

MASTER MANAGEMENT DES ORGANISATIONS – M2
SPECIALITE PROFESSIONNELLE : MANAGEMENT DES SYSTEMES
D'INFORMATION ET DE CONNAISSANCE

Mémoire

**« EVOLUTION DE LA MATURETE DU PROCESSUS DE
MAINTENANCE DU LOGICIEL DANS UNE
ORGANISATION EN MODE PROJET »**

REDIGE ET SOUTENU PAR :

PHILIPPE GALAUP

PROMOTION JB 2009

DIRECTEUR DE MEMOIRE :

SELMIN NURCAN

DATE DE LA SOUTENANCE :

L'UNIVERSITE N'ENTEND DONNER AUCUNE APPROBATION NI IMPROBATION AUX
OPINIONS EMISES DANS CE MEMOIRE : CES OPINIONS DOIVENT ETRE
CONSIDEREES COMME PROPRES A LEUR AUTEUR.

REMERCIEMENTS

Je tiens en premier lieu, à remercier le Dr Alain April, professeur de génie logiciel à l'École de technologie supérieure de l'Université du Québec. Son aide a été pour moi comme à l'image de sa gentillesse : le diamant dont l'éclat a illuminé 15 ans d'expérience dans le métier encore peu reconnu de la maintenance du logiciel. J'aimerais y associer le Dr Alain Abran, directeur du laboratoire de recherche au sein de la même école et co-auteur du livre « *Améliorer la maintenance du logiciel* » qui a constitué ma bible de référence tout au long de la rédaction de ce mémoire. Leur accent québécois résonnera encore longtemps dans les circonvolutions de mon tympan comme dans mon cœur, merci à eux.

Je remercie, mon directeur ou plutôt ma directrice de Mémoire, Selmin Nurcan, Maître de Conférences à l'Université de Paris 1, Sorbonne. Son regard et ses connaissances « aiguisées » ont su pointer et orienter ce travail sur la juste voie d'une recherche universitaire.

Je remercie également Diane, ma « belle » sœur, dont les différents documents empruntés à ses cours de terminale STG (Sciences et Technologies de la Gestion) m'ont été d'un grand secours. Ses précieuses remarques à la lecture du mémoire ont encore une fois prouvé ses capacités incontestables à asseoir son statut de professeur au sein de l'enseignement technique secondaire.

Je remercie, le lieutenant-colonel Bretegnier et le commandant Quemerais ainsi que l'ensemble du bureau « JANUS », chargé de la maintenance du logiciel du même nom. Ils n'ont pas hésité à participer activement aux investigations et se sont spontanément prêtés au jeu du questionnaire. J'y associe également Zmirdine Mari et le lieutenant-colonel Cazoulat. J'espère que cette « étude » un peu particulière aura apporté à tous un éclairage nouveau sur le métier et dans leur travail au quotidien.

Enfin, je ne peux parachever ces remerciements sans avoir une pensée toute particulière pour la promotion JB2009. Les travaux en communs, les repas conviviaux un peu « arrosés », les échanges amicaux et « collaboratifs » ininterrompus, ont fait de cet exercice, comme de l'année universitaire dans son ensemble, une expérience unique, singulière et d'une profondeur que je n'aurais su imaginer. J'espère sincèrement que les liens résisteront à l'érosion inévitable du temps. J'y associe aussi l'ensemble des professeurs et l'ensemble de l'équipe de l'IAE. Un grand merci à tous.

RESUME

L'objectif de ce mémoire est de mettre en avant la maintenance du logiciel dans un contexte économique qui tend à démontrer une augmentation croissante des coûts de maintenance. La maintenance du logiciel est souvent perçue comme une des étapes d'un projet logiciel. Parfois, du fait d'un certain nombre d'activités similaires, la maintenance peut être considérée comme un projet à part entière.

Nous voulons montrer, ici, que la maintenance du logiciel et le projet de développement d'un logiciel sont deux processus différents qui nécessitent des techniques de gestion différentes : la gestion de projet (planning, jalons, équipe projet, ...) d'un côté et la gestion de files d'attente (acceptation ou rejet, priorité, un ou deux mainteneurs par modification, ...) de l'autre. Ces processus ne peuvent pas se confondre. Il est donc nécessaire de considérer les spécificités de la maintenance et d'évaluer leur rapport d'influence dans un contexte organisationnel orienté « projet logiciel ».

Un état de l'art en matière de pratiques exemplaires nous permet de parcourir les référentiels et les normes en vigueur dans le domaine des technologies de l'information. Nous déterminons ainsi que le modèle S^{3m} est plus adapté au domaine de la maintenance applicative que ne le sont les modèles CMMi ou ITIL.

Sur la base du modèle S^{3m} et des méthodes d'évaluation de processus qui s'y rattachent, nous décidons d'améliorer la maturité du processus de maintenance logicielle d'une unité de maintenance de l'Armée de Terre et préparer le changement organisationnel dans une structure orientée « projet logiciel ».

Notre démarche s'appuie sur le cadre de référence (« framework ») de la méthode EKD-CMM pour définir les étapes du changement de l'organisation et se déroule en trois temps :

- Evaluation de la maturité du processus de maintenance logicielle à l'aide des méthodes S^{3m} pour prendre la mesure objective de la réalité actuelle de l'organisation
- Formalisation du processus de maintenance à l'aide des modèles de EKD-CMM pour définir le modèle actuel (« As-Is ») de l'organisation de la maintenance
- Définition des axes d'amélioration possibles par rapport au processus de référence et préparation du changement à deux niveaux de l'organisation, celui de l'unité de maintenance et celui de la structure dans laquelle elle opère

Le périmètre du mémoire s'arrête à la définition du changement sans le conduire réellement pour aboutir à une réalité future. Cette étape de concrétisation du changement constitue les perspectives possibles de ce mémoire.

Abstract

The purpose of this document is to make the Software Maintenance more visible than it is, in a context of increasing software maintenance costs. Usually, the software maintenance is defined as a simple step of the software life cycle. And, because some activities are pretty close to each other, software maintenance and software development project are considered as one.

Here, we want to show that software maintenance and software development are two different processes which both need different management techniques: project management (planning, project team ...) versus request management (acceptance or rejection, priority, one or two persons by modification ...). These processes can't be understood as the same. So, it's necessary to consider the specific aspect of software maintenance and evaluate the influence on a project-oriented organization context.

The software practices state of the art gives us the standards and the best practices used in Information Technology. In consequence, we find S^{3m} model is better adapted than CMMi or ITIL model in software maintenance domain.

We decide to improve the software maintenance process maturity in a French Army maintenance unit with the S^{3m} methodology and prepare the project-oriented organization change.

We use the EKD-CMM framework to define the three steps of the organization change:

- Software maintenance process maturity assessment, with S^{3m} method, to objectively define the actual reality of the maintenance organization
- Software maintenance process formalization with EKD-CMM models to define de actual model ("As-Is")
- Defining the way of improvement to the reference process ("To-Be") and preparing the maintenance organization and the organization above change

This study stop the work at "defining the change" without conduct it to create a new reality of the organization. This step can be the further work outside this document.

SOMMAIRE

INTRODUCTION	2
I. PRESENTATION DU SUJET	3
1. Le domaine : la maintenance du logiciel	4
1.1. Pourquoi s'intéresser à la maintenance du logiciel ?	4
1.2. Les problèmes liés à la maintenance	4
2. Le terrain d'investigation : l'armée de Terre	6
2.1. Présentation de l'organisme	6
2.2. Le processus « <i>Gérer les projets</i> »	7
2.3. Le bureau chargé de la maintenance du logiciel	8
3. La problématique émergente	9
3.1. Maintenance logicielle et projet logiciel	9
3.2. Les questions clés	9
II. ETAT DE L'ART DES CONCEPTS ET DES PRATIQUES	11
4. Définition des concepts	12
4.1. La maintenance du logiciel	12
4.2. Projet de développement logiciel	14
4.3. Processus et maturité	16
5. Référentiels de pratiques exemplaires	19
5.1. CMMi (<i>Capability Maturity Model intégration</i>)	19
5.2. ITIL (<i>Information Technology Infrastructure Library</i>)	23
5.3. S ^{3m} (<i>Software Maintenance Maturity Model</i>)	26
6. Méthodes d'évaluation des processus	30
6.1. La méthode SCAMPI	30
6.2. La méthode S ^{3m} Assessment	32
6.3. Conclusion sur les concepts et les pratiques	34
III. DEMARCHE DE MODELISATION DU CHANGEMENT	37
7. Les hypothèses	38
7.1. Les questions liées au contexte de la maintenance	38
7.2. Les hypothèses retenues	39
8. Le cadre de référence de la démarche	41
8.1. Présentation de la méthode EKD-CMM	41
8.2. Proposition d'une approche personnalisée	43
9. Les étapes de la démarche	46
9.1. Le recueil, le traitement et l'analyse des données	46
9.2. La formalisation du processus	47
9.3. La définition du changement organisationnel	48
IV. ETUDE DE CAS	49
10. Evaluation de la maturité du processus de maintenance	50
10.1. Application de la méthode S ^{3m} Assessment	50
10.2. Présentation des résultats	53
11. Formalisation du processus de maintenance	55
11.1. Modèle intentionnel du présent	55
11.2. Les acteurs et leurs rôles	57
11.3. Modèle intentionnel du futur	58
12. Définition du changement organisationnel	59
12.1. Définition du changement organisationnel de la maintenance	59
12.2. Définition du changement organisationnel global	61
CONCLUSION	63
BIBLIOGRAPHIE	66

Tableaux et figures

Tableau 1.1 Coût de la maintenance en pourcentage du budget logiciel [SCH09].....	4
Tableau 1.2 Part du budget alloué à la maintenance selon le sondage [MET07].....	4
Tableau 1.3 Perceptions des problèmes liés à la maintenance [DEK92].....	5
Tableau 4.1 Catégories de la maintenance normalisées par l'ISO 14764 [ISO98].....	13
Tableau 5.1 Objectifs spécifiques du CMMi niveau 2 [BAS06]	22
Tableau 5.2 Les niveaux de maturité CMMi [JOU06].....	22
Tableau 5.3 Domaines et secteurs clés de la maintenance [APR06]	28
Tableau 5.4 Les niveaux d'aptitude de la maintenance [APR06].....	29
Tableau 6.1 Les 3 phases et les 11 processus de la méthode SCAMPI [SCA06]	31
Tableau 6.2 Différences entre maintenance logicielle et développement logiciel.....	34
Tableau 9.1 Les étapes de l'investigation de terrain	46
Tableau 10.1 Résultats de l'audit interne sur la maturité du processus de maintenance	53
Figure 2.1 Organisation du CDEF [CDE09].....	6
Figure 2.2 Le processus « <i>Gérer les projet</i> » de la DSRO [DSR09]	7
Figure 2.3 Organisation du bureau de la maintenance du logiciel « <i>JANUS</i> »	8
Figure 4.1 La maintenance dans le cycle vie du Système Informatique [CNR01]	12
Figure 4.2 Modèle du métaprocessus de la maintenance du logiciel dans ISO 14764 [ISO98].....	13
Figure 4.3 Le projet de développement dans le cycle vie du Système Informatique	14
Figure 4.4 Les objectifs de la gestion de projet [REI05].....	15
Figure 4.5 Un processus générique [ISO08]	16
Figure 4.6 Typologie des processus de l'ISO 9001 [ISO08]	16
Figure 4.7 Pilotage des processus de projets et des processus récurrents [LOR03]	17
Figure 4.8 Représentation des niveaux d'aptitude et des niveaux de maturité [SEI06].....	18
Figure 5.1 Les 3 dimensions critiques d'une organisation [SEI06].....	19
Figure 5.2 Architecture fondamentale du CMMi dans ses 2 représentations [BAS06]	20
Figure 5.3 Le principe de Deming [QUA09].....	23
Figure 5.4 Cadre de référence d'ITIL [ITI04]	24
Figure 5.5 Le Centre de Services [ITI04].....	24
Figure 5.6 Architecture du modèle S ^{3m} [APR06]	26
Figure 5.7 Diagramme de contexte du modèle S ^{3m} [APR06]	27
Figure 6.1 Les grandes étapes des méthodes d'évaluation [APR06]	30
Figure 6.2 L'approche étagée de la méthode S ^{3m} Assessment [LEB08]	32
Figure 6.3 Le passage de la méthode SCAMPI à S ^{3m} Assessment [LEB08].....	33
Figure 6.4 Domaine d'application des pratiques exemplaires.....	35
Figure 6.5 Les pratiques exemplaires dans le cycle de vie du SI	35
Figure 7.1 Procédure relative à un projet réalisée avec une MOE interne [DSR09].....	38
Figure 8.1 Le processus de changement organisationnel selon Jackson [NUR02].....	41

Figure 8.2 La vision du changement organisationnel dans EKD-CMM [NUR02].....	41
Figure 8.3 Représentation d'une organisation et de ses systèmes d'information [NUR02].....	42
Figure 8.4 Cartographie de la démarche de niveau global	43
Figure 8.5 Les objectifs de l'évaluation de la maturité du processus de maintenance	44
Figure 8.6 Construction de la hiérarchie des buts et formalisation des processus	45
Figure 8.7 Comparaison des processus actuels de la maintenance et ceux du modèle S ^{3m}	45
Figure 9.1 Exemple de questions d'évaluation du processus de maintenance – niveau 0 [PAQ06] ...	47
Figure 9.2 Les deux niveaux du changement organisationnel	48
Figure 10.1 Exemple d'un itinéraire S3M décliné aux niveaux 0, 1 et 2	51
Figure 10.2 Exemple de degré d'accomplissement d'un domaine de processus S3M.....	52
Figure 10.3 Evaluation du niveau d'aptitude du processus de maintenance de notre organisation ...	54
Figure 11.1 La vision du changement organisationnel de la maintenance	55
Figure 11.2 Modèle intentionnel de l'organisation actuelle de la maintenance.....	56
Figure 11.3 Cartographie des processus de l'organisation actuelle de la maintenance.....	56
Figure 11.4 Diagramme Acteur-Rôle du bureau de maintenance du logiciel actuel	57
Figure 11.5 Cartographie cible des processus de la maintenance des logiciels.....	58
Figure 12.1 Différents niveaux d'impact organisationnel.....	59
Figure 12.2 Processus de changement de l'organisation de la maintenance du logiciel.....	59
Figure 12.3 Cartographie actuelle et future du processus de maintenance.....	60
Figure 12.4 Procédures appliquées dans l'organisation	61

Glossaire

CDEF	<i>Centre de la Doctrine d'Emploi des Forces</i> (Organisme de l'armée de Terre)
CMMi	<i>Capability Maturity Model integration</i> (Modèle de maturité des processus de projets de développement logiciels)
DSRO	<i>Division Simulation et Recherche Opérationnelle</i> (Organisation métier dans laquelle s'opère notre étude de cas)
EKD-CMM	<i>Enterprise Knowledge Development - Change Management Method</i> (Méthode pour la définition du changement organisationnel)
ISO/SEI	<i>International Standard Organization/Software Engineering Institute</i> (Branche ingénierie logicielle de l'organisme international de standardisation)
ITIL	<i>Information Technology Infrastructure Library</i> (Référentiel des meilleures pratiques pour les opérations informatiques)
KPA	<i>Key Process Area</i> (Secteur clés couvert par un processus - ex : planification, documentation)
MCO	<i>Maintien en Condition Opérationnelle</i> (Maintenance matérielle et logicielle d'un système opérationnel)
S3M	<i>Software Maintenance Maturity Model</i> (Modèle de maturité de la maintenance du logiciel)
S3M Assmt	<i>Software Maintenance Maturity Model Assessment</i> (Méthode pour l'évaluation de la maturité de la maintenance du logiciel)
SCAMPI	<i>Standard CMMI Appraisal Method for Process Improvement</i> (Méthode pour l'évaluation de la maturité des processus du modèle CMMi)
SIOC	<i>Système d'Information Opérationnel et de Commandement</i> (Système destiné à être utilisé sur un théâtre d'opérations militaires)
SMQ	<i>Système de Management de la Qualité</i>

Bureau JANUS

Bureau chargé de la maintenance du système de simulation « *JANUS* »
(Organisation objet de notre étude)

But

Objectif qu'une organisation veut atteindre. Il exprime une vision ou une direction à suivre

But opérationnalisable

But directement associé à un micro-processus d'entreprise

Capacité d'un processus

Aptitude d'un ensemble d'activités à accomplir des pratiques exemplaires

Domaine de processus

Le plus haut niveau de regroupement de processus utilisé par les modèles de maturité. Un domaine de processus contient des KPAs et, pour certains modèles, des itinéraires

Itinéraire

Ensemble de pratiques pouvant couvrir plusieurs niveaux de maturité et ordonnées entres-elles selon leur degré de priorité dans la maîtrise du processus

Maturité d'un processus

« Représentation des attributs clés, d'entités organisationnelles choisies, qui ont un effet sur la progression de ces unités organisationnelles vers la pleine réalisation de leur potentiel et de leur développement » [GAR93]

Méthode

Ensemble de règles raisonnablement exhaustif qui établit une façon précise et répétable de réaliser une tâche et d'obtenir le résultat désiré

Modèle

Représentation schématique ou simplifiée de certains aspects du monde réel

Modèle Acteur/Rôle

Modèle de produit de la méthode EKD-CMM. Il permet de représenter les processus d'entreprise du point de vue des acteurs et des responsabilités qu'ils exercent en remplissant des rôles dans les processus

Modèle intentionnel

Représentation sous forme de hiérarchie buts dont chaque but est une instance d'un objectif visé par une organisation

Opérations informatiques

Activité de maintien opérationnel, des infrastructures informatiques et des services applicatifs d'une organisation, aux profit des utilisateurs des services

Petite maintenance

Activité de correction/évolution du logiciel dont la charge estimée varie de quelques jours à quelques semaines par opposition au *projet de maintenance* estimé à quelques mois

Pratique exemplaire

Une activité de management ou d'ingénierie du logiciel qui contribue à améliorer la capacité d'un processus

Processus

« Un processus est un système dynamique orienté vers la réalisation d'un objectif » [MOR07]

Projet

« Démarche spécifique qui permet de structurer méthodiquement et progressivement une réalité à venir » [REI05]

INTRODUCTION

Ce mémoire s'inscrit dans un contexte de 15 années d'expérience dans le domaine de la maintenance du logiciel ou plus exactement du maintien en condition opérationnelle de systèmes de simulation de natures diverses, et ce, pour le bénéfice des armées françaises. Si le besoin de traiter ce sujet ambitieux visant à améliorer la maintenance du logiciel au sein d'un organisme militaire, émerge d'un contexte professionnel, il n'en est pas moins une volonté personnelle de questionner la manière d'aborder les problématiques liées à cette activité particulière.

De plus, le contexte économique actuel tend à montrer que « maintenir les applications mises en service » devient un des principaux défis à relever.

Nous l'aborderons dans les premiers chapitres, le métier de la maintenance est souvent peu valorisé et le manque de littérature consacrée au sujet ne fait que renforcer la méconnaissance des problématiques particulières qui s'y rattachent. Pourtant, l'objectif de développer des logiciels performants ou de réaliser des systèmes informatiques qui répondent parfaitement aux besoins, ne suffit pas en soi. Ce serait occulter le fait que, même si un système informatisé répond pleinement aux besoins des clients et des utilisateurs, ce besoin est changeant dans le temps, du fait de l'évolution technologique d'une part et de l'évolution du contexte professionnel d'autre part, nécessitant ainsi, une adaptation constante des technologies et des fonctionnalités des systèmes.

La maintenance ne peut donc pas être considérée comme une étape mineure dans le cycle de vie des logiciels tant elle va conditionner l'adaptation de ces logiciels aux nouveaux besoins.

A l'heure où les entreprises sont principalement organisées en « mode projet » pour conduire leurs missions, comment rendre visible, dans une telle organisation, les activités récurrentes telle que la maintenance du logiciel ? Par ailleurs, comment conduire les améliorations du processus et préparer le changement sans remettre en question l'ensemble de l'organisation ?

Telles sont les questions auxquelles, ce document, essaiera de répondre.

Dans le 1^{er} chapitre du mémoire, je m'attache à présenter le sujet de ma recherche : l'amélioration de la maintenance du logiciel. J'explique l'intérêt de se pencher sur ce sujet et je définis le contexte professionnel qui a suscité mon questionnement quant à la confusion qui demeure entre les activités de la maintenance et celles du projet de développement.

Améliorer les processus de la maintenance du logiciel implique de connaître les contours du concept et les pratiques de référence en la matière. Cet état de l'art fait l'objet du 2^{ème} chapitre. J'y compare les concepts de maintenance et de projet et expose l'éventail des référentiels connus dans le domaine des technologies de l'information pour en déduire le référentiel le mieux adapté à la maintenance.

Dans le 3^{ème} chapitre, je présente une démarche personnalisée et le cadre de référence sur lequel je m'appuie pour définir le changement organisationnel induit par l'évolution de la maturité du processus de maintenance du logiciel.

La démarche est appliquée dans une étude de cas que je présente dans le dernier chapitre. L'objectif recherché est d'accompagner l'évolution de la maturité du processus de maintenance et d'évaluer l'influence de ce changement sur un organisme structuré autour d'une démarche orienté projet logiciel.

CHAPITRE I

PRESENTATION DU SUJET

Dans ce premier chapitre nous introduisons la maintenance du logiciel, sujet traité dans ce mémoire, et son contexte professionnel.

La première partie aborde l'intérêt du sujet et les problématiques managériales et techniques liées à la maintenance du logiciel.

Nous présentons dans la seconde partie le contexte professionnel qui a suscité le questionnement et la problématique traitée dans le mémoire. Nous décrivons l'organisation d'une unité de maintenance de l'Armée de Terre, ainsi que la structure du département auquel elle est rattachée.

Enfin, dans la troisième partie, nous abordons la problématique issue du contexte professionnel qui est liée aux activités spécifiques de la maintenance du logiciel. Nous y développons les interrogations auxquelles notre étude devra répondre.

1. Le domaine : la maintenance du logiciel

Qu'est-ce que la maintenance du logiciel ? Pourquoi maintenir le logiciel ? Quels sont les problèmes spécifiques à la maintenance ? Ces questions sont souvent peu abordées pour un métier finalement peu reconnu et qui représente pourtant, une part importante des budgets des services informatiques.

1.1. Pourquoi s'intéresser à la maintenance du logiciel ?

Dans un article paru en septembre 2007 sur le site *01netPro* [PRO07], il apparaît d'après le cabinet d'analystes *Forrester Research* que la part dédiée aux nouveaux projets représente seulement 20% des ressources d'un service informatique, contre 80% dédiée à la maintenance et l'exploitation des applications existantes.

De plus, différentes études montrent que les coûts de la maintenance n'ont cessé d'augmenter ces dernières années :

2000's	80 – 90%
1990's	70 – 80%
1980's	40 – 60%
1970's	35 - 40%

Tableau 1.1 Coût de la maintenance en pourcentage du budget logiciel [SCH09]

Toutefois, ces estimations sont aujourd'hui pondérées, comme le montre le dernier sondage effectué par *Methods & Tools*, qui révèle que la majorité des participants estiment que leur budget maintenance est inférieur à 50% du budget alloué au cycle de vie total du logiciel :

25% or less of the budget	37%
26% to 50% of the budget	27%
51% to 75% of the budget	24%
more than 75% of the budget	12%

Participants: 433
Ending date: October 2007

Tableau 1.2 Part du budget alloué à la maintenance selon le sondage [MET07]

Si cette inflexion dans la courbe de progression des coûts de la maintenance peut être imputée à diverses évolutions et pratiques dans le domaine (outsourcing, externalisation de la maintenance, utilisation accrue de progiciels, transfert vers des technologies web plus évolutives, ...etc.), ces chiffres et la volonté de plus en plus croissante de réduire ces coûts font de la maintenance du logiciel un sujet d'actualité sérieux qui ne peut être éclipsé par l'intérêt souvent « trop » focalisé sur les projets de développement de nouveaux logiciels.

1.2. Les problèmes liés à la maintenance

Comme nous venons de le voir, la maintenance du logiciel représente une part importante des activités d'un service informatique. De plus, nous observons que très peu d'ouvrages universitaires ont traité du sujet, et bien souvent, les théories ont présenté la maintenance comme une discipline équivalente au développement de logiciels sur des systèmes existants. Elle ne nécessiterait donc pas d'approches ou de qualités particulières. Pourtant,

les acteurs de la maintenance du logiciel sont quotidiennement confrontés à des problèmes spécifiques liés à cette activité particulière.

Le tableau 1.3 montre les résultats d'un sondage mené auprès de spécialistes de la maintenance du logiciel. Cette étude donne une classification des problèmes perçus.

On peut aborder ces problèmes selon les 2 dimensions liées à la gestion des systèmes d'information :

- la dimension managériale
 - ✓ Le pilotage de la performance
 - ✓ L'organisation
 - ✓ Les facteurs humains
- la dimension technique
 - ✓ Méthodologie de développement, de test et de validation
 - ✓ Plate-forme système et outils logiciels
 - ✓ Documentation

Rang	Problèmes de la maintenance
1	Gestion des priorités changeantes
2	Techniques d'essais inadéquates
3	Difficultés à mesurer la performance
4	Documentation du logiciel incomplète ou absente
5	Adaptation aux changements rapides des organisations d'utilisateurs
6	Nombre important de requêtes de changements en attente
7	Difficulté à mesurer et démontrer la contribution de l'équipe de maintenance
8	Moral bas dû au manque de reconnaissance et de respect de la maintenance
9	Peu de professionnels du domaine et encore moins avec expérience
10	Peu de méthodologie, de standards, de procédures et d'outils spécifiques
11	Le code source des logiciels existants est complexe et non structuré
12	Intégration, recoupement et incompatibilité des systèmes existants
13	Les employés de la maintenance ont peu de formation disponible
14	Pas de plans stratégiques concernant la maintenance
15	Difficulté à comprendre les attentes des utilisateurs et d'y répondre
16	Manque de compréhension et de support de la part des gestionnaires
17	Les logiciels de la maintenance opèrent sur des systèmes désuets
18	Peu de volonté et de support pour rénover les applications existantes
19	Perte d'expertise quand les employés quittent l'équipe
	<i>Problème de management</i>
	<i>Problème technique</i>

Tableau 1.3 Perceptions des problèmes liés à la maintenance [DEK92]

2. Le terrain d'investigation : l'armée de Terre

Cette partie nous permet de définir le terrain sur lequel va s'opérer notre travail de recherche.

2.1. Présentation de l'organisme

Le Centre de la Doctrine d'Emploi des Forces (CDEF) est un organisme sous la tutelle directe du chef de l'Etat-major de l'Armée de Terre, chargé en tant qu'expert d'établir et d'améliorer la doctrine d'emploi des forces. On entend par doctrine les principes fondamentaux selon lesquels l'Armée de Terre entend s'organiser, s'entraîner et s'engager dans tout type d'opération.

Le CDEF est composé de 4 divisions dont la Division Simulation et Recherche Opérationnelle (DSRO) chargée de conduire des projets de simulation voués à la formation et à l'entraînement des forces, ainsi que des projets en recherche opérationnelle dans le domaine de l'aide à la décision. C'est au sein de cette division, et plus particulièrement du bureau chargé du Maintien en Condition Opérationnelle (MCO) d'un système de simulation de combat, que nous mènerons notre investigation de terrain.

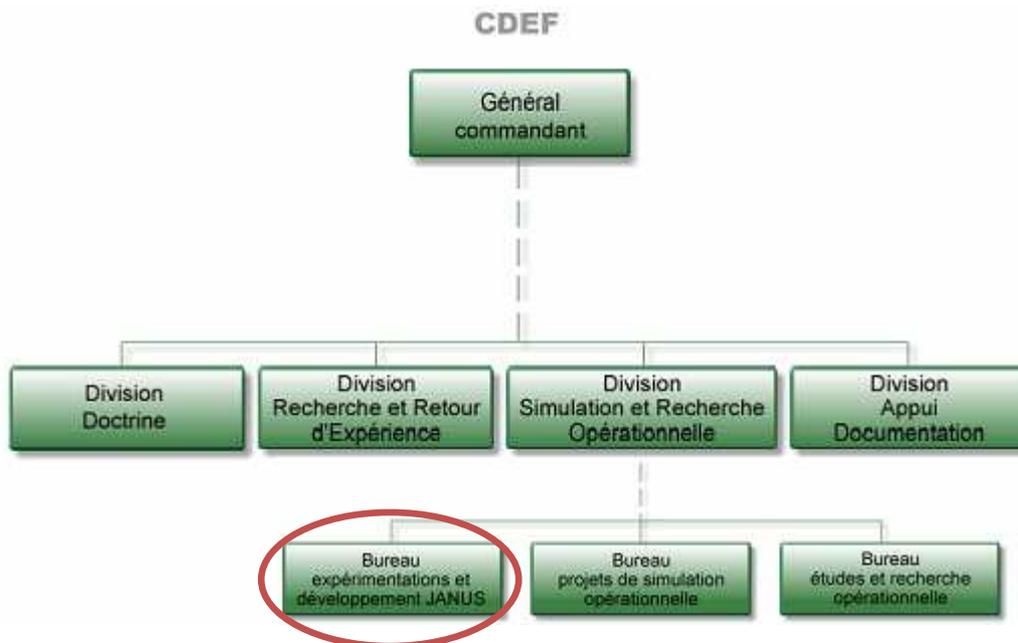


Figure 2.1 Organisation du CDEF [CDE09]

En 2006, le CDEF a été certifié ISO 9001 version 2000. Les missions qui lui sont dévolues sont les suivantes :

- Etablir la doctrine d'emploi des forces
- Améliorer la capacité des forces par le retour d'expérience des opérations françaises et étrangères
- Assurer l'expertise dans le domaine de la simulation de combat et la recherche opérationnelle (aide à la décision)

Dans ce mémoire nous nous intéressons uniquement à l'organisation de la DSRO (*Division Simulation et Recherche Opérationnelle*) et particulièrement à celle chargée de la maintenance des logiciels du système « JANUS » (bureau Expérimentations et développement « JANUS »).

2.2. Le processus « Gérer les projets »

« Un processus est un système dynamique orienté vers la réalisation d'un objectif » [MOR07].

Pour harmoniser et gérer les activités des 3 bureaux qui la composent, la DSRO a décidé de créer un unique processus : le processus « Gérer les projets ».

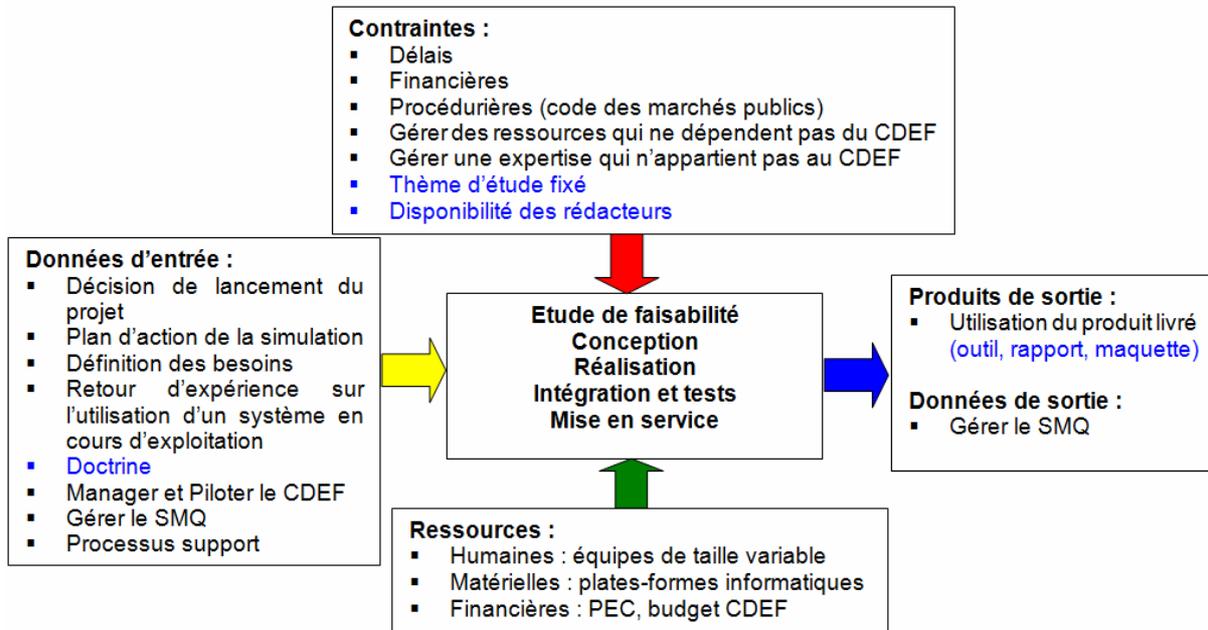


Figure 2.2 Le processus « Gérer les projet » de la DSRO [DSR09]

Toutefois, les missions propres à chacun des 3 bureaux sont quelque peu différentes :

- Le bureau « *MCO JANUS* » est chargé de maintenir en condition opérationnelle (MCO) le système de simulation « *JANUS* »
- Le bureau « *projet de simulation* » est chargé d'assurer la maîtrise d'ouvrage des grands projets de simulation de l'armée de Terre
- Le bureau « *recherche opérationnelle* » est chargé de développer des outils d'aide à la décision

De ce fait, 3 procédures distinctes ont été créées pour guider les activités des différents bureaux selon le type de maîtrise d'œuvre (MOE), interne ou externe :

- ✓ PR1 (projet MOE externe) → pour le bureau « *projet de simulation* »
- ✓ PR2 (projet MOE interne) → pour le bureau « *MCO JANUS* »
- ✓ PR3 (projet MOE réduite) → pour le bureau « *recherche opérationnelle* »

Le bureau « *JANUS* », dont le métier est la maintenance des logiciels de simulation, est tenu de respecter les itinéraires dictés par la procédure PR2 (projet en MOE interne) du processus « Gérer les projets ».

2.3. Le bureau chargé de la maintenance du logiciel

Situé au sein d'une division « métier » (DSRO), le bureau « JANUS » est chargé de maintenir en condition opérationnelle le système de simulation tactique « JANUS » destiné à la formation et à l'entraînement des postes de commandement des forces terrestres.

Les missions du bureau sont de deux types :

- Maintenir le logiciel de simulation
- Développer et maintenir l'interopérabilité du système de simulation avec les Systèmes d'Information Opérationnels et de Commandement (SIOC)

Le bureau est structuré de la façon suivante :

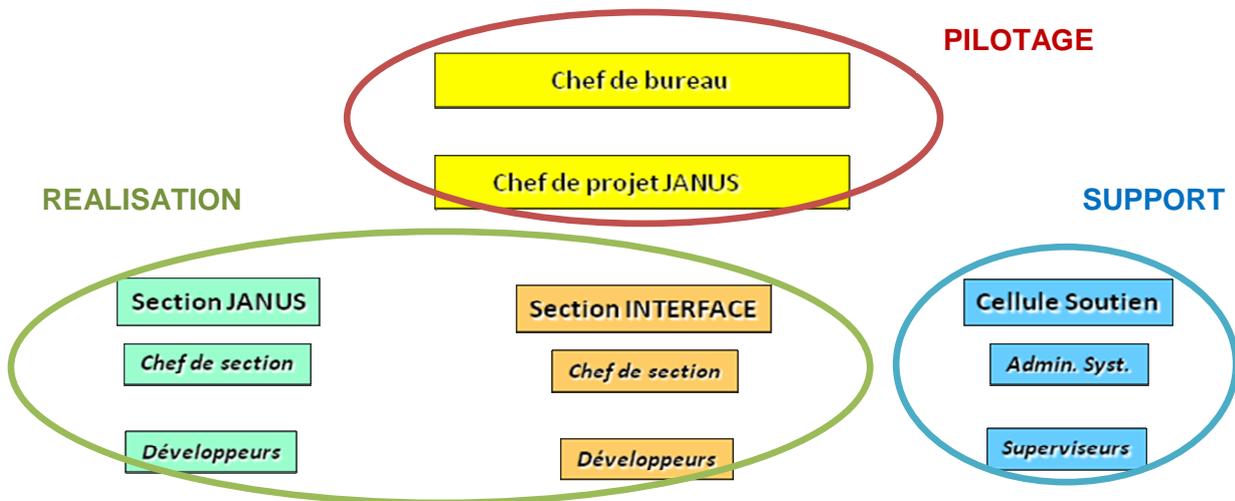


Figure 2.3 Organisation du bureau de la maintenance du logiciel « JANUS »

Ce schéma organisationnel révèle 3 grands acteurs de la maintenance, ainsi que 3 rôles qui leur sont respectivement dévolus :

- *La cellule de pilotage*

Elle est chargée d'organiser l'ensemble du bureau et de définir les orientations fonctionnelles du logiciel.

- *Les cellules de maintenance et développement*

Elles sont chargées d'organiser, conduire et contrôler les travaux de développement et de maintenance du logiciel.

- *La cellule support*

Elle est chargée d'assurer le soutien des centres de formation (notamment le support opérationnel et le déploiement des nouvelles versions), comme le soutien interne des différentes équipes (notamment l'administration du réseau et les tests d'intégration).

3. La problématique émergente

3.1. Maintenance logicielle et projet logiciel

Nous avons vu dans les précédentes parties que les problèmes de la maintenance du logiciel sont de natures diverses, mais surtout, qu'ils font de la maintenance une problématique particulière et différente de la problématique du développement d'un logiciel dans le cadre d'un nouveau projet.

Dans le contexte professionnel qui nous intéresse, le fait de gérer les activités de la division simulation en mode projet (processus « *Gérer les projets* »), semble effacer complètement le processus de maintenance au profit du processus de gestion de projet. Les difficultés que rencontre le bureau chargé de la maintenance du logiciel « *JANUS* » sont diverses :

- Difficulté à piloter efficacement les activités de maintenance
- Difficulté à rendre visible les travaux effectués
- Baisse de la motivation du personnel
- Difficulté à appliquer et suivre le processus de gestion de projet

D'une manière générale, il semble que le processus de maintenance, dont les activités sont récurrentes, ne doit pas être assimilé à un processus de projet dont les activités sont jalonnées et bornées dans le temps.

3.2. Les questions clés

Les questions que ces observations suscitent sont les suivantes :

- Les activités de maintenance du logiciel peuvent-elles s'apparenter ou se confondre avec les activités de développement du logiciel ?
- Si oui, sous quelles conditions, quelles en sont les limites ?
- Si non, alors comment inscrire l'activité de maintenance dans une organisation en mode projet ?
 - la formalisation du processus de maintenance remet-elle nécessairement en cause, l'organisation en place ?
 - si oui, dans quelle proportion ?
 - si non, quels ajustements sont-ils nécessaires ?

Voilà en quelques mots les questions auxquelles nous essaierons d'apporter des réponses tout au long de ce document.

Pour cela, nous nous appuierons dans un premier temps sur une étude approfondie des concepts et des *pratiques exemplaires* [APR06] dans les domaines de la gestion de la maintenance et de la gestion de projet.

Dans un second temps, nous nous efforcerons de définir la transformation de l'organisation actuelle vers un futur souhaitable qui répond aux problèmes posés dans le domaine de la maintenance du logiciel.

CHAPITRE II

ETAT DE L'ART DES CONCEPTS ET DES PRATIQUES

Dans le premier chapitre du document, nous avons mis en exergue les problèmes de la maintenance du logiciel et notre questionnement quant à l'amalgame entre activités de maintenance et activités de projet logiciel.

Dans ce second chapitre, nous nous efforçons de définir les concepts de maintenance du logiciel et de projet logiciel pour en extraire les différences majeures. C'est l'objet de la première partie.

La seconde partie, quant à elle, propose un éventail des référentiels de pratiques exemplaires et des normes en vigueur pour déterminer le référentiel le plus à même de répondre aux problématiques de la maintenance. Ainsi nous naviguons du référentiel CMMi, en passant par ITIL et jusqu'au référentiel S^{3m} spécifique à la petite maintenance du logiciel.

Dans la troisième partie nous abordons, les méthodes d'évaluation des processus. Nous explorons ainsi une méthode de référence, la méthode SCAMPI, puis une méthode qui s'en inspire et qui intéresse plus directement notre étude : S^{3m} Assessment.

4. Définition des concepts

4.1. La maintenance du logiciel

Le SWEBOK (Software Engineering Body of Knowledge) définit la *maintenance* comme « la totalité des activités qui sont requises afin de procurer un support, au meilleur coût possible, d'un logiciel. Certaines activités débutent avant la livraison du logiciel, donc pendant sa conception initiale, mais la majorité des activités ont lieu après la livraison finale (l'équipe de développement ayant maintenant terminée son travail et étant affectée à d'autres travaux) » [ABR05].

La norme ISO 12207 la définit comme « les changements du logiciel et de sa documentation causés par un problème ou le besoin de l'améliorer » [ISO95]. La norme ISO 14764, propre à la maintenance, reprend la même définition.

Pour Robert REIX [REI99], c'est l' « ensemble des opérations permettant de maintenir un système informatique en bon état de fonctionnement. En ce qui concerne le matériel informatique, on distingue la maintenance préventive systématique et la maintenance curative (traitement des pannes). En ce qui concerne le logiciel, la maintenance consiste à modifier les programmes pour les adapter à l'évolution des matériels d'une part, à l'évolution des problèmes d'autre part ».

La maintenance du logiciel peut donc se définir comme une activité de changement, de modification des programmes pour répondre à des besoins de correction ou d'évolution du logiciel.

Où positionner la maintenance dans le cycle de vie du système informatique ?

Historiquement, la maintenance du logiciel est perçue comme la phase consécutive à la phase de développement. Elle débute au moment de la livraison finale et s'arrête lorsque la décision est prise de mettre à la retraite le logiciel, cette décision étant liée à différents facteurs, économiques, technologiques, stratégiques, ...etc. PIGOSKY [PIG94] propose aussi qu'il y ait plusieurs activités de la maintenance qui débutent avant l'opérationnalisation du logiciel.

Autrement dit, il est extrêmement difficile de borner la période de maintenance et de prédire la date de renouvellement d'un système informatique.

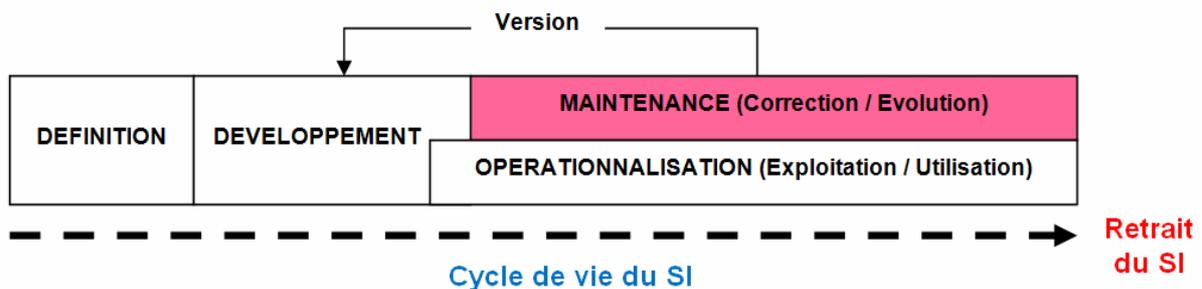


Figure 4.1 La maintenance dans le cycle vie du Système Informatique [CNR01]

Le cycle de vie de la maintenance

Le cycle de la maintenance s'organise autour d'une boucle composée de 3 étapes clés :

- Analyse d'impact (de la correction ou de l'évolution)
- Modification (implémentation de la solution)
- Validation (comprenant les tests de non régression du logiciel)

Comme le montre la figure 4.2, la maintenance débute au moment du déploiement du logiciel (implantation) et entre dans une boucle répétitive de traitement des modifications (analyse, implémentation et acceptation). Ce cycle peut être interrompu momentanément par une migration du logiciel vers une autre plate-forme (nouveau système d'exploitation par exemple) ou un autre langage de programmation (par exemple : réécriture du code dans un langage orienté objet). Il peut également être interrompu définitivement par la mise à la retraite du système ou du logiciel.

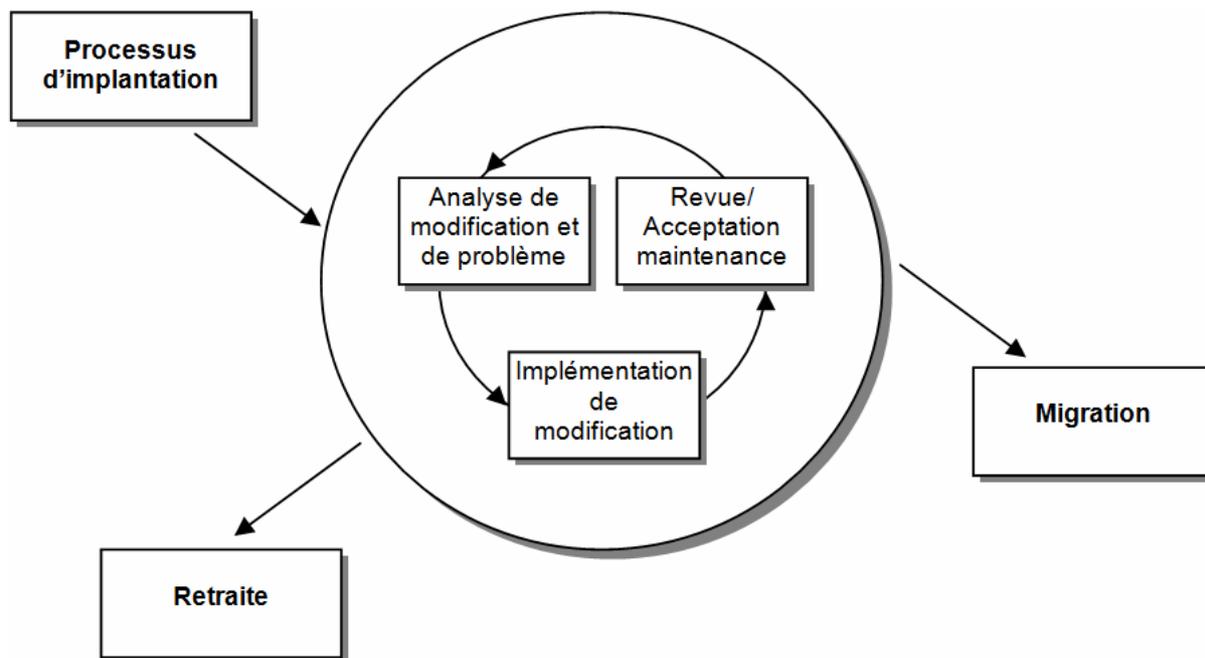


Figure 4.2 Modèle du méta processus de la maintenance du logiciel dans ISO 14764 [ISO98]

Les catégories de la maintenance

La maintenance relève de deux grands besoins distincts que sont le besoin de remédier à un problème du logiciel, d'une part, et le besoin d'améliorer le logiciel, d'autre part. L'ISO 14764 en propose une classification internationalement normalisée :

	Correction	Amélioration
Proactif	Préventif	Perfectif
Réactif	Correctif	Adaptatif

Tableau 4.1 Catégories de la maintenance normalisées par l'ISO 14764 [ISO98]

Cette classification distingue les actions de maintenance en réaction par rapport à une requête des utilisateurs (*réactif*), des actions de maintenance initiées par le groupe de maintenance lui-même et qui tendent à anticiper les problèmes ou les améliorations qu'il estime nécessaires (*proactif*).

La gestion de la maintenance

Le particularisme de la gestion de la maintenance tient à quelques caractéristiques propres à ce domaine. Certains auteurs en ont identifié quelques unes [ABR93] :

- Les requêtes en modification parviennent de manière plus ou moins aléatoire empêchant une planification annuelle stricte
- Les requêtes sont étudiées et classées par ordre de priorité
- La taille et la complexité des requêtes ne nécessitent l'intervention que d'une ou deux personnes
- L'objectif principal recherché par l'équipe de maintenance est le bon fonctionnement quotidien du ou des logiciels en opération (on parle aussi de Maintien en Condition Opérationnelle)
- Les priorités peuvent changer à tout moment (par exemple : la correction d'une anomalie dite « bloquante » passera avant toute autre modification)

Tous ces exemples, dont la liste n'est pas exhaustive, font que la maintenance est gérée par des techniques particulières dites de *gestion de file d'attente*. Ces files d'attente (des requêtes) peuvent être supportées par un logiciel ou un bureau de support de type *help desk* (un bureau d'aide qui réceptionne les requêtes, les vérifie et les oriente vers le gestionnaire ou les mainteneurs).

4.2. Projet de développement logiciel

« [...] tout projet vise à atteindre un but, la plupart du temps non répétitif, dans des situations d'incertitudes fortes, avec une démarche bornée dans le temps [...] » [PIC05].

Chantal MORLEY définit ainsi le mode projet : « Le mode projet se distingue d'une activité répétitive ou d'une mission permanente » [MOR08].

L'AFNOR le définit « comme une démarche spécifique qui permet de structurer méthodiquement et progressivement une réalité à venir » [REI05].

Le projet est donc défini comme une activité non répétitive et bornée dans le temps, c'est à dire, ayant un début et une fin. Dans le cas qui nous intéresse, le projet de développement d'un logiciel débutera par l'expression d'un besoin et se terminera par la livraison d'un produit logiciel, réalisé progressivement en appliquant des méthodes de gestion de projet et répondant parfaitement à ce besoin.

Où positionner le projet dans le cycle de vie du système informatique ?

Comme nous venons de le voir, le projet constitue l'étape primaire dans le cycle de vie du système informatique, de sa définition à sa livraison.

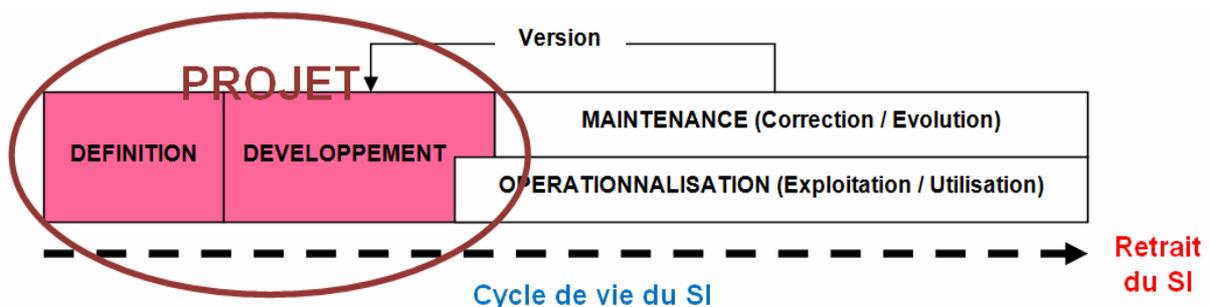


Figure 4.3 Le projet de développement dans le cycle vie du Système Informatique

Le cycle de vie du projet logiciel

Tout projet de développement d'un logiciel se décompose typiquement en 4 grandes étapes [REI05] :

- La *définition* (études préalables, cahier des charges, spécifications)
- La *conception* (conception préliminaire, conception détaillée)
- Le *développement* (codage, tests unitaires, tests d'intégration)
- L'*implantation* (livraison, validation, déploiement)

La gestion de projet

Au-delà de la satisfaction du client, la gestion d'un projet est tournée essentiellement vers la recherche d'un équilibre fragile entre la qualité, les coûts et les délais :

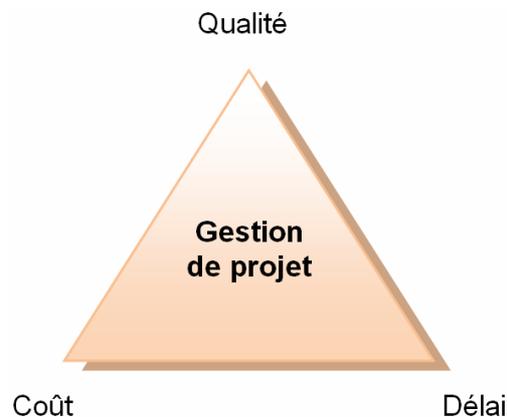


Figure 4.4 Les objectifs de la gestion de projet [REI05]

De plus, le fait que le projet soit borné dans le temps confère à sa gestion un caractère structurant et planifiable. On peut donner quelques exemples d'activité :

- Estimation (des ressources, des moyens et des délais)
- Constitution d'une équipe projet
- Découpage du projet (structurel ou temporel)
- Planification (définition de jalons temporels)
- Gestion des risques (respect du périmètre)

La gestion de projet s'appuie sur des méthodes classées en 2 catégories :

- Les méthodes dites *classiques* (cycle en cascade, en V, en W) se basent sur une définition des exigences au début du projet. Il est difficile de les remettre en cause en cours de réalisation
- Les méthodes dites *agiles* (RUP, XP, SCRUM) se basent sur le principe d'itération. Le produit logiciel est construit brique par brique et fait l'objet de livraisons successives. Le produit livré à chaque itération est fonctionnel et peut entraîner une revue des exigences sur l'itération suivante

4.3. Processus et maturité

Alain BOUVIER [BOU06] définit le processus comme un « ensemble d'activités, de ressources et de compétences plus ou moins indépendantes, organisées autour de la mise en œuvre d'un objectif stratégique, pour fournir, à travers une série d'interactions entre les acteurs, les groupes et les sous-systèmes, un service ou un produit ».

L'ISO 9001 considère qu'un processus est « un ensemble d'activités corrélées ou interactives qui transforment les éléments d'entrée en éléments de sortie ».

Comme le montre la figure 4.5, un processus se définit donc comme un système constitué d'un ensemble d'activités, d'acteurs, de rôles... etc., en interrelation, qui, à partir d'éléments entrants (expression d'un besoin, requêtes en correction/modification) tend à atteindre un but. Ce but peut être un produit (produit logiciel, nouvelle version) ou un service (assistance technique, hotline).

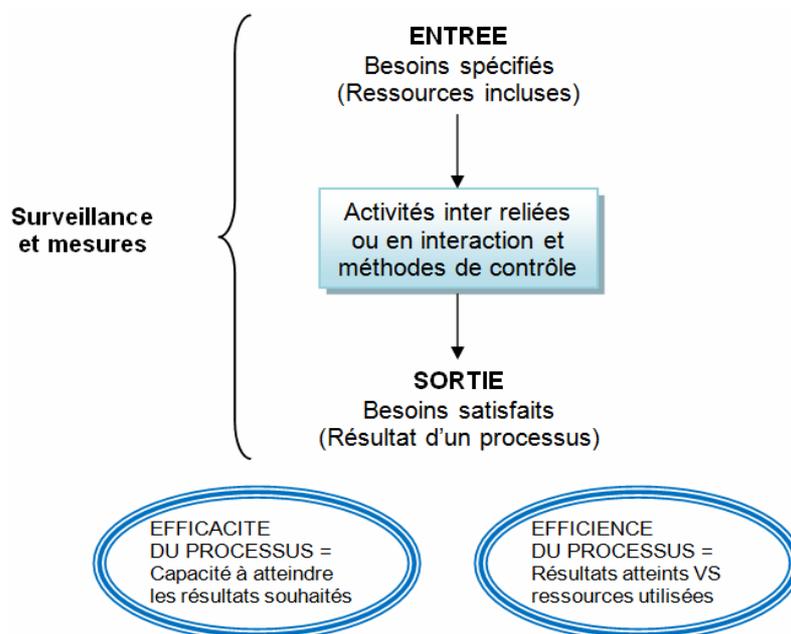


Figure 4.5 Un processus générique [ISO08]

Typologie des processus

L'ISO 9001 [ISO08] propose la typologie des processus suivante :

- Les processus de management des organisations (Processus de pilotage)
 - Les processus de management des ressources (Processus de support)
 - Les processus de réalisation (Processus opérationnels)
 - Les processus de mesure, d'analyse et d'amélioration
- ISO 9001-v2000
+
Mesure des processus

ISO 9001-v2008

Figure 4.6 Typologie des processus de l'ISO 9001 [ISO08]

Philippe LORINO [LOR03] étudie la performance des processus et propose une typologie des processus, ajoutant deux perspectives supplémentaires :

- *Les processus récurrents* pour lesquels le pilotage peut s'appuyer sur des *indicateurs moyens* pour caractériser le fonctionnement du processus (durée moyenne, coût moyen, ratios statiques de qualité)
- *Les processus de projets* dont le pilotage s'appuie sur les techniques de la gestion de projet (calendrier de production, jalons temporels, gestion individualisée des outputs)

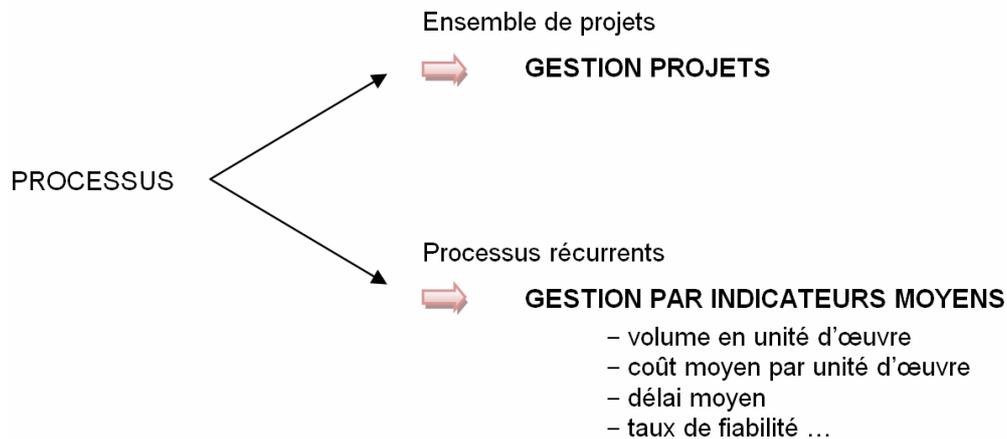


Figure 4.7 Pilotage des processus de projets et des processus récurrents [LOR03]

Si l'on considère la maintenance comme une activité continue et non bornée dans le temps, alors elle correspond à un processus récurrent et, en ce sens, elle diffère d'un processus de projet. Comme le présente Philippe LORINO, le pilotage de ces processus est différent, car ces processus sont de nature différente.

Maturité des processus

Richard BASQUE [BAS06] présente le concept de l'évolution de la maturité d'un processus d'une manière analogique : « *Le concept de maturité, dont il est question ici, rejoint la compréhension intuitive que le lecteur aura sans doute développée face au mot 'maturité' appliquée à l'échelle d'une personne par exemple* ».

GARCIA [GAR93] qui travaille pour le SEI (*Software Engineering Institute*), le premier organisme à proposer *un modèle de référence pour l'évolution de la maturité* des processus du logiciel, définit ce modèle comme « *une représentation des attributs clés, d'entités organisationnelles choisies, qui ont un effet sur la progression de ces unités organisationnelles vers la pleine réalisation de leur potentiel et de leur développement* ».

La maturité d'un processus peut donc être comprise, à l'image de la maturité d'un individu, comme une amélioration progressive des pratiques liées au bon fonctionnement et au bon pilotage des activités qui incarnent le processus. Ici, nous nous intéressons aux activités qui caractérisent la maintenance du logiciel et donc à l'amélioration du pilotage, des méthodes et des procédures qui permettent de les réaliser.

Dans ce cadre, et nous le verrons dans les prochaines parties, il est important de distinguer le concept de *maturité* du concept d'*aptitude*. En effet, le principe de *maturité* peut être vu comme un escalier où, pour atteindre une marche, la marche précédente doit être complètement accomplie. Un processus de niveau 3 doit donc vérifier complètement les exigences du niveau 1 et du niveau 2. Il n'en est pas de même pour l'*aptitude* des

processus. Dans ce dernier cas, les processus d'un organisme peuvent être considérés séparément et une évaluation peut être conduite pour déterminer le niveau d'accomplissement des exigences de chacun des domaines de processus. Par exemple, un domaine de processus peut être évalué au niveau 1, un autre au niveau 3 et un autre au niveau 0 (ici le niveau 0 qui n'existe pas pour la maturité des processus, indique que le processus évalué ne répond à aucune exigence du domaine visé). Certains modèles de maturité utilisent le terme de *capacité* pour définir l'*aptitude* d'un processus.

Autrement dit, la *maturité* est une évaluation verticale par pallier (on parle d'évaluation *étagée*) et l'*aptitude* (ou *capacité*) est une évaluation transversale par domaine de processus (on parle aussi d'évaluation *continue*). La figure 4.8 caractérise les différences entre les 2 concepts :

<i>Niveau</i>	<i>Représentation continue</i> <i>Niveaux d'aptitude</i>	<i>Représentation étagée</i> <i>Niveaux de maturité</i>
Niveau 0	Incomplet	N/A
Niveau 1	Basique	Initial
Niveau 2	Discipliné	Discipliné
Niveau 3	Ajusté	Ajusté
Niveau 4	Géré quantitativement	Géré quantitativement
Niveau 5	En optimisation	En optimisation

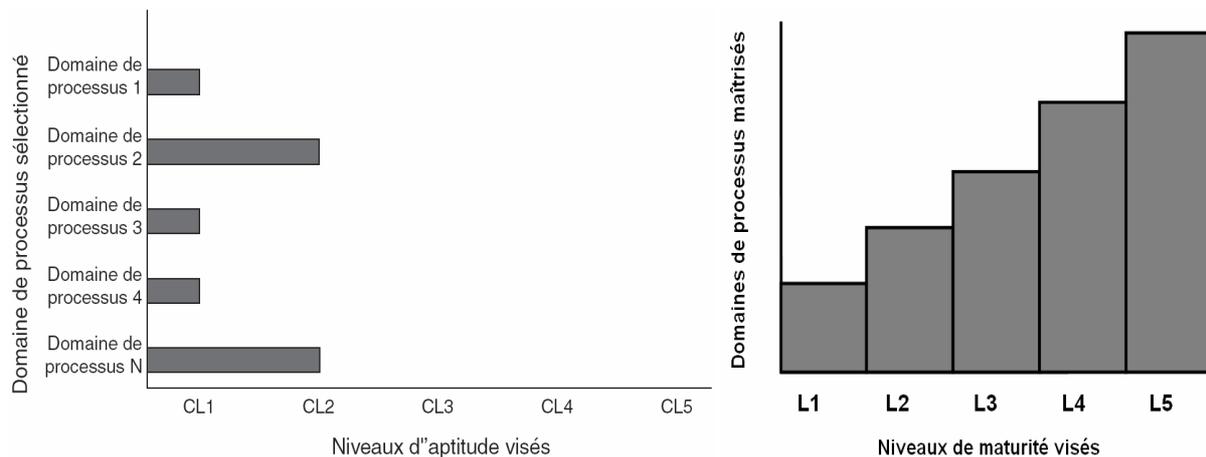


Figure 4.8 Représentation des niveaux d'aptitude et des niveaux de maturité [SEI06]

5. Référentiels de pratiques exemplaires

Un référentiel de pratiques est un socle méthodologique permettant de se repérer dans un domaine d'activité spécifique. Ce socle définit les « bonnes pratiques » à suivre pour atteindre les objectifs liés au domaine d'activité. Nous nous pencherons ici sur 3 modèles du domaine informatique : CMMi, ITIL et S3M. Et nous déterminerons ce qui fait leur différence.

5.1. CMMi (*Capability Maturity Model intégration*)

Le référentiel CMMi repose sur une approche processus, fondement du cercle vertueux mettant en relation les 3 dimensions critiques sur lesquelles peuvent se concentrer les organisations pour améliorer leur fonctionnement :

- Les personnes
- Les procédures et les méthodes
- Les outils et équipements

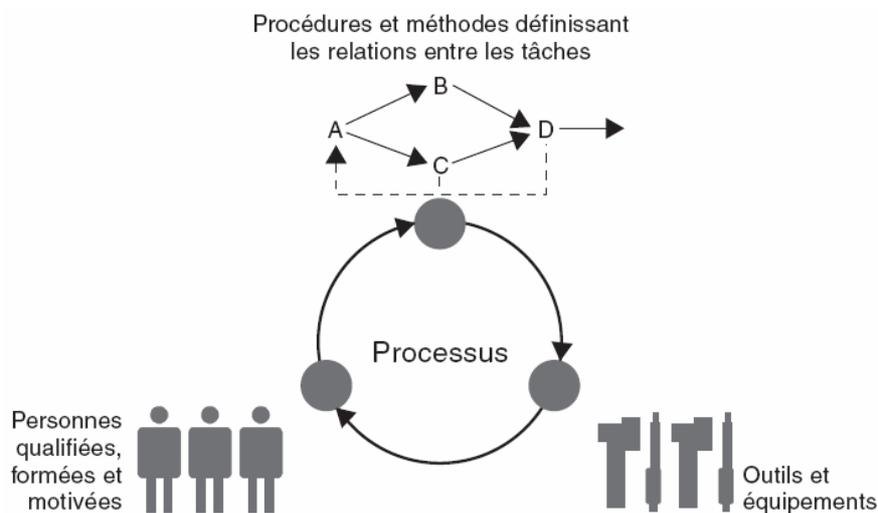


Figure 5.1 Les 3 dimensions critiques d'une organisation [SEI06]

Les origines de CMMi [BAS06]

- Dans les années 70, Watts HUMPHREY redresse le projet 360 d'IBM, en s'appuyant sur une approche innovante basée sur la maturation des processus de développement
- 1980, un rapport commandé par le département de la défense américaine (DoD) révèle que moins de 5% des projets informatiques aboutissent dans les délais
- En 1984, sous l'impulsion du DoD, le SEI (Software Engineering Institute) est créé et Watts Humphrey y dirige l'équipe qui va créer le CMM (Capability Maturity Model)
- 1991, publication de la 1^{ère} version du modèle SW-CMM « CMM for Software »
- Le succès du SW-CMM pousse le modèle à s'élargir à d'autres aspects : SE-CMM « *System Engineering* », SA-CMM « *Software Acquisition* » et P-CMM « *People* »
- 1998, arrêt des travaux sur SW-CMM et démarrage controversé du projet CMMi « *Integration* »
- 2000, publication du CMMi version 1.0
- 2003, publication du CMMi version 1.1
- 2006, publication du CMMi version 1.2
- La prochaine version, CMMi V1.3, est attendue pour novembre 2010

Qu'est-ce que CMMi ?

CMMi se présente comme un référentiel de bonnes pratiques permettant d'aider à augmenter la maturité des processus de développement du logiciel dans une organisation afin de mieux gérer et diminuer les risques d'échec des projets informatiques. Le développement du logiciel s'appuie également sur la norme ISO 12207.

L'architecture fondamentale du référentiel CMMi repose sur 2 représentations, l'une dite « étagée » et l'autre dite « continue ». Chacune de ces représentations donne lieu à une perspective différente d'évaluation, respectivement de la maturité (processus maîtrisé) et de l'aptitude (objectifs et pratiques respectées) :

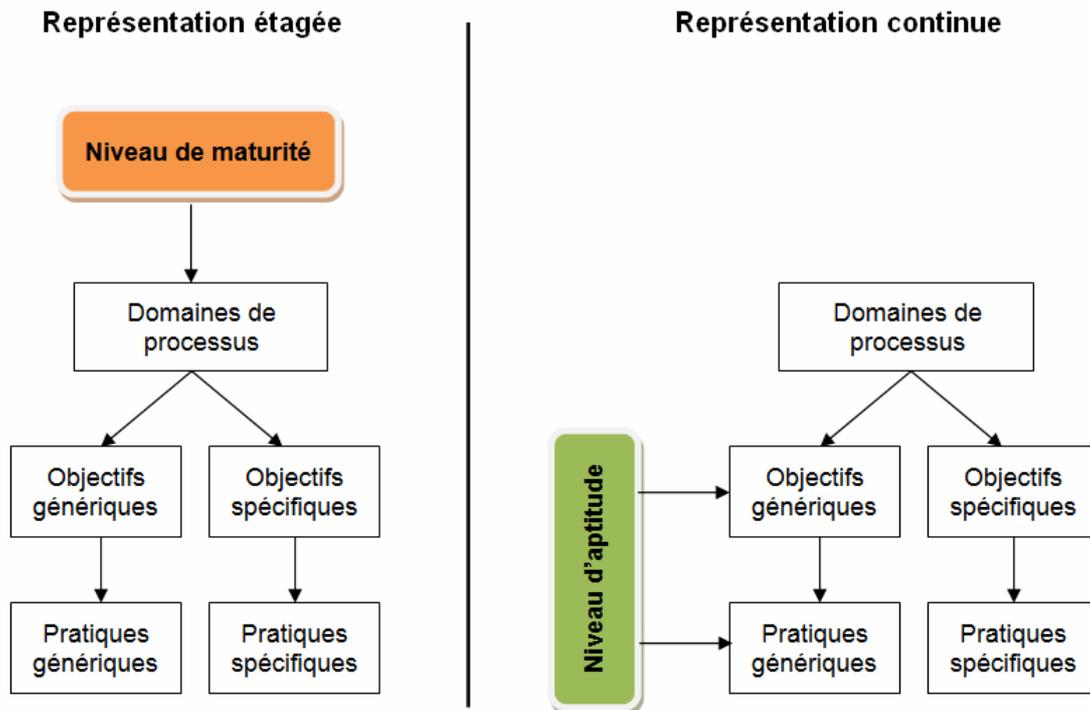


Figure 5.2 Architecture fondamentale du CMMi dans ses 2 représentations [BAS06]

Aujourd'hui, le cadre générique des référentiels processus soutenus par CMMi, se décline en 3 modèles :

- CMMi-DEV pour le développement de systèmes
- CMMi-ACQ pour la maîtrise des activités d'achat
- CMMi-SVC pour la fourniture de service

La version CMMi-V1.2 de 2006 ne définit que la constellation CMMi-DEV.

Zoom sur CMMi-DEV [SEI06]

La constellation CMMi-DEV est constituée d'un ensemble de 22 processus, eux-mêmes regroupés en 4 grandes familles :

- Process Management (Gestion de processus)
 - ✓ Innovation et déploiement organisationnel
 - ✓ Définition du processus organisationnel
 - ✓ Focalisation sur le processus organisationnel

- ✓ *Performance du processus organisationnel*
- ✓ *Formation organisationnelle*
- *Project Management* (Gestion de projet)
 - ✓ *Planification de projet*
 - ✓ *Gestion des accords avec les fournisseurs*
 - ✓ *Surveillance et contrôle de projet*
 - ✓ *Gestion des risques*
 - ✓ *Gestion de projet intégré*
 - ✓ *Gestion de projet quantitative*
- *Engineering* (Ingénierie)
 - ✓ *Gestion des exigences*
 - ✓ *Développement des exigences*
 - ✓ *Solution technique*
 - ✓ *Intégration de produit*
 - ✓ *Vérification*
 - ✓ *Validation*
- *Support* (Support)
 - ✓ *Gestion de configuration*
 - ✓ *Mesure et analyse*
 - ✓ *Assurance qualité processus et produits*
 - ✓ *Analyse et prise de décision*
 - ✓ *Analyse causale et résolution*

Chaque domaine de processus est accompagné d'objectifs et pratiques génériques mais également d'objectifs et pratiques spécifiques comme le montre l'exemple suivant, emprunté au niveau de maturité 2 :

Catégorie	Domaine de processus	Objectifs Spécifiques	
Gestion de processus	Définition du processus organisationnel
	

Gestion de projet	Planification de projet	SG1	<i>Etablir les estimations</i>
		SG2	<i>Développer un plan projet</i>
		SG3	<i>Obtenir l'engagement sur le plan</i>
	Surveillance et contrôle de projet	SG1	<i>Surveiller le projet par rapport au plan</i>
		SG2	<i>Gérer l'action corrective jusqu'à clôture</i>
	Gestion des accords avec les fournisseurs	SG1	<i>Etablir les accords avec les fournisseurs</i>
		SG2	<i>Se conformer aux accords avec les fournisseurs</i>
...	
Ingénierie	Gestion des exigences	SG1	<i>Gérer les exigences</i>
	Solution technique

Support	Mesure et analyse	SG1	<i>Aligner les activités de mesure et analyse</i>
		SG2	<i>Fournir des résultats de mesure</i>
	Assurance qualité processus et produit	SG1	<i>Évaluer de manière objective des processus et des produits d'activités</i>
		SG2	<i>Fournir une image objective</i>
	Gestion de configuration	SG1	<i>Etablir des référentiels</i>
		SG2	<i>Suivre et contrôler les modifications</i>
		SG3	<i>Etablir l'intégrité</i>
...	

Tableau 5.1 Objectifs spécifiques du CMMi niveau 2 [BAS06]

Concernant la maturité, tous les processus sont définis selon 5 niveaux. Les processus d'un même niveau ne peuvent être stabilisés que si les processus de niveaux inférieurs sont eux-mêmes stabilisés, respectant ainsi un principe étagé ou d'empilement :

Niveaux de maturité CMMI	
Niveau	Signification
Niveau 1 : Initial	Le niveau le plus bas montre que l'organisation n'est pas prête, et le projet pas stable. Ce dernier dépend d'une poignée de personnes, qui ne font pas appel à des processus éprouvés. Il se peut cependant que le projet aboutisse, mais en dépassant certainement le budget et le temps alloués. Le projet ne construit pas sur les succès passés.
Niveau 2 : Reproductible	Le projet construit sur ce qui a été appris précédemment, en faisant appel à une certaine discipline et à une gestion de projet basique. De fait, le projet est géré selon les plans, avec étapes-clefs et vérification des coûts et des fonctionnalités.
Niveau 3 : Défini	Ce n'est plus le projet qui dispose d'une bonne discipline, mais l'ensemble de l'organisation, de manière cohérente. Tous les projets s'en trouvent améliorés.
Niveau 4 : Maîtrisé	Les efforts de mesure et de gestion autorisent un contrôle sans effort du développement, avec capacité d'ajuster et adapter des projets précis sans troubler les autres. Les performances des processus sont prévisibles en quantité comme en qualité.
Niveau 5 : optimisation	Les processus sont constamment améliorés de manière incrémentale et innovante. Les objectifs sont revus en permanence pour rester proches des besoins du marché. Les évolutions sont anticipées et gérées de bout en bout.

Tableau 5.2 Les niveaux de maturité CMMi [JOU06]

En conclusion, comme le stipule Richard BASQUE [BAS06] dans son livre, le référentiel CMMi s'inscrit dans le paradigme du projet (de développement du logiciel) borné dans le temps par opposition au paradigme de l'activité continue qui correspond le mieux à la maintenance du logiciel.

De ce fait, le référentiel CMMi ne répond pas aux spécificités de la maintenance [APR06] :

- Les problèmes liés à la maintenance n'y sont pas pris en compte
- La maturité du processus de maintenance n'y est pas traitée spécifiquement
- Les mécanismes d'amélioration de processus ignorent en partie les pratiques propres à la maintenance du logiciel

5.2. ITIL (*Information Technology Infrastructure Library*)

Le référentiel ITIL, comme celui du CMMi, est fondé sur l'approche processus et le principe de l'amélioration continue de DEMING :

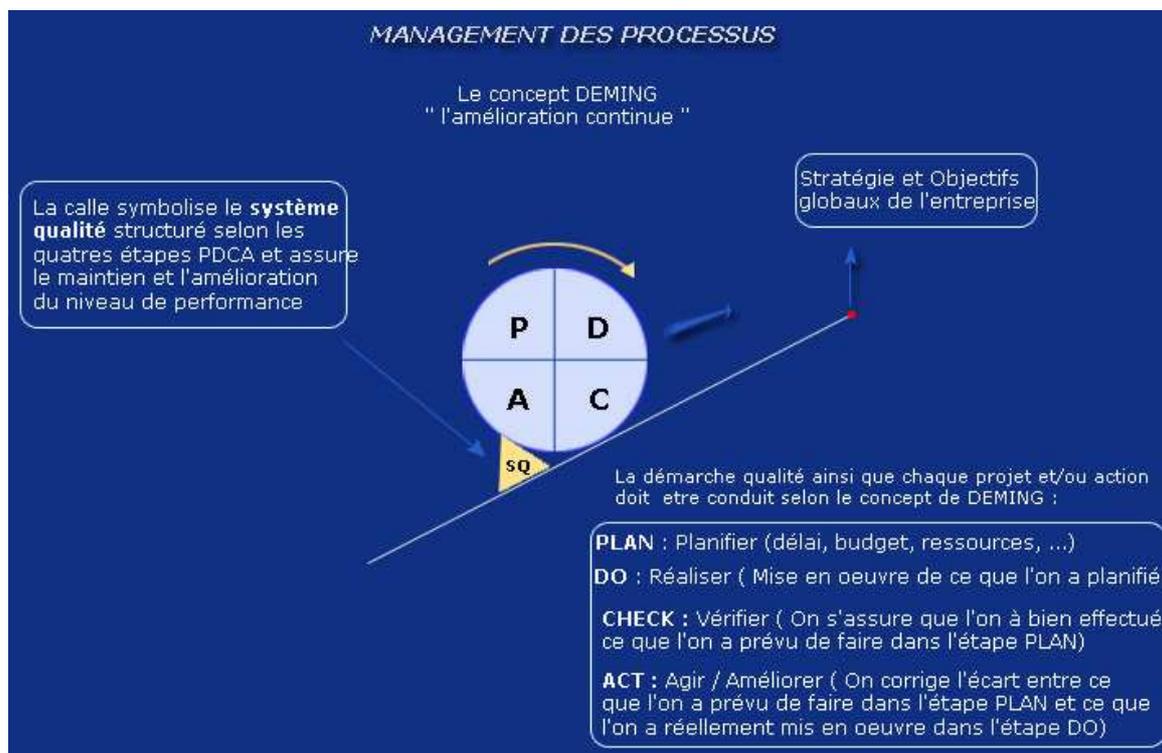


Figure 5.3 Le principe de Deming [QUA09]

Historique [MOU06]

- 1980, le gouvernement britannique décide de réduire les coûts des systèmes d'information des entreprises publiques
- Introduction de la notion de service (production immatérielle d'une activité humaine) destiné à satisfaire un besoin
- Initiée dès 1986, cette approche prend son essor vers 1988
- L'objectif recherché est de créer une méthode universelle mais finalement elle définit les principes généraux et les meilleures pratiques
- 1990, achèvement de la publication d'ITIL version 1, constitué de 40 livres
- 2004, achèvement de la publication d'ITIL version 2, constitué de 9 livres
- 2007, achèvement de la publication d'ITIL version 3, constitué d'un noyau de 6 livres

Qu'est-ce que le référentiel ITIL ?

ITIL définit un cadre de référence décrivant les meilleures pratiques de gestion des services de l'information dans lequel est incluse la notion *d'opérations informatiques*. Dans une approche orientée *services*, les opérations informatiques se focalisent sur la disponibilité des applications mises en service et le maintien de l'infrastructure qui les soutient.

La version 2 d'ITIL a fondé son succès sur deux des livres qui la composent : le soutien des services « *Service Support* » et la fourniture de services « *Service Delivery* ». Elle propose un cadre de référence pour définir un vocabulaire commun et normaliser les processus qui régissent ces services :

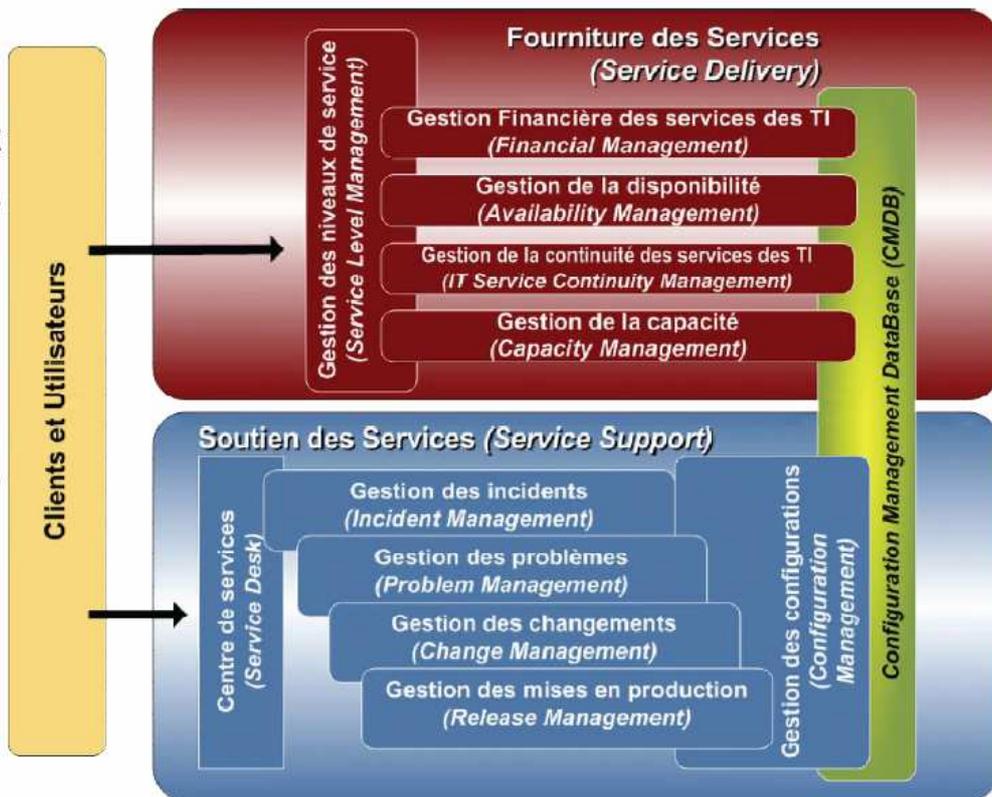


Figure 5.4 Cadre de référence d'ITIL [IT104]

Zoom sur le centre de services

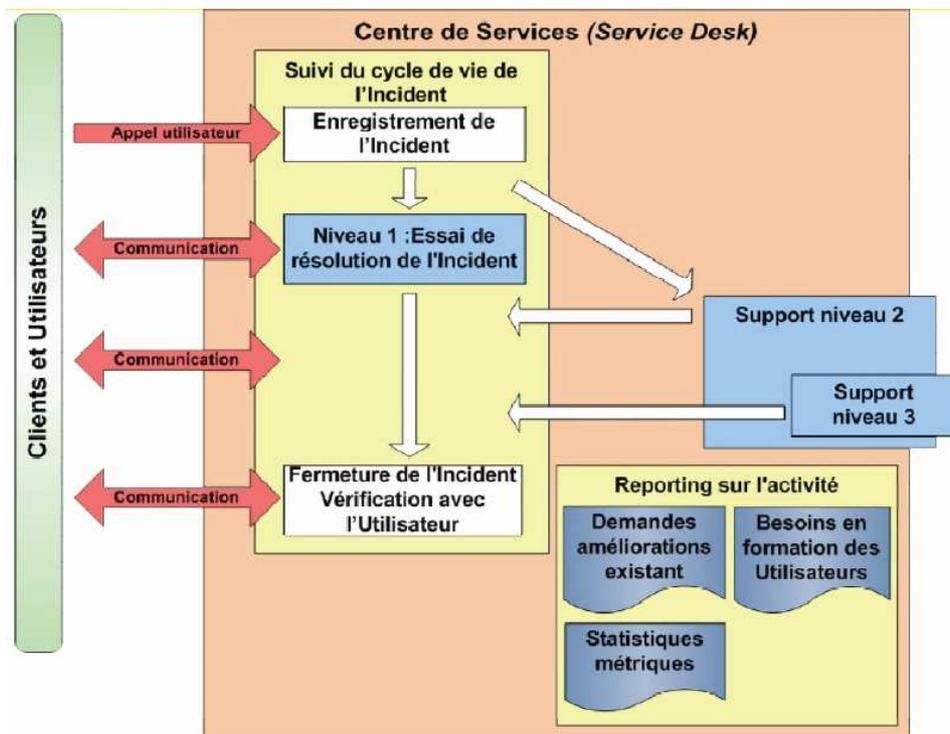


Figure 5.5 Le Centre de Services [IT104]

La figure 5.5 présente le *centre de services*, un des processus décrit par ITIL. Les objectifs du centre de services [ITI04] sont de :

- « *Fournir un point de contact unique pour les Clients* »
- « *Faciliter le retour à la normale d'un service avec l'impact minimum sur les activités des Clients en restant dans le cadre des contrats de services et des priorités métier* »

Les rôles et responsabilités du centre de services mettent en évidence 2 éléments clés :

- Le rôle principal est de gérer le cycle de vie des *incidents* qui affectent les services fournis aux utilisateurs et aux clients
- Si l'incident est un dysfonctionnement, il est transféré vers un niveau de support plus élevé (niveau 2 ou 3)

Les niveaux de support sont au nombre de 3 et définissent le niveau d'intervention pour résoudre l'incident :

- au niveau 1 l'incident est enregistré et classifié et une solution immédiate est recherchée. S'il n'y a pas de solution ad-hoc l'incident est transféré vers une équipe d'expert (support niveau 2)
- au niveau 2 les incidents non résolus par le 1^{er} niveau de support sont pris en charge et un support externe peut intervenir si nécessaire (fournisseurs de matériels et logiciels). Si aucune solution rapide n'est possible, l'incident passe en gestion des problèmes
- le niveau 3 se situe typiquement chez les fournisseurs de matériels et de logiciels. Ce support est initié par le support de niveau 2

ITIL ne gère pas la maintenance du logiciel mais se préoccupe essentiellement de la restauration, le plus rapidement possible, des services interrompus (incidents ou problèmes).

En résumé, ITIL se situe dans le paradigme de l'activité continue. Le périmètre des changements gérés couvre aussi bien les matériels, les équipements, les logiciels réseaux et les logiciels systèmes que les applications en production. Cependant, ITIL s'intéresse au maintien des infrastructures et des services plutôt qu'au maintien des logiciels applicatifs. Ce référentiel vise spécifiquement le domaine des *opérations informatiques* (anciennement *exploitation* et *production*) tel que l'on peut le concevoir pour un service informatique chargé d'assurer un haut niveau de disponibilité des serveurs et des applications utilisés par un service de gestion de la paye par exemple.

Ainsi, l'ISO 14764 [ISO98] précise que les activités d'opérations informatiques (copies de sauvegarde, recouvrement et administration des ordinateurs, ...etc.) sont exclues de la portée de la norme ISO traitant de la maintenance du logiciel.

Les opérations informatiques et les services associés entrent plutôt dans le périmètre de la norme ISO 20000 :2005 [ISO05] de certification des services informatiques.

Mais, si les « *développements* » et les « *opérations* » informatiques sont couverts par des référentiels de bonnes pratiques, qu'en est-il réellement de la « *maintenance du logiciel* » ?

5.3. S^{3m} (Software Maintenance Maturity Model)

Le référentiel S^{3m} est un modèle d'évolution « qui décrit le chemin évolutif nécessaire pour passer progressivement d'un processus individuel à un processus organisationnel plus mature, c'est-à-dire à un processus bien structuré et déployé dans l'organisation » [APR06].

Ce modèle de maturité est tout particulièrement applicable aux processus de la maintenance du logiciel. Comme le montre la figure 5.6, S^{3m} s'appuie sur l'architecture de type « continue » du CMMi pour définir les domaines de processus spécifiques à la *petite maintenance*. Pour chaque domaine de processus, le modèle propose les niveaux d'aptitude (ou de capacité) et les pratiques exemplaires pour améliorer de manière continue et spécialisée la maintenance du logiciel.

On entend par *petite maintenance* les opérations de maintenance qui ne sont pas gérées comme un projet de maintenance. En effet, une application en opération peut, au cours de son existence et pour des raisons diverses, exiger l'implémentation par exemple d'une nouvelle fonctionnalité. Si cette implémentation s'étale sur plusieurs mois et nécessite la constitution d'une équipe, elle pourra être traitée comme un projet de développement avec une date de début et une date de fin. On parle alors de projet de maintenance; on fait simplement évoluer cette application, on n'en développe pas une nouvelle comme dans le cas d'un nouveau projet de développement logiciel. La petite maintenance regroupe toutes les autres opérations de correction et d'amélioration du logiciel.

Par ailleurs, le modèle S^{3m} est fondé sur le concept d'un itinéraire à suivre. Un *itinéraire* se définit comme un ensemble de pratiques liées qui couvrent plusieurs niveaux sur l'échelle du modèle.

Le concept de *facette*, comme à l'image d'un cube, introduit les différents aspects d'un itinéraire, par exemple : « l'ingénierie du changement devra prendre en compte un certain nombre de facettes telles que : a) la documentation technique ; b) la gestion des données » [APR06].

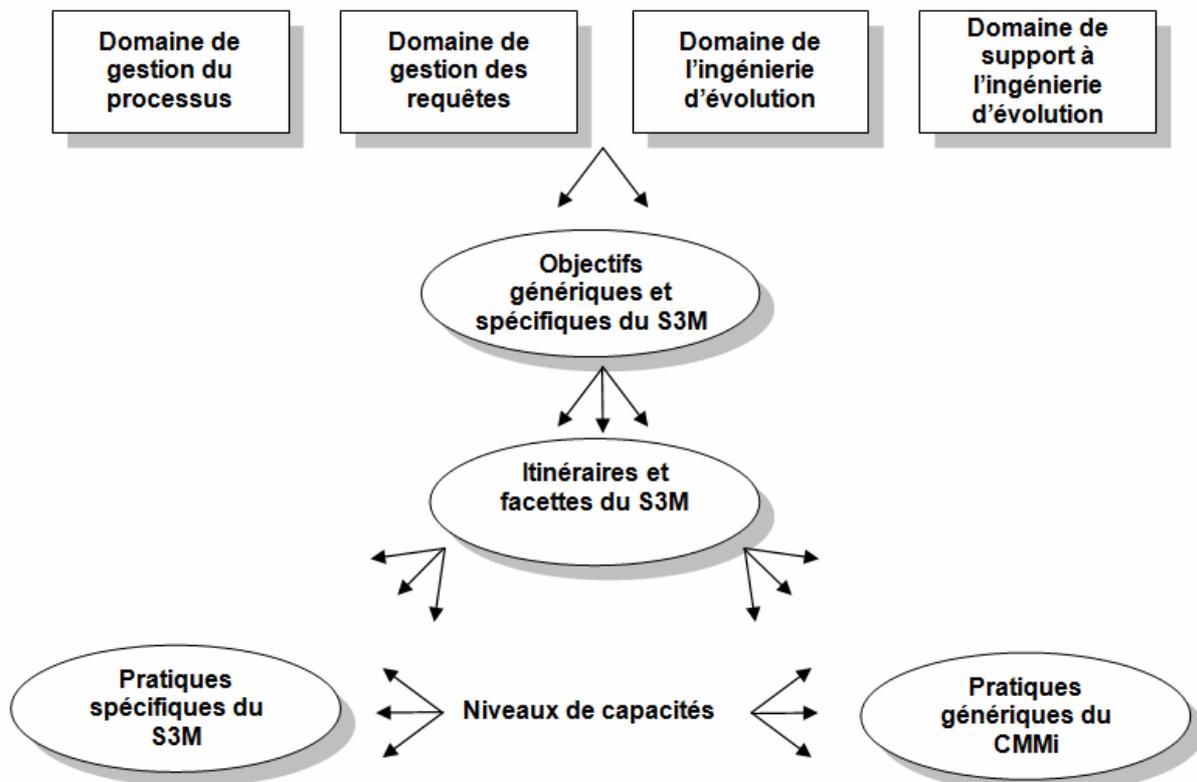


Figure 5.6 Architecture du modèle S^{3m} [APR06]

Comment a été construit le modèle S^{3m} ?

Pour construire le modèle S^{3m} Alain APRIL et Alain ABRAN se sont appuyés sur 2 piliers incontournables et 4 modèles de références.

Les 2 piliers sont le modèle CMMi qui a été présenté dans la partie 5.1 et l'interprétation de la norme ISO9001 :2000 pour le logiciel.

Les 4 modèles de référence sont :

- le modèle de maturité *Camélia* qui compte 9 domaines de processus, 30 itinéraires et 500 pratiques environ afin d'évaluer les produits informatiques [CAM94]
- le modèle CM³ qui s'intéresse à la maintenance corrective pour tenter de compenser les manques du CMMi dans le domaine de la maintenance du logiciel [KAJ01]
- le modèle de ZITOUNI qui propose plusieurs processus spécifiques à la maintenance du logiciel [ZIT96]
- le modèle de maturité CobiT spécialisé dans la gestion, le contrôle et l'audit des technologies de l'information [COB00]

Le modèle de maturité S^{3m} est donc un *mapping* (une transposition) du modèle de maturité CMMi enrichi des processus et pratiques directement inspirés des modèles de la maintenance logicielle et ceux des technologies de l'information (tel que CM³ et CobiT). Comme le CMMi le modèle S^{3m} compte 4 domaines de capacité (ou d'aptitude). Par ailleurs, le modèle complet a retenu 17 itinéraires ou *secteurs clés de la maintenance* (détaillés plus loin dans cette partie), 75 facettes et environ 500 pratiques.

Les caractéristiques que nous venons d'énoncer font donc du modèle de maturité S^{3m} un modèle complémentaire du modèle de maturité CMMi dans le cycle de vie du logiciel. En effet, les 2 modèles réunis définissent l'ensemble des processus et des pratiques exemplaires à respecter pour améliorer les activités de gestion du logiciel tout au long de son cycle de vie (des spécifications à la livraison jusqu'à la maintenance).

Le contexte de la maintenance du logiciel

Le modèle S^{3m} définit parfaitement le contexte de la maintenance et les échanges de flux d'information et de service :

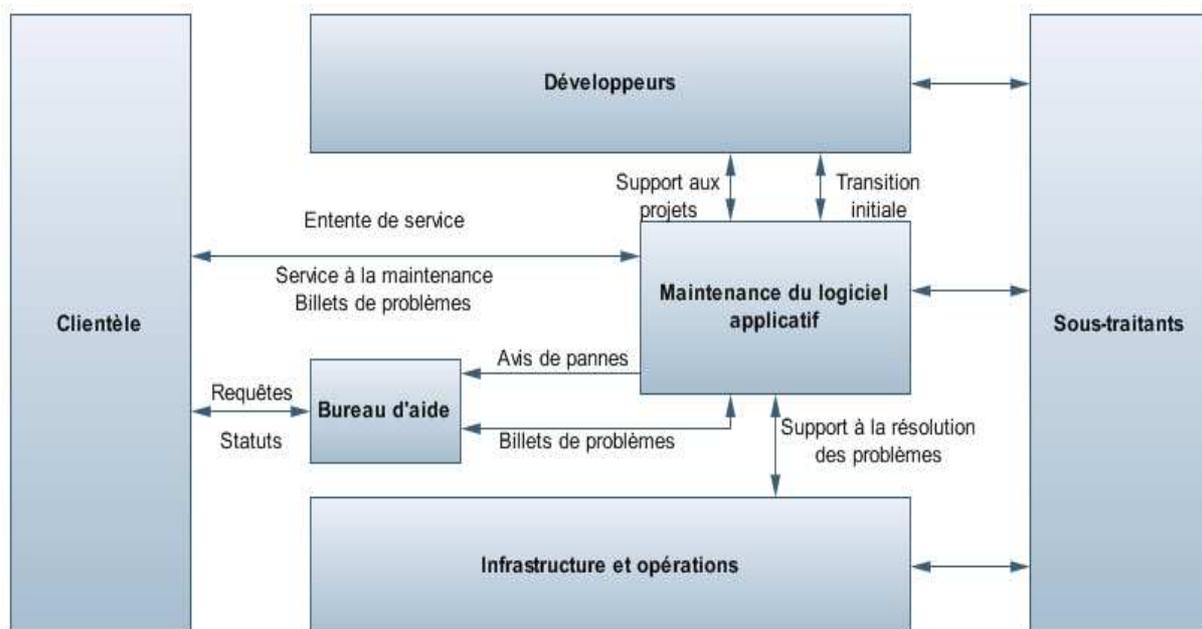


Figure 5.7 Diagramme de contexte du modèle S^{3m} [APR06]

Le diagramme de contexte de la figure 5.7 met en scène différents acteurs et leurs interfaces organisationnelles autour de la maintenance du logiciel applicatif :

- l'interface avec la *clientèle* est la plus importante puisque les demandes ou *requêtes* émanent directement des utilisateurs du logiciel pour être satisfaites.
- le *bureau d'aide* est en charge de cette interface. Il assure le support opérationnel et transmet les requêtes (*billets*) de problèmes vers le groupe de maintenance. Des négociations sont fréquentes pour déterminer les priorités, les budgets et planifier.
- une autre interface entre les *développeurs* et les mainteneurs facilite la *transition* du logiciel, c'est-à-dire, le passage du logiciel du groupe projet vers le groupe de maintenance. Cette transition peut débuter avant la livraison du logiciel.
- certains problèmes, liés aux infrastructures plutôt qu'au logiciel applicatif lui-même peuvent être traités en collaboration avec l'unité des *infrastructures et opérations*.
- enfin, l'organisation peut être amenée à solliciter une *sous-traitance* pour développer, soutenir ou configurer ses logiciels ou progiciels.

Les domaines de processus couverts par S^{3m}

S^{3m} se réfère aux 4 domaines de processus du CMMi (*Gestion du processus, Gestion du projet, Ingénierie, Support*) adaptés à la maintenance pour définir les itinéraires qui rassemblent les bonnes pratiques du modèle :

4 domaines de processus de la maintenance S3M	Secteurs clés de la maintenance
Gestion du processus	1- Focalisation sur les processus 2- Définitions des processus / services 3- Formation organisationnelle 4- Performance des processus 5- Innovation et déploiement
Gestion des requêtes	1- Gestion des requêtes de services et événements 2- Planification de la maintenance 3- Suivi et supervision des requêtes 4- Gestion d'ententes de service et de sous-traitance
Ingénierie d'évolution	1- Coordination avant livraison et transition 2- Services de support opérationnel 3- Services d'évolution et de correction du logiciel 4- Vérification et validation
Support à l'ingénierie d'évolution	1- Gestion de version et de configuration 2- Assurance qualité (services, processus et produits) 3- Analyse causale et résolution de problèmes 4- Rajeunissement, retraite et migration

Tableau 5.3 Domaines et secteurs clés de la maintenance [APR06]

Pour plus de clarté, nous pouvons illustrer le concept de domaine de processus, d'itinéraire, de facette et de pratiques exemplaires de la manière suivante :

- Prenons par exemple le domaine de processus → Gestion des requêtes
Il se compose des itinéraires

- ✓ *Gestion des requêtes de services et événements*
- ✓ *Planification de la maintenance*
- ✓ *Suivi et supervision des requêtes*
- ✓ *Gestion d'ententes de service et sous-traitance*

➤ Considérons maintenant l'itinéraire particulier → *Suivi et supervision des requêtes*

Il se compose des facettes

- ✓ *Suivi des activités planifiées et approuvées*
- ✓ *Réviser et analyser les écarts*
- ✓ *Prendre des mesures correctives*

➤ Regardons maintenant la facette spécifique → *Réviser et analyser les écarts*

Au niveau de capacité 2, elle se compose de la pratique exemplaire

- ✓ *Des revues techniques et des échanges intergroupes ont lieu périodiquement avec les responsables des logiciels de la maintenance*

Les niveaux d'aptitude

Alain APRIL et Alain ABRAN [APR06] définissent les niveaux de capacité (ou d'aptitude) de la maintenance, de la manière suivante :

Niveau 0 : Inexistant	Le processus n'est pas effectué par l'organisation, l'organisation n'est pas consciente de l'existence de ce processus ou il n'en existe pas d'évidence
Niveau 1 : Initial, improvisé	Reconnaissance de la pratique mais elle est faite de façon informelle
Niveau 2 : Discipliné – Répétable mais intuitif	Conscience de la bonne pratique, mise en place de cette pratique ou exécution d'une pratique similaire
Niveau 3 : Processus personnalisé	La pratique exemplaire est comprise et effectuée conformément à une procédure organisationnelle documentée
Niveau 4 : Géré quantitativement et mesurable	La pratique exemplaire est effectuée formellement, est gérée quantitativement, conformément à un objectif et avec des limites établies
Niveau 5 : En optimisation	La pratique exemplaire est sous contrôle statistique, conformément à un objectif et à l'intérieur des limites établies

Tableau 5.4 Les niveaux d'aptitude de la maintenance [APR06]

Du *niveau 0* au *niveau 2* l'organisation progresse d'une approche immature à une approche disciplinée de la maintenance. Les activités se rapprochent peu à peu des pratiques exemplaires. Les processus qui caractérisent les activités commencent à être formalisés et documentés.

Du *niveau 3* au *niveau 5* l'accent est mis sur la formation des employés pour interpréter concrètement les processus et les personnaliser. Peu à peu les processus sont contrôlés et mesurés pour tendre vers une pratique d'amélioration continue des processus de la maintenance.

6. Méthodes d'évaluation des processus

L'évaluation des processus est une démarche qui ne s'improvise pas. Elle doit nécessairement répondre à des principes clairement définis. Ces principes reposent généralement sur une méthode éprouvée pour garantir l'exploitabilité des résultats.

Le SWEBOK [ABR05] définit deux critères essentiels qui permettent de caractériser une méthode. Une méthode doit être :

- *Répétable* : La méthode fournit des résultats similaires si la même équipe d'évaluateurs évalue deux fois la même organisation
- *Reproductible* : La méthode est applicable dans des milieux différents et par des évaluateurs différents tout en obtenant des résultats similaires

La plupart des méthodes d'évaluation sont organisées selon des étapes bien distinctes qui peuvent se présenter comme suit :

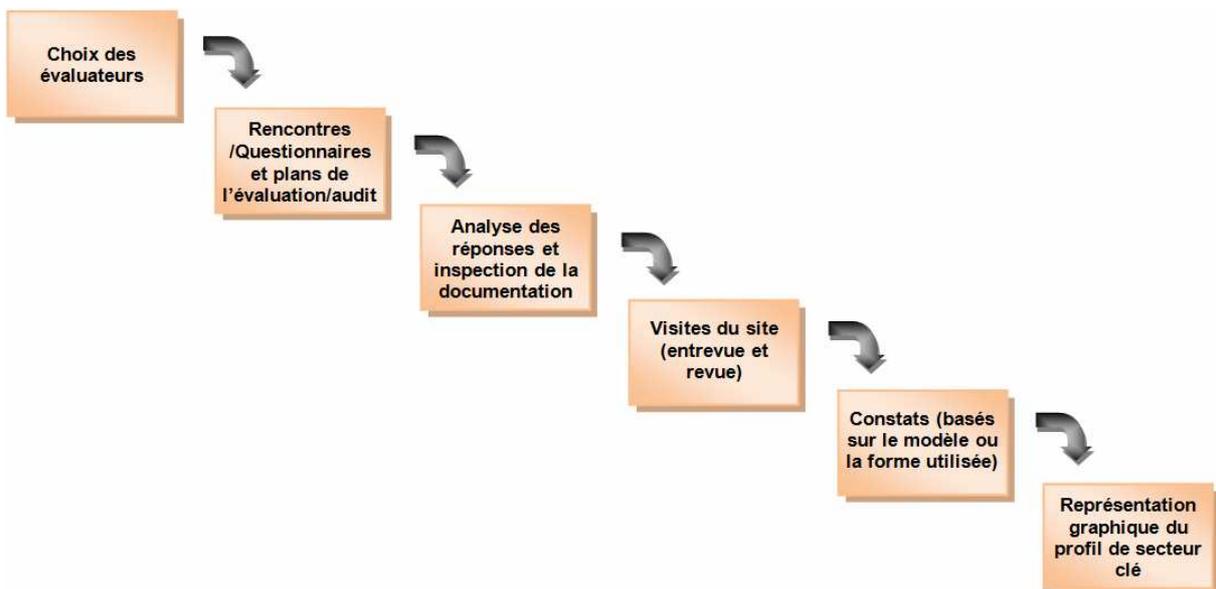


Figure 6.1 Les grandes étapes des méthodes d'évaluation [APR06]

6.1. La méthode SCAMPI

SCAMPI (« *Standard CMMI Appraisal Method for Process Improvement* ») est une méthode développée par le SEI. Elle s'appuie sur le modèle CMMi pour évaluer la maturité des processus déployés par un organisme.

La méthode SCAMPI propose 3 types (classes) d'évaluation :

- **CLASSE C** : est généralement utilisée pour évaluer la maturité des processus avant d'initier une démarche CMMi. Elle permet d'appréhender le niveau de maturité
- **CLASSE B** : est utilisée dans une logique de déploiement du référentiel et fournit de précieuses indications à l'organisme quant à ses forces et faiblesses, ainsi que les actions à mener pour atteindre un niveau de *Maturité* ou de *Capacité* supérieur
- **CLASSE A** : est utilisée pour valider (certifier) la mise en œuvre des pratiques d'un niveau de *Maturité* ou de *Capacité* donné

La démarche se décompose en 11 processus, eux-mêmes répartis sur 3 phases :

Phases	Processus
<i>Planification et préparation de l'évaluation</i>	Analyse des exigences
	Développement du plan d'évaluation
	Sélection et préparation de l'équipe
	Obtenir et analyser des preuves objectives initiales
	Préparation à la collecte de données objectives
<i>Évaluation</i>	Revue des éléments objectifs
	Vérification et validation des éléments objectifs
	Documenter les éléments objectifs
	Générer les résultats de l'évaluation
<i>Présentation des résultats</i>	Livraison des résultats de l'évaluation
	Archiver les bénéfices de l'évaluation

Tableau 6.1 Les 3 phases et les 11 processus de la méthode SCAMPI [SCA06]

Chaque processus se compose d'activités qui organisent les travaux de(s) évaluateur(s) :

➤ *Analyse des exigences*

Le but est de comprendre les objectifs visés par l'organisation pour définir de façon cohérente les objectifs et contraintes de l'évaluation.

➤ *Développement du plan d'évaluation*

L'analyse précédente doit être documentée pour regrouper les exigences, les accords entre participants, les estimations, une étude de risque, le planning de l'évaluation et les contraintes techniques.

➤ *Sélection et préparation de l'équipe*

Ce processus permet de s'assurer que l'équipe chargée de l'évaluation satisfait l'exigence d'une expérience et d'une formation suffisante.

➤ *Obtenir et analyser des preuves objectives initiales*

Le but de cette étape est de recueillir les informations clés permettant de comprendre comment l'organisation peut s'accorder au modèle d'évaluation.

➤ *Préparation à la collecte de données objectives*

Il s'agit ici de définir la stratégie selon laquelle les données seront collectées sur le site.

➤ *Revue des éléments objectifs*

A cette étape, les interviews, l'exploitation des documents et outils peuvent être conduits pour évaluer l'organisation.

➤ *Vérification et validation des éléments objectifs*

Les évaluateurs cherchent ici à expliquer pourquoi et comment les éléments objectifs

s'accordent avec les pratiques du modèle.

➤ *Documenter les éléments objectifs*

Les activités de ce processus consistent à consolider de manière formelle les éléments collectés pour faire apparaître les forces et faiblesses de l'organisation.

➤ *Générer les résultats de l'évaluation*

Il s'agit ici, à partir des travaux précédents, de déterminer les niveaux de chaque processus clé ainsi que les niveaux de capacité et/ou maturité.

➤ *Livraison des résultats de l'évaluation*

Tous les résultats sont consignés dans un document et présentés aux intéressés.

➤ *Archiver les bénéfices de l'évaluation*

La méthode préconise de terminer la démarche par une boucle d'auto-évaluation pour en retenir des leçons à archiver et à exploiter pour les prochaines évaluations.

6.2. La méthode S^{3m} Assessment

La méthode S^{3m} Assessment a été spécialement conçue pour des organisations de petite ou moyenne taille. Elle s'appuie sur la norme ISO 15504 [ISO98a] qui encadre l'évaluation des processus inhérents aux services informatiques et en particulier sur les exigences d'une méthode d'évaluation.

Ces exigences sont organisées sur 4 grands domaines :

- *Définition des données d'entrées de l'évaluation*
- *Responsabilités*
- *Le processus d'évaluation*
- *Enregistrement des résultats de l'évaluation*

Pour s'adapter au contexte économique et aux faibles moyens des petites organisations, la méthode propose 2 approches d'évaluation :

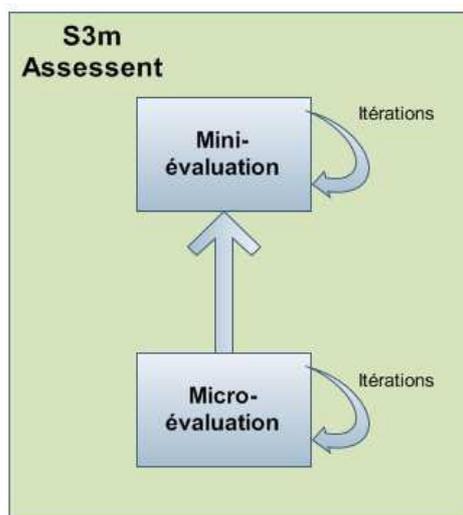


Figure 6.2 L'approche étagée de la méthode S^{3m} Assessment [LEB08]

La micro évaluation est une évaluation minimaliste qui donne la « couleur » des processus évalués. Elle se réduit uniquement à 3 étapes :

- *Le choix des évaluateurs, rencontres et plan d'évaluation*

Il s'agit de déterminer la personne la plus apte à mener l'évaluation et définir les objectifs et le plan d'évaluation.

- *Questionnaire, entrevues et revues*

L'étape consiste à préparer les interviews sur la base d'un questionnaire pour collecter le maximum d'information pertinente qui permettront d'évaluer la capacité des processus.

- *Constats et représentation du profil par secteur clé*

Le but est de rédiger un rapport de synthèse et de représenter les résultats sous une forme la plus visuelle possible.

La mini évaluation, quant à elle, se réfère largement à la méthode SCAMPI, sans toutefois, en retenir l'exhaustivité. Plus complète et plus précise que la précédente, elle offre la possibilité d'évaluer efficacement chaque secteur clé de la maintenance afin de garantir la fiabilité des résultats :

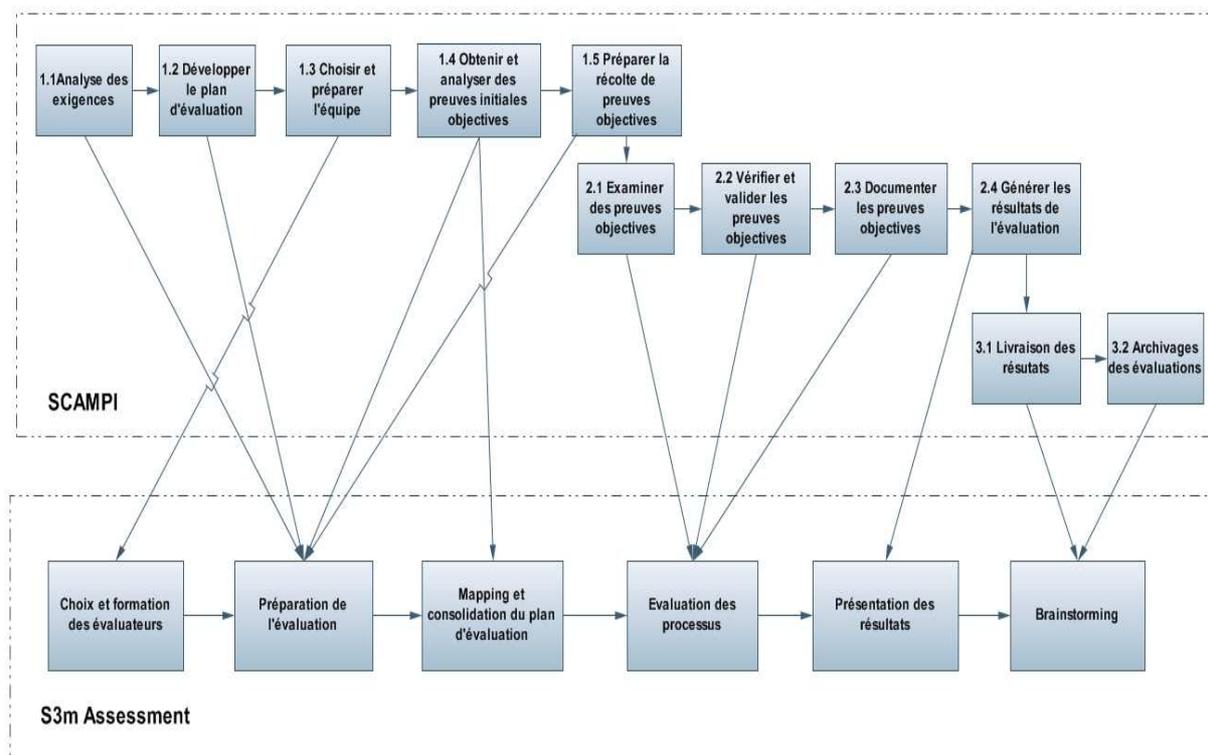


Figure 6.3 Le passage de la méthode SCAMPI à S^{3m} Assessment [LEB08]

- *Choix et formation des évaluateurs*

Ici, l'évaluateur doit être compétent et formé. Il doit posséder une certaine maîtrise du modèle S^{3m}.

- *Préparation de l'évaluation*

Les rôles des participants, les objectifs et le planning de l'évaluation doivent être définis dans un cadre bien précis.

➤ *Mapping et consolidation du plan d'évaluation*

Il s'agit de prendre la mesure des pratiques du modèle qui seraient exclus du cadre de l'organisation afin d'adapter le questionnaire aux réalités de l'organisation.

➤ *Evaluation des processus*

A cette étape, l'évaluateur attribue une cote à chaque pratique évaluée en fonction du degré d'accomplissement de la pratique exemplaire.

➤ *Présentation des résultats*

Cette étape vise à représenter les résultats de l'évaluation de manière complète, objective et intelligible.

➤ *Brainstorming*

L'étape finale consiste à présenter les forces et les faiblesses de l'organisation et à définir un plan d'amélioration continue des processus.

6.3. Conclusion sur les concepts et les pratiques

Pour terminer le chapitre II, il est intéressant de résumer ce que nous avons appris de l'état de l'art des concepts et des pratiques exemplaires, du projet de développement du logiciel à la maintenance du logiciel.

Maintenance logicielle VS Projet de développement logiciel

Comme nous l'avons vu, les principales différences entre maintenance et projet résident dans le fait que la maintenance est un processus récurrent alors que le projet est borné dans le temps.

Dans le tableau présenté ici nous proposons un résumé, issu de notre recherche, des principales différences entre maintenance et projet logiciel :

Maintenance	Projet
Processus récurrent	Processus borné dans le temps
Une à deux personnes par requête	Une équipe projet constituée ad-hoc
Technique de gestion par file d'attente	Techniques de gestion de projet
Gestion d'impact et des priorités	Gestion des risques
Objectifs focalisés sur le bon fonctionnement opérationnels	Objectifs focalisés sur la qualité, les coûts et les délais

Tableau 6.2 Différences entre maintenance logicielle et développement logiciel

Les activités spécifiques de la maintenance montrent qu'elles ne peuvent pas être assimilées à celles d'un projet de développement logiciel. De ce fait, la maintenance du logiciel doit être étudiée et considérée de manière spécifique.

Comparaison des référentiels de pratiques exemplaires

La présentation des pratiques exemplaires nous permet d'identifier les domaines d'application de CMMi, ITIL et S^{3m} ainsi que les normes ISO qui s'y rattachent :

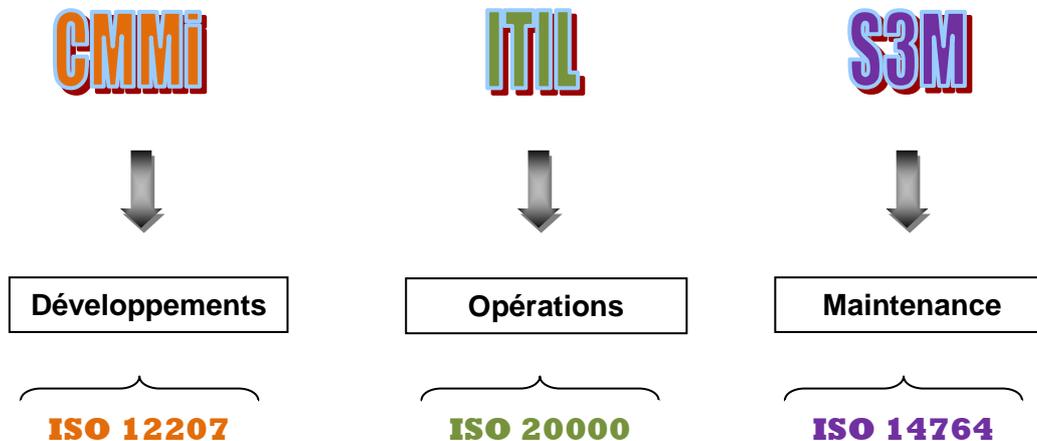


Figure 6.4 Domaine d'application des pratiques exemplaires

Nous pouvons les situer également dans le cycle de vie du système informatique :

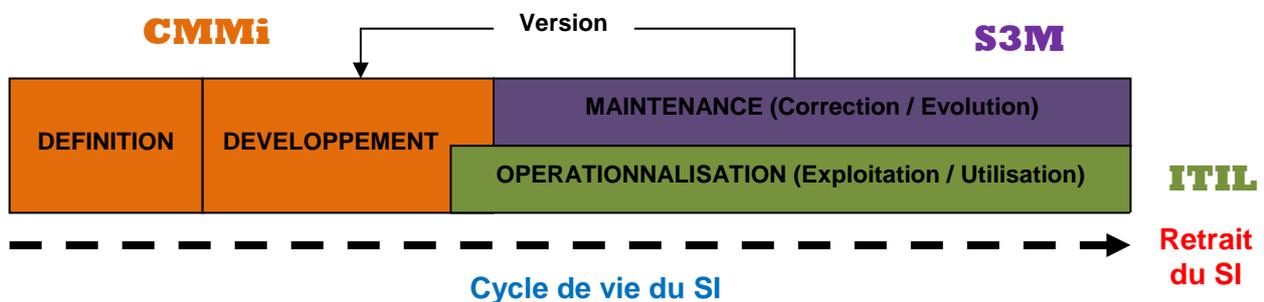


Figure 6.5 Les pratiques exemplaires dans le cycle de vie du SI

Nous voyons d'après ces représentations que chacun des 3 référentiels CMMi, ITIL et S^{3m} trouve une place qui lui est spécifique dans le cycle de vie du système informatique :

- CMMi pour la phase de développement du logiciel (projet logiciel)
- ITIL pour la phase de soutien des infrastructures informatiques (opérations informatiques)
- S^{3m} pour la maintenance du logiciel (hors projet de maintenance, comme c'est le cas lors d'une évolution majeure du logiciel qui est conduite comme un projet logiciel)

De plus, le référentiel le plus adapté à la maintenance du logiciel est celui du modèle S^{3m} car il traite spécifiquement des activités de maintenance du logiciel alors que les deux autres ont été conçus pour accompagner, soit les projets de développement logiciel (CMMi), soit la maintenance des infrastructures informatiques (ITIL).

Nous choisissons donc le modèle S^{3m} pour conduire notre investigation de terrain.

CHAPITRE III

DEMARCHE DE MODELISATION DU CHANGEMENT

Après avoir déterminé dans le chapitre précédent que le modèle S^{3m} constitue une référence pour traiter des activités de maintenance logicielle, nous présentons dans ce chapitre la démarche que nous allons suivre pour définir les changements organisationnels induits par l'évolution de la maturité du processus de maintenance.

Dans la première partie nous posons les hypothèses sur lesquelles reposent notre démarche, et notamment les particularités de la maintenance du logiciel par rapport à une approche projet.

Pour définir le changement organisationnel nous nous appuyons sur le cadre de référence de la méthode EKD-CMM que nous présentons dans la seconde partie.

La dernière partie présente l'approche personnalisée que nous décidons d'adopter pour conduire notre étude de cas, de l'évaluation de la maturité du processus de maintenance actuel à la modélisation du processus envisagé.

7. Les hypothèses

7.1. Les questions liées au contexte de la maintenance

Comme nous l'avons vu lors de la présentation de l'organisme dans le 1^{er} chapitre de ce document, le fonctionnement des bureaux qui composent l'organisation de la simulation, est sensé respecter les principes d'un processus de gestion de projet et d'appliquer les procédures qui en découlent :

- Procédure de gestion de projet en maîtrise d'œuvre externe
- Procédure de gestion de projet en maîtrise d'œuvre interne

Pour le bureau « JANUS », objet de notre étude et chargé de la maintenance du logiciel « JANUS », la procédure à suivre est la procédure de *gestion de projet en maîtrise d'œuvre interne*, dont voici l'organigramme :

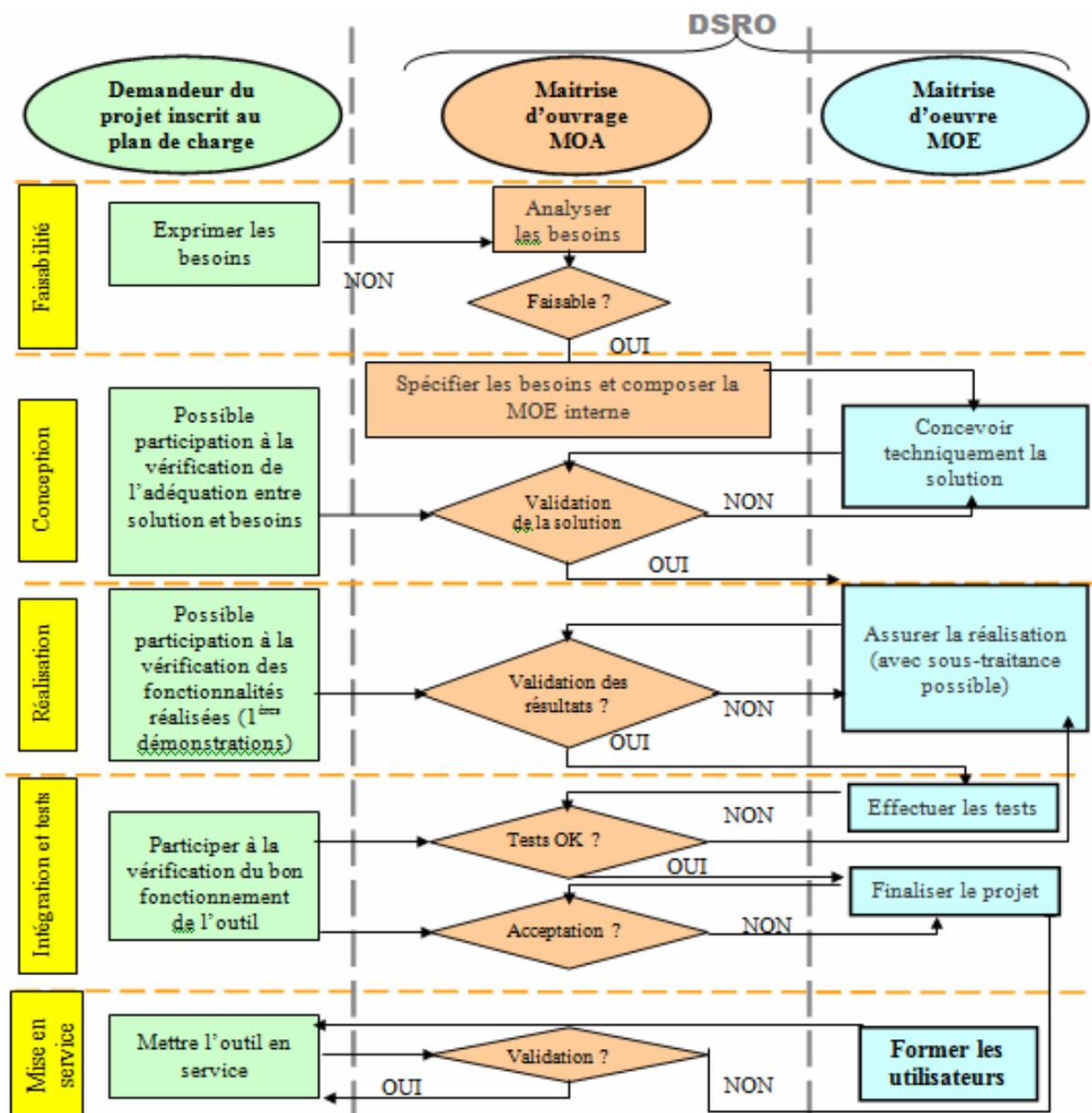


Figure 7.1 Procédure relative à un projet réalisée avec une MOE interne [DSR09]

Cette procédure qui découle du processus de gestion de projet, est organisée selon les grandes étapes de la gestion de projet (*faisabilité, conception, réalisation, intégration et tests, mise en service*). Les outils de suivi des activités, l'organisation de l'espace de travail partagé sont conçus selon une approche orientée projet.

Cependant, nous constatons que le processus de gestion de projet ne nous permet pas de piloter réellement les activités quotidiennes du bureau de maintenance :

- Gestion des requêtes en correction/évolution
- Gestion des priorités
- Gestion des délais
- Gestion du support opérationnel ...

7.2. Les hypothèses retenues

L'état de l'art des concepts et les pratiques exemplaires des référentiels CMMi, ITIL et S^{3m} nous conduisent à considérer les spécificités de la maintenance du logiciel face aux activités de la gestion de projet. Comme nous l'avons vu dans le chapitre II, l'un appartient au paradigme de l'activité continue et l'autre à celui du projet, borné dans le temps.

Cependant, aujourd'hui, les services informatiques sont souvent structurés autour de leurs projets et, conséquemment, utilisent le mode projet pour gérer l'ensemble de leurs activités.

Dans ces conditions, à moins de remettre en cause cette approche qui est maintenant insérée dans la culture organisationnelle, il est nécessaire de trouver le moyen de formaliser le processus de maintenance des logiciels et de le rendre visible dans l'organisation actuelle.

D'autre part, pour remettre en cause une organisation, il est indispensable d'avoir été mandaté pour cela et d'obtenir un soutien hiérarchique fort. Or, ce n'est absolument pas le cas de notre étude car, s'il est possible d'envisager une amélioration du processus de maintenance dans l'organisation (bureau de maintenance « JANUS »), il semble difficile de remettre en cause la structure qui l'encadre (la division simulation).

De ce fait, nous poserons les hypothèses suivantes :

- La maintenance du logiciel ne peut être traitée comme un projet de développement du logiciel
- Le processus de maintenance ou les procédures qui en découlent peuvent s'inscrire dans une organisation en mode projet sans la remettre en cause nécessairement

De plus, nous réduisons le périmètre de notre étude à la *petite maintenance*, qui s'attache à ne traiter que les modifications dont la charge de travail est évaluée de quelques jours à quelques semaines. Les modifications dont la charge de travail est importante (de quelques semaines à quelques mois) sont considérées comme les activités d'un projet de maintenance assimilées à une évolution importante du logiciel :

- Seule la *petite maintenance* fait l'objet de notre étude : chaque modification est inférieure à un mois de travail
- Les modifications dont la charge est estimée à plus d'un mois de travail sont considérées comme des projets de maintenance et traitées comme un projet de développement logiciel

Concernant l'évaluation de la maturité du processus de maintenance, nous considérons que l'organisation étudiée n'étant pas consciente de l'existence de pratiques exemplaires et, n'ayant pas formalisé son processus de maintenance, ne peut pas avoir atteint, de fait, dans quelque domaine de processus que ce soit, le niveau de capacité 3 (où *la pratique exemplaire est comprise et effectuée conformément à une procédure organisationnelle documentée*) :

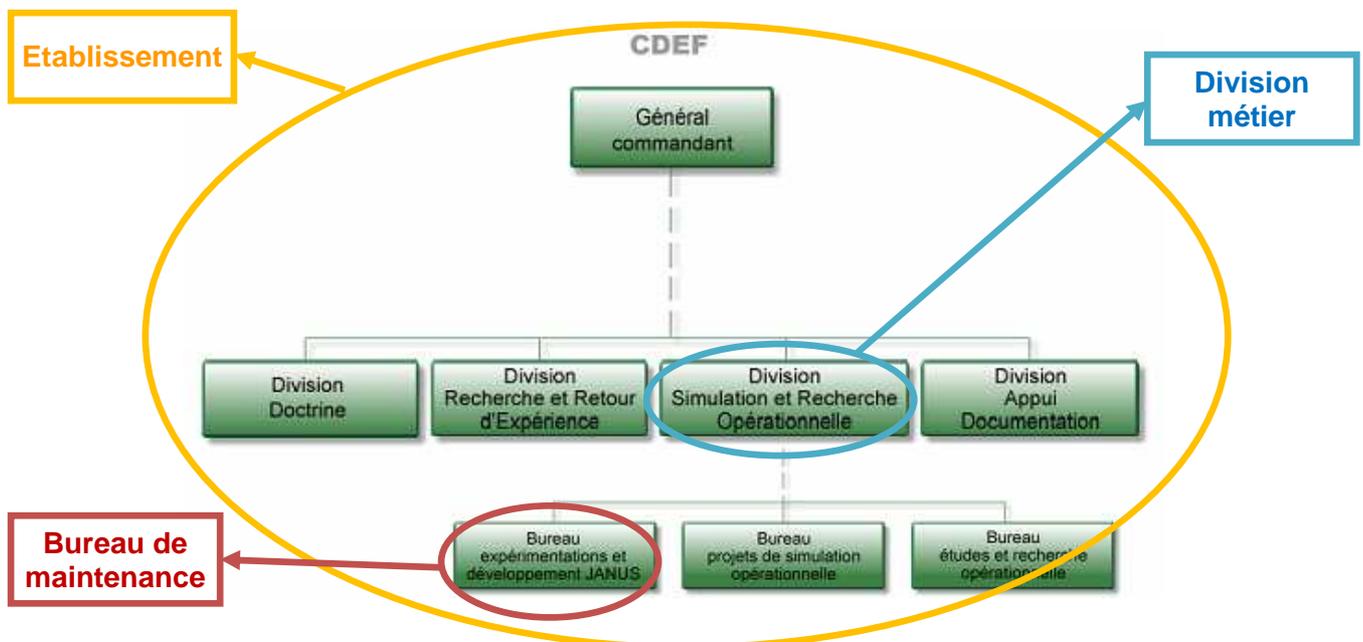
- Le niveau de maturité de l'organisation de la maintenance, objet de notre étude, ne peut pas avoir atteint le niveau 3
- Nous admettons que le niveau de capacité, atteint par chacun des itinéraires du modèle S3m est compris entre 0 (*inexistant*) et 2 (*Discipliné – Répétable mais intuitif*)

Notre étude se base sur ces hypothèses pour conduire une évaluation des niveaux de capacité 0, 1 et 2 de l'ensemble des itinéraires S^{3m}. De plus, nous considérons que l'objectif visé dans premier temps est d'atteindre le niveau 2, puis ensuite le niveau 3.

Enfin, par expérience, nous constatons qu'au sein des armées, les groupes chargés de maintenir un système informatique opérationnel sont souvent structurés autour d'un réseau indépendant et donc d'un système d'information indépendant du système d'information de l'établissement auquel ils sont rattachés. D'autre part, ces groupes opèrent au sein d'une division métier de l'établissement puisque les logiciels qu'ils soutiennent entrent dans le périmètre des produits et des services fournis.

En conséquence, il nous semble important d'envisager le changement organisationnel à deux niveaux :

- Nous considérons le bureau JANUS (bureau de maintenance du logiciel JANUS) comme une organisation à part entière avec son propre système d'information
- Nous considérons la DSRO qui héberge le bureau de maintenance comme une autre organisation soutenue par le système d'information de l'établissement



Nous rappelons ici que notre objectif est d'améliorer l'organisation de la maintenance et de la rendre plus visible à l'échelon supérieur sans remettre en cause la structure actuelle. Le changement s'opère bien aux deux niveaux organisationnels.

8. Le cadre de référence de la démarche

8.1. Présentation de la méthode EKD-CMM

Dans le cadre de la gestion du changement organisationnel, la méthode EKD-CMM (*Enterprise Knowledge Development - Change Management Method*) se propose de définir des formalismes permettant de représenter « le processus de changement, la vision du changement, l'impact sur l'organisation et/ou le(s) résultat(s) possible(s) du processus de changement » [CRI04].

Généralement, la gestion du changement se décline en 3 étapes [LEW58]:

- Définition du changement
- Mise en œuvre du changement
- Consolidation du changement

JACKSON est le premier à proposer un cadre de modélisation du processus de changement. Il définit une représentation des situations organisationnelles avant et après le changement :

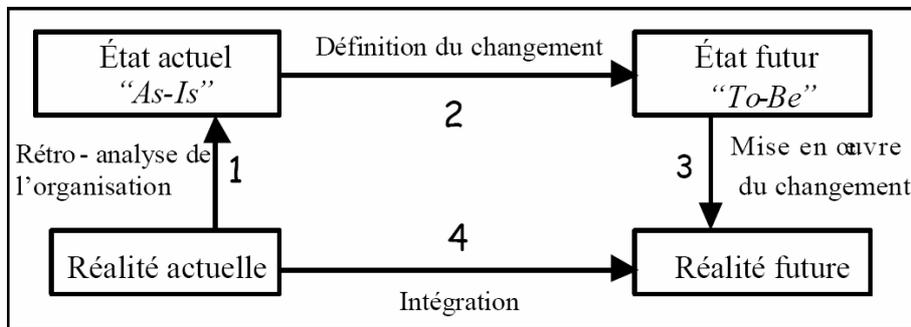


Figure 8.1 Le processus de changement organisationnel selon Jackson [NUR02]

La méthode EKD-CMM s'intéresse particulièrement à la phase la plus importante du processus : la phase de *définition du changement*. En s'appuyant sur les principes de JACKSON, elle s'attache à mettre en place un cadre de référence (un ensemble de modèles) pour représenter les états présents et futur de l'organisation ainsi que les scénarios alternatifs d'impact du changement :

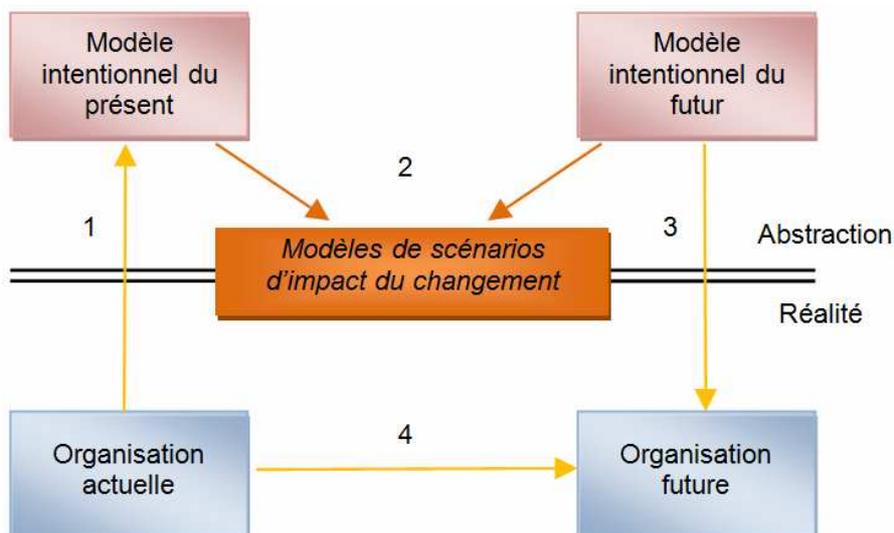


Figure 8.2 La vision du changement organisationnel dans EKD-CMM [NUR02]

Le changement s'opère du niveau intentionnel (les objectifs de l'entreprise) vers le niveau opérationnel (les processus métier de l'entreprise), ce dernier étant soutenu par le système d'information. Ce qui signifie qu'un changement qui intervient sur l'un des 3 niveaux aura un impact sur les deux autres.

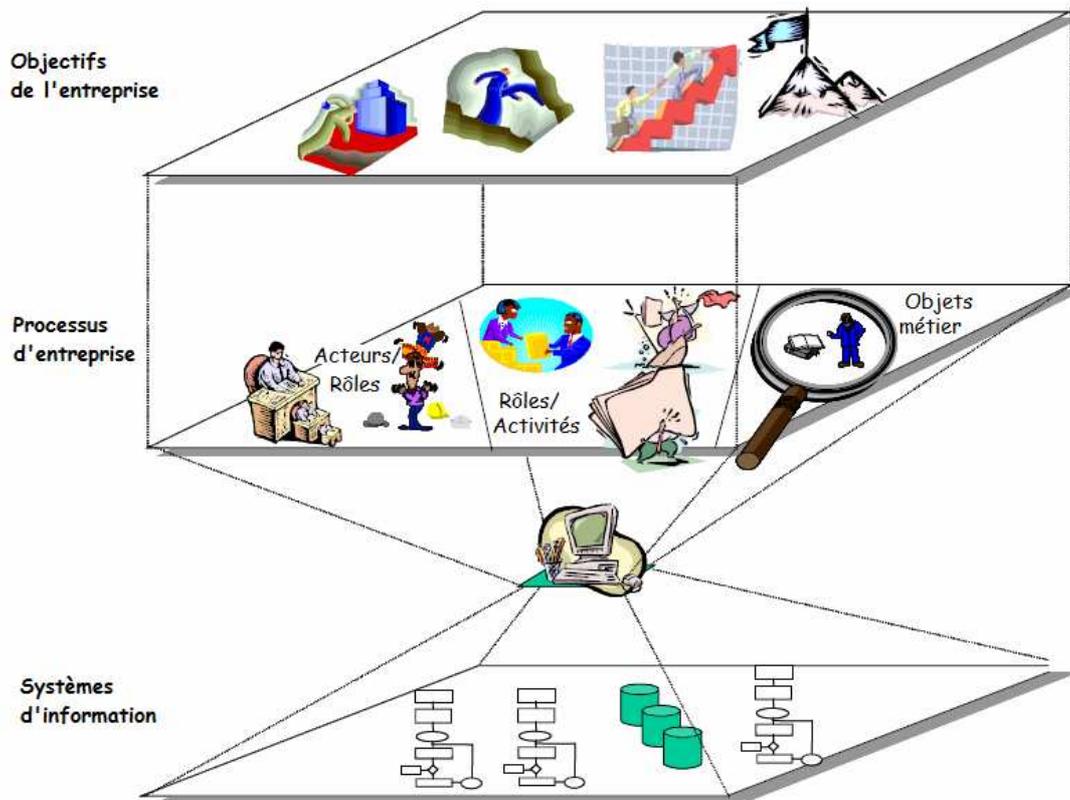


Figure 8.3 Représentation d'une organisation et de ses systèmes d'information [NUR02]

Pour obtenir une modélisation du changement, la réflexion repose à chaque niveau sur des modèles bien distincts :

- Les *modèles intentionnels* (qui décrivent ce que l'entreprise veut accomplir)
- Les *modèles d'entreprise* (qui décrivent les processus d'entreprise, les acteurs qui y prennent part, leurs rôles et les activités qui en découlent)
 - ✓ modèles acteurs/rôles
 - ✓ modèles rôles/activités
 - ✓ modèles d'objet
- Les *modèles de système d'information* (qui décrivent la structure et le fonctionnement des systèmes d'information)

La force de la méthode réside dans la capacité d'analyser et d'évaluer les différentes solutions alternatives pour une situation de changement organisationnel. Pour cela, elle se base sur deux autres modèles :

- Le *modèle des forces contextuelles* (qui décrit les facteurs externes et/ou internes qui se trouvent à l'origine du processus de changement)
- Le *modèle de scénarios d'impact* (qui indique les changements à apporter sur les objectifs organisationnels courants pour mettre en œuvre le changement envisagé et pour atteindre la vision du futur envisagé)

8.2. Proposition d'une approche personnalisée

« Ce que les gens disent aux analystes c'est ce qu'ils croient devoir faire, ce qu'ils parviennent à se rappeler ou ce qu'on leur a dit de répondre ; ils ne disent pas ce qu'ils font réellement [...] il vaut mieux les regarder faire ou mieux encore, le faire soi-même » [HAM03].

Nous utilisons le principe de l'observation, évoqué ici par HAMMER & CHAMPY, comme support de base à notre étude. Cependant, nous avons besoin de valider ces observations par des données objectives. C'est pourquoi nous décidons de débiter notre démarche par une évaluation de la maturité du processus de maintenance logicielle.

De plus, nous faisons généralement le constat que les organisations structurées autour de processus non formalisés, adoptent des pratiques dites « de bon sens » qui se rapprochent des pratiques de référence. L'évaluation des pratiques proposées par le modèle S3m doit donc nous permettre de mettre en évidence les pratiques qui ont été adoptées et celles qui ne l'ont pas été, celles qui sont formalisées et celles qui ne le sont pas.

La carte qui suit définit les intentions de notre démarche accompagnées des différentes stratégies possibles. Elle se base sur la notion de *contexte* : <situation ; intention> introduite par ROLLAND et GROSZ [ROL94]. Ainsi, nous nous plaçons dans une *situation de modélisation*, et nous devons prendre une décision correspondant à l'*intention* que nous voulons accomplir :

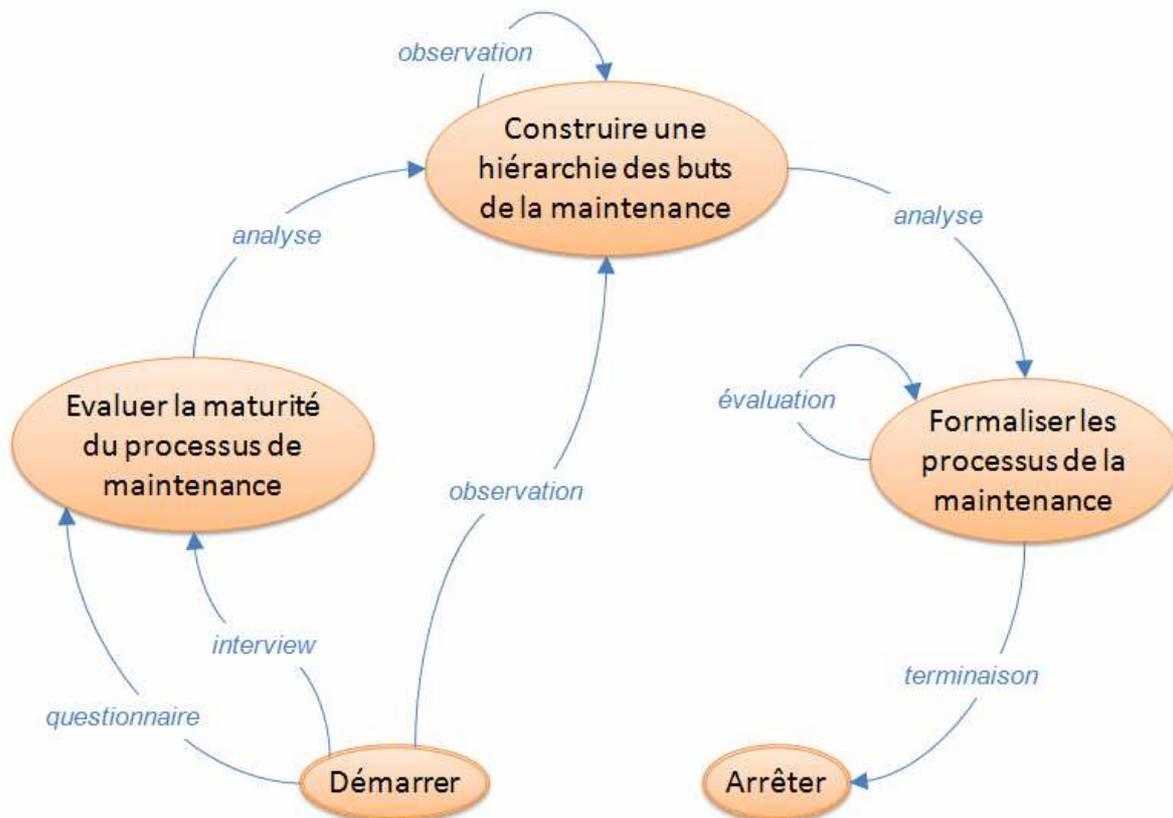


Figure 8.4 Cartographie de la démarche de niveau global

Comme nous l'avons expliqué précédemment, nous choisissons de conduire des interviews sur la base de la méthode S^{3m} Assessment plutôt que de baser la modélisation sur notre seule observation. D'autre part, la stratégie du questionnaire sans l'accompagnement de

l'auditeur est difficilement envisageable compte tenu de la complexité du modèle S^{3m} pour un non initié. L'auditeur a également la possibilité d'utiliser ce moment privilégié de l'interview pour recueillir des informations qui viennent conforter la compréhension des buts visés par l'organisation.

L'évaluation de la maturité du processus a un double intérêt : celui de fournir des données objectives pour modéliser les objectifs visés par l'organisation de la maintenance et celui de mettre en évidence les forces et les faiblesses de l'organisation. Ces éléments permettent d'orienter les améliorations du processus.

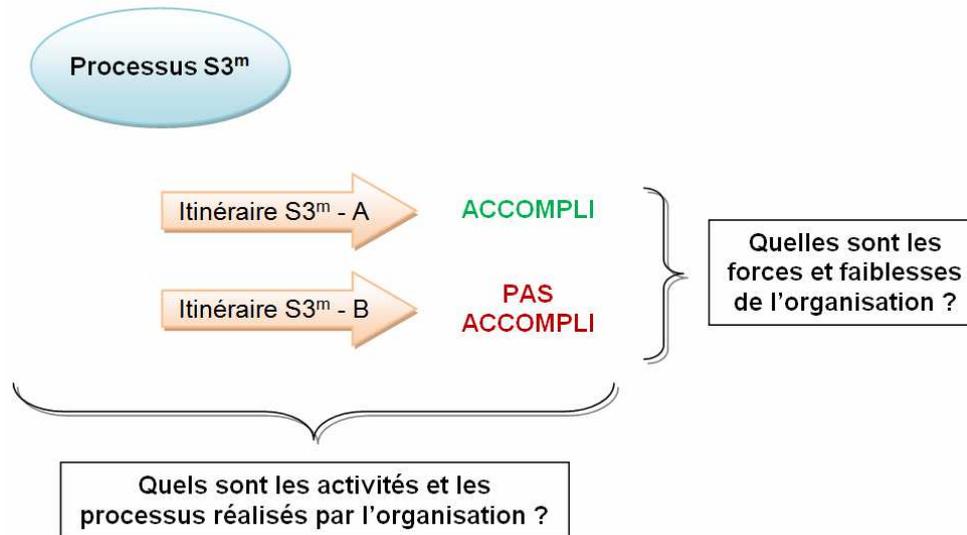


Figure 8.5 Les objectifs de l'évaluation de la maturité du processus de maintenance

L'analyse de l'évaluation nous aide à déterminer ce qui est réellement accompli par l'organisation. Elle donne des informations précieuses sur les buts de plus bas niveau de l'organisation (les buts *opérationnalisables*). Par ailleurs, les commentaires qui sont recueillis favorisent la compréhension à un niveau plus stratégique, l'observation venant appuyer cette compréhension.

Il est ainsi possible de construire une hiérarchie des buts recherchés par l'organisation, du niveau le plus stratégique au niveau le plus opérationnel. La représentation de cette hiérarchie des buts utilise le *modèle intentionnel* de la méthode EKD-CMM.

Lorsque la hiérarchie des buts est construite, le niveau *opérationnalisable* (le plus bas) nous permet d'en déduire les micro-processus de l'organisation de la maintenance. Il est alors possible de formaliser le processus global de la maintenance du logiciel sous la forme d'une cartographie des processus comme le montre la figure 8.6 de la page suivante.

Pour finir, nous pouvons comparer la cartographie des processus actuelle (« As-Is ») à la cartographie des processus du modèle S^{3m} vers lequel nous souhaitons tendre (« To-Be »). Cette comparaison nous permet d'identifier les axes d'amélioration possible de notre processus de maintenance comme le montre la figure 8.7.

Enfin, l'identification des processus à améliorer et l'identification des forces et faiblesses de l'organisation devraient nous permettre de modéliser les *forces contextuelles* et les *scénarios d'impact* selon la méthode EKD-CMM. Cependant, le périmètre de notre étude s'arrêtera simplement à l'identification des changements sans en modéliser complètement le processus.

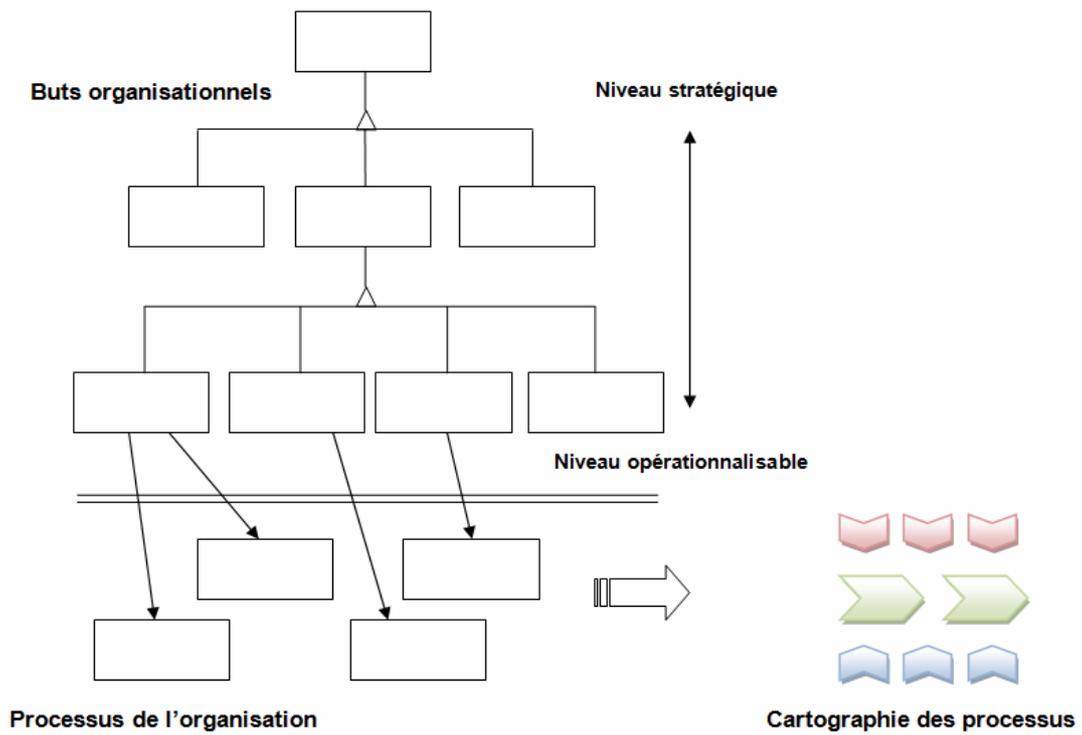


Figure 8.6 Construction de la hiérarchie des buts et formalisation des processus

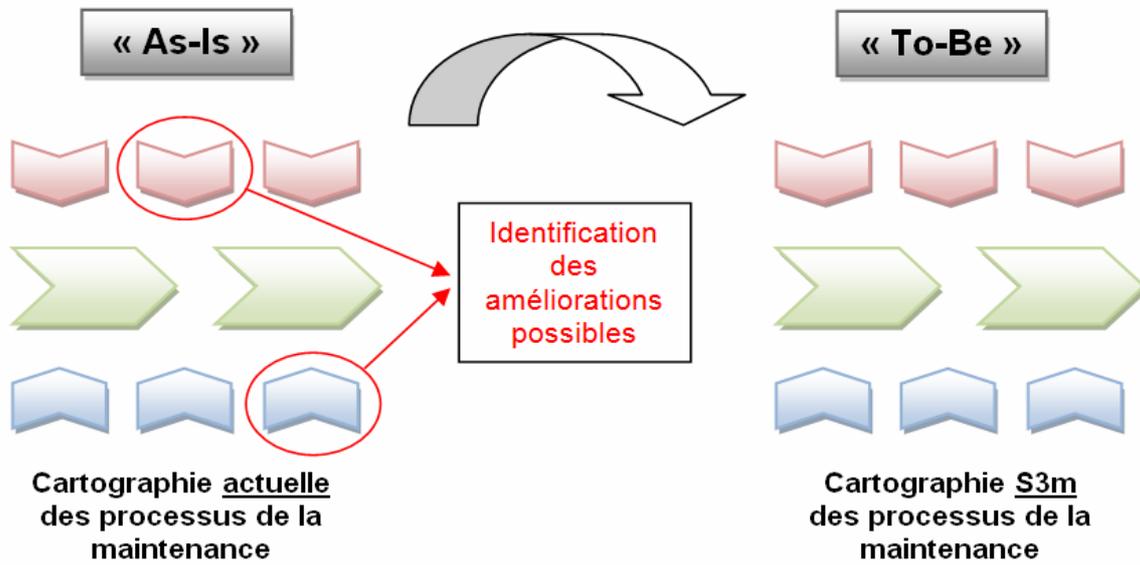


Figure 8.7 Comparaison des processus actuels de la maintenance et ceux du modèle S^{3m}

9. Les étapes de la démarche

L'investigation sur le terrain et le travail sur le processus de maintenance se déroulent en trois phases :

- 1^{ère} phase : Recueil, traitement et analyse des données (observation et interview)
- 2^{ème} phase : Formalisation du processus de maintenance
- 3^{ème} phase : Définition du changement organisationnel

Phase 1	Audit interne à partir du questionnaire d'évaluation de la maturité
	Traitement et analyse des résultats obtenus
Phase 2	Modélisation du processus de maintenance actuel
	Analyse des axes d'amélioration possible
Phase 3	Définition du changement organisationnel de la maintenance
	Définition du changement organisationnel global

Tableau 9.1 Les étapes de l'investigation de terrain

Les méthodes utilisées pour ce travail d'investigation peuvent se résumer ainsi :

- *Recueil, traitement et analyse de données*

Observation et audit interne sur la base d'un questionnaire élaboré à partir du modèle de maturité S^{3m}. Traitement des données à partir de formulaires Excell.

- *Modélisation du processus*

Les principes du modèle EKD-CMM (modèle intentionnel du présent, diagramme Acteurs-Rôles, modèle intentionnel du futur).

- *Définition du changement*

Bilan de la modélisation selon le cadre de référence fourni par le modèle EKD-CMM et définition du changement aux différents niveaux de l'organisation.

9.1. Le recueil, le traitement et l'analyse des données

Le recueil des données se base sur deux actions complémentaires que sont l'observation et l'interview.

Pour conduire une observation, il est nécessaire de définir une grille d'observation. Celle-ci est définie de manière classique pour répondre aux questions :

- QUI (qui fait ? quels acteurs ?)
- QUOI (quels sont les rôles tenus ? quelles sont les activités exécutées ?)
- OU (quel est le contexte spatial ? de l'enquête, des activités ?)
- QUAND (quel est le contexte temporel ? de l'enquête, des activités ?)
- COMMENT (quels outils, méthodes, procédures, documents sont utilisés ?)
- POURQUOI (pourquoi ce qui est fait, est-il fait ? pourquoi est-ce fait comme cela ?)

Pour conduire l'interview, nous nous appuyons sur la méthodologie S3m Assessment et prenons la posture de l'auditeur. L'évaluation que nous conduisons est la micro-évaluation,

définie par les étapes suivantes :

- *Choix des évaluateurs, rencontres et plan d'évaluation*
- *Questionnaire, entrevues et revues*
- *Constats et représentation du profil par secteur clé*

Une réunion de présentation permet de définir le plan d'évaluation (qui, où et quand). Chaque personnel est interviewé sur le lieu de son activité quotidienne pour une durée fixée à 2 heures.

Ainsi, les observations sont validées par les interviews réalisées auprès de l'ensemble du service de la maintenance (10 personnes) et deux personnes extérieures au service mais impliquées dans la conduite du processus « *gérer les projets* » propre à l'ensemble de la division.

Event/request management	Event/request management	1.0.1	Non	0	L'unité organisationnelle de la maintenance du logiciel ne fait pas d'activités de gestion des requêtes et des événements sur les différents logiciels en maintenance
	Maintenance planning	2.0.1	Non	20	L'unité organisationnelle de la maintenance du logiciel ne fait pas d'activités de planification
	Requests/software monitoring and control	3.0.1	Non	10	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.
	SLA and supplier agreements management	4.0.1	Oui	70	L'unité organisationnelle de la maintenance du logiciel n'a pas encore reconnu le besoin d'une entente de service et de contrats formels.
				25	
Evolution Engineering	Pre-delivery and transition services	1.0.1	Oui	70	L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de transition du logiciel vers la maintenance
	Operational support services	2.0.1	Non	0	L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de support opérationnel.
	Software evolution and correction services	3.0.1	Non	0	L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de correction et d'évolution du logiciel
	Verification and validation	4.0.1	Non	0	L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de vérification et de validation du logiciel.
				18	

Figure 9.1 Exemple de questions d'évaluation du processus de maintenance – niveau 0 [PAQ06]

Le questionnaire repose sur un outil d'évaluation de la maturité du processus de maintenance du modèle S^{3m} mis au point par David-Alexandre PAQUETTE [PAQ06]. Les niveaux évalués, seront les niveaux : 0, 1 et 2. Nous nous appuyons sur ce questionnaire pour vérifier et valider ce que le service de la maintenance fait ou ne fait pas et jusqu'à quel point il le fait ou ne le fait pas. Il est donc aisé de déterminer les forces et les faiblesses de l'organisation.

9.2. La formalisation du processus

Le travail de formalisation consiste, à partir des informations recueillies sur le terrain, à mettre en évidence les processus et les activités de la maintenance ainsi que les rôles tenus par les acteurs du processus.

Dans un premier temps il est indispensable de définir les objectifs visés par le bureau chargé de la maintenance du logiciel JANUS afin d'identifier les micro-processus mis en place dans l'organisation de façon informelle.

Il s'agit dans un deuxième temps de faire l'inventaire des acteurs en présence et des rôles qu'ils assument dans le cycle de la maintenance du logiciel.

Dans le cas qui nous intéresse, nous utilisons les *modèles intentionnels* et les *modèles d'entreprise (diagrammes Rôles-Activités, cartographie des processus)* de la méthode EKD-CMM pour représenter l'état actuel (« *As-Is* ») et l'état futur (« *To-Be* ») de notre organisation de la maintenance du logiciel.

9.3. La définition du changement organisationnel

Cette étape doit faciliter la mise en évidence de l'impact du changement au niveau de l'organisation de la maintenance (dans notre cas, le bureau « JANUS ») mais également au niveau de la structure à laquelle est rattaché le bureau (la Division Simulation et Recherche Opérationnelle). Cette division étant organisée autour d'un unique processus de gestion de projet, nous devons répondre à la question : comment rendre visible le processus de maintenance dans cette organisation orientée « projet » ?

Cette analyse commence par un bilan, nécessaire, de l'étape précédente qui consiste, comme nous venons de le voir ci-dessus, à modéliser l'état actuel de notre organisation de la maintenance du logiciel (le processus de maintenance actuel) et son état futur (le processus de maintenance proposé par le référentiel S^{3m}). Ce bilan doit montrer les améliorations à conduire pour augmenter la maturité du processus de maintenance. Autrement dit, on évalue l'impact de l'évolution de la maturité du processus de maintenance sur l'organisation du bureau « JANUS ».

Enfin, le bureau « JANUS » faisant parti d'une structure plus vaste et composée de 3 bureaux organisés autour d'un unique processus de gestion de projet, il est indispensable de placer notre travail dans une double perspective: impact sur l'organisation de la maintenance et sur la structure dont elle dépend.

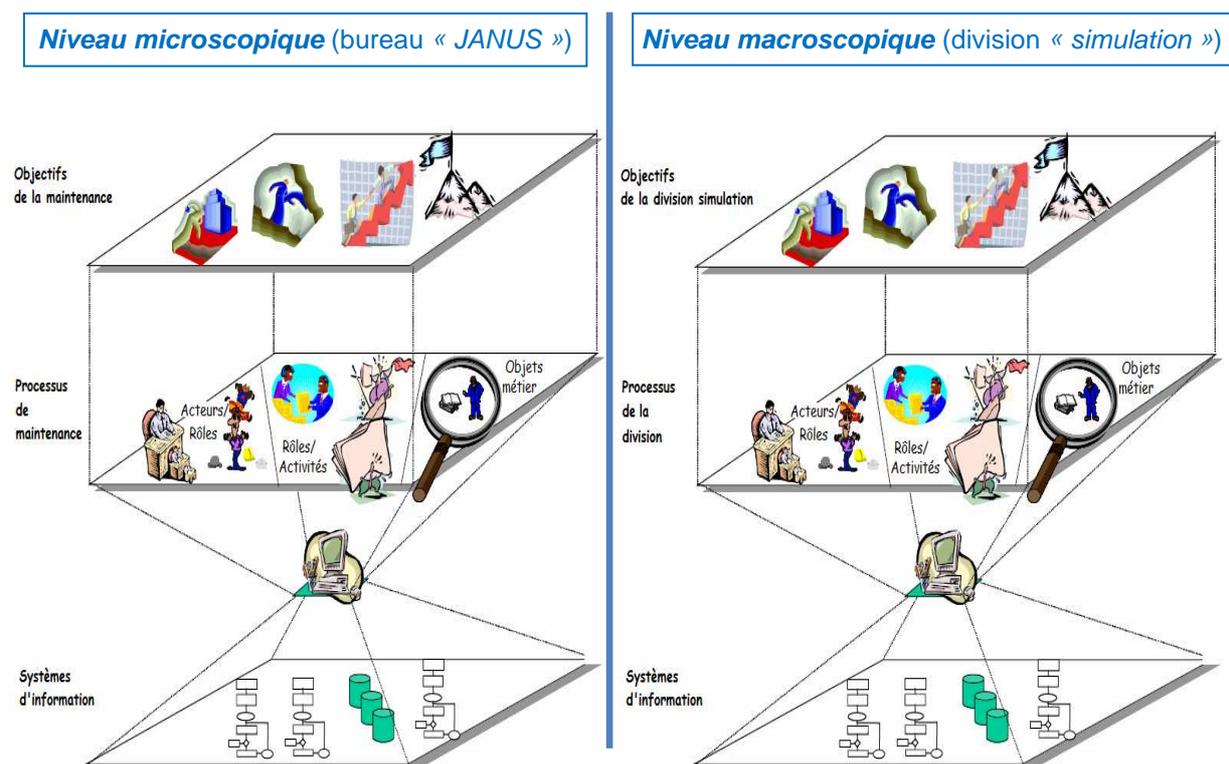


Figure 9.2 Les deux niveaux du changement organisationnel

CHAPITRE IV

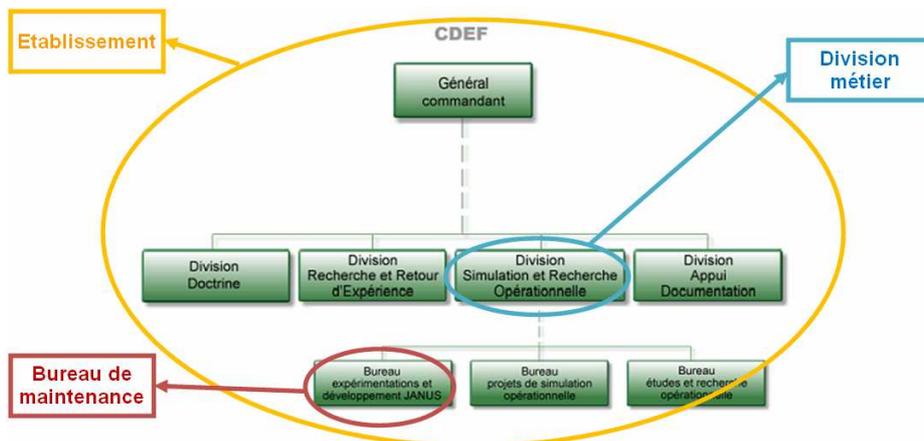
ETUDE DE CAS

Après avoir exposé notre démarche, dans le chapitre précédent, nous débutons notre étude de cas afin d'appréhender les changements organisationnels induit par l'évolution de la maturité du processus de maintenance. Placés en contexte professionnel, nous nous appuyons sur le cadre de référence de la méthode EKD-CMM pour accomplir cette étude.

Ainsi, dans la première partie, nous utilisons les outils de la méthode S3m Assessment pour conduire une évaluation de la maturité de notre processus de maintenance actuel.

Cette évaluation objective nous aide à formaliser le processus de maintenance et les activités qui en découlent, utilisant en partie les modèles d'EKD-CMM. C'est l'objet de notre seconde partie.

La dernière partie aborde la préparation du changement organisationnel à deux niveaux, celui de l'organisation de la maintenance et celui de la structure à laquelle elle se rattache qui est basée sur une approche de type *projet*.



10. Evaluation de la maturité du processus de maintenance

10.1. Application de la méthode S^{3m} Assessment

L'évaluation porte sur le niveau d'aptitude, atteint par notre bureau de maintenance, pour chacun des itinéraires définis par le modèle de maturité S^{3m}:

- ✓ Focaliser sur les processus (*Gestion du Processus*)
- ✓ Définition des processus et des services (*GP*)
- ✓ Formation organisationnelle des ressources (*GP*)
- ✓ Performance des processus (*GP*)
- ✓ Innovation et déploiement (*GP*)
- ✓ Gestion des requêtes de services et des événements (*Gestion des Requêtes*)
- ✓ Planification de la maintenance (*GR*)
- ✓ Suivi et supervision des requêtes de la maintenance (*GR*)
- ✓ Gestion d'ententes de services et de la sous-traitance (*GR*)
- ✓ Coordination avant livraison et transition du logiciel (*Ingénierie d'Evolution*)
- ✓ Service de support opérationnel (*IE*)
- ✓ Services d'évolution et de correction du logiciel (*IE*)
- ✓ Vérification et validation (*IE*)
- ✓ Gestion de versions et de la configuration (*Support à l'Ingénierie d'Evolution*)
- ✓ Assurance qualité des services, des processus et des produits (*SIE*)
- ✓ Mesure et analyse de la maintenance (*SIE*)
- ✓ Rajeunissement, retraite et migration du logiciel (*SIE*)

A partir de nos connaissances préexistantes du fonctionnement du bureau, et compte tenu que le processus de maintenance n'y est pas formalisé en tant que tel, nous posons l'hypothèse suivante :

- Les niveaux d'aptitude de notre bureau de maintenance, pour chacun des itinéraires S^{3m}, ne peuvent atteindre le niveau 3 (*Personnalisé et orienté processus*)

Pour réaliser cette évaluation nous nous appuyons sur un questionnaire construit autour des itinéraires des niveaux 0, 1 et 2 :

- **Niveau 0** : le processus est incomplet. Certaines pratiques ne sont pas réalisées.
- **Niveau 1** : le processus est exécuté mais plutôt de manière informelle.
- **Niveau 2** : le processus est discipliné (existence de documents d'organisation) et orienté requête.

Les questions, relatives à un itinéraire, sont de plus en plus précises au fur et à mesure que le niveau augmente. Le niveau d'aptitude est évalué en pourcentage d'accomplissement de la pratique et s'appuie sur les caractéristiques de la norme ISO 15504 [ISO98a]:

- N : *not reached* = 0 – 15%
- P : *partially reached* = >15 – 50%
- L : *largely reached* = >50 – 85%
- F : *fully reached* = >85 – 100%

Dans notre cas, pour simplifier les calculs nous utilisons une moyenne correspondant à la réponse donnée :

- « *Non accompli* » = 0%
- « *Partiellement accompli* » = 33%
- « *Largement accompli* » = 68%
- « *Totalement accompli* » = 93%

Voici 3 exemples pour un même itinéraire selon chacun des niveaux :

Niveau 0

Event/request management	Event/request management	1.0.1	Non	0	L'unité organisationnelle de la maintenance du logiciel ne fait pas d'activités de gestion des requêtes et des événements sur les différents logiciels en maintenance
--------------------------	--------------------------	-------	-----	---	---

Niveau 1

Event/request management	Event/request management	1.1.1	N: Not Achieved	0	La gestion des requêtes et des événements, dans l'organisation de maintenance du logiciel, se fait de manière informelle
		1.1.2	N: Not Achieved	0	Une approche individuelle de gestion des requêtes et des événements est principalement basée sur des relations personnelles entre le programmeur de la maintenance et une ressource de l'unité organisationnelle du client

Niveau 2

Event/request management	Event/request management	1.2.1	F: Fully Achieved	93	Il y a un point de contact unique pour fournir une aide directe aux clients
		1.2.2	F: Fully Achieved	93	Chaque requête et événement 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.
		1.2.3	F: Fully Achieved	93	Chaque requête et événement font l'objet de catégorisation, de choix, de priorités et d'une estimation préliminaire de leur taille et de leur ampleur
		1.2.4	L: Largely Achieved	68	Les requêtes de modifications acceptés sont assignées. D'une manière préliminaire, à une version future du logiciel

Figure 10.1 Exemple d'un itinéraire S3M décliné aux niveaux 0, 1 et 2

Nota :

Le niveau 0 est légèrement différent des autres niveaux. En effet, à ce stade initial, on cherche à déterminer les itinéraires qui sont ou ne sont pas accomplis, dans une perspective globale. De ce fait, les réponses possibles ne peuvent être que oui ou non. Comme le montre la figure 10.1, au niveau 0 les questions sont posées à la forme négative pour vérifier que les pratiques ou les processus ne sont pas reconnus dans l'organisation. Un *oui* sera donc côté 100% (la pratique n'est effectivement pas accomplie) et un *non* à 0% (la pratique est accomplie même partiellement) :

- *Oui* = 100% (la pratique n'est pas faite → « oui » = 100%)
- *Non* = 0% (la pratique n'est pas faite → « non » = 0%)

Autrement dit, si aucune pratique n'est accomplie (« oui » à toutes les réponses) la moyenne du niveau 0 sera à 100%. On dit alors que la capacité de l'organisation de la maintenance est 100% de niveau 0 dans tous les domaines de processus (l'organisation n'est pas consciente de ses processus ni des pratiques en matière de maintenance du logiciel).

Ainsi, peut être calculé le pourcentage moyen d'accomplissement des itinéraires d'un domaine de processus. Ce calcul est fait pour chacun des niveaux 0, 1 et 2 et donne la tendance du niveau d'aptitude pour chacun des domaines. Il donne également une tendance générale de la maintenance (tendance niveau 0 ou 1 ou 2 en fonction des pourcentages moyens obtenus).

La figure 10.2 de la page suivante montre un exemple des moyennes calculées pour chacun des domaines de processus considérés.

Exemple : le cas du domaine de processus « Ingénierie d'évolution » (Evolution Engineering)

Niveau 0

Evolution	Pre-delivery and transition services	1.0.1	Oui	70	L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités
Engineering	Operational support services	2.0.1	Non	0	L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités
Engineering	Software evolution and correction services	3.0.1	Non	0	L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités
Engineering	Verification and validation	4.0.1	Non	0	L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités
				18	

Niveau 1

Evolution	Pre-delivery and transition services	1.1.1	L: Largely Achieved	68	La transition dans l'unité organisationnelle de maintenance du logiciel, est
Engineering	Operational support services	2.1.1	F: Fully Achieved	93	Le support opérationnel, dans l'unité organisationnelle de maintenance du
Engineering	Software evolution and correction services	3.1.1	N: Not Achieved	0	L'unité organisationnelle de maintenance du logiciel ne fait pas d'activités de
Engineering	Verification and validation	4.1.1	N: Not Achieved	0	L'unité organisationnelle de
				40	

Pourcentage moyen de l'aptitude du domaine de processus à ce niveau

Niveau 2

Evolution Engineering	Pre-delivery and transition services	1.2.1	L: Largely Achieved	68	L'unité organisationnelle de
		1.2.2	L: Largely Achieved	68	L'unité organisationnelle de la maintenance s'assure de sélectionner avec
		1.2.3	N: Not Achieved	0	L'unité organisationnelle de la maintenance communique avec l'unité
		1.2.4	P: Partially Achieved	33	L'unité organisationnelle de la maintenance communique avec les clients pour
		1.2.5	N: Not Achieved	0	L'unité organisationnelle de la maintenance développe, adapte et utilise une
		1.2.6	L: Largely Achieved	68	Les ressources de la maintenance du logiciel qui prendront en charge le
		1.2.7	L: Largely Achieved	68	L'entité organisationnelle de la maintenance s'assure de l'efficacité de la
		1.2.8	P: Partially Achieved	33	L'entité organisationnelle de la maintenance évalue l'efficacité de la formation
		1.2.9	P: Partially Achieved	33	Les ressources de la maintenance du logiciel qui prendront en charge le
		1.2.10	L: Largely Achieved	68	Les ressources de la maintenance du logiciel qui prendront en charge le
		1.2.11	L: Largely Achieved	68	Les ressources de la maintenance du logiciel qui prendront en charge le
	Operational support services	2.2.1	L: Largely Achieved	68	Le calendrier d'opération existe et est élaborée sur la base de la capacité des
		2.2.2	N: Not Achieved	0	Le mainteneur et l'unité organisationnelle de la maintenance connaissent
		2.2.3	N: Not Achieved	0	Le mainteneur produit des rapports de surveillance des caractéristiques des
		2.2.4	N: Not Achieved	0	L'horaire de support est créé et publié, et on s'assure de la couverture du
		2.2.5	P: Partially Achieved	33	Le mainteneur possède l'expertise de la fonctionnalité et des règles d'affaires
		2.2.6	L: Largely Achieved	68	Le mainteneur produit des rapports ad hoc sur requête
	Software evolution and correction services	3.2.1	N: Not Achieved	0	La conception détaillée élabore les solutions aux exigences de l'utilisateur et
		3.2.2	N: Not Achieved	0	Les activités d'investigation et de conception d'une défaillance identifiées sur
		3.2.3	N: Not Achieved	0	Les activités de mise en œuvre (programmation) sont effectuées
		3.2.4	L: Largely Achieved	68	Le partage des rôles et des responsabilités en matière de données est
		3.2.5	L: Largely Achieved	68	Les langages de programmation originaux du logiciel sont utilisés pour les
		3.2.6	N: Not Achieved	0	Les interrelations (tracabilité) entre les procédures administratives, les
		3.2.7	L: Largely Achieved	68	Les essais unitaires doivent être exécutés pour chaque composant qui fait
		3.2.8	L: Largely Achieved	68	Les essais d'intégration doivent être effectués pour chaque composant qui
		3.2.9	P: Partially Achieved	33	La documentation de l'utilisateur est soumise à l'essai par le mainteneur
		3.2.10	N: Not Achieved	0	La documentation interne du logiciel est élaborée conformément à une
		3.2.11	N: Not Achieved	0	La documentation externe du logiciel est élaborée conformément à une
Verification and validation	4.2.1	N: Not Achieved	0	Les activités de vérification et de validation sont documentés, connus et	
	4.2.2	P: Partially Achieved	33	On procède à des révisions des activités des fournisseurs et des sous-	
	4.2.3	N: Not Achieved	0	Un horaire à des révisions des activités des fournisseurs et des sous-	
	4.2.4	N: Not Achieved	0	Un essai des systèmes est effectué conformément à une procédure	
	4.2.5	N: Not Achieved	0	Des essais de régression sont systématiquement et formellement effectués	
	4.2.6	P: Partially Achieved	33	La coordination de l'installation, de l'essai sur place et de la maintenance du	
	4.2.7	P: Partially Achieved	33	Les pratiques relatives à l'installation des modifications de logiciel (y compris	
	4.2.8	N: Not Achieved	0	Le mainteneur effectue des essais d'acceptation à la livraison du produit du	
				30	

Figure 10.2 Exemple de degré d'accomplissement d'un domaine de processus S3M

Dans l'exemple ci-dessus, la tendance d'aptitude du domaine « Ingénierie d'évolution » peut être évaluée de la façon suivante :

- niveau 0 : 18% niveau 1 : 40% niveau 2 : 30%

A ce stade, nous pouvons en déduire que la tendance de notre aptitude à accomplir les itinéraires du domaine de processus « ingénierie d'évolution » est plutôt de niveau 1. D'autre part, puisque notre processus de maintenance n'est pas encore formalisé, aucun des domaines de processus ne peut atteindre le niveau 3 même partiellement. Ainsi, nous pouvons vérifier que la somme des pourcentages obtenus avoisine les 100% :

- Dans notre exemple, 18 + 40 + 30 = 88 %

Ce résultat peut être considéré comme acceptable ou donner lieu à des ajustements en s'appuyant sur les arguments notés au moment de l'interview. Tous les résultats sont validés ou ajustés de cette manière.

10.2. Présentation des résultats

Finalement, 10 personnes ont été interviewées (9, internes au bureau de maintenance et une externe au bureau). Le tableau suivant, présente les résultats de l'enquête (audit interne) en pourcentage d'accomplissement global du processus de maintenance et réparti sur les 3 niveaux d'aptitude :

	Niveau 0	Niveau 1	Niveau 2	
BTR	46,2	40,74	19,860	106,798
QMS	41,0	41,05	20,233	102,283
ZLT	48,9	31,19	19,910	100,000
RCR	47,8	34,27	29,981	112,049
GLP	29,6	38,92	31,143	99,663
OLO	46,1	39,38	22,327	107,808
GNT-RCE	23,1	44,15	33,511	100,760
LBN-BZA	25,8	29,84	38,518	94,154
DMA-CRR	41,0	42,48	21,455	104,937
MRI	21,8	54,11	35,267	111,177
%	37,13	39,6124	27,22048	103,96288

La somme des pourcentages en ligne doit avoisiner les 100%

Pourcentage global pour un niveau d'aptitude donné

Tableau 10.1 Résultats de l'audit interne sur la maturité du processus de maintenance

Nous déduisons de ces résultats que le niveau d'aptitude de notre processus de maintenance atteint, selon les tendances :

- Niveau 0 ~ 35% niveau 1 ~ 40% niveau 2 ~ 25%

Cela signifie que :

- Le processus est réalisé en grande partie de manière informelle (niveau 1)
- Certaines pratiques exemplaires ne sont pas du tout ou très peu réalisées (niveau 0)
- Un petit nombre de pratiques sont structurées et documentées (niveau 2)

Il est alors intéressant de rechercher les axes d'amélioration possible en fonction de chaque itinéraire des 4 domaines de processus de la maintenance (voir la figure 10.3 de la page suivante).

Pour représenter le niveau atteint par chacun des itinéraires nous adoptons la convention ci-après. Deux couleurs utilisées pour un même itinéraire signifie que nous estimons qu'il se situe entre deux niveaux (par exemple lorsqu'une partie seulement des pratiques d'un itinéraire est accomplie) :

- L'itinéraire n'est pas ou peu réalisé (**niveau 0**)
- L'itinéraire est réalisé de manière informelle (**niveau 1**)
- L'itinéraire est réalisé de façon structurée et documentée (**niveau 2**)

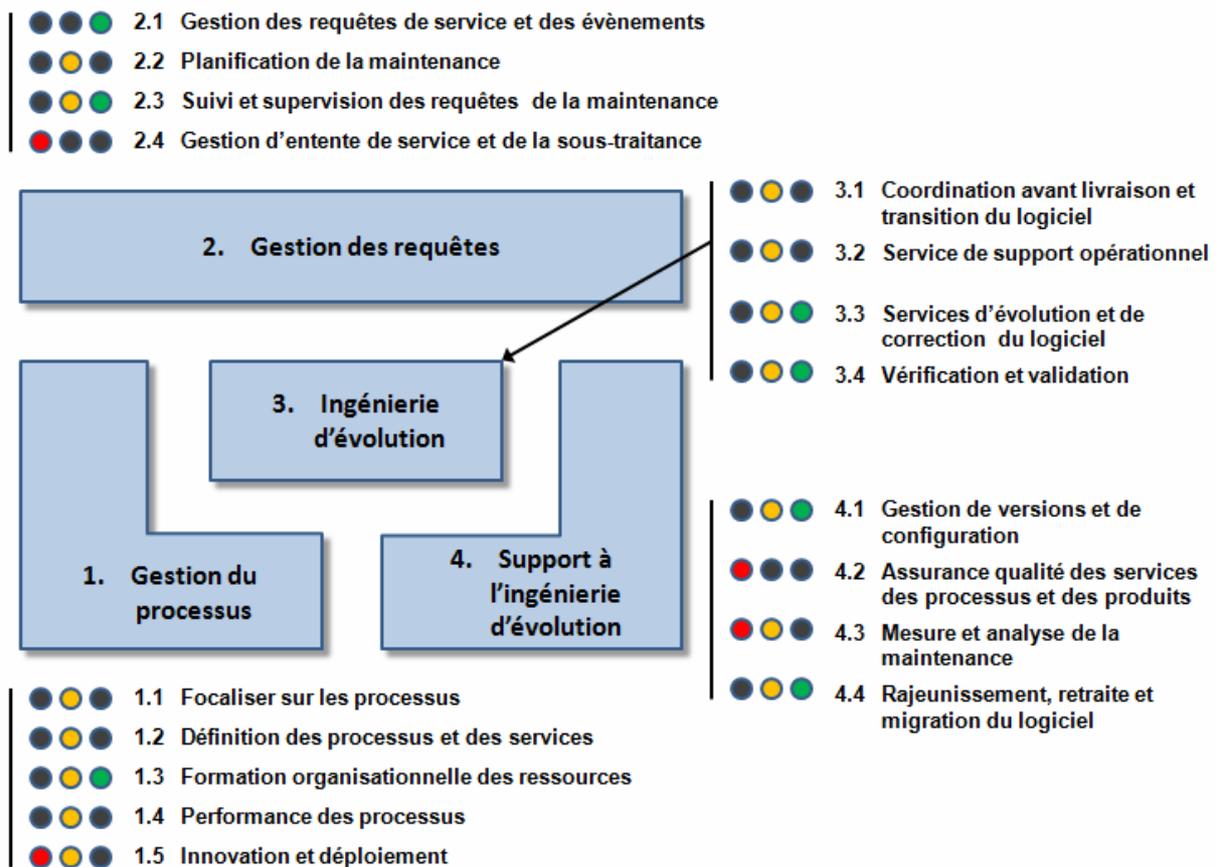


Figure 10.3 Evaluation du niveau d'aptitude du processus de maintenance de notre organisation

Nous déduisons de cette représentation que certains itinéraires sont peu ou pas pratiqués (les éléments notés en « rouge »). Ils constituent donc des axes d'amélioration certains et concernent les pratiques liées à :

- L' « innovation et déploiement »
- La « gestion d'entente de service »
- L' « assurance qualité des services des processus et des produits »
- La « mesure et l'analyse de la maintenance »

Les éléments qui apparaissent en « jaune » sont également perfectibles. Il sera alors intéressant de définir quelles améliorations sont estimées les plus importantes. Par exemple, on préférera peut-être améliorer en priorité :

- La « définition des processus et des services »
- La « coordination avant livraison et transition du logiciel »
- La « mesure et l'analyse de la maintenance »

Les itinéraires apparaissant en « vert » peuvent être assimilés aux points forts de l'organisation.

Enfin, si l'on adopte une approche plus globale, le bureau de la maintenance « souffre » du fait que le processus de maintenance n'est pas encore formalisé et reconnu au sein de l'entité.

11. Formalisation du processus de maintenance

La formalisation d'un processus est une action de modélisation de la connaissance d'entreprise. Comme nous l'avons expliqué au chapitre précédent, nous nous appuyons sur les principes de la méthode EKD-CMM. Cette méthode sera appliquée à notre organisation de la maintenance du logiciel.

Que cherchons-nous à faire ici ? Quel est notre objectif ?

Notre démarche consiste à modéliser notre organisation actuelle (le processus de maintenance applicative) en vue de l'améliorer (augmenter le niveau de maturité du processus) et de tendre vers une nouvelle organisation :

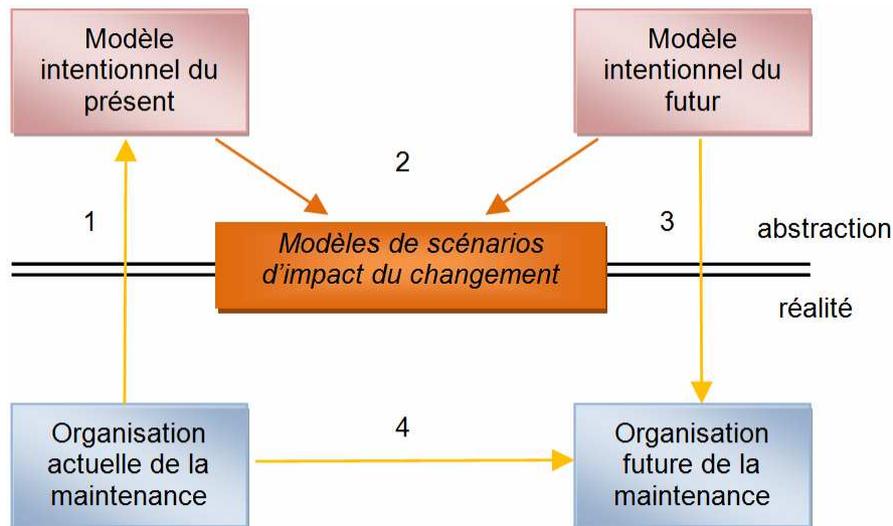


Figure 11.1 La vision du changement organisationnel de la maintenance

Dans cette démarche de changement organisationnel, nous cherchons à définir dans un 1^{er} temps le modèle intentionnel du présent puis ensuite une cartographie des processus de la maintenance. Le modèle intentionnel du futur nous est donné par les pratiques exemplaires définies dans le modèle S3M.

11.1. Modèle intentionnel du présent

Le modèle intentionnel du présent définit les objectifs visés par l'organisation. Cette modélisation se présente sous la forme d'un arbre où chaque branche correspond à un but. Un but de niveau supérieur peut donner naissance à des buts de niveau inférieur qui sont connectés entre eux par un lien de composition (connecteur de type « **ET** »).

Ici, la modélisation de l'organisation de la maintenance du logiciel donne naissance à un arbre des buts composé de liens de type « **ET** » (représenté graphiquement par un \triangle).

Il est également important de noter que les feuilles de l'arbre apparaissant de manière ombrées représentent les buts dit « *opérationnalisables* ». On entend par buts *opérationnalisables* les objectifs qui peuvent directement se traduire par des micro-processus d'entreprise [BAR01].

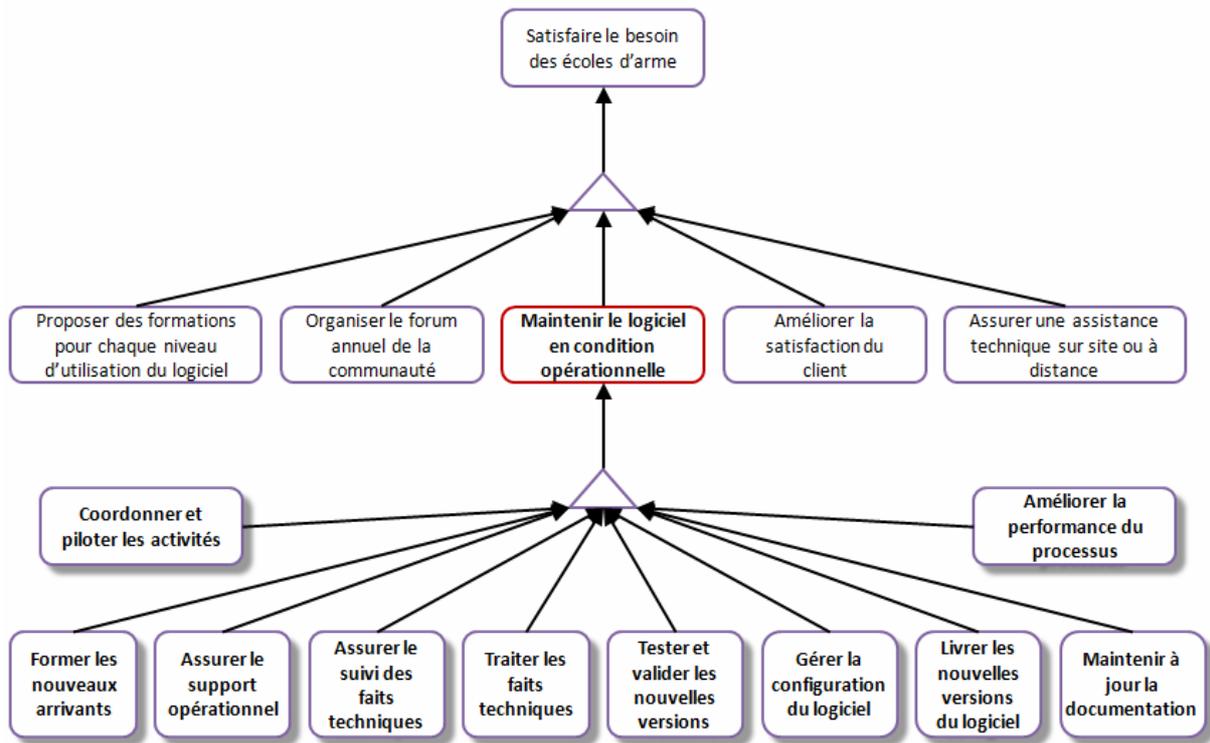


Figure 11.2 Modèle intentionnel de l'organisation actuelle de la maintenance

A partir de cette modélisation mettant en évidence les processus de la maintenance du logiciel, nous pouvons en déduire une représentation conventionnelle de la cartographie des processus :

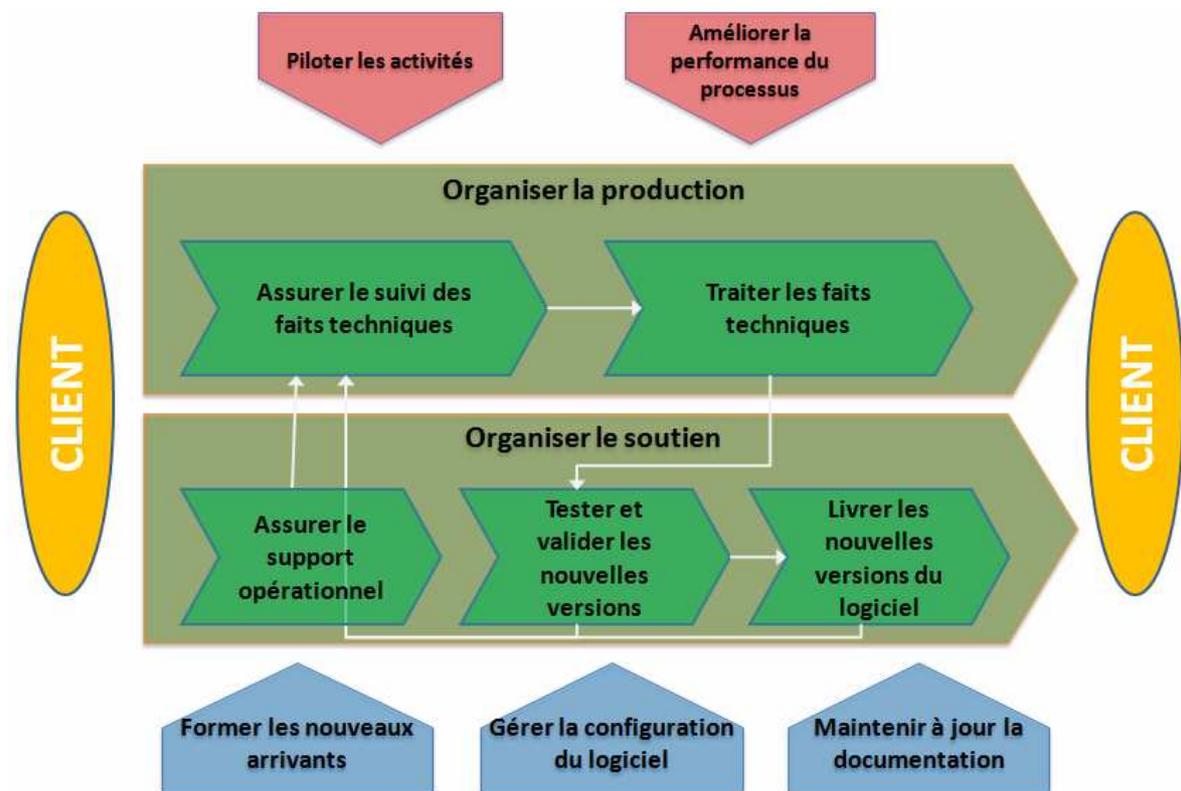


Figure 11.3 Cartographie des processus de l'organisation actuelle de la maintenance

Cette cartographie se réfère à l'approche processus (vu dans la partie 4.3 du chapitre II) pour représenter les processus de « *pilotage* » (en haut de la figure), les processus « *opérationnel* » (au centre) et les processus de « *support* » (au bas de la figure).

Elle fait apparaître les deux grandes activités métier du bureau « *JANUS* » que sont l'ingénierie *logicielle* (pour réaliser les corrections et évolutions du logiciel) et le *soutien aux utilisateurs* (pour assurer le support opérationnel, la validation et la livraison des nouvelles versions).

Le soutien, interne au bureau, est ici considéré comme une activité de « *support* » selon l'approche processus et non comme une activité « *opérationnelle* ».

11.2. Les acteurs et leurs rôles

Dans le cas d'étude qui nous intéresse, quels sont les acteurs en présence dans le cycle de la maintenance du logiciel ?

On peut distinguer 4 grands acteurs qui ont des rôles bien distincts :

- Les centres de formation (les utilisateurs)
- La cellule de pilotage du bureau de la maintenance
- La cellule de soutien opérationnel
- La cellule d'ingénierie logicielle (développement des versions logicielles)

Pour modéliser les rôles tenus par chaque acteur et les liens de dépendance entre chacun d'eux, nous utilisons le diagramme « *Acteurs-Rôles* » proposé dans la méthode EKD-CMM [BAR01].

Les liens de dépendance définissent la nature de la dépendance entre deux acteurs :

- **A** → dépendance d'autorité
- **O** → dépendance d'objectif
- **C** → dépendance de coordination
- **R** → dépendance de ressource

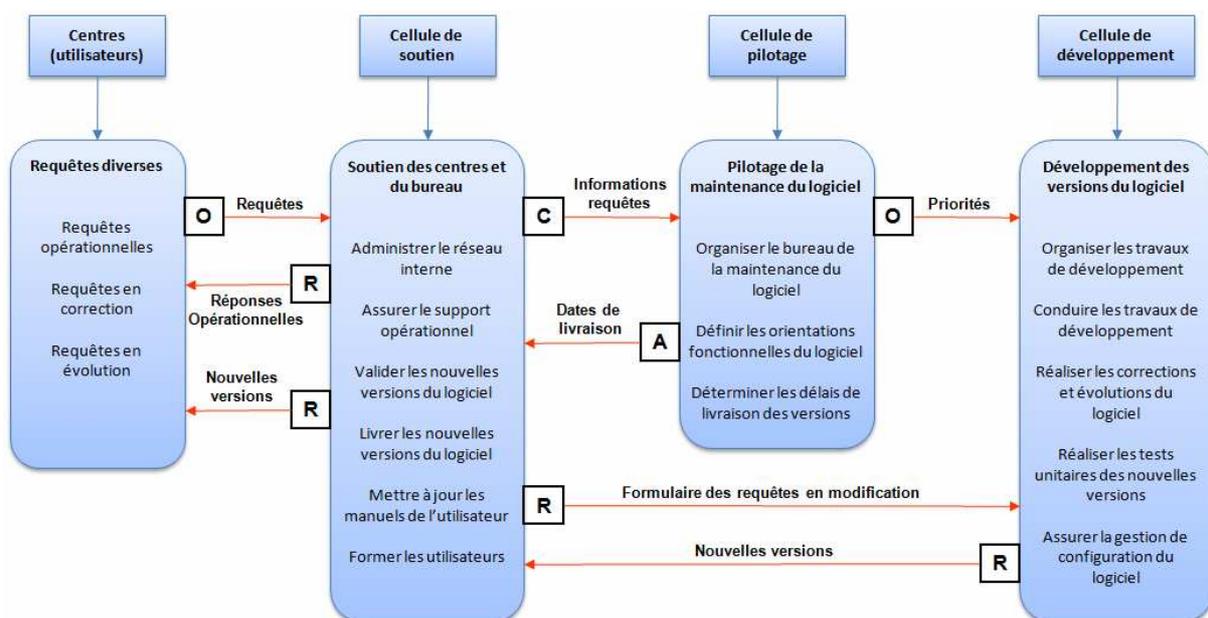


Figure 11.4 Diagramme Acteurs-Rôles du bureau de maintenance du logiciel actuel

11.3. Modèle intentionnel du futur

Dans l'optique de l'amélioration du fonctionnement du bureau JANUS, le changement visé par cette organisation de la maintenance du logiciel doit tendre vers le référentiel proposé par le modèle S3M que nous avons décrit au chapitre II.

Dans leur livre, Alain APRIL et Alain ABRAN [APR06] proposent une cartographie des processus de la *petite maintenance* du logiciel. La représentation qui en est faite ici a été personnalisée tout en respectant les processus définis par les deux auteurs :

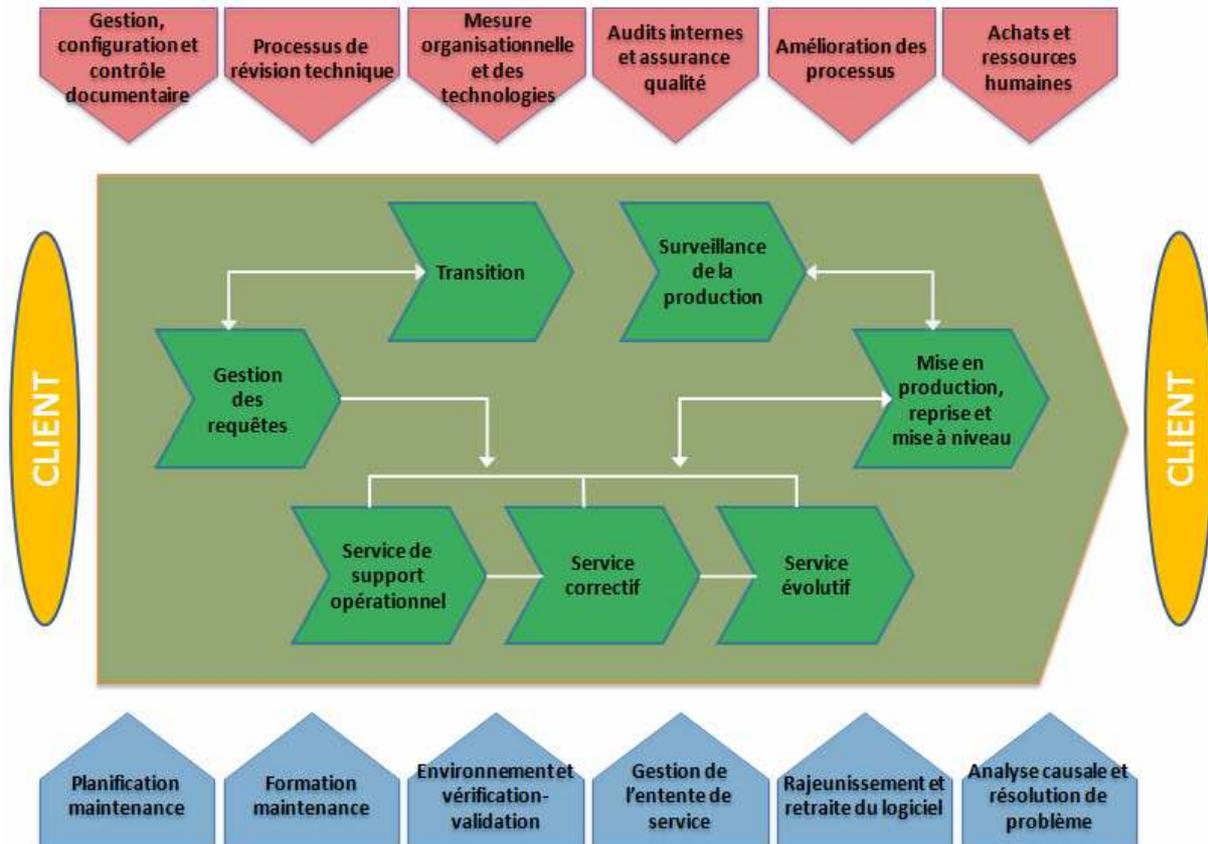


Figure 11.5 Cartographie cible des processus de la maintenance des logiciels

A partir de ces éléments, nous pouvons commencer une analyse de l'impact des changements envisagés sur l'organisation actuelle orientée projet de développement logiciel.

12. Définition du changement organisationnel

Dans notre cas d'étude, la définition du changement organisationnel peut s'opérer à deux niveaux différents :

- Niveau microscopique : impact sur l'organisation du bureau en charge de la maintenance du logiciel (le bureau « JANUS »)
- Niveau macroscopique : impact sur l'organisation globale de la division regroupant 3 bureaux distincts dont le bureau « JANUS »

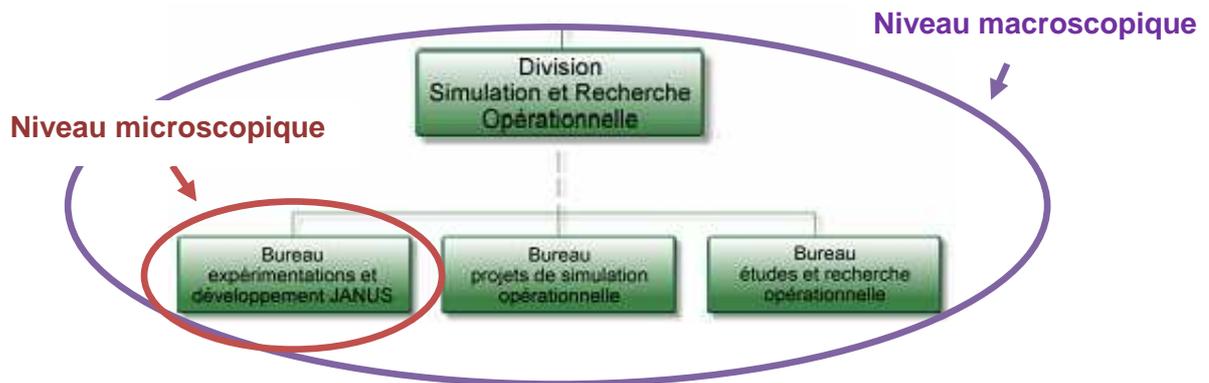


Figure 12.1 Différents niveaux d'impact organisationnel

12.1. Définition du changement organisationnel de la maintenance

En s'appuyant sur la représentation du processus de changement proposé par JACKSON (chapitre III, partie 8.1), notre processus de changement de l'organisation de la maintenance du logiciel peut se résumer ainsi :

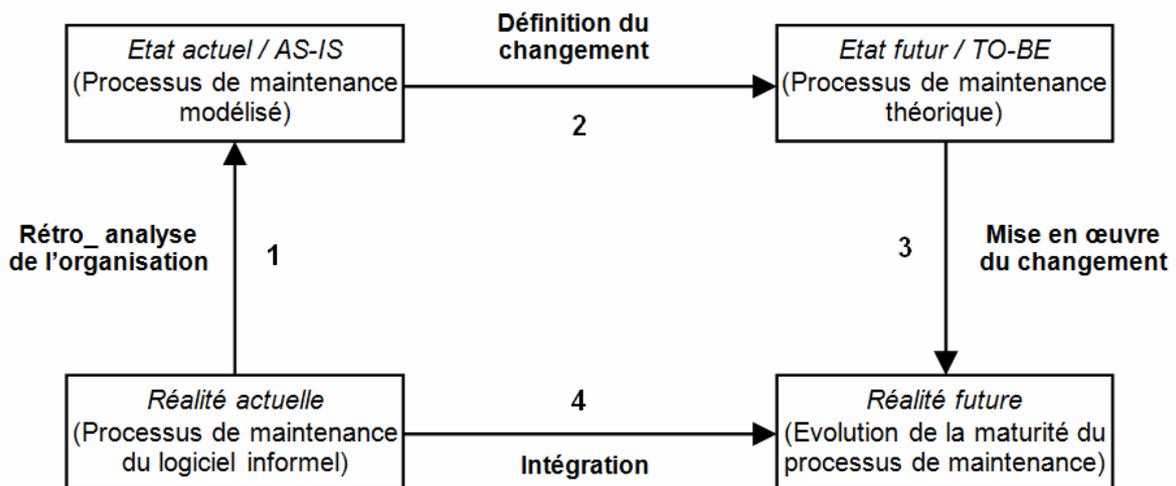
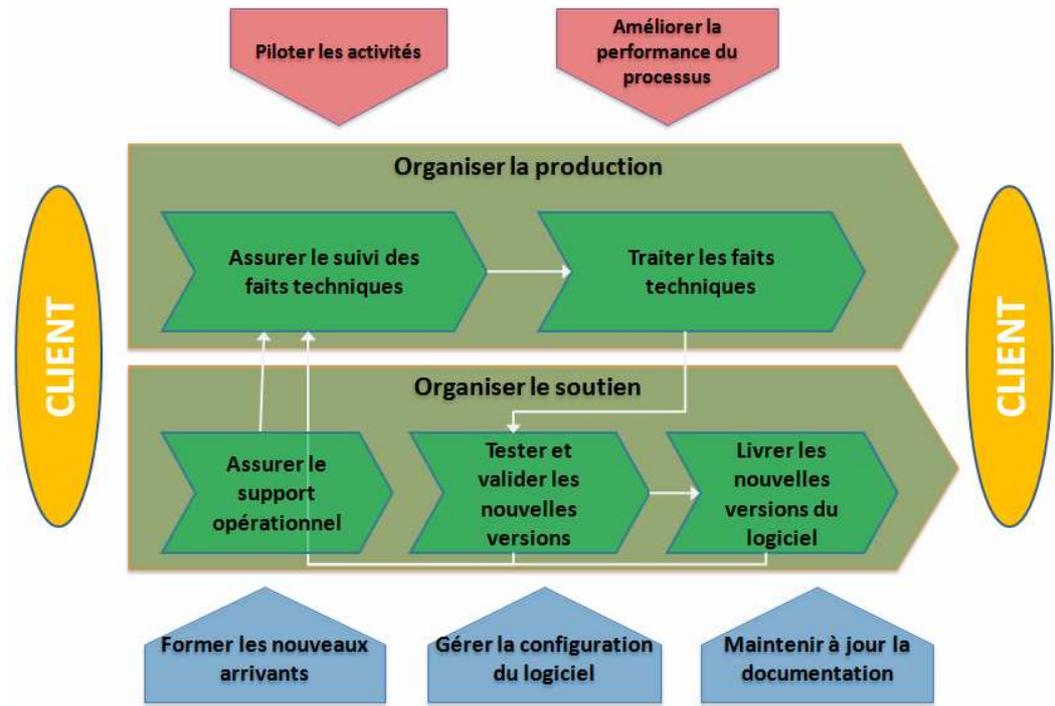


Figure 12.2 Processus de changement de l'organisation de la maintenance du logiciel

D'autre part, pour approfondir cette analyse, nous pouvons comparer la cartographie actuelle des processus de maintenance du logiciel au sein du bureau « JANUS » avec la cartographie théorique proposée par Alain APRIL et Alain ABRAN [APR06]. Cette comparaison nous permet d'identifier les processus manquants dont certains sont mis en évidence ici :

Cartographie « As-Is »



Cartographie « To-Be »

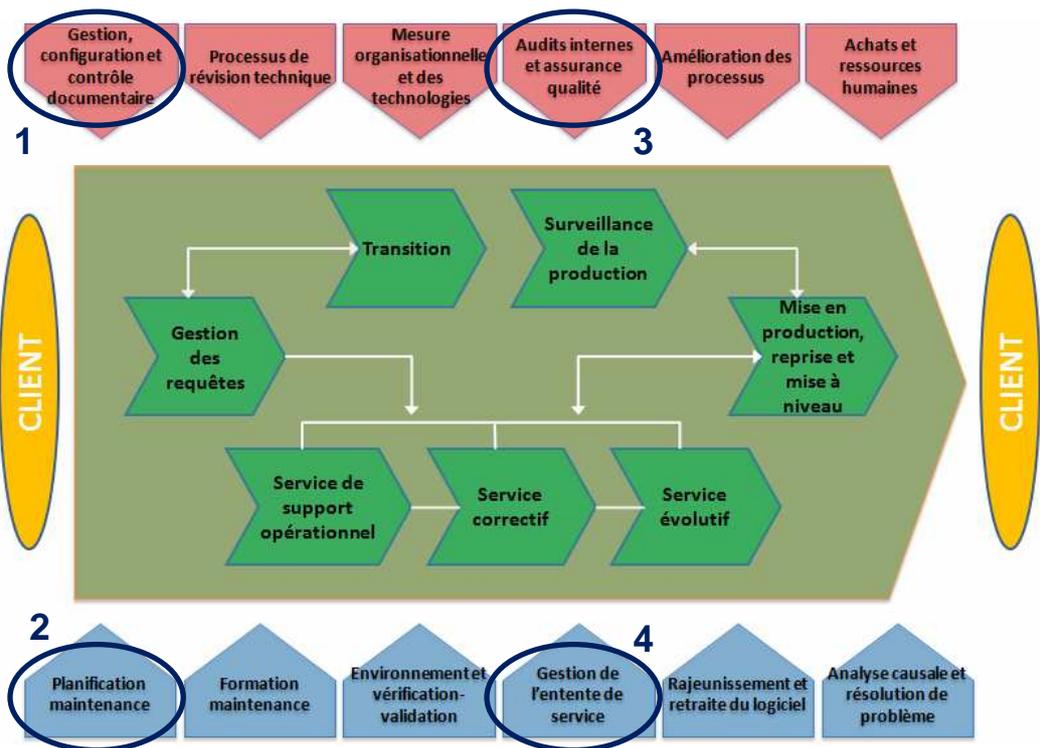


Figure 12.3 Cartographie actuelle et future du processus de maintenance

Nous tirons plusieurs enseignements de cette comparaison :

- D'une manière générale, les *processus métier* (gestion des requêtes, support opérationnel, traitement des modifications, ...etc.) sont effectués

- Les *processus de pilotage* sont perfectibles (par exemple : la gestion documentaire est inexistante ainsi que l'assurance qualité spécifique à la maintenance)
- Les *processus de support* peuvent être améliorés pour rendre l'activité plus visible vis-à-vis du client (par exemple : grâce à une entente de service et une planification mieux gérées)

Mais si l'impact du changement semble évident au niveau de l'organisation de la maintenance du logiciel, qu'en est-il au niveau de la structure dans laquelle les activités de maintenance prennent place au côté de projets de développement logiciels de diverses natures ?

12.2. Définition du changement organisationnel global

Nous l'avons présenté dans le premier chapitre de ce mémoire, le bureau « JANUS » fait parti d'un ensemble composé de trois bureaux, organisés autour des missions (ou métiers) confiés à la division « simulation ». Le processus « gérer les projets » définit les activités de la division et trois procédures ont été créées pour répondre aux besoins spécifiques de chaque bureau :

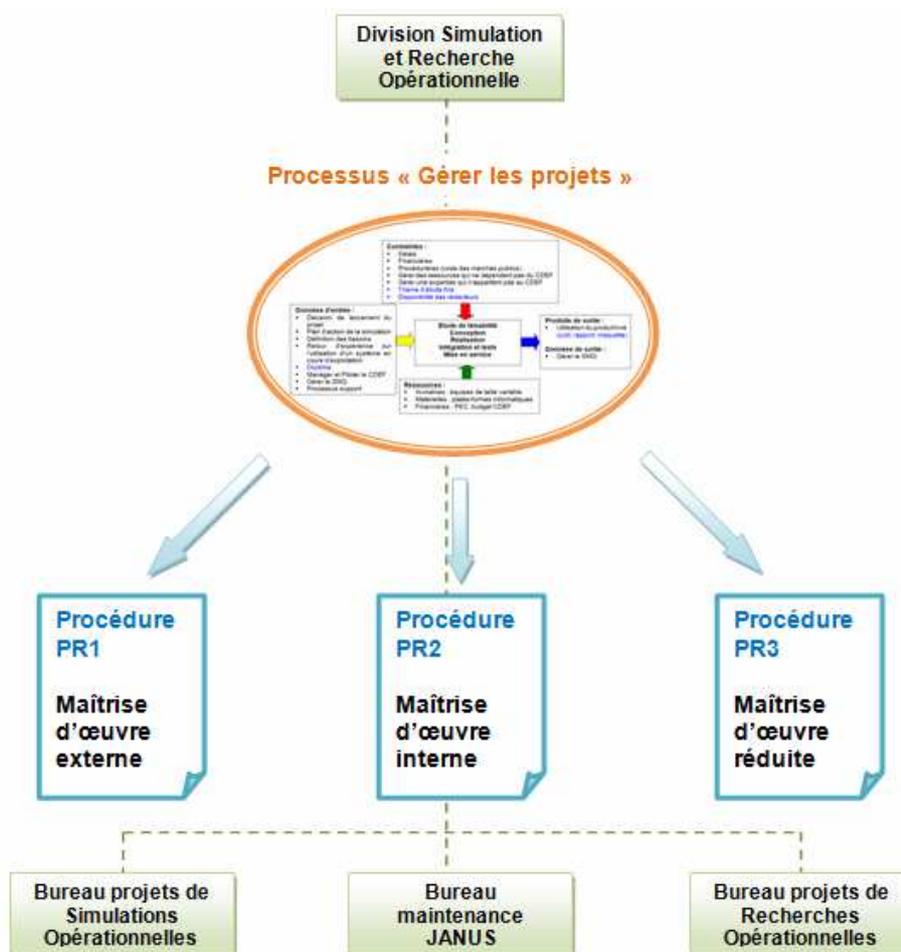


Figure 12.4 Procédures appliquées dans l'organisation

Nous rappelons, ici, que notre démarche est de rendre visible le processus de maintenance mieux adapté aux différents travaux du bureau « *JANUS* » que le processus de gestion de projets sur lequel reposent les activités de la division.

Dans ce cadre, deux solutions pourraient être envisagées :

- a) La création d'une procédure spécifique pour la maintenance
- b) La création d'un processus de maintenance parallèlement au processus de projet

Le 1^{er} cas rejoint ce qui est généralement fait dans les organisations : une procédure spécifique à la maintenance et personnalisée à l'organisation est créée.

L'avantage est de formaliser les activités de maintenance sans remettre en cause l'organisation globale. Par ailleurs, le processus de maintenance peut être formalisé et documenté à l'intérieur du groupe de maintenance indépendamment du reste de la structure.

Le risque est de garder la maintenance applicative dans une confusion récurrente par rapport aux projets logiciels. Et d'autre part, il y a un risque fort de contribuer à maintenir ce domaine d'activité dans l'invisibilité et l'ignorance des problématiques qui lui sont propres.

Le 2^{ème} cas s'apparente à une activité de *réingénierie*. Autrement dit, on s'engage ici dans une refonte d'une partie des processus de l'entreprise.

L'avantage est de prendre la mesure d'une réalité qui ne répond pas aux besoins métier et de s'inscrire dans une logique d'amélioration continue de la qualité comme le préconise l'ISO 9001 : v2000. Cependant, pour mener une telle restructuration, il est indispensable d'être mandaté par la direction de l'entreprise.

Le risque est de désorganiser la structure en place. En effet, dans notre cas d'étude, la division « *simulation* » est organisée autour de son unique processus (« *gérer les projets* »). Cela signifie que l'espace de travail commun a été créé autour des grandes étapes d'un projet, les outils d'accompagnement et de suivi sont ceux d'un projet.

Par exemple, les fiches d'étude sont ouvertes au début du projet et refermées à la fin. On mesure les écarts par rapport aux estimations, ...etc.

Ce qui implique que la démarche de *réingénierie* n'est pas une simple affaire de création d'un processus d'entreprise dans la cartographie des processus mais bel et bien une révision générale du fonctionnement, au moins, en ce qui concerne la division « *simulation* ».

CONCLUSION

Lors de la conférence *Computing Professionals*, en avril 2010 à Montréal (CP2010), Alain APRIL introduisait l'atelier (workshop) « *Software Maintenance* » par une présentation intitulée :

« *A maintainer's nightmare* » (Le cauchemar des mainteneurs)

Si la formule impressionne, elle évoque néanmoins, de manière simple, l'ampleur des difficultés que rencontrent les professionnels de la maintenance du logiciel dans un univers essentiellement tourné vers les projets de développement.

Comme nous l'avons vu dans le premier chapitre de ce mémoire, les problèmes spécifiques à la maintenance sont divers. Ils affectent la dimension technique (peu de documentation, absence de méthodologie, langage et système d'ancienne génération ...etc.) comme la dimension managériale (absence d'indicateurs de mesure, démotivation du personnel, turn-over important, manque de compétences, etc.).

En nous plaçant dans le contexte d'une unité de maintenance applicative de l'armée de Terre, nous nous sommes interrogés, à travers ce mémoire, sur la façon de caractériser les activités de ce métier afin de les rendre plus visibles au sein de l'organisation. Nous estimions, ainsi, répondre en partie aux problèmes de la maintenance du logiciel et susciter une reconnaissance qui fait tant défaut dans ce domaine.

Pour répondre à nos interrogations, nous avons parcouru l'état de l'art des référentiels de pratiques qui s'appliquent dans les métiers des technologies de l'information. Nous avons déterminé que les référentiels CMMi, ITIL et S^{3m} se destinent à un pan de l'informatique qui leur est propre. Ainsi, CMMi a vocation à améliorer la gestion des *projets* informatiques, ITIL, des *opérations* informatiques et S^{3m} de la *petite maintenance* du logiciel.

Par ailleurs, nous en avons définitivement conclu que le processus de maintenance ne peut pas être assimilé à un processus de projet. Le premier est, en effet, constitué d'activités récurrentes et particulières (gestion des requêtes et des priorités, changement de priorité en cours de modification, etc.) alors que le deuxième est constitué d'activités bornées dans le temps et planifiables à l'avance (estimation des coûts et délais, découpage du projet, planification des étapes, etc.).

Fort de ces nouvelles connaissances, nous avons élaboré une démarche pour définir le changement induit par l'amélioration de la maintenance du logiciel. Pour cela nous nous sommes appuyés sur le cadre de référence de la méthode EKD-CMM dont le but est de modéliser le processus de changement d'une organisation pour en extraire les scénarios d'impact organisationnel.

Notre démarche posée, nous l'avons mise en pratique dans un cas d'étude (au sein de notre unité de maintenance). Nous avons débuté notre travail par une évaluation de la maturité du processus actuel sur la base du modèle S^{3m}. Cette évaluation nous a ensuite aidé objectivement à décrire le processus de maintenance sous la forme de modèles empruntés à la méthode EKD-CMM.

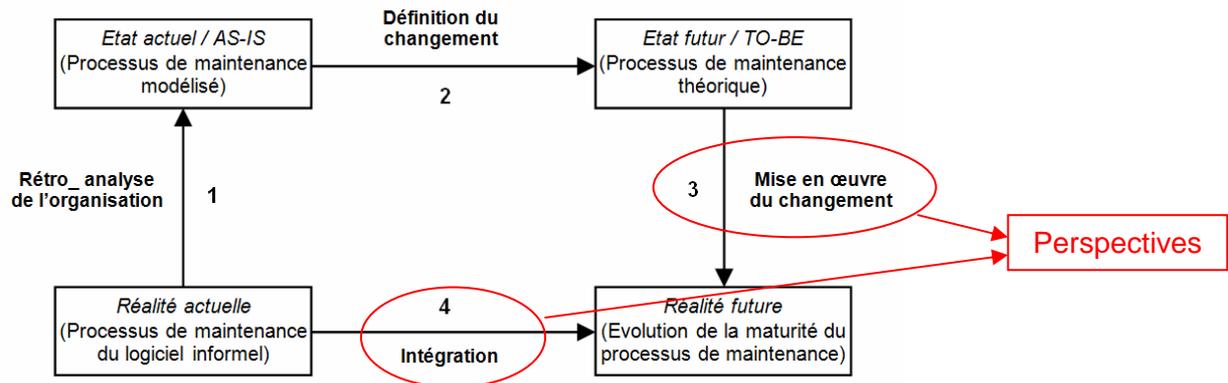
Enfin, nous avons envisagé le changement à deux niveaux organisationnels. Le premier correspond à l'organisation de la maintenance. Et le deuxième concerne la structure, dans laquelle opère l'unité de maintenance, dont l'organisation est orientée *projet*.

Du changement organisationnel de la maintenance, nous en avons déduit des pistes d'améliorations possibles.

Du changement organisationnel de l'entreprise nous avons conclu que la création d'une procédure adaptée à la maintenance aurait un moindre impact sur l'organisation de l'entreprise que la mise en évidence d'un processus de gestion de la maintenance du logiciel.

Quelles perspectives pouvons-nous envisager ?

Notre travail a consisté à modéliser le processus de maintenance du logiciel et à préparer le changement organisationnel. Si nous reprenons le processus de changement de JACKSON, nous pouvons en déterminer les étapes suivantes :



➤ Modéliser les forces contextuelles de changement

Les améliorations qui ont été mises en évidence dans notre étude, peuvent être modélisées sous la forme d'une hiérarchie de buts de changement (*améliorer, introduire, maintenir, ...*).

➤ Mettre en œuvre le changement

Créer les nouvelles procédures et les documenter. Définir les outils de pilotage et de mesure. Introduire les principes de la qualité de la maintenance du logiciel.

➤ Intégrer le changement

Reconsidérer le système d'information qui soutient l'unité de maintenance. Réviser et documenter les procédures existantes. Améliorer les outils techniques dans la mesure du possible.

Quelles nouvelles interrogations en découlent ?

Si ce mémoire s'inscrit dans une démarche de modélisation de la connaissance d'entreprise, il soulève potentiellement de nouvelles interrogations quant à divers domaines :

➤ Mesure de la maintenance et prise de décision

La création d'indicateurs de pilotage est indispensable pour une gestion efficace. Elle contribue, par ailleurs, à améliorer la reconnaissance du travail effectué. Comment définir les indicateurs de la maintenance du logiciel ? Comment les valider ?

➤ Posture de l'auditeur et du référent qualité

Notre étude a suivi une démarche de type *bottom-up* (des réalités du terrain vers les processus d'entreprise). La posture de l'auditeur comme du « qualicien » nous paraît importante. Elle ne s'improvise pas et le travail à accomplir doit être mandaté pour attribuer aux intervenants une réelle autorité, sans quoi, les efforts restent vains.

➤ Qualité, compétences et certification

On ne peut s'inscrire dans une démarche qualité, quelque soit le domaine d'activité, sans s'interroger sur les compétences inhérentes aux différentes activités du domaine et sur la certification des organisations, des personnes, des produits et des services. Comment définir les compétences de la maintenance du logiciel ? Quelle certification pour quelles organisations, quelles personnes, quels produits et services de la maintenance ?

Pour finir, j'aimerais terminer ce mémoire sur une note tout à fait personnelle. Il y a quelques mois de cela, je n'aurais jamais imaginé prendre autant de plaisir à « décortiquer » les standards, les référentiels de pratiques pour déterminer les grands principes qui définissent la maintenance du logiciel. Certainement parce que j'exerce ce métier depuis longtemps et souvent, au sein d'organisations de la maintenance bâties sur une démarche intuitive ou dite de « bon sens ».

La recherche, la littérature spécialisée, la théorie, apportent cet éclairage à la pratique qui font que subitement deux univers fusionnent : la « theoria » et la « praxis », pour enfanter d'une nouvelle connaissance. Cette nouvelle connaissance n'est pas une simple connaissance, elle donne du « sens », elle ouvre la porte magique de la « logique » des choses. Elle offre la possibilité de passer du « je sais faire mais je ne sais pas ce que je fais » au « je sais faire et je sais ce que je fais ».

Ce mémoire a été pour moi une formidable aventure qui m'a conduit jusqu'à Montréal pour exposer un cas concret des problèmes liés à la maintenance du logiciel (merci Alain). J'espère qu'il suscitera de nouvelles vocations, peut-être, sur les chemins vertueux de la qualité...

BIBLIOGRAPHIE

[ABR93] ABRAN A., NGUYENKIM H. *Measurement of the Maintenance Process From a Demand Based Perspective*, Journal of Software Maintenance: Research and Practice, Vol.5, n°2, 1993

[ABR05] ABRAN A., MOORE J. W. : Executive Editors; BOURQUE P., DUPUIS R. : Editors ; L.L. Trip Chair of IAB, *Guide to the Software Engineering Body of Knowledge – SWEBOOK – 2004 Version*, IEEE Computer Society Press, 2005

[APR06] APRIL A., ABRAN A. *Améliorer la maintenance du logiciel*, Loze Dion, Québec, 2006

[BAR01] BARRIOS J. *Une méthode pour la définition de l'impact organisationnel du changement*, Thèse de Doctorat de l'Université de Paris 1, 2001

[BAS06] BASQUE R. *CMMI, un itinéraire fléché vers le Capability Maturity Model Integration Version 1.2*, Dunod, Paris, 2006, 2è éd.

[BOU06] BOUVIER A. *Management et sciences cognitives*, PUF, Paris, 2006, 2è éd.

[CAM94] Camélia *Modèle d'évaluation de la capacité à développer, entretenir et à exploiter des produits informatiques*, Modèle conjoint Groupe de travail France Québec, octobre 1994

[CDE09] CDEF *Manuel de management de la qualité*, Centre de Doctrine d'Emploi des Forces, Armée de Terre, novembre 2009

[CNR01] CNRS *Maintenance/évolution d'un système d'information*, Centre National de la Recherche Scientifique, mai 2001, [en ligne]: < <http://www.dsi.cnrs.fr/conduite-projet/phasemaintenance/presentation/default.htm> >, (consulté le 15 avril 2009)

[COB00] IT Governance Institute *CobiT, Governance, Control and Audit for Information and Related Technology*, juillet 2000, 3è éd.

[CRI04] CRI *EKD-CMM (Enterprise Knowledge Development - Change Management Method)*, Centre de Recherche en Informatique, 1999, [en ligne]: < <http://crinfo.univ-paris1.fr/id44.htm> >, (consulté le 10 janvier 2010)

[DEK92] DEKLEVA S. *Delphi study of Software Maintenance Problems*, Proceedings of the IEEE Conference on Software Maintenance (ICSM), Orlando, 1992, (cité dans [APR06])

[DES97] DESHARNAIS J-M., PARE F., ST-PIERRE D. *Implementing a Measurement Program in Software Maintenance: An Experience Report Based on Basili's Approach*, Presented at IFPUG Conference, Cincinnati, 1997

[DSR09] DSRO *Procédure relative à un projet réalisé avec une MOE constituée par une équipe de développement interne*, Centre de Doctrine d'Emploi des Forces/ Division Simulation et Recherche Opérationnelle, Armée de Terre, octobre 2009

[GAR93] GARCIA S. *Principles of Maturity Modeling*, Proceedings of SEI Symposium, Pittsburgh, 1993, (cité dans [APR06])

[HAM03] HAMMER M., CHAMPY J. *Le Reengineering*, Dunod, Paris, 2003, 2è éd.

[ISO95] ISO/IEC 12207 *Information Technology: Software Life Cycle Processes*, International Organization for Standardization, Geneva, 1995

[ISO98] ISO/IEC 14764 *Software Engineering - Software maintenance*, International Organization for Standardization, Geneva, 1998

[ISO98a] ISO/IEC 15504 *Information Technology: Software process assessment*, International Organization for Standardization, Geneva, 1998

[ISO05] ISO/IEC 20000 *Information Technology: Service Management*, International Organization for Standardization, Geneva, 2005

[ISO08] ISO *Guide sur le concept et l'utilisation de l'approche processus pour les systèmes de management*, Organisation Internationale de Normalisation, 2008, [en ligne] : < http://www.iso.org/iso/fr/iso_catalogue/management_standards/iso_9000_iso_14000/iso_9001_2008/ >, (consulté le 23 février 2010)

[ITI04] ITIL France *Information Technology Infrastructure Library – version 2*, 2004, [en ligne] : < <http://www.itilfrance.com/>>, (consulté le 28 septembre 2009)

[JOU06] LeJOURNALduNET *Expliquez-moi... Le CMMi*, 2006, [en ligne] : < <http://www.journaldunet.com/developpeur/tutoriel/theo/060524-modele-cmmi.shtml> >, (consulté le 1er octobre 2009)

[KAJ01] KAJKO M. *Corrective Maintenance Maturity Model*, Partial fulfillment of the requirements for P.H.D, Stockholm University, 2001

[LEB08] LEBRUN V. *Méthodes d'évaluation des processus de la maintenance des logiciels*, Faculté Universitaire Notre-Dame de la Paix, Namur, Belgique, 2008

[LOR03] LORINO P. *Méthodes et pratiques de la performance*, Editions d'Organisation, Paris, 2003, 3è éd.

[LEW58] LEWIN K. *Group Decision and Social Change*, Readings in Social Psychology, E. Maccoby, T.Newcomb, E. Hartley (Eds), Hold, Rinehart & Winston, New York, 1958

[MET07] METHODS & TOOLS *A Trend Towards Lower Software Maintenance Budgets?*, 2007, [en ligne] : < <http://www.methodsandtools.com/dynpoll/oldpoll.php?Maintenance> >, (consulté le 27 décembre 2009)

[MOR07] MORLEY C., HUGUES J., LEBLANC B., HUGUES O. *Processus métiers et systèmes d'information*, Dunod, Paris, 2007, 2è éd.

[MOR08] MORLEY C. *Management d'un projet SI*, Dunod, Paris, 2008, 6è éd.

[MOU06] MOULINEC J., LOYER L., CHRISTENSEN F., GUYADER H. *Management des opérations informatiques et ITIL*, Lavoisier, Paris, 2006

[NUR02] NURCAN S., BARRIOS J., ROLLAND C. *Une méthode pour la définition de l'impact organisationnel du changement*, Numéro Spécial de la Revue Ingénierie des Systèmes d'Information « Connaissances Métier dans l'Ingénierie des SI, 7:4 », 2002

[PAQ06] PAQUETTE D-A., APRIL A., ABRAN A. *Assessment Results Using the Software Maintenance Maturity Model (S3m)*, Proceedings of the 16th International Workshop on Software Measurement (IWSM-Metrikom), Postdam, Germany, 2006

[PIC05] PICQ T. *Manager une équipe projet*, Dunod, Paris, 2005, 2^e éd.

[PIG97] PIGOSKY T. *Practical Software Maintenance: Best Practice for Managing your software investment*, Wiley, New York, 1997, (cité dans [APR06])

[PRO07] 01NETPRO *Réduire les coûts de maintenance du logiciel*, 2007, [en ligne] : < <http://pro.01net.com/editorial/358862/reduire-les-couts-de-maintenance-du-logiciel/> >, (consulté le 15 octobre 2009)

[QUA09] QUALITEONLINE *Approche processus*, 2009, [en ligne] : < http://www.qualiteonline.com/rubriques/rub_3/dossier-19.html >, (consulté le 12 février 2010)

[REI99] REIX R. *Dictionnaire des systèmes d'information*, Vuibert, Paris, 1999

[REI05] REIX R. *Systèmes d'information et management des organisations*, Vuibert, Paris, 2005, 5^e éd.

[ROL94] ROLLAND C., GROSZ G. *A general framework for describing The Requirements Engineering Process*, Proceedings of the Conference on Systems Man and Cybernetics (CSMC), San Antonio, Texas, 1994

[SCA06] SCAMPI Upgrade team *Appraisal Requirements for CMMi, version 1.2 (ARC, v1.2)*, Technical Report, CMU/SEI, 2006

[SCH09] SCHAPPELLE S. *Software Maintenance*, cours de l'Université privée de recherche John Hopkins, Montgomery Center, Rockville (Maryland), 2009, [en ligne] : < <http://www.apl.jhu.edu/Classes/Notes/Schappelle/CS401/> >, (consulté le 7 janvier 2010)

[SEI06] SEI/CMU *CMMi pour le développement, version 1.2*, 2006, [en ligne] : < <http://www.sei.cmu.edu/cmmi/tools/translations/french.cfm> >, (consulté le 28 septembre 2009)

[ZIT96] ZITOUNI M. *Elaboration d'un outil d'évaluation et d'amélioration du processus de maintenance des logiciels*, Ecole de Technologie Supérieure, Montréal, 1996