logicielle qui ne parvient pas à mettre sur pieds des versions utilisables avec des groupes de données pouvant être utilisées lors de la réalisation et des tests, on obtient un contexte où les acteurs de la maintenance réalisent des correctifs qu'ils ne peuvent pas tester eux-mêmes vu la lourdeur que représente la phase de test, lourdeur principalement due à l'absence de batteries de tests automatisés pouvant être exécutées par la maintenance, ainsi qu'à l'absence d'environnements permettant d'exécuter les fonctionnalités dans un contexte se rapprochant des groupes de données exécutables tel que lors de cas problématiques rencontrés en production. L'état provoqué par cette séquence de tâches réalisées en silo par des équipes surspécialisées provoque un échange « Volley-Ball » des cas de maintenance entre la maintenance et l'équipe d'assurance qualité. L'absence de batteries de tests automatisés pouvant être exécutés lors de la maintenance décourage aussi les développeurs dans la réalisation de leurs tâches de réingénierie et plus de provoquer des contextes ou de nouveaux bogues sont introduits dans le système sans qu'ils ne soient détectés avant de monter en production. Bref, la correction des bogues est ralentie par la lourdeur du processus et l'absence d'automatisation empêche les développeurs de s'assurer que leur réalisation lors de la correction des bogues n'en provoque pas davantage qu'elle n'en corrige.

CHAPITRE 5

« MAPPING » DU PROCESSUS ET IDENTIFICATION DES SOURCES DE GASPILLAGE

Suite aux observations que j'ai faites du « *Mapping* » actuel du processus, celui-ci correspond au modèle suggéré par la norme ISO14764 : suite aux étapes de création et de priorisation des cas de maintenance il y a une étape d'analyse du problème qui requiert une modification, suivie de la réalisation et au final l'étape de révision puis de migration qui est composée des tests fonctionnels, des tests d'intégration, des tests du groupe d'assurance qualité puis ensuite la migration. Ce modèle en cascade est intégré au modèle utilisé dans TFS. Dans un cas où la transition entre ses étapes est interrompue par un problème hérité de l'étape précédente les sources de gaspillages s'accumulent. J'ai identifié les sources de gaspillage suivantes :

- Changement de tâches
- Temps d'attente
- Manutention
- Défauts

Ces sources de gaspillage se manifestent dans le processus de maintenance logiciel lorsqu'un défaut est livré à l'étape des tests : la réalisation et les tests sont sous la responsabilité de personnes différentes et aussi d'équipes différentes. La réalisation est sous la responsabilité des développeurs, les tests fonctionnels et d'intégration sont manuels et sous la responsabilité de l'analyste fonctionnel, la stratégie de tests est complétée par l'équipe GCQ. Lors de cette séquence, j'ai identifié que l'équipe de maintenance ne respecte pas les 3 principes « LEAN » suivants:

- Décider au dernier moment possible
- Impliquer l'équipe
- Avoir une vue d'ensemble du processus

La surspécialisation des activités de tests ne permet pas aux développeurs de la maintenance de pouvoir décider de la livraison de leur réalisation au dernier moment possible. Les scénarios de tests et les stratégies d'essai de GCQ ne sont pas connus des développeurs lors de la réalisation, les tests fonctionnels eux ne sont pas communiqués au développeur ou pas conçus lorsqu'il entame la réalisation. Le développeur n'étant pas un spécialiste du fonctionnel ni des stratégies de test, il n'arrive pas dans tous les cas à pouvoir identifier la totalité des scénarios de tests devant s'exécuter afin de considérer son travail comme complété. Dans les cas où tous les tests possibles seraient connus du développeur, l'absence d'automatisation des tests ne lui permettra pas de pouvoir valider son travail efficacement sans perdre de temps.

Cette réalisation en silo empêche les différents intervenants de la séquence d'avoir une vue d'ensemble du processus leur permettant de pouvoir s'impliquer en aval ou en amont des étapes auxquelles ils sont assignés afin de pouvoir faire profiter aux autres leur expertise. Par exemple, le spécialiste des scénarios fonctionnels n'a pas accès à l'expertise du spécialiste des tests lorsque vient le temps de concevoir les tests qui pourrait être automatisés par le développeur afin qu'ils puissent être par la suite exécutés facilement par ceux qui ne possèdent pas l'expertise pour les réaliser.

Cette réalisation en cascade provoque des défauts qui ne seront détectés par GCQ qu'après avoir été livrés par le développeur : ces échanges entre l'assurance qualité provoquent des changements de tâches coûteux et des temps d'attente non planifiés qui nuisent au reste du processus.

Suite à l'identification d'une source potentiellement responsable d'une partie de la problématique, avant d'envisager d'entamer les travaux de réalisation d'une solution il faut se munir de mesures permettant d'observer la zone du processus impliqué dans les changements à venir afin de pouvoir observer l'impact des solutions sur le processus.

Dans le cas de la problématique abordée dans ce projet, bien qu'il soit possible de fixer des objectifs à atteindre pour parvenir à régler davantage de cas de maintenance et ainsi voir le

nombre de nouveaux cas être inférieur au nombre de cas réglés, il faut pouvoir mesurer le processus et relier les objectifs à atteindre avec les résultats des mesures pour parvenir à déterminer si les objectifs sont en voie d'être atteints ou non. Dans le tableau qui suit j'ai rédigé un plan de mesure qui permettrait d'effectuer un suivi des échanges entre l'équipe de maintenance et l'équipe d'assurance qualité lors des livraisons afin de pouvoir déterminer si des objectifs de diminution du gaspillage atteignent leurs cibles

Tableau 5.1 - Proposition d'un programme de mesure

Programme de mesure		
Problématique	Les échanges entre les étapes de réalisations et de validation par les tests sont une source de gaspillage.	
Objectif	Permettre d'observer les activités impliquant l'étape de réalisation et des tests fonctionnels, d'intégration et GCQ afin de répertorier les défauts et les manquements qui nuisent au processus.	
Portée	La séquence : analyse → réalisation → tests, du processus de maintenance logicielle.	
Programme proposé		
<ul style="list-style-type: none"> • Calculer et répertorier les retours de GCQ • Calculer et répertorier les bogues livrés • Identifier les tests manquants devant être réalisés • Calculer le temps écoulé pendant cette partie du processus 		
Processus	<ul style="list-style-type: none"> • Documenter une stratégie de tests fonctionnels permettant de valider les critères d'acceptation d'une réalisation. • Mettre sur pied un environnement de tests avec des groupes de données permettant d'exécuter les scénarios de tests. • Réalisation des tests automatisés • Répertorier cette séquence dans TFS, les dates incluant l'heure doivent être obligatoires donc bloquante au processus. Les retours doivent être enregistrés comme un nouveau bogue et joint au bogue initial en incluant la description pour pouvoir ajouter un test manquant au besoin. 	
Outils	<ul style="list-style-type: none"> • L'outil de gestion des cas de maintenance: TFS dans le cas actuel. • De la documentation fonctionnelle à jour et intégrée à l'environnement préférablement. • Outils pour générer les groupes de données nécessaires aux tests • Librairie permettant de réaliser et d'exécuter des tests fonctionnels automatisés. 	
Risques	Manque de collaboration du personnel impliqué	
	Contingence	Mitigation

	Plan de communication adéquat pour les informer sur leurs rôles, les enjeux et les gains espérés.	Avoir l'appui de la direction afin de diminuer la résistance au besoin.
	Outils non présents actuellement dans le processus	
	Contingence	Mitigation
	Acquérir les outils nécessaires	Développer ceux qui ne peuvent pas être acquis.
	Manque d'expertise	
	Contingence	Mitigation
	Former le personnel impliqués	Embaucher des experts du domaine.

CHAPITRE 6 CONCLUSION

La problématique de départ qui a motivé ce projet de maîtrise consistait à trouver une solution permettant de renverser la tendance de la courbe d'évolution des cas de maintenance afin de parvenir à résoudre davantage de problèmes qu'il ne s'en crée.

Suite aux premières observations de la situation chez ACME, les premières avancées faites dans ce projet consistaient en une planification d'activités à réaliser en entreprise afin d'atteindre l'objectif. Bien que les activités planifiées étaient connues dans le domaine, il ne m'a pas été possible de démontrer et convaincre qu'elles allaient permettre d'atteindre l'objectif fixé. Suite à ce constat, j'ai modifié la méthodologie de travail afin d'orienter celle-ci de sorte que mes observations me permettent d'avoir une meilleure vue d'ensemble du processus de maintenance chez ACME et ensuite pouvoir l'analyser et comprendre ce qui, en premier lieu, m'a empêché de proposer une solution permettant d'atteindre l'objectif avec ma première approche.

Suite à l'information que j'ai acquise de la littérature couvrant le domaine de la maintenance logicielle, de mes observations chez ACME et suite à l'application d'un modèle d'évaluation du niveau de maturité du processus de maintenance, j'en suis venu à la conclusion que le niveau de maturité du processus de maintenance ainsi que la taille du projet ne me permettaient pas d'avoir une vue d'ensemble qui me permettrait de proposer une solution qui influencerait la tendance d'évolution de la courbe des cas de maintenance dans un délai suffisamment court pour que les résultats puissent être observés en cours de projet. La complexité du processus de maintenance et le manque de données permettant d'observer son évolution ainsi que les résultats à différentes périodes dans le temps ne me permettaient pas de pouvoir tirer des conclusions suite aux changements ni de prédire de quelconques résultats pouvant être obtenus suite à un changement.

Pour parvenir à faire la trace des résultats obtenus suite à des changements faits au processus de maintenance, le processus doit être mesuré en fonction de la planification du changement et des zones impactées par le changement. Par exemple, avant de mettre en place des mesures pour diminuer le nombre de retours suites aux livraisons des maintenances correctives, il serait possible de sélectionner un échantillon de maintenances correctives réalisées par le passé et identifier les retours qui sont reliés à c'est maintenance lorsqu'il y en a eu et de calculer un ratio. Cette information pourrait ensuite être comparée au ratio obtenu après l'application de la mesure perfective et aider à évaluer l'effet que l'amélioration à eux sur le processus.

Les entreprises qui désirent bonifier leur processus de maintenance avec des activités leur permettant de collecter de l'information en cours de processus afin de pouvoir l'observer ont intérêt à se munir de plans de mesures pour les aider à orienter leurs décisions. La complexité d'un processus de maintenance tend à augmenter lorsque le projet prend de l'ampleur et il devient alors difficile d'avoir une vue d'ensemble sur la totalité du processus et les conclusions basées strictement sur l'expérience des individus sont influencées par plusieurs facteurs, dont le poste occupé et le nombre d'années d'expérience à l'intérieur et à l'extérieur de l'organisation. Ajouté à un roulement élevé d'employés dans un contexte où l'information sur le processus est déficiente il en résulte un risque de perte de contrôle; lorsque le problème est identifié ça reste difficile d'élaborer une stratégie, de proposer des solutions permettant d'atteindre des objectifs à haut niveau, telle que la diminution d'une courbe d'évolution des cas de maintenance et aussi de démontrer les impacts que cette solution peut avoir sur le processus. La présence de plan de mesures aiderait à diminuer le risque de perte de contrôle du processus, à intervenir plus tôt et à élaborer développer des stratégies plus ciblées.

CHAPITRE 7 RECOMMANDATIONS

Suite à ce projet de maîtrise, je m'avance à rédiger quelques recommandations qui pourraient augmenter les chances de succès lors d'activités d'amélioration du processus de maintenance logiciel.

- Définir une planification détaillée des activités d'amélioration du processus devant être réalisées, comme il est fait dans le cas d'un projet afin de s'assurer d'avoir l'aval de la direction en tout temps et permettre d'éviter de gaspiller des efforts sur des tâches dont la valeur ajoutée au processus n'a pas été reconnue par la direction.
- Identifier des activités d'information et de formation permettant aux employés d'avoir les connaissances requises pour comprendre les enjeux et les objectifs visés par les activités qui impliquent du changement et diminuer le risque de résistance au changement.
- Procéder avec des changements de petite taille afin de faciliter la réalisation des tâches d'analyse et de réalisations dans le but de diminuer la charge d'efforts et de temps requis pour réaliser le changement.
- Se munir de plan de mesure pour diminuer l'aspect approximatif des changements et l'appuyer avec des informations factuelles permettant de prendre des décisions éclairées et non arbitraires.
- Faire un suivi de la collecte d'informations sur le processus afin de s'assurer que tous les employés soient rigoureux lors de l'utilisation des outils permettant de collecter cette information au fil du processus dans le but de permettre d'analyser le processus au fil du temps.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- 1- IEEE Guide for Adoption of ISO/IEC TR 14471:2007 Information Technology -- Software Engineering -- Guidelines for the Adoption of CASE Tools," in IEEE Std 14471-2010 , vol., no., pp.1-41, Sept. 9 2010 doi: 10.1109/IEEESTD.2010.5585616 keywords: {computer aided software engineering;programming languages;ISO/IEC TR 14471:2007;computer-aided software engineering;programming language;software process;Computer aided software engineering;IEC standards;IEEE standards;ISO standards;Information technology;CASE tools;computer-aided software engineering (CASE);information technology;pilot project processes;process;tool adoption;tool evaluation;tool selection}, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5585616&isnumber=5585615>
- 2- D-A Paquette, A April, A Abran (2006) Assessment Results Using the Software Maintenance Maturity Model (S3m) International Workshop on Software Measurement – IWSM-Metrikom 2006, Postdam, Germany, Nov. 2-3, Shaker Verlag, pp.147-160. [Refereed Conference Proceedings Papers] .
- 3- International Standard - ISO/IEC 14764 IEEE Std 14764-2006 Software Engineering ♂ Software Life Cycle Processes ♂ Maintenance," in ISO/IEC 14764:2006 (E) IEEE Std 14764-2006 Revision of IEEE Std 1219-1998) , vol., no., pp.0_1-46, 2006 doi: 10.1109/IEEESTD.2006.235774 keywords: {IEEE standards;ISO;Software engineering}, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1703974&isnumber=35960>
- 4- 'Software Maintenance Management: Evaluation & Continuous Improvements' - with co-author A. April URL: <http://ca.wiley.com/WileyCDA/WileyTitle/productCd-0470258020.html>
- 5- “Kanban et Scrum - tirer le meilleur des deux” écrit par Henrik Kniberg et Mattias Skarin, Préfacé par Mary Poppendieck et David Anderson Traduit par Claude Aubry, Frédéric Faure,Antoine Vernois et Fabrice Aimetti, © 2010 C4Media Inc. Tous droits réservés. C4Media, Éditeur de InfoQ.com. ISBN: 978-0-557-13832-6

- 6- “Lean software development : an agile toolkit” Auteur(s): Mary Poppendieck, Thomas David Poppendieck, Éditeur: Boston : Addison-Wesley, Date de publication: c2003 Numéro: ISBN 0321150783 (pbk.)
- 7- “Choisir l'agilité : du développement logiciel à la gouvernance” Auteur(s): Mathieu Boisvert et Sylvie Trudel, Éditeur: Paris : Dunod, Date de publication: c2011, ISBN 9782100558506 ISBN 2100558501
- 8- Software Maintenance Management: Evaluation and Continuous Improvement
Copyright Year: 2008 Author(s): Alain April; Alain Abran Publisher: Wiley-IEEE Press, ISBN : 9780470258033 EISBN : 9780470258033 URL:
<http://ieeexplore.ieee.org/servlet/opac?bknumber=6129685>
- 9- ISO/IEC TR 19759:2015: Software Engineering -- Guide to the software engineering body of knowledge (SWEBOK), Publication date : 2015-10, Edition : 2 Number of pages : 336 Technical Committee : ISO/IEC JTC 1/SC 7 Software and systems engineering ICS : 35.080 Software URL: <https://www.iso.org/standard/67604.html>

Site internet consulté sur le sujet

1. Site internet : Tool for Agile Blog : How Agile team handle maintenance work
URL: <http://toolsforagile.com/blog/archives/390/how-agile-teams-handle-maintenance-work> consulté en ligne le 5 juin 2016
2. Site internet Piloter la performance URL: <http://www.piloter.org/six-sigma/lean-management.htm> Consulté en ligne le 5 juin 2016

ANNEXE 1



S3M_Short_Assessment_detailed.xls