

ÉTS
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

RAPPORT DE MÉMOIRE
PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE LOGICIEL
Maîtrise en génie logiciel

PAR
NIKIEMA, Wendpouire Edmond

EXPÉRIENCE DE FORMALISATION D'UN PROCESSUS DE DÉVELOPPEMENT
LOGICIEL

MONTREAL, LE 8 NOVEMBRE 2008

© Nikiema, 2008

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE :

M. Alain April, directeur de projet
Département de génie logiciel et des technologies de l'information à l'École de Technologie
Supérieure

M. Claude Laporte, président du jury
Département de génie logiciel et des technologies de l'information à l'École de Technologie
Supérieure

REMERCIEMENTS

J'aimerais saisir l'opportunité d'adresser mes sincères remerciements aux personnes qui m'ont apporté leur aide et qui ont ainsi contribué à l'élaboration de ce mémoire.

Tout d'abord à Monsieur Alain April, mon directeur de projet de maîtrise, pour les conseils, le temps et la patience qu'il a bien voulu me consacrer tout au long de ce projet.

J'aimerais remercier également TELUS Solutions d'Affaires (TSA) particulièrement les membres de mon équipe qui ont bien voulu répondre à mes questions, mes sondages, ... avec gentillesse.

Je voudrais exprimer toute ma gratitude à mes parents et mes sœurs pour leurs soutiens inconditionnels.

Enfin, un merci spécial à Alizèta, Auriane et tous mes amis qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce projet.

EXPÉRIENCE DE FORMALISATION D'UN PROCESSUS DE DÉVELOPPEMENT LOGICIEL

NIKIEMA, Wendpouire Edmond

RÉSUMÉ

Ce projet consiste à cartographier les processus de développement logiciel d'une équipe TI au sein de TELUS Solutions d'Affaires. La démarche adoptée est calquée sur le cadre de Basili, et se subdivise en quatre étapes formant la structure de ce document. Ainsi :

- La première partie du document présente la définition du projet à savoir la motivation, l'objet, l'objectif, le domaine et les utilisateurs potentiels de ce projet.
- La deuxième partie du document, décrit la portée de la revue de littérature sur le domaine des processus de développement logiciel agile (Scrum, XP, DSDM), sur les méthodes de représentation des processus notamment celle de Qualigram et sur les normes ISO 9001 :2000 et ISO 12207. Elle présente aussi le contexte et les objectifs de notre étude.
- La troisième partie de ce document présente une proposition de formalisme des processus de développement du logiciel d'une équipe de TELUS à Montréal.
- Dans la dernière étape de ce projet, les résultats de l'expérimentation du formalisme et les interprétations sont présentés.

EXPÉRIENCE DE FORMALISATION D'UN PROCESSUS DE DÉVELOPPEMENT LOGICIEL

NIKIEMA, Wendpouire Edmond

ABSTRACT

This project consists of engineering TELUS Business Solutions IT Unit's software development process. The used approach inspires from Basili's framework, and builds on four main parts:

- The first part of this document depicts the project's scope, motivations, goals, and stakeholder's interests.
- The second part covers literature review of existing agile software development processes (Scrum, XP, DSDM), exposes process representation as recommended by Qualigram and on ISO 9001: 2000 and ISO 12207. It also presents the background and objectives of our study.
- The third part proposes a formal software process development for TELUS Montreal's IT Team.
- The last part of this document discusses the implementation of the pilot formal process, analyses the collected results.

TABLE DES MATIÈRES

	Page
INTRODUCTION	13
CHAPITRE 1 REVUE LITTÉRAIRE	18
1.1 Processus de développement agile.....	19
1.1.1 DSDM : Dynamic Systems Development Method.....	19
1.1.2 SCRUM	24
1.1.3 XP : extrême programming.....	28
1.2 Représentation des processus.....	31
1.2.1 Qualigram	31
1.3 Représentation d'un processus avec Qualigram manager	36
1.4 Formes proposées par l'outil Qualigram au niveau 1 et 2 de la pyramide	38
1.5 ISO/IEC 12207	39
1.6 ISO 9001:2000.....	41
1.7 Conclusion	44
CHAPITRE 2 PÉRIMÈTRE DE LA SITUATION ACTUELLE	46
2.1 Présentation de l'entreprise : TELUS Solutions d'Affaires (TSA)	46
2.2 TELUS Solutions d'Affaires et ISO 9001	47
2.2.1 Méthodologie : cadre de gestion de projet de TELUS Agilité	47
2.2.2 Cycle de vie d'un projet de TSA	48
2.2.3 Processus actuels.....	51
2.3 Forces et faiblesses de TSA	52
2.4 Objectifs de la recherche.....	52
2.5 Conclusion	53
CHAPITRE 3 PROCESSUS APPLICABLE ET DÉMARCHE DU PROJET	55
3.1 Introduction.....	55
3.2 Analyse des processus Scrum et XP dans le contexte TSA.....	55
3.3 Solution de cartographie de processus de développement logiciel.....	56
3.3.1 Étape 1: Niveau 1 Qualigram.....	57
3.3.2 Étape 2: Niveau 2 Qualigram.....	62
3.3.2.1 P1.1: kick-off meeting	64
3.3.2.2 P1.2: mises en place de l'environnement de développement	67
3.3.2.3 P1.3: assignation des ressources	70
3.3.2.4 P2: planification du sprint.....	73
3.3.2.5 P3: planification hebdomadaire du sprint	76
3.3.2.6 P4: meeting hebdomadaire du sprint	79
3.3.2.7 P5.1: analyse	82
3.3.2.8 P5.2: architecture	86
3.3.2.9 P5.3: réalisation ou codage	89
3.3.2.10 P5.4: test.....	92

3.3.2.11 P5.5: intégration.....	95
3.3.2.12 P6: révision du sprint	97
3.3.2.13 P15: troubles shooting	100
3.4 Déploiement du processus	103
3.4.1 Évaluation de la culture du groupe de TI.....	103
3.4.2 Former le groupe de TI au nouveau processus	104
3.4.3 Assurer un contrôle et un suivi du processus.....	104
3.4.4 Réviser le processus.....	105
3.5 Conception de l’outil de collecte de données	105
3.6 Outil de mesure de l’état avant et après l’implantation du processus.....	106
3.7 Conclusion	108
CHAPITRE 4 INTERPRÉTATION DES RÉSULTATS	109
4.1 Introduction.....	109
4.2 Résultats de l’expérimentation.....	111
4.3 Sommaire des forces et faiblesses	113
4.3.1 Sommaire des forces.....	113
4.3.2 Sommaire des faiblesses	114
4.4 Travaux futurs.....	115
4.4.1 À court terme	115
4.4.2 À moyen et long terme :	115
4.5 Conclusion	115
CONCLUSION.....	117
ANNEXE I MODÈLES DE DOCUMENTS ET OUTILS DÉVELOPPÉS	119
ANNEXE II PROCESSUS TELUS AGILITÉ EXISTANT	123
ANNEXE III COLLECTE DE DONNÉES.....	130
ANNEXE IV QUALIGRAM	134
ANNEXE V FORMULAIRE DE DESCRIPTION DE PROCESSUS-MODÈLE ETVX ...	143
LISTE DE RÉFÉRENCES	144

LISTE DES TABLEAUX

	Page
Tableau 0.1:cadre de Basili adapté à la recherche	17
Tableau 1.1:Livrable du projet DSDM.....	23
Tableau 1.2:Objectifs du processus Scrum.....	26
Tableau 1.3: Règles et pratiques du XP adaptées	30
Tableau 3.1:Objectifs du processus	61
Tableau 3.2: Formulaire de description de processus.....	63
Tableau 3.3: kick-off meeting.....	65
Tableau 3.4: Mises en place de l'environnement de développement.....	68
Tableau 3.5: Assignation des ressources	71
Tableau 3.6: Planification du sprint.....	74
Tableau 3.7: Planification Hebdomadaire du sprint	77
Tableau 3.8: Meeting hebdomadaire du sprint	80
Tableau 3.9: description du processus d'analyse	83
Tableau 3.10: architecture.....	87
Tableau 3.11: Processus de codage.....	90
Tableau 3.12: Processus de Test.....	93
Tableau 3.13: Processus d'intégration.....	95
Tableau 3.14: Processus de révision du sprint.....	98
Tableau 3.15: processus de troubles shooting	101
Tableau 3.16: Analyse du processus.....	105
Tableau 3.17: Formulaire d'évaluation du processus en cours	108
Tableau 4.1: Sommaire des résultats obtenus avant et après l'implantation du processus	111

Tableau 4.2: Critique des processus	112
Tableau 4.3: backlog-général.....	119
Tableau 4.4: backlog-détaillé.....	121
Tableau 4.5: Données du burndown	122
Tableau 4.6: Expérience de l'équipe vis à vis du projet.....	130
Tableau 4.7: Sondage avant le déploiement du processus.....	131
Tableau 4.8: Sondage après le déploiement du processus.....	133

LISTE DES FIGURES

	Page
Figure 1.1: Processus de développement logiciel DSDM adapté.....	21
Figure 1.2: Processus Scrum.....	26
Figure 1.3: Vue d'ensemble du processus XP.....	28
Figure 1.4: Différents types de formes du langage QUALIGRAM	32
Figure 1.5: Mises en œuvre du langage graphique QUALIGRAM.....	32
Figure 1.6: Notion zoom sur la pyramide Qualigram.....	35
Figure 1.7: représentation d'un processus au niveau 2 avec Qualigram	37
Figure 1.8: Légende utilisé dans la représentation Qualigram	38
Figure 1.9: Processus logiciels ISO 12207:2008 Adaptés en français	40
Figure 1.10:Processus ISO 9001:2000 -processus de réalisation et les autres processus.....	42
Figure 2.1: Structure du projet dans le cadre de gestion de projet TELUS Agilité	48
Figure 2.2: Cycle de vie de projet dans le cadre de gestion de projet de TELUS Agilité	49
Figure 2.3: Processus de mise en œuvre	51
Figure 3.1: processus de développement TELUS.....	58
Figure 3.2: kick-off meeting	66
Figure 3.3:Mise en place de l'environnement de développement.....	69
Figure 3.4:assignation des ressources.....	72
Figure 3.5:planification du sprint	75
Figure 3.6:planification hebdomadaire du sprint.....	78
Figure 3.7:meeting hebdomadaire du sprint	81
Figure 3.8:analyse.....	85
Figure 3.9:architecture	88

Figure 3.10:réalisation ou codage.....	91
Figure 3.11: Test.....	94
Figure 3.12:intégration	96
Figure 3.13:révision du sprint.....	99
Figure 3.14:troubles shooting.....	102
Figure 3.15: Évaluation de la culture du groupe de TI.....	103
Figure 4.1: Structure et composition de l'équipe TI	109
Figure 4.2: Processus d'initialisation	123
Figure 4.3: Processus de prise en charge	124
Figure 4.4: Processus de mise en œuvre.....	125
Figure 4.5: Processus de satisfaction de la clientèle et de fermeture.....	126
Figure 4.6: Processus d'évolution du cadre de gestion	127
Figure 4.7: Processus d'audit	128
Figure 4.8: Processus d'évolution des ressources	129
Figure 4.9: Différents types de formes du langage Qualigram.....	135
Figure 4.10: Approche global au plus détaillée et inversement.....	137
Figure 4.12: Notion zoom sur la pyramide Qualigram.....	139
Figure 4.13: Différents types de navigation Qualigram	140

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

CMMI	Capability Maturity Model and Integration
CobiT	Control Objectives for Business and Related Technology
DSDM	Dynamic systems development method
ETVX	Entry-Task-Validation-Exit
ISO	International Organisation for standardization
ITIL	Information Technology Infrastructure Library
OMG	Object Management Group
OSSAD	Office Support Systems Analysis and Design
RAD	Rapid Application Development
SQUARE	Security Quality Requirements Engineering
SEI	Software Engineering Institute
SVN	Subversion
SWEBOK	Software Engineering Body of Knowledge
TSA	TELUS Solutions d'Affaires
XP	Extreme programming

INTRODUCTION

De nos jours une entreprise de développement se démarque des autres par la qualité du logiciel réalisé, le temps et le budget consacrés à cette réalisation. Ces défis poussent beaucoup d'entreprises à rechercher de nouveaux moyens pour y faire face et par la même occasion, pour mieux se positionner par rapport à la concurrence. Certaines choisissent d'améliorer leur façon de réaliser leurs produits logiciels en élaborant des processus logiciels ou en les révisant lorsqu'ils existent. Que ce soit pour l'élaboration ou l'amélioration des processus, ces entreprises se basent sur des normes de processus logiciels existants tels que : l'ISO 12207 et l'ISO 15288, des guides de bonnes pratiques tels que : le CMMI, CoBiT, ITIL et des méthodes agiles qui sont des techniques émergentes.

Le SEI (Software Engineering Institute) stipule qu'une entreprise ne peut prétendre améliorer ses processus de développement logiciel tant que ces processus ne sont pas définis formellement [JUN-01]. Selon le SEI, l'absence de description formelle de processus logiciels empêche d'avoir des résultats reproductibles. La réussite d'un projet dans ce cas, repose uniquement sur les connaissances informelles de l'équipe. Il devient donc très difficile d'assurer le succès des projets logiciels, dans de telles conditions.

Un processus logiciel est une succession d'activités utilisées par une entreprise pour planifier, gérer, exécuter, surveiller, contrôler et améliorer le logiciel [LEP-01].

Tout comme la technologie, le processus logiciel est lié étroitement au type de projet [Ali-01]. Il existe plusieurs types de logiciels et une seule méthode de développement ou de processus logiciel ne peut couvrir ce vaste nombre [Ali-01]. Un processus logiciel fonctionne dans un contexte spécifique et sa définition ou son amélioration nécessite non seulement de prendre connaissance des modèles existants mais aussi de comprendre les types de projets de l'entreprise [Ali-01].

La recherche, réalisée dans le cadre de la maîtrise en génie logiciel, vise à expérimenter la formalisation du processus de développement de logiciel au sein d'une équipe de

développement de TELUS Solutions d'Affaires (TSA) à Montréal. La démarche méthodologique suivie dans le cadre de cette recherche, est fondée sur le cadre de Basili [BASI-01] adapté pour la recherche exploratoire en génie logiciel [BASI-02]. En effet ce cadre fournit une méthodologie aidant à planifier un projet de recherche en quatre étapes, dans le but de mieux décrire le travail qui sera effectué à chaque étape de la recherche [BASI-02]. La première étape vise la définition du projet de recherche et consiste à définir la motivation, l'objet, l'objectif, le domaine et les utilisateurs potentiels de cette recherche (voir tableau 0.1).

La deuxième étape de la recherche, décrite au tableau 0.1, comporte trois phases. La première phase décrit la portée de la revue de littérature. Cette revue permettra, dans un premier volet, de présenter l'état de l'art dans le domaine des processus de développement logiciel. Dans un deuxième volet nous étudierons les méthodes de représentation des processus dont principalement la méthode proposée par l'outil QUALIGRAM. Dans le cadre de cette recherche l'entreprise Taktika Management nous a offert d'utiliser son outil de description de processus et sa méthode. Après vérification, cette méthode semble répondre aux besoins immédiats de cartographie de processus et nous nous limiterons uniquement à l'étude de QUALIGRAM comme méthode et langage de description de processus. Enfin dans le troisième volet, nous survolerons les normes ISO 9001:2000 et ISO 12207 qui décrivent les exigences nécessaires pour le processus de management de qualité et les processus qui s'appliquent au cycle de vie du logiciel. La deuxième phase de la deuxième étape du cadre de Basili présente le contexte et les objectifs de notre étude. Nous effectuerons la description de l'organisation et le contexte d'intervention. C'est à cette étape que nous mesurons l'efficacité du processus informel actuel. Cette mesure de départ nous servira de point de départ pour l'évaluation de l'amélioration visée. Ensuite nous présenterons les problèmes, objectifs et hypothèses de la recherche.

La troisième étape de cette recherche consistera à formaliser les processus de développement du logiciel d'une équipe de TELUS à Montréal. Par la suite nous expliquons comment la validation de ce processus a été effectuée. Pour se faire nous allons réaliser un outil de collecte des données qui servira à mesurer l'exactitude du processus à bien représenter les activités qui sont effectuées informellement suite à sa formalisation.

Dans la dernière étape de ce projet, nous présentons les résultats de l'expérimentation en interprétant les résultats obtenus. À la suite de ces résultats nous pourrions conclure sur l'atteinte de nos objectifs. Ainsi nous pourrions suggérer des changements à apporter et guider les dirigeants de l'entreprise avec les résultats de cette première expérience de formalisation des processus de développement.

ÉTAPE DE DÉFINITION	
Motivation	Permettre l'amélioration des processus de développement logiciel de l'équipe de développement de TSA par la formalisation du processus actuel.
Objet	Faire un pilote de définition de processus pour aider la direction à comprendre les enjeux de la définition et de l'amélioration des processus TI.
Objectif	Identifier, documenter et cartographier les processus de développement logiciel dans un groupe de travail actuel de TSA. Étudier les résultats et émettre des recommandations.
Domaine	Amélioration des processus logiciel.
Utilisateurs	Équipe de développement (chef de projet, analyste, architecte, développeurs, testeurs, mainteneurs). Responsable de l'assurance qualité chez TELUS. Chercheurs en amélioration des processus.

ÉTAPE DE PLANIFICATION		
Étapes du projet	Intrants	Livrables
Phase I : Revue de littérature	<p>-Revue littéraire du domaine des processus de développement du logiciel.</p> <p>-Littérature de la représentation des processus, dont l'outil et la méthode QUALIGRAM.</p> <p>-Normes du domaine du logiciel : ISO 9001:2000 et ISO 12207.</p>	Synthèse et état de l'art dans le domaine des processus de développement logiciel (Chapitre 1 du mémoire).
Phase II : Périmètre de la situation actuelle	<p>Description de l'entreprise.</p> <p>Mesure de l'état de la situation actuelle (sans processus formel).</p> <p>Problématique et objectifs de la recherche.</p>	Contexte et objectifs du travail de recherche (Chapitre 2 du mémoire).

ÉTAPE D'EXÉCUTION		
Conception d'un processus de développement	<ul style="list-style-type: none"> -Élaboration de la démarche de conception. -Cartographie du processus dans une équipe à l'aide de la méthode et de l'outil Qualigram. -Évaluation de la culture de l'entreprise. -Déployer le processus. -Conception de l'outil de collecte de données. -Collecte de données. 	Analyse (imperfections, avantages) (Chapitre 3 du mémoire).
ÉTAPE D'INTERPRÉTATION		
Contexte d'interprétation	Interprétation des résultats	Travaux futurs
	Sommaire des forces et des faiblesses.	Apporter de l'amélioration dans tous les autres processus (Chapitre 4 du mémoire).

Tableau 0.1:cadre de Basili adapté à la recherche

CHAPITRE 1

REVUE LITTÉRAIRE

La fin des années 80 est marquée par le formalisme des processus en entreprise [QUALI-01]. De nombreuses entreprises de développement se sont basées sur des modèles de référence pour définir leurs processus. Des standards comme ISO 12207, ISO 15288 ont connu déjà un succès en entreprise mais sont critiqués pour leurs lourdeurs (plus de documentation à produire,...). Au cours des dernières années une nouvelle approche des processus logiciels a vu le jour et vise à décrire le processus logiciel le plus simplement possible [JEN-01] [WHM-01]. Il s'agit de la proposition de méthodes dites agiles. Bien que critiquées pour la légèreté de la documentation qui accompagne les logiciels, beaucoup d'entreprises adoptent ces nouvelles méthodes de développement logiciel et réussissent à satisfaire leurs clients [NVD-01][WHM-01]. En effet ces méthodes offrent de nombreux avantages tant pour le client que pour le fournisseur du logiciel. L'objectif principal des méthodes agiles est d'augmenter la satisfaction des clients en offrant une livraison continue et rapide des produits logiciels. Ces méthodes facilitent l'intégration des changements qui surviennent notamment pendant la réalisation des solutions logicielles. Ceci est une valeur ajoutée pour le client qui obtient toujours une solution logicielle de qualité c'est-à-dire une solution utile [AGIL-01]. De plus, beaucoup de ces méthodes agiles offrent un processus logiciel aidant à atteindre les objectifs mais s'efforcent de les garder légers [RRR-01]. Les entreprises peuvent ainsi représenter leurs processus avec un niveau de détail approprié à leur contexte pour atteindre leurs objectifs.

En effet les processus peuvent être définis selon plusieurs niveaux d'abstraction et sont toujours présentés au moyen d'une représentation. Il y a de nombreuses techniques de représentation enseignées (diagramme de flow, diagramme d'état, ETVX, OSSAD,...) et utilisées par les ingénieurs du logiciel [AACL-01]. Pour cette recherche nous utiliserons la méthode OSSAD avec l'outil Qualigram qui nous a été fourni gracieusement par l'entreprise Taktika Management pour notre projet pilote. Le livre de Cédric Berger et de Serge Guillard

intitulé « *la rédaction graphique des procédures: démarches et techniques de description de processus* » sera notre principale source de référence en ce qui concerne la méthode et l’outil QUALIGRAM. Nous utiliserons aussi les concepts de la norme ISO 12207 qui est la norme de référence internationale pour la définition des processus de cycle de vie du logiciel. Une autre norme, l’ISO 9001 version 2000 sera utilisée puisque TELUS possède cette certification dans d’autres processus de l’entreprise.

1.1 Processus de développement agile

Comme nous l’avons déjà indiqué, développer et/ou maintenir des logiciels avec des contraintes de courts délais de développement constituent des défis pour les fournisseurs logiciels d’où l’intérêt grandissant des méthodes agiles comme : DSDM, Scrum et XP pour n’en nommer que quelques unes.

1.1.1 DSDM : Dynamic Systems Development Method

Le DSDM (Dynamic Systems Development Method) est une méthode agile populaire de développement de logiciel. Cette méthode est issue d’une initiative britannique en 1994 [DSDM-01]. Cette méthode est fondée sur les meilleures pratiques du moment et intègre un cycle d’amélioration tenant compte des leçons apprises. Le DSDM fournit un canevas couvrant l’ensemble du cycle de développement du produit logiciel. La méthode agile DSDM est axée surtout sur les personnes, le processus et la technologie. Le processus est supporté par un ensemble de technique de gestion du processus qui est fourni par la méthode; de plus la méthode définit des rôles et responsabilités pour les personnes impliquées dans le projet mais ne prescrit aucune technique qui doit être utilisée. L’objectif de cette nouvelle méthode de développement de logiciel est de fournir des produits logiciels dans un délai fixe (et court) avec des besoins imprécis qui peuvent être précisés d’une manière itérative.

Le DSDM propose des principes méthodologiques. Ces principes, au nombre de neuf (9), constituent le fondement de la méthode. Ces principes sont:

- 1) L’implication active des utilisateurs est indispensable [DSDM-01].
- 2) Pouvoir de décision des équipes DSDM [DSDM-01].

- 3) Livraison fréquente de produits [DSDM-01].
- 4) L'adéquation aux besoins est le critère essentiel pour que les produits livrés soient acceptés [DSDM-01].
- 5) Il est nécessaire d'utiliser un développement itératif et incrémental pour obtenir une solution adaptée aux besoins [DSDM-01].
- 6) Toutes les modifications effectuées au cours du développement sont réversibles [DSDM-01].
- 7) Les besoins sont définis à un niveau global [DSDM-01].
- 8) Les tests sont intégrés à toutes les étapes du cycle de vie [DSDM-01].
- 9) Une approche basée sur la collaboration et la coopération entre toutes les personnes intéressées par le projet est essentielle [DSDM-01].

Les principes numéros 1, 3, 5 et 8 ont contribué à élaborer un processus de développement logiciel ayant sept phases (voir figure 1.1) :

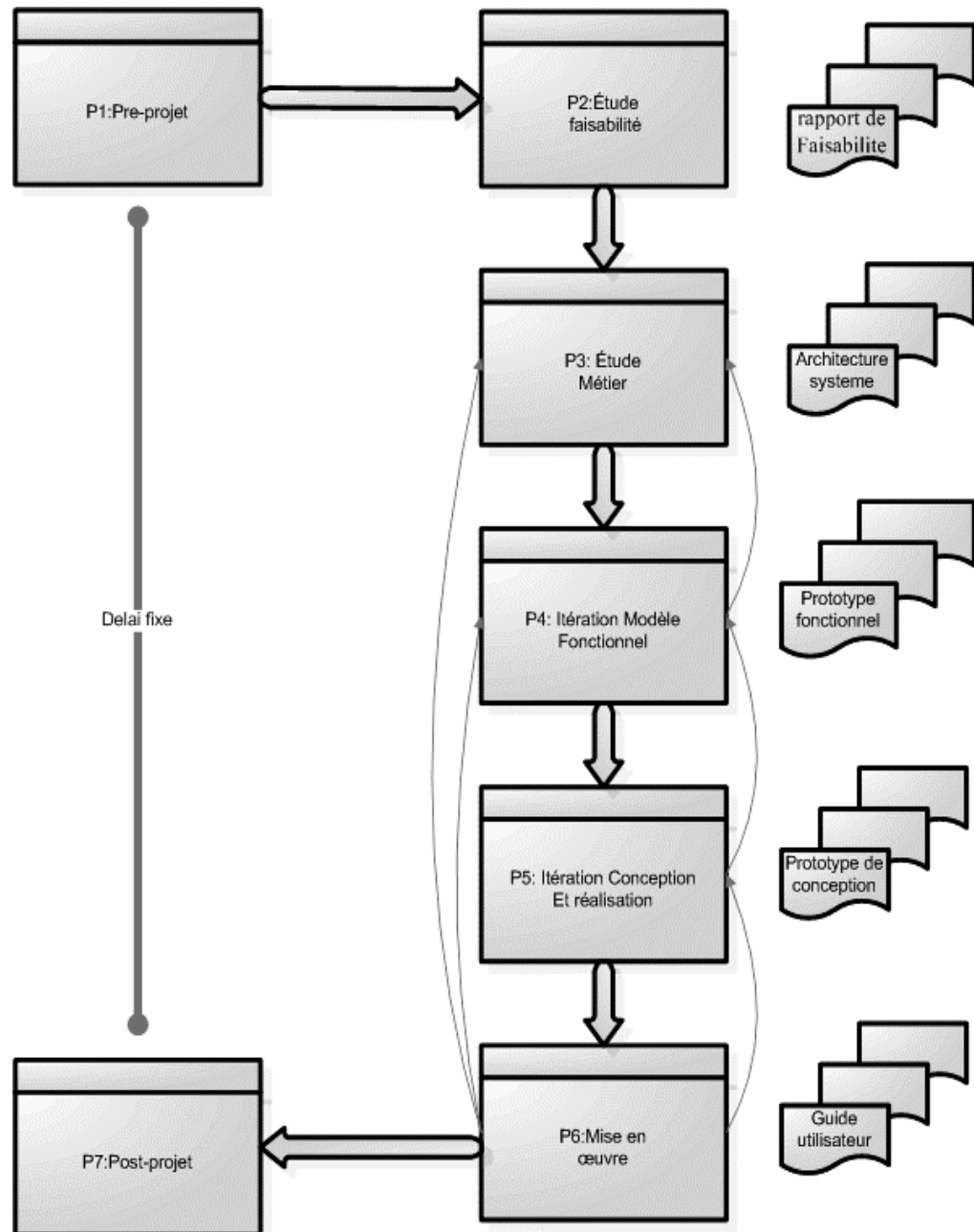


Figure 1.1: Processus de développement logiciel DSDM adapté

Chaque phase a un objectif et produit un certain nombre de livrables qui sont résumés dans le tableau ci-dessous (Tableau 1.1)

Phase du projet	Livrable produit
<p>P1 :Pré-projet</p> <p>Comment les besoins du projet seront identifiés et par qui?</p>	<p>Non formalisé par DSDM</p>
<p>P2 : Étude de faisabilité</p> <ul style="list-style-type: none"> • S’assurer du retour sur investissement. • Adapter le processus de développement. 	<p>Rapport de faisabilité.</p> <p>Faisabilité du prototype (optionnel).</p> <p>Plan global.</p>
<p>P3 : Étude métier</p> <ul style="list-style-type: none"> • Domaine d’application du besoin d’affaire. • Les intervenants du projet. • Prioriser les exigences. • Évaluer le risque. • Exigences non fonctionnelles et maintenance du produit. 	<p>Définition du domaine d’activité industrielle.</p> <p>Liste par ordre de priorité des besoins.</p> <p>Plan de développement.</p> <p>Définition de l’architecture du système.</p>
<p>P4 : Itération du modèle fonctionnel</p> <ul style="list-style-type: none"> • Définition des spécifications du système logiciel (fonctionnel et non fonctionnel). • Planification du test d’acceptation. 	<p>Modèle fonctionnel.</p> <p>Prototypes fonctionnels.</p> <p>Listes des besoins non fonctionnels.</p> <p>Enregistrement de révision du modèle fonctionnel.</p> <p>Plan de mise en œuvre.</p> <p>Rapport d’analyse du risque de développement.</p>
<p>P5 : Itération, conception et réalisation</p> <ul style="list-style-type: none"> • Conception du système logiciel. • Options techniques du système. 	<p>Conception de prototypes.</p> <p>Enregistrement des modifications aux prototypes.</p> <p>Systèmes testés.</p>

	Enregistrement des tests.
P6 : Mise en œuvre <ul style="list-style-type: none"> • Déployer une version du logiciel. • Former les utilisateurs et le personnel de support. • Éliciter les nouveaux besoins. 	Documentation pour les utilisateurs. Population d'utilisateurs formés. Fourniture du système. Document de capitalisation du projet.
P7 : Post-projet Garder le système en production opérationnel Améliorer le processus utilisé	rapport de capitalisation du projet

Tableau 1.1: Livrable du projet DSDM

Le DSDM définit douze (12) rôles (avec responsabilités) qui contribuent au succès du projet. Le DSDM restreint chaque équipe de développement à six personnes. Ces personnes peuvent jouer un ou plusieurs rôles. Ces rôles sont : *Executive Sponsor, Visionary, Ambassador User, Advisor User, Project Manager, Technical Co-ordinator, Team Leader, Developer, Tester, Scribe, Facilitator, Specialist Roles*. [DSDM-01]

Le DSDM, version 4.2, traite également de la maintenance du logiciel. La méthode propose de regrouper les réparations non urgentes et les améliorations sous forme de projet et de les réaliser en suivant le processus de développement présenté [DSDM-01] ci-dessus (Tableau 1.1). Cette étape constitue cependant une autre itération du système. Pour chaque changement à apporter dans le système, il faut évaluer la possibilité de réécrire le changement en fonction du coût et du temps. La méthode DSDM prétend réduire le coût de la maintenance car elle est prise en compte très tôt dans le cycle de vie du produit logiciel. La méthode DSDM a déjà prouvé son efficacité dans des domaines comme le prototypage du processus, la gestion de changement et la nouvelle culture dans l'entreprise [DSDM-01].

La méthodologie DSDM fournit un cadre de gestion de projet et un processus de développement qui tente de couvrir toutes les étapes de réalisation d'un logiciel. Elle propose des techniques, comme le « timeboxing », qui tentent d'aider à mieux contrôler le projet de développement. Elle propose aussi les « règles MoSCoW » qui visent à aider à prioriser les besoins du client. Cependant DSDM ne fournit pas de technique pour supporter le développement logiciel.

DSDM est surtout utilisé pour le développement accéléré d'application (RAD) mais peut aussi être utilisé comme une démarche pour l'adaptation d'un processus dans une entreprise. En effet Nancy R [NVD-01] a intégré grâce au DSDM, la méthode SQUARE (Security Quality Requirements Engineering) dans une entreprise de développement. Cette entreprise a déjà prouvé que l'intégration des nouveaux processus comme SQUARE représentait un défi. Ainsi cette nouvelle expérience guide à la réussite de cette intégration avec les méthodes agiles notamment le DSDM [NVD-01].

1.1.2 SCRUM

Tout comme DSDM, la méthode SCRUM est une méthode agile. Introduit en 1995 par Ken Schwaber et Jeff Sutherland, la méthode SCRUM propose un cycle de vie de développement logiciel en trois phases (voir tableau 1.2). Cette méthode propose de découper un projet de développement logiciel en versions appelées « releases ». Chaque release ou version est subdivisée en itérations appelées « sprints » qui ont d'une durée maximale de 30 jours.

Le tableau (Tableau 1.2) décrit objectivement le processus proposé par la méthode Scrum.

Phases	Processus	Objectif
Pregame Phase	Planning	<ul style="list-style-type: none"> • Définir la liste de fonctionnalités par priorité (backlog) • Comprendre les différents risques • Estimer l'effort et les ressources nécessaires • Établir un calendrier du projet
	architecture	<ul style="list-style-type: none"> • Comprendre le domaine • Définir l'architecture du système • Mettre à jour le backlog
Development Phase	Sprint planning	<ul style="list-style-type: none"> • Établir l'objectif du sprint • Assigner le sprint à une équipe
	Sprint Development	<ul style="list-style-type: none"> • Analyser, concevoir, réaliser les exigences fixées dans l'objectif du sprint (délai=30 jours) • Assurer et contrôler les activités du sprint (réunion quotidienne de 15 mn par jour) • Maintenir la motivation de l'équipe • Résoudre les problèmes rapidement
	Sprint Review (révision du sprint)	<p>Vérifier que les objectifs ont été atteints</p> <p>Identifier les nouveaux besoins</p> <p>Mettre à jour l'architecture du système</p>

Postgame Phase	Intégrer les incréments des différents sprints Réaliser les tests système et de recette Élaborer la documentation pour les utilisateurs Préparer la formation

Tableau 1.2: Objectifs du processus Scrum

La figure 1.2 décrit les différentes interactions entre les différents processus définis dans le tableau 1.2.

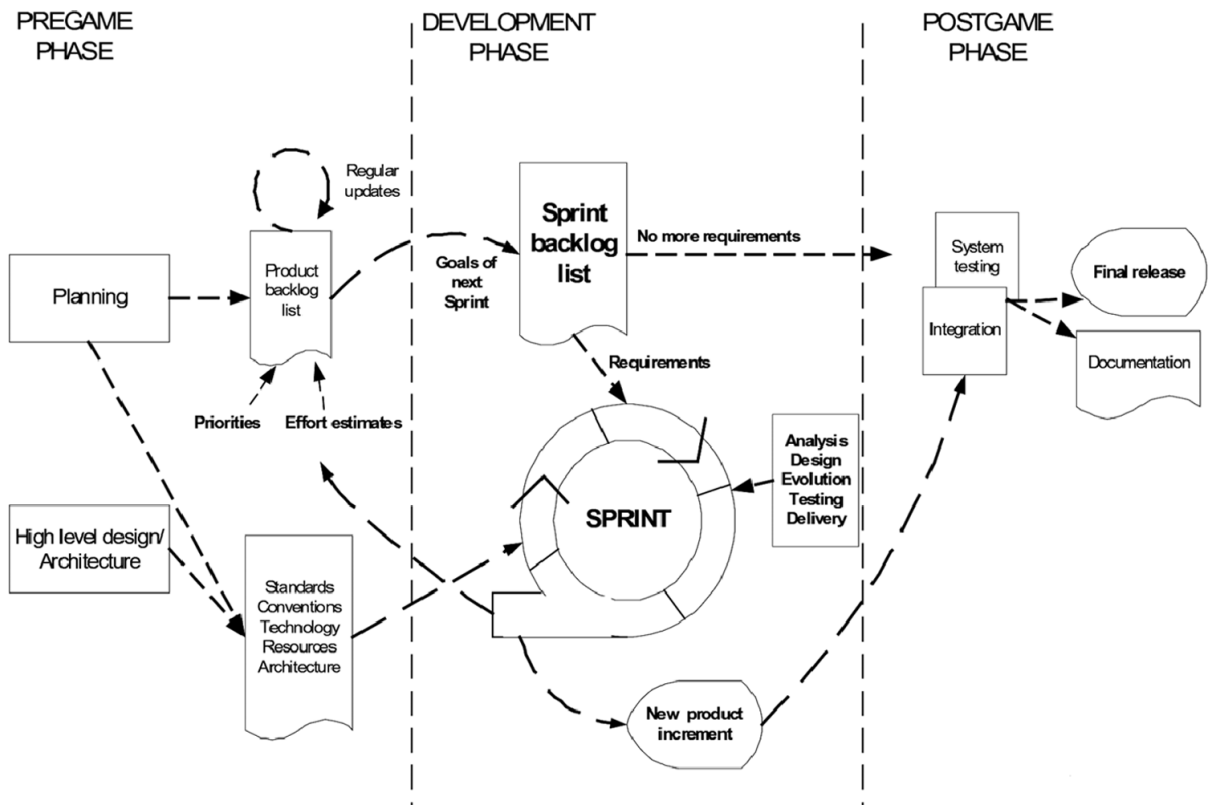


Figure 1.2: Processus Scrum

Extrait de [ARA-01]

La méthode Scrum définit trois acteurs qui contribuent à la réalisation d'un logiciel. Le **propriétaire du produit** (Product Owner), qui représente le client dans le processus. Le client connaît les besoins d'affaires, il définit et priorise aussi les besoins fonctionnels du produit. Le rôle du client est essentiel pour le succès d'un projet Scrum. Le deuxième acteur est le **Scrum Master**, dont le rôle consiste à optimiser la capacité de production de l'équipe dont il est membre. Le Scrum Master est responsable de la bonne marche du processus Scrum. Enfin, le troisième acteur est l'**équipe**. L'équipe est de petite taille pouvant compter jusqu'à 9 personnes et est responsable du développement du logiciel. Les rôles ne sont pas définis spécifiquement dans l'équipe.

Le processus Scrum définit trois livrables :

- 1) le « Product backlog » ou la liste des fonctionnalités du produit.
- 2) le « sprint backlog » ou une partie de la liste des fonctionnalités du produit à réaliser dans un sprint.
- 3) le « BurnDown Chart ».

Le livrable product backlog est un document essentiel au processus Scrum qui décrit toutes les fonctionnalités du produit. Il est sous la responsabilité du propriétaire du produit. Dans ce document, l'effort estimé pour développer les fonctionnalités est défini, les fonctionnalités sont priorisées et décrites sous forme de cas d'utilisation simplifiés, appelés « user stories ». Lors d'une planification de sprint, le propriétaire du produit et l'équipe de développement élaborent un sprint backlog qui contient la liste des fonctionnalités à développer pendant le sprint. Le sprint backlog est une version simplifiée du product backlog. Il contient des fonctionnalités à réaliser lors du sprint. Enfin le BurnDown Chart qui permet de représenter graphiquement l'évolution du projet.

La méthodologie Scrum fournit un cadre de gestion de projet. Son processus de développement ne couvre pas la conception, le codage, les tests unitaires. De plus elle ne fournit pas de techniques pour ces mêmes étapes.

1.1.3 XP : extrême programming

Introduit en 1994 par Kent Beck, la méthode ‘extrême programming’ est aussi une méthode agile. Dans l’extrême programming, la réalisation d’un produit logiciel passe par six phases (voir figure 1.3).

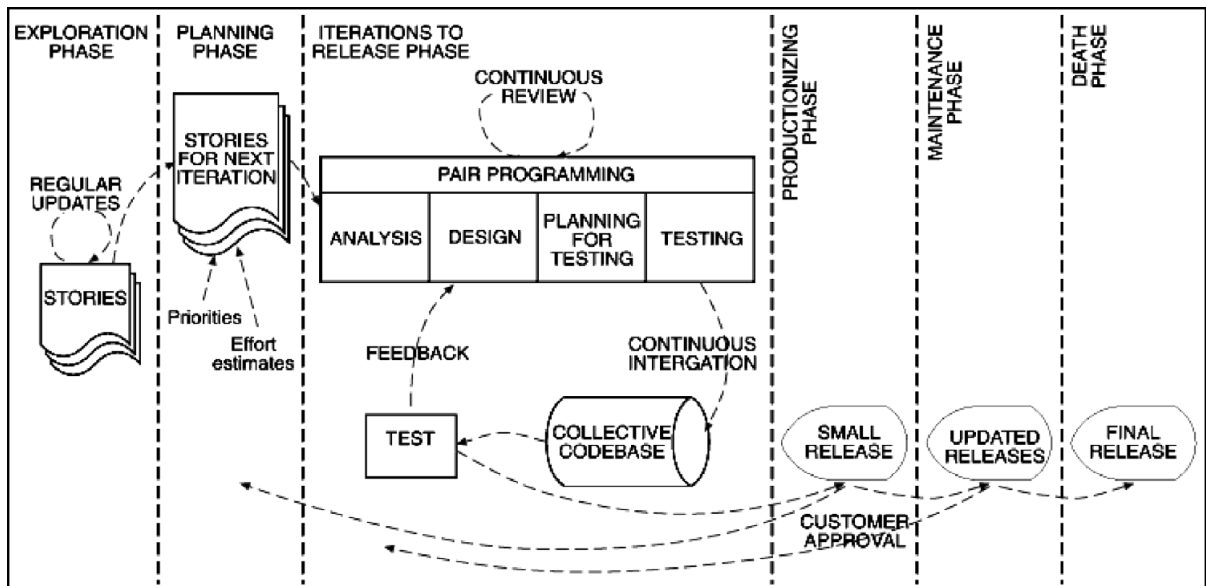


Figure 1.3: Vue d'ensemble du processus XP

Extrait de [Ara-01]

La phase d’exploration consiste à élaborer les exigences du logiciel et à déterminer l’architecture globale du système. La phase de planification se focalise sur l’estimation et la priorisation des exigences logicielles identifiées précédemment. Elle établit un calendrier pour la première version. La phase de développement itérative assure le développement des versions du logiciel en respectant les règles et pratiques fournies par le processus XP.

Le tableau ci-dessous (tableau 1.3) décrit ces règles et pratiques de chaque étape de cette phase.

Étapes	Règles et pratiques
PLANIFICATION	<ul style="list-style-type: none"> • Écrire les cas d'utilisation (user stories) • Établir un calendrier des versions (releases) • Faire fréquemment des petites releases • Mesurer la vélocité du projet • Diviser le projet en itérations • Au début de chaque itération, favoriser la circulation de l'information en déplaçant les gens au sein du projet • Faire une réunion chaque jour • Adapter le processus XP
CONCEPTION	<ul style="list-style-type: none"> • Simplicité • Choisir un système de métaphore • Utiliser les CRC cards (Class Responsibilities and Collaboration) pour le design • Créer un programme simple pour explorer les solutions et réduire les risques • Pas de nouvelles fonctionnalités très tôt • Effectuer le factoring dès que possible

CODAGE	<ul style="list-style-type: none"> • Le client est toujours disponible • Le code doit être formaté suivant les standards de la programmation • Réaliser d’abord les tests unitaires • Tout le code en production a été réalisé à deux • Ne pas intégrer plusieurs parties en même temps • Intégrer souvent • Encourager la collaboration • Laisser l’optimisation du code à la fin • S’en tenir au temps estimé.
TEST	<ul style="list-style-type: none"> • Tout le code doit avoir des tests unitaires • Chaque code doit passer tous les tests avant d'être ajouté à la release • En cas de bug, créer des tests • Faire souvent des tests de conformité aux besoins du client

**Tableau 1.3: Règles et pratiques du XP adaptées
[Beck-01]**

Chaque itération a une durée de 1 à 3 semaines. La phase de production qui correspond à la mise en production d’une version fonctionnelle du logiciel dans l’environnement de l’utilisateur final. Les tests de recette sont réalisés afin de vérifier et valider que le logiciel répond au besoin. À la suite de cette phase, le logiciel passe en phase de maintenance. Cette phase permet d’intégrer les fonctionnalités manquantes ou ajoutées lors de la mise en production. La dernière phase du processus XP est la phase de fermeture du projet. Elle correspond à la livraison finale du logiciel et à la révision de la documentation accompagnante.

XP définit plusieurs rôles qui interviennent dans le processus.

- 1) Le *programmeur* écrit le code et les tests unitaires. Il veille à écrire un code simple et fonctionnel en collaboration avec les autres programmeurs [ARA-01].
- 2) Le *client* écrit les users stories (exigences logicielles) et les tests fonctionnels. Il est responsable de la priorisation des exigences logicielles [ARA-01].

- 3) Le « *tracker* » fait un suivi réel des différentes itérations et veille à l'amélioration des estimations futures [ARA-01].
- 4) Le « *coach* » quant à lui est responsable du processus dans tout son ensemble. Il guide à la bonne utilisation du processus.
- 5) Enfin le « *manager* » qui prend des décisions. Il communique régulièrement avec l'équipe pour connaître la situation courante, comprendre les difficultés et les défaillances dans le processus [ARA-01].

La méthodologie XP ne fournit pas de méthode de gestion de projet. Par contre elle fournit un processus et des techniques de développement logiciel comme le « pair programming » [BECK-01], le « code reviews » ou révision du code [BECK-01], qui couvre la planification d'une version du logiciel, la conception, le codage et les tests.

1.2 Représentation des processus

Plusieurs notations existent et sont utilisées pour représenter les processus [SPC-01]. Ces notations diffèrent selon les symboles utilisés [AACL-01], leurs canevas pour définir, capturer et utiliser l'information [SWBK-1]. Ces notations sont peu à peu normalisées et les plus importantes à connaître par un ingénieur logiciel, selon le SWEBOK, sont: Le Diagramme de flux de Yourdon et DeMarco [SWBK-1], le Diagramme d'état [Har-01], le ETVX [Rad85], la méthode OSSAD [OSSAD-04] et plus récemment le BPMN [OMG]. Dans notre recherche aucune comparaison ne sera faite sur ces différentes notations. Nous utiliserons la méthode OSSAD avec l'outil Qualigram.

1.2.1 Qualigram

Qualigram est un outil qui supporte la notation OSSAD [OSSAD-04]. L'outil Qualigram propose « *des concepts fondamentaux qui permettent de reproduire fidèlement, de manière pragmatique l'organisation et les savoirs faire d'une entreprise* » [QUALI-02]. Ces concepts (décrits à l'annexe II) sont les piliers de la notation qui est encadré par: la méthode OSSAD,

le langage graphique, le flux d'information, les pyramides de l'entreprises, la notion de poupées russes, la pyramide Qualigram et sa dynamique.

Le **langage graphique** proposé par Qualigram est simple et opérationnel. Il est soutenu par un vocabulaire graphique, des règles de structurations, une grammaire et une syntaxe. Le vocabulaire graphique de base est très simple et se compose de quatre types de formes de bases (voir figure 1.4). Cette simplicité lui permet d'être utilisable et compréhensible à tous les niveaux de l'entreprise donc adapté pour la cartographie de processus de développement. [QUALI-02]

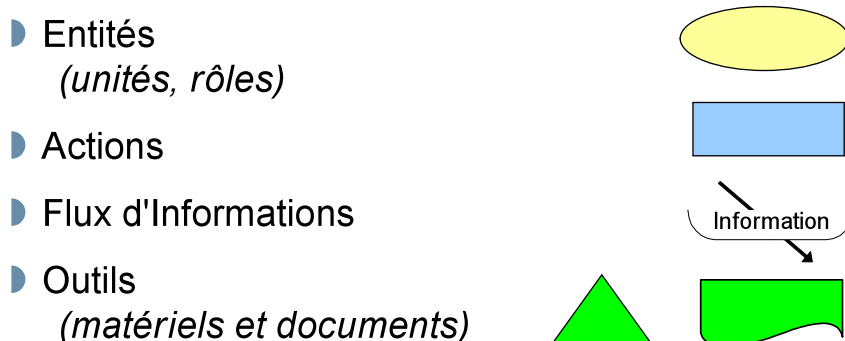


Figure 1.4: Différents types de formes du langage QUALIGRAM

Extrait de [Quali-03]

Avec cette notation, un processus est représenté et interprété comme l'indique la figure 1.5:

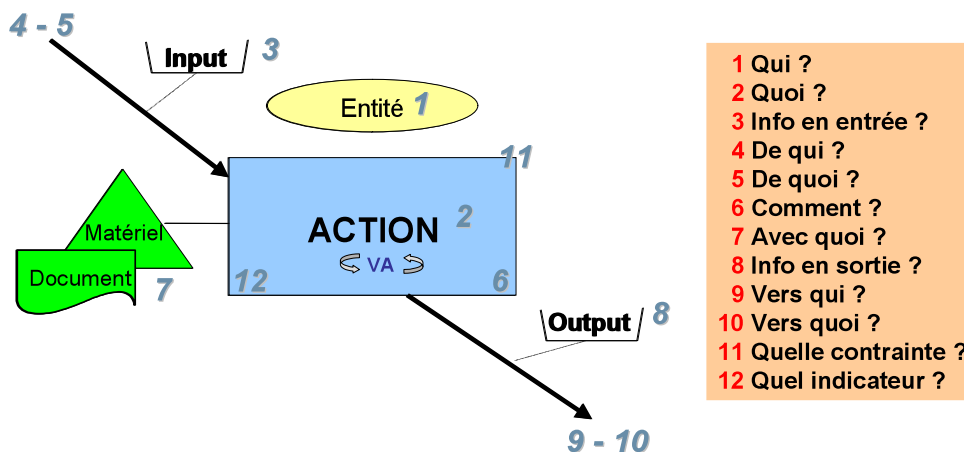


Figure 1.5: Mises en œuvre du langage graphique QUALIGRAM

[Quali-03]

Les éléments 3 et 8, de la figure 1.5, sont identifiés comme des **flux d'informations** et doivent être représentés pour chaque action qui sera décrite. La représentation graphique (aussi appelée cartographie) de Qualigram présente le savoir-faire suivant trois niveaux de détails avec la possibilité de passer d'un niveau à l'autre facilement.

Le **premier niveau de représentation graphique** permet de représenter les processus à un niveau macroscopique. À ce niveau la représentation permet de comprendre pourquoi et vers quoi un processus a été créé. Les règles de structuration pour la cartographie de processus du niveau 1 sont :

- Pas de rôle interne;
- Pas d'outils;
- Pas de chronologie.

Ainsi trois des quatre types de formes de base sont utilisés à l'exception de la forme représentant l'outil (voir figure 1.4).

Le vocabulaire accompagnant les formes de base et permettant la description des processus se compose de :

- L'ovale: utilisé pour représenter une **entité externe**;
- Le rectangle: utilisé pour représenter le **processus**, le **sous processus**, ou une **procédure**;
- Les **flèches d'informations** : utilisées pour représenter les informations échangées entre les ovales et les rectangles.

Grâce à la syntaxe fournie par Qualigram, les processus peuvent être représentés sous quatre formes :

- La cartographie macroscopique;
- La cartographie relationnelle;

- La cartographie détaillée;
- Le processus transversal.

Le **deuxième niveau de représentation graphique** de Qualigram met en exergue le « qui fait quoi ». Ce niveau de description des processus est le niveau que nous utiliserons le plus dans notre projet de recherche. L'objectif principal de ce niveau est de décrire chaque action accompagnée de ses entrées/sorties et ce pour chaque membre de l'équipe de développement. Ce niveau de cartographie possède les règles de structuration suivantes :

- Pas de rôle interne;
- Des moyens principaux (outils matériels et documents);
- Une chronologie verticale des actions.

À ce niveau de détail nous retrouvons les quatre types de formes de base (voir figure 1.4) mais avec un vocabulaire graphique plus vaste. Il en est de même pour la syntaxe. La figure 1.8 présente les formes et la syntaxe utilisée pour la description des procédures à ce niveau [QUALI-06].

Le troisième niveau de représentation graphique permet de représenter les instructions de travail. Ce niveau ne sera pas utilisé dans notre projet de recherche. En effet l'instruction de travail est le dernier niveau de la pyramide documentaire Qualigram. Une instruction de travail permet de connaître exactement «comment réaliser une tâche ou une activité particulière, assignée à un rôle clairement identifié». Sa représentation se réalise toujours avec les mêmes formes de base mais avec un vocabulaire graphique aussi plus large. Les règles de structurations sont :

- Un seul rôle pour une instruction;
- Tous les moyens nécessaires sont représentés;
- Les opérations s'enchaînent de façon chronologique.

Grâce à la notion de « Zoom », Qualigram permet de créer des liens dynamiques (hypertexte) entre tous les trois niveaux de représentations graphiques d'un même processus. La figure 1.6 ci-dessous décrit ces relations.

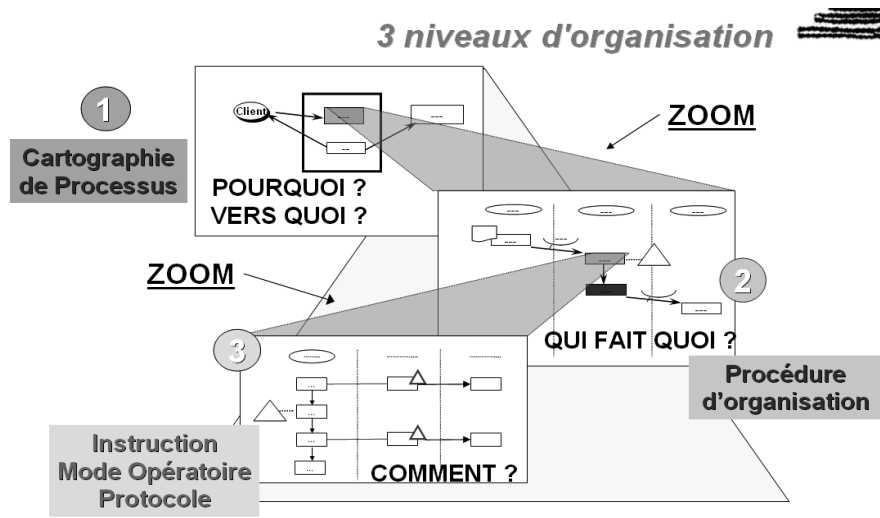


Figure 1.6: Notion zoom sur la pyramide Qualigram

Pour réaliser un projet de cartographie de processus, Qualigram propose deux démarches : une première adaptée à la refonte des processus et une seconde centrée sur la création de nouveaux processus. Lors d'une refonte de processus, une documentation existe généralement. L'approche Qualigram dans ce cas, suggère de faire un passage progressif du système textuel à un système graphique proposé par la pyramide Qualigram [QUALI-08]. Dans les cas où il n'y a pas de documentation existante, il est recommandé de s'appuyer sur la méthode de structuration à trois niveaux du langage Qualigram. Il est possible de commencer la cartographie par le niveau de notre choix. On peut donc choisir entre :

- **L'approche stratégique:** Elle débute au niveau 1 de la pyramide Qualigram. Elle consiste à identifier les processus principaux de l'entreprise puis ensuite identifier les procédures et les instructions de travail nécessaires pour maîtriser les processus et atteindre leurs objectifs [QUALI-08]. Cette approche est bien en adéquation avec la norme ISO 9000 version 2000 [QUALI-08]. L'inconvénient de cette approche c'est que certains savoir-faire peuvent être oubliés. [QUALI-08]

- **L'approche organisationnelle:** Cette approche débute au niveau 2 de la pyramide Qualigram. Elle consiste à identifier les procédures à décrire à partir des activités et produits de l'entreprise, ou des exigences normatives. Ensuite il est plus simple soit de passer au niveau 1 de la pyramide ou au niveau 3. [QUALI-08].
- **L'approche opérationnelle:** L'approche opérationnelle débute au niveau 3 de la pyramide Qualigram. Elle se concentre sur les opérations par un rôle plutôt que les échanges d'informations entre les différents rôles. Dans le cadre des normes actuelles comme ISO 9000 cette approche s'avère peut satisfaisante car elle trop restrictive [QUALI-08].
- **L'approche combinée :** C'est un mélange des trois approches ci-dessus. Elle nécessite la création de trois équipes distinctes ou chacune d'elle adopte une des approches citées précédemment. Cette approche comporte aussi des risques qui apparaissent surtout lors de la liaison des trois niveaux de la pyramide Qualigram, entraînant de fois un double travail sur l'écriture des procédures [QUALI-08].

Pour supporter le langage graphique, Qualigram fournit une suite d'outils qui aident à représenter les processus selon la méthode OSSAD. L'outil, « QUALIGRAM MANAGER » nous a été fournit gratuitement pour une utilisation dans le cadre de ce projet de recherche.

1.3 Représentation d'un processus avec Qualigram manager

La figure 1.7 ci-dessous est une illustration d'un processus « Écrire un mémoire » avec l'outil Qualigram. Il s'agit d'une représentation au niveau 2 de la pyramide Qualigram.

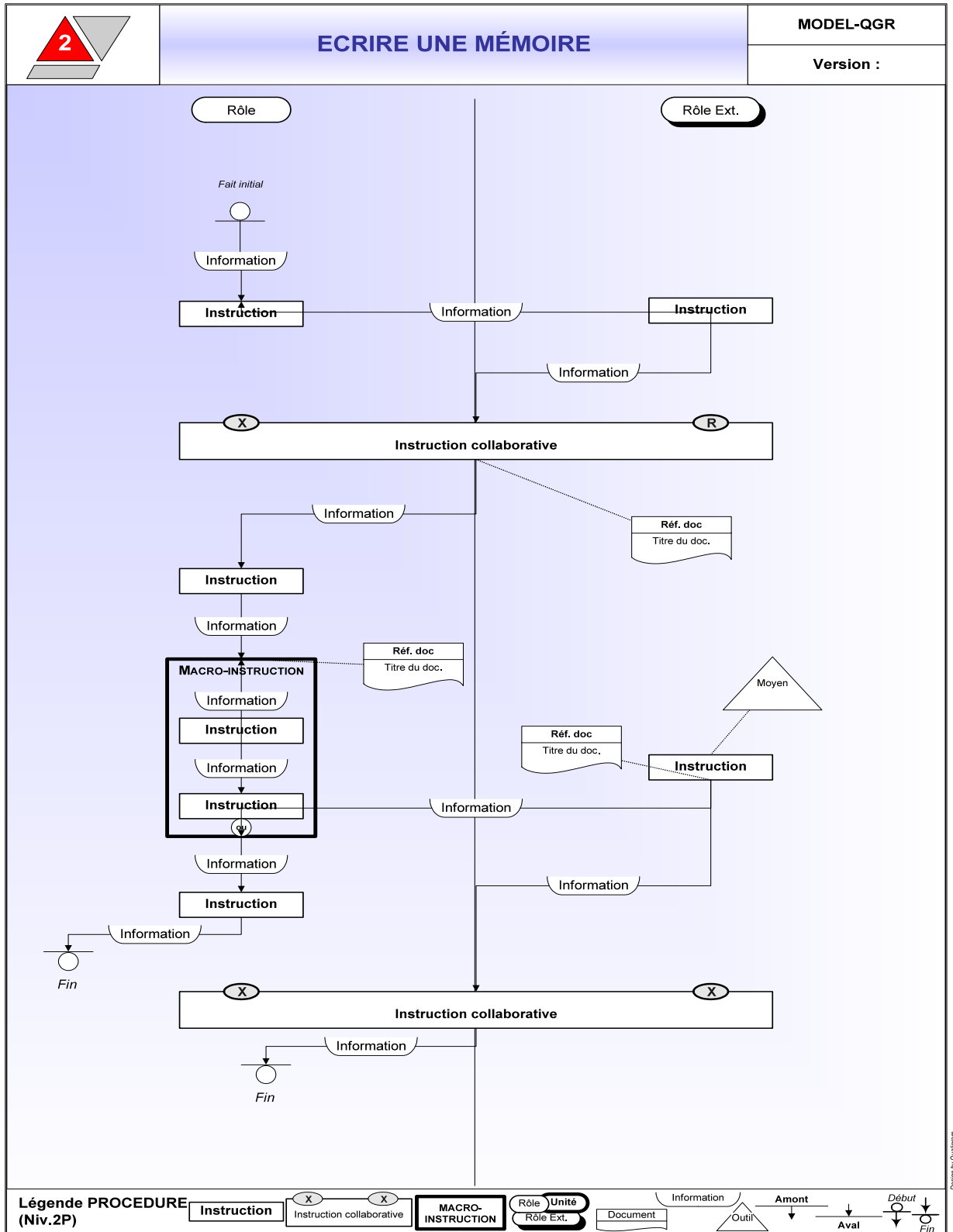


Figure 1.7: représentation d'un processus au niveau 2 avec Qualigram

1.4 Formes proposées par l'outil Qualigram au niveau 1 et 2 de la pyramide

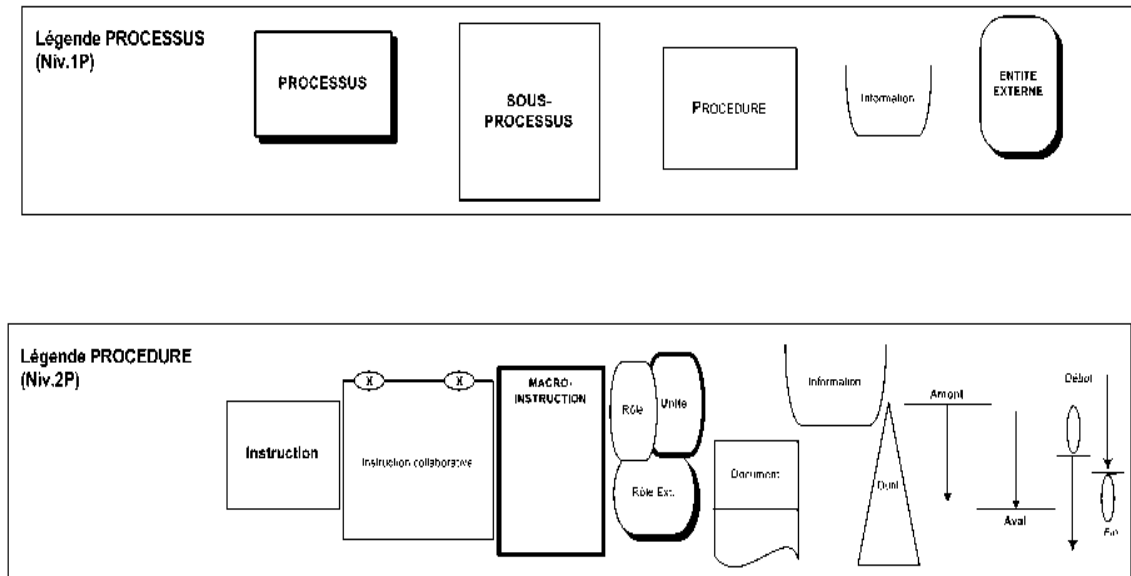


Figure 1.8: Légende utilisé dans la représentation Qualigram

1.5 ISO/IEC 12207

La norme ISO/IEC 12207 est la norme internationale pour la définition des processus et activités du cycle de vie du logiciel. Publié en 2008, ISO/IEC couvre sept processus qui se regroupent en processus spécifique au logiciel et en processus systèmes. Les processus logiciels sont au nombre de trois et comprennent:

- Les **processus d'implémentation** qui définissent les activités liées à la réalisation d'un composant spécifique (ou le logiciel lui-même) pour le logiciel [ISO/IEC12207-1].
- Les **processus de support** qui traitent de la maîtrise et de l'amélioration de la qualité des projets, de la gestion de configuration, de la documentation,... [ISO/IEC12207-1].
- Enfin les **processus de réutilisation** qui aident l'organisation à réutiliser les composants du logiciel. [ISO/IEC12207-1].

Ces différents processus sont définis respectivement dans la section 7.1, 7.2, et 7.3 comme l'indique la figure 1.9 ci-dessous :

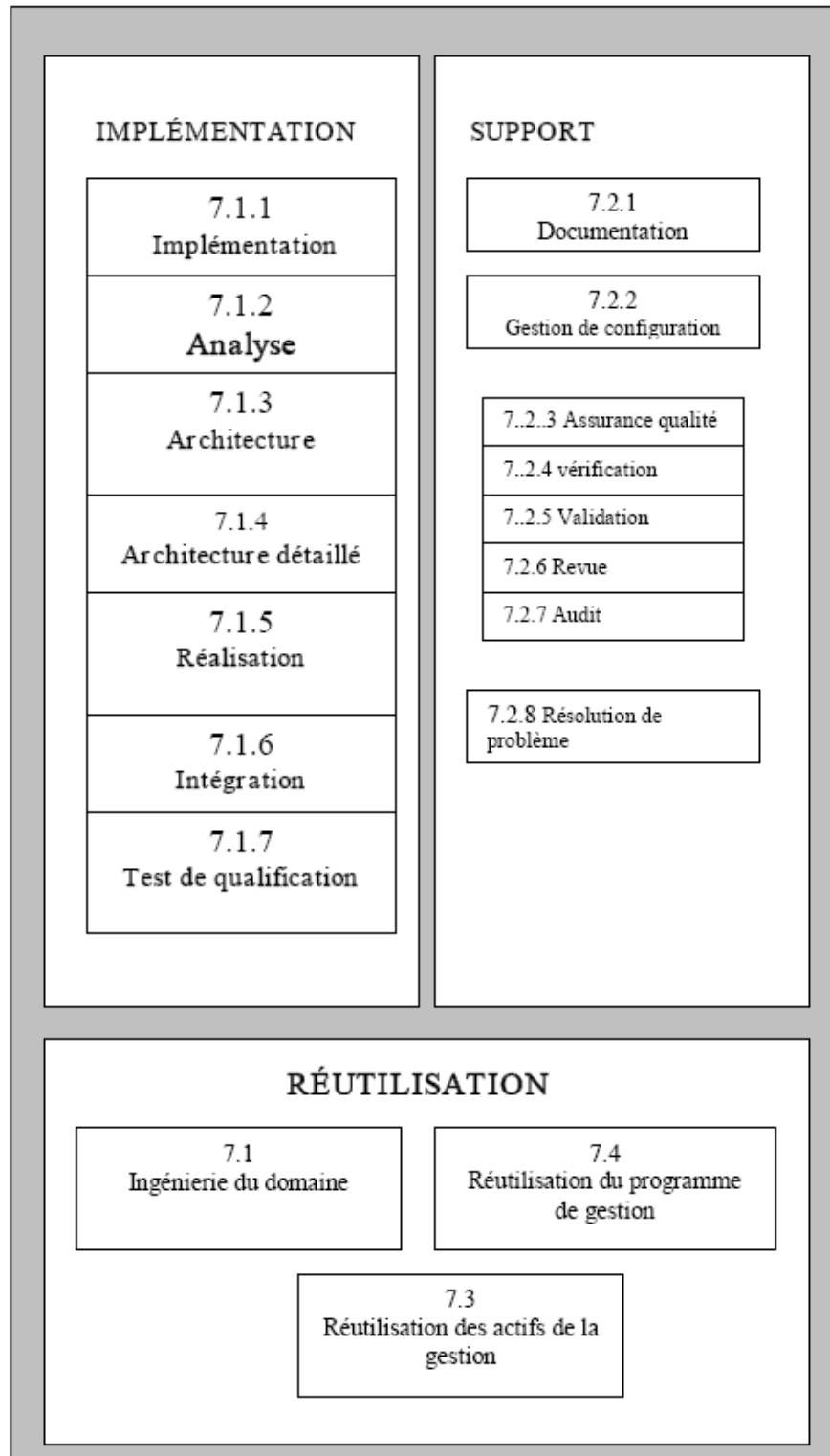


Figure 1.9: Processus logiciels ISO 12207:2008 Adaptés en français
[ISO/IEC12207-1]

Dans la version 2008, des modifications ont été aussi apportées au niveau de la structure du processus. Le processus se décompose en sous processus, puis en activités puis en tâches. Cependant, elle ne donne toujours pas d'indication sur la mise en place des activités et des tâches de chacun des processus, ni de modèle particulier de cycle de vie [ISO/ IEC12207-3].

Dans notre projet nous nous limiterons au processus de développement décrit dans la section 7.1 de la norme. Le processus de développement comprend les sous processus suivants:

1. Implémentation ou mise en œuvre du processus
2. Analyse des exigences du logiciel
3. Conception de l'architecture du logiciel
4. Conception détaillée du logiciel
5. Codage du logiciel
6. Intégration du logiciel
7. Essais de qualification du logiciel

Les processus systèmes ont été séparés des processus logiciels dans la version 2008. La norme ISO 12207 définit un seul rôle générique qui intervient dans tous ces sous processus. Il s'agit du rôle « développeur ». Dans une entreprise, le développeur peut être un analyste, un architecte, un programmeur,...

La norme a connu d'énorme succès dans les grandes entreprises et continue d'être la référence pour les standards de développement émergents. En effet le groupe 24 de M. Claude Laporte se réfère toujours à cette norme pour l'adapter à de très petites entreprises [CL-01].

1.6 ISO 9001:2000

ISO 9001:2000 est la norme « *qui définit les exigences pour les systèmes de management de la qualité* » [ISO9001-1]. C'est une norme générique applicable à toute les sociétés quelles qu'en soient le type, la taille et le produit fourni. Elle remplace les normes ISO 9001:1994 [ISO9001-3], ISO 9002:1994 [ISO9001-4] et ISO 9003:1994 [ISO9001-5] et couvre quatre

axes à savoir la responsabilité de la direction [ISO9001-2], la gestion des ressources [ISO9001-2], la gestion des processus [ISO9001-2] et enfin la mesure, l'analyse et l'amélioration continue [ISO9001-2]. L'axe orienté gestion des processus couvre également les processus de conception et de développement définis comme «un ensemble de processus qui transforme des exigences en caractéristiques spécifiées ou en spécifications d'un produit, d'un processus ou d'un système » [ISO9001-5].

La figure 1.10 décrit les différentes relations entre les différents processus ISO 9001:2000. Un accent est mis plus sur le processus de réalisation.

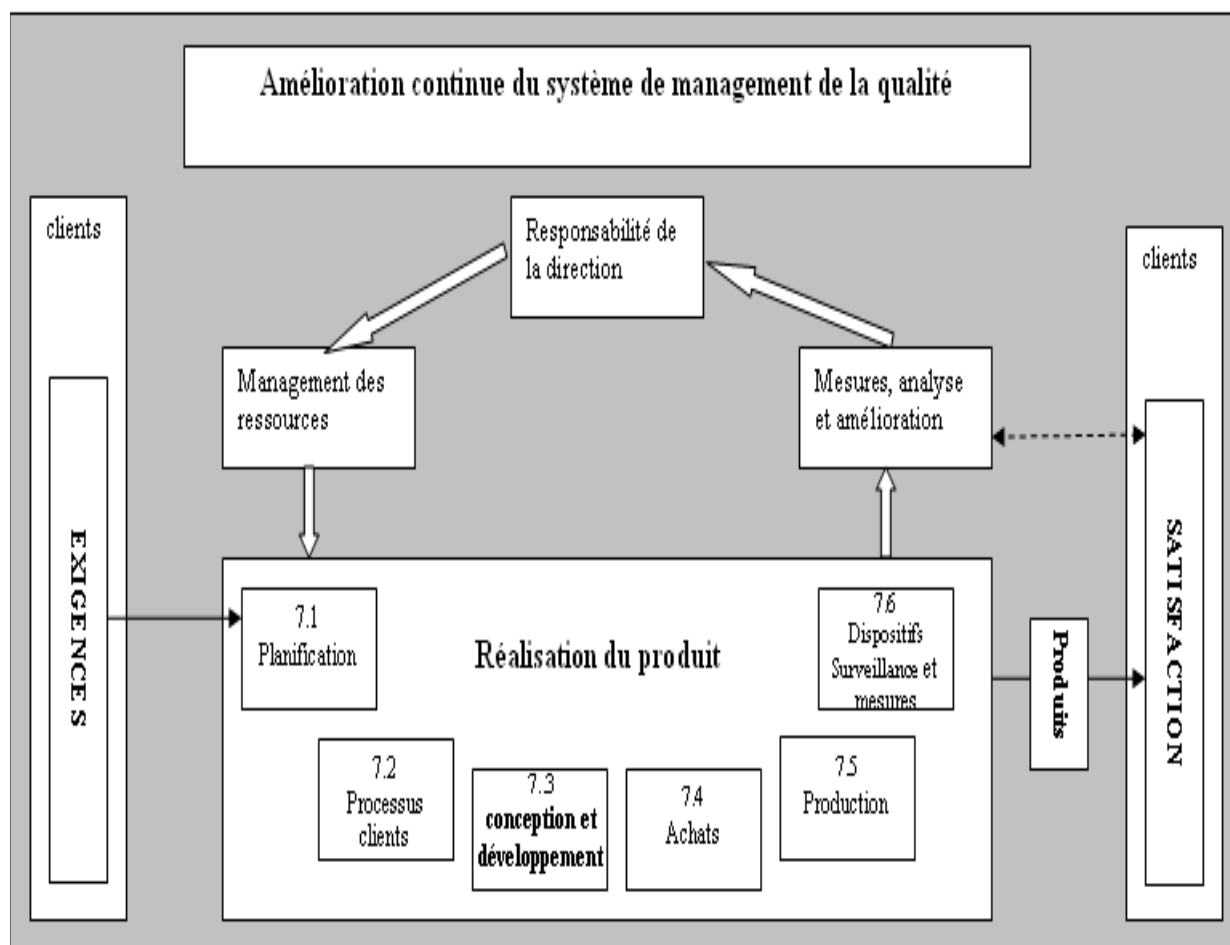


Figure 1.10:Processus ISO 9001:2000 -processus de réalisation et les autres processus

Le processus de conception et de développement mis en relief sur la figure 1.10, n'est pas spécifique aux produits logiciels. C'est un processus générique qui s'applique à n'importe quel produit autre que logiciel. Cependant une application plus spécifique de la norme ISO 9001:2000 sur les produits logiciels est guidée par la norme ISO 90003 et est facilitée par la norme ISO 12207 [ISO9001-7].

ISO 9001 n'est pas le modèle de référence pour décrire les processus de développement et de maintenance des produits logiciels, mais un modèle de qualité auquel le produit doit répondre [ISO9001-2]. Dans le domaine logiciel, ISO 9001 a pour rôle de fournir des orientations sur les points à prendre sur la mise en œuvre de l'ISO / IEC 12207[ISO9001-2].

1.7 Conclusion

DSDM est une méthodologie agile de développement logiciel. Il divise le développement logiciel en sept phases à savoir, le pré-projet, l'étude de faisabilité, l'étude du domaine d'affaire, le modèle fonctionnel itératif, la conception et la réalisation itérative, la mise en œuvre et enfin le post-projet. DSDM définit les douze (12) rôles avec des responsabilités qui contribuent au succès du projet. Chaque équipe doit compter au plus six personnes. Chaque phase du processus de développement libère un certain nombre de produits livrables ce qui peut augmenter le temps de réalisation du logiciel et rendre le processus assez fastidieux. DSDM fournit un processus complet de gestion de projet, un processus de développement logiciel mais aucune technique pour réaliser le développement logiciel. Cependant cette insuffisance peut se combler en l'utilisant conjointement avec XP. La méthode DSDM peut être déployée de façon modulaire car certaines de ses pratiques peuvent être utilisées sans les autres. Cette modularité lui offre un avantage car le déploiement d'un processus en entreprise doit se faire de façon graduelle. Cependant la méthode n'est pas très connue en Amérique du nord car peu d'articles dans les bases de données comme « ACM Digital Library », ou « IEEE Xplore » en font de DSDM leur sujet principal.

Scrum est aussi une méthodologie agile de plus en plus connue dans le domaine du développement logiciel. Elle définit un cycle de vie du logiciel plus court que DSDM composé de trois phases. La première phase couvre la planification et l'architecture du logiciel. La deuxième phase est la phase de sprint ou phase de développement. Enfin la troisième phase couvre l'intégration et les tests. Scrum définit également trois rôles principaux à savoir le directeur de produit, le Scrum Master, et l'équipe. Scrum fournit un processus de gestion de projet et un processus de développement qui couvre les différentes phases de la réalisation d'un logiciel à partir des exigences jusqu'à l'intégration du système. Par contre Scrum ne fournit aucune aide ni pour le concept du logiciel, ni pour les tests d'acceptations contrairement à DSDM. Scrum fournit des techniques qui aident à la définition des exigences logicielles, responsables de la plupart des échecs dans les projets de

développement logiciel. Pour la phase de conception, de codage, des tests unitaires, Scrum ne fournit aucune technique appropriée.

L'extrême programming (XP) fournit six phases pour la réalisation d'un produit logiciel. Contrairement à DSDM et Scrum, XP n'est pas une méthode de gestion de projet mais plutôt un processus de développement. Il fournit des techniques pour assurer la conception, le codage, les tests d'un logiciel. L'un des avantages du XP est d'amener l'utilisateur à écrire ses besoins. Malheureusement, un grand nombre d'entreprises ne sont pas prêtes à réaliser cette tâche et attendent toujours que le fournisseur réécrive les besoins. Ces trois méthodes agiles décrites ont pour objectif de fournir des logiciels de qualité en conformité avec les exigences de qualité de ISO 9001:2000.

ISO 9001:2000 est une norme qui fournit les exigences nécessaires pour le processus de management de qualité. La norme est indépendante du domaine d'activité mais aussi de la taille de l'organisation. Elle n'est pas suffisante pour définir les processus de développement logiciel. Cette insuffisance peut être couverte avec la norme ISO 12207 qui aide à la définition des processus et des activités du cycle de vie du logiciel. La mise à jour de la norme 12207 en 2008 permet d'avoir une structure du processus semblable à la décomposition fournie par l'outil de Qualigram. Un processus se décompose en activités ou procédures, qui se décomposent en actions ou instructions de travail. Bien que la norme ait été améliorée, elle reste lourde pour les fournisseurs de logiciel web qui adoptent de plus en plus une approche agile. Ces entreprises pour faciliter le transfert de connaissance utilise des outils de modélisations connus pour représenter leurs savoir faire comme Qualigram. Qualigram n'est pas seulement un outil mais aussi une technologie de modélisation des processus qui possède son langage et sa méthodologie. Elle offre quatre approches de formalisation des processus basées sur les trois niveaux de la pyramide Qualigram. L'approche stratégique qui permet de cartographier les processus en débutant du niveau 1 vers le niveau 3 sera une démarche que nous adopterons dans ce projet.

CHAPITRE 2

PÉRIMÈTRE DE LA SITUATION ACTUELLE

TELUS Solutions d’Affaires nous a donné une opportunité de réaliser notre recherche au sein d’un groupe de TI. Il s’agit de l’équipe du middleware actuellement assignée au projet de Aéroplan. Le groupe comprend six personnes et chacun a manifesté un intérêt pour ce projet. TELUS Solution d’Affaires est une unité de TELUS qui possède une certification ISO 9001. Le groupe utilise également un cadre de gestion de projet appelé « cadre de gestion de projet de TELUS agilité».

Ce chapitre décrit le groupe de TELUS Solutions d’Affaires, la portée de la certification ISO 9001 et son processus de gestion de projet.

2.1 Présentation de l’entreprise : TELUS Solutions d’Affaires (TSA)

TELUS est une entreprise de télécommunication canadienne dont l’organisation consiste en quatre unités d’exploitation axées sur la clientèle [TSA]. Ce sont, TELUS Solutions Consommateurs, TELUS Solutions d’Affaires (TSA), TELUS Québec, TELUS Solutions Partenariats. Ces unités offrent des services de données filaires et sans fil, et des services voix et divertissement aux ménages et aux particuliers partout au Canada. D’autres services comme des services de technologies de l’information (TI) et d’affaires électroniques sont offert en plus par TSA à de grandes entreprises multinationales et autres, et à d’importants clients du secteur public. En effet, TSA était une société privée qui offrait des services technologiques avant d’être détenue à 100 % depuis juin 2000 par TELUS. Cette appartenance permet à TSA de mettre au point pour ses clients des solutions qui bénéficient de toutes les expertises de la famille TELUS (télécommunications, mobilité, hébergement, etc.) et d’offrir ses services spécifiques à l’ensemble du marché canadien, grâce à son lien privilégié avec le reste de la famille TELUS. Dans le cadre de notre projet, TSA se limite

uniquement à l'équipe de développement logiciel qui offre des services liés aux produits logiciels.

2.2 TELUS Solutions d'Affaires et ISO 9001

TELUS détient et compte maintenir sa certification ISO 9001. TELUS Solutions d'Affaires a instauré des processus de gestion de la qualité, capables d'évoluer selon les besoins du marché et contribuant à la satisfaction de sa clientèle. Son système de qualité répond aux critères de la norme ISO 9001:2000 et englobe le cadre de gestion de projet de TELUS agilité (voir 2.3) [TSA3]. La certification délivrée à TSA s'étend de la période du 8 août 2000 au 5 juillet 2011. Elle couvre les activités de gestion de projet encadrant les services conseils dans l'élaboration des solutions de technologie de l'information. Elle a été réalisée pour le siège social à Québec au 330 st-vallier et à Montréal au 11 Duke.

2.2.1 Méthodologie : cadre de gestion de projet de TELUS Agilité

TELUS Solutions d'Affaires a développé sa propre méthodologie de gestion de projet intitulée « cadre de gestion de projet TELUS Agilité ». Cette méthodologie est générique et est utilisée dans toutes les unités TELUS que ce soit pour faire du développement logiciel ou pour implanter des solutions de télécommunications. C'est une méthodologie qui s'inspire directement des concepts de l'agilité plaçant le client au centre du projet. La méthodologie offre des activités aidant à identifier les objectifs du projet, les échéanciers, les budgets enfin de s'assurer qu'ils seront respectés. De plus la méthodologie permet d'identifier, d'encadrer et de gérer les risques liés au projet. Le chargé de projet a la responsabilité d'en assurer le bon déroulement dans la structure présentée dans la figure 2.1.

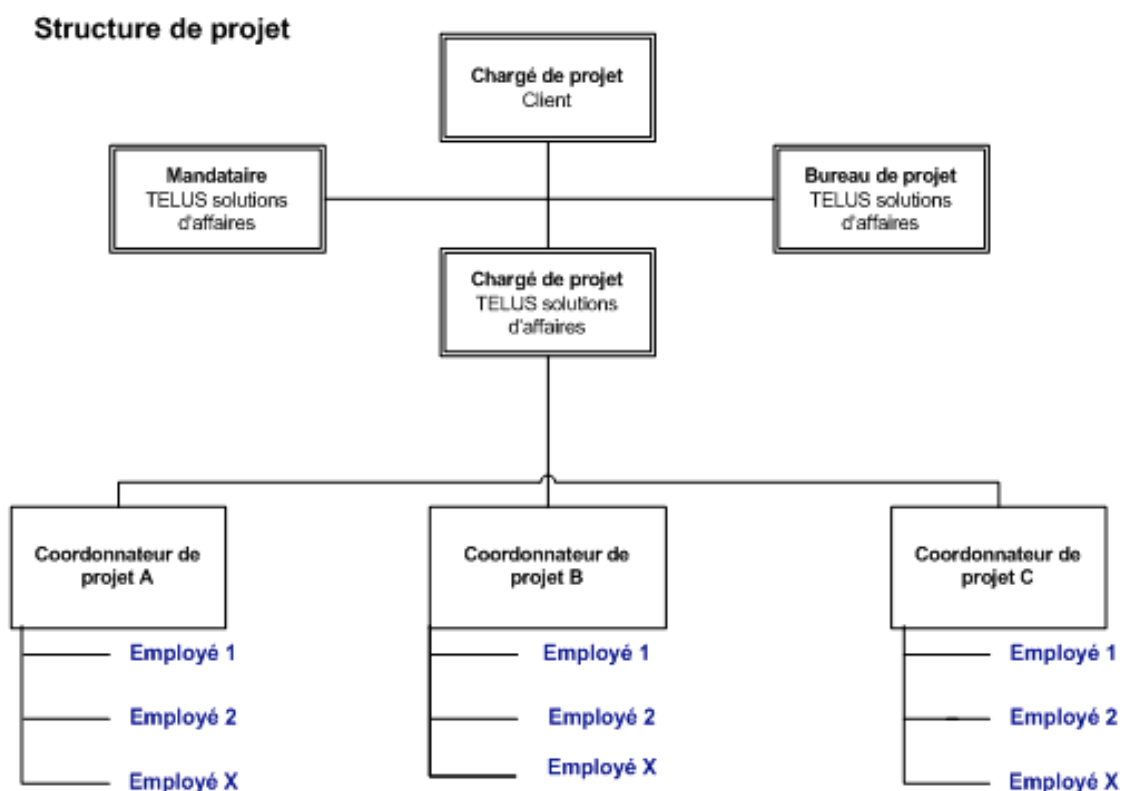


Figure 2.1: Structure du projet dans le cadre de gestion de projet TELUS Agilité
Extrait de [TSA3]

La méthodologie définit plusieurs étapes qui sont des étapes essentielles pour assurer un succès du projet. Ces étapes sont présentées dans la section 2.2.2 [TSA3].

2.2.2 Cycle de vie d'un projet de TSA

Le cycle de vie d'un projet dans le cadre de TELUS Agilité consiste en quatre étapes opérationnelles et une cinquième qui correspond plus au suivi du projet. La figure 2.2 décrit ces étapes.

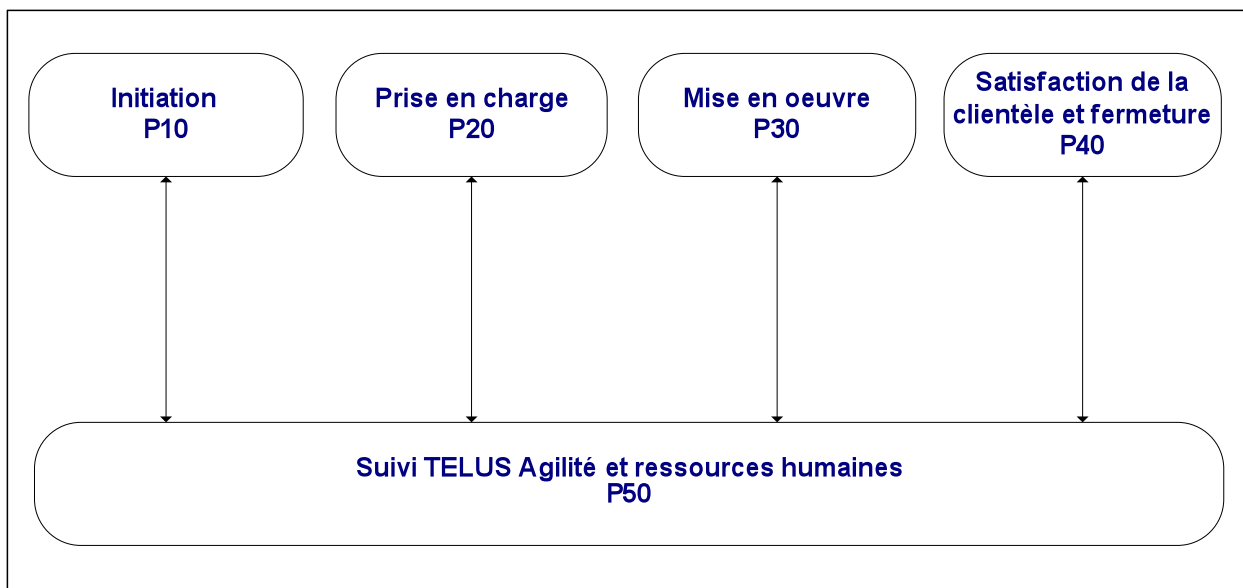


Figure 2.2: Cycle de vie de projet dans le cadre de gestion de projet de TELUS Agilité
Extrait de [TSA-01]

Étape 1: Initiation (P10)

Elle consiste au démarrage du projet en partenariat avec le client. Lors de cette étape une proposition détaillée reflétant notre compréhension du mandat et exposant les coûts associés aux différentes étapes de sa réalisation, est présentée au client. L'équipe des ventes en collaboration avec les analystes et les experts des domaines impliqués, s'occupe de cette étape et est responsable de la livraison d'une proposition d'affaires [TSA3].

Étape 2: Prise en charge du projet (P20)

Cette étape commence par une rencontre de démarrage du projet réunissant le chargé de projet et les représentants du client. Cette rencontre vise à définir plus précisément les besoins et les attentes des deux parties, la portée des travaux, le détail des livrables, les facteurs de risques, les critères de succès, les rôles et les responsabilités de l'équipe de projet ainsi que les niveaux d'approbation. Lors de cette étape, le manuel d'organisation de projet (MOP) et la planification sont élaborés et soumis pour approbation avant le lancement de projet. Une fois l'approbation obtenue, l'équipe est réunie et le projet démarre. Lors de cette

étape, le chargé de projet, en collaboration avec le bureau de projet, détermine de quelle façon le cadre de gestion de projet sera appliqué [TSA3].

Étape 3: Mise en œuvre (P30)

Cette étape vise à réaliser le projet, tel que prévu et défini à l'étape 2, en appliquant ce qui est prévu au cadre de gestion de projet [TSA3].

Étape 4: Satisfaction de la clientèle et fermeture (P40)

Cette étape s'effectue après la livraison d'un projet au client. Elle vise à s'assurer que tout est à la satisfaction de celui-ci, conformément à l'entente conclue entre les deux parties et de la fermeture du projet conformément au cadre de gestion. Dans le cas des projets, cette étape permet au directeur de compte de contacter ou de rencontrer le client après la fin des travaux afin de s'assurer de sa satisfaction. Si des problèmes notables sont identifiés par le client, le directeur de compte s'assurera de les formuler au mandataire du projet. Les informations sont ensuite colligées et un bilan statistique sera dressé dans le but d'en faire rapport à la prochaine revue de direction. Cette approche permet de s'assurer que le cadre de gestion de projet répond toujours aux besoins de la clientèle.

Étape 5 (Support): Suivi TELUS Agilité et ressources humaines (P50)

Cette étape n'est pas une étape de gestion de projet. Elle sert à supporter le suivi et l'évolution du cadre de gestion et à suivre l'évolution des ressources, tant au niveau de leur carrière que de leur formation.

2.2.3 Processus actuels

Dans ce projet nous nous intéressons uniquement au processus P30 ou processus de mise en œuvre. Les autres processus sont présentés en annexe II. La figure 2.3 présente plus en détail le processus de mise en œuvre [TSA3].

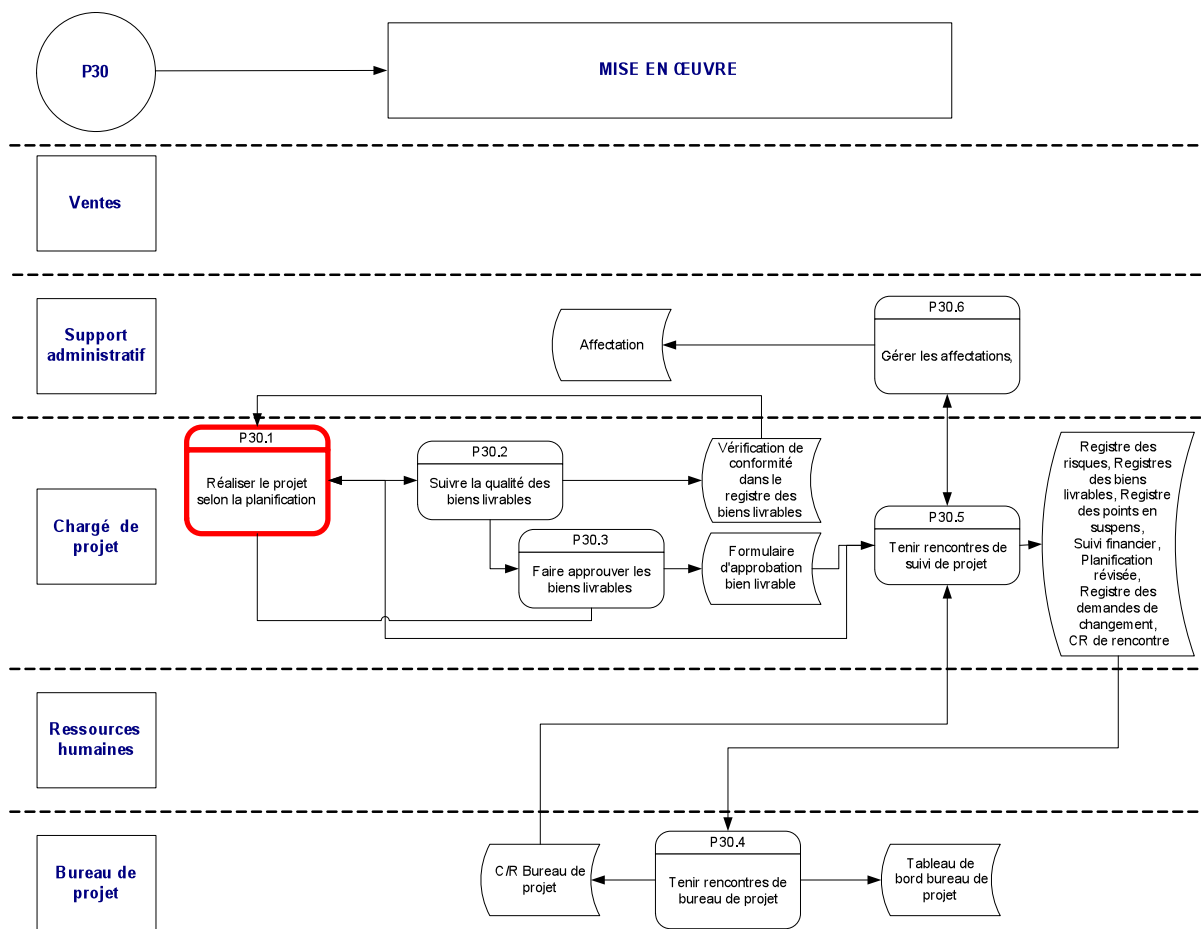


Figure 2.3: Processus de mise en œuvre

Extrait et modifié de [TSA3]

Comme nous le constatons à la figure 2.3, le processus actuel de mise en œuvre s'adresse uniquement au chargé de projet. De plus le processus est générique et est utilisé dans toutes les unités TELUS que ce soit pour faire du développement logiciel ou pour implanter des solutions de télécommunications. Il serait utile d'avoir plus de précision au niveau du processus P30.1: «réaliser le projet selon la planification» pour des projets de nature différente. Du point de vue du développement logiciel, la représentation actuelle du processus P30.1 est superficielle et ne fournit pas de processus de gestion de projet logiciel, ni de processus et d'activités précises de développement de logiciel.

2.3 Forces et faiblesses de TSA

TSA est une entreprise en pleine croissance caractérisée par de petites équipes dynamiques et motivées. La direction est ouverte aux suggestions d'amélioration mais compte tenu des délais de livraisons, elle n'alloue pas suffisamment de temps pour les projets d'amélioration. Les processus de développement sont à l'état embryonnaire ce qui laisse place aux mauvaises pratiques de développement, une caractéristique des entreprises de faible niveau de maturité [SEI].

2.4 Objectifs de la recherche

Notre recherche consiste à proposer une définition du processus P30.1 pour la gestion de projet logiciel. Nous allons compléter ce processus en décrivant également le processus de développement du logiciel d'une manière plus détaillée. Nous décrirons toutes les activités effectuées actuellement au sein de l'équipe TI et y ajouteront les meilleures pratiques de la méthode Scrum. Notre objectif à plus long terme est d'en arriver à sensibiliser la direction à l'amélioration de ses processus logiciels.

Une représentation plus formelle de ce processus permettra aussi d'atteindre certains objectifs d'amélioration à court terme. L'expérience vécue dans l'entreprise, précisément au sein de mon équipe, permet aussi de fixer comme objectifs à court terme:

- Une diminution des reprises de travail « rework»: Il arrive dans certaines situations que des changements surviennent dans le projet logiciel sans que la documentation ne soit mise à jour. En conséquence, la fonctionnalité implémentée par le développeur est bien différente de celle attendue par le testeur qui se base sur la documentation pour effectuer son travail;
- Une amélioration de la communication et des délais de livraison: Plusieurs fois dans divers projets, des employés ont été obligés de faire des heures supplémentaires pour livrer les projets à temps. Des problèmes non signalés (comme les règles d'affaires mal spécifiées, une analyse d'impact mal effectuée, une incompréhension du système) sont toujours à l'origine de ces problèmes;
- Une amélioration de la formation des nouveaux employés: TSA est une entreprise en pleine croissance dans le domaine du développement logiciel. L'entreprise embauche de plus en plus des nouveaux développeurs. Cependant certaines technologies rares utilisées dans certains projets augmentent la courbe d'apprentissage des nouveaux employés.

Comme procédure de mesure de l'amélioration de la situation actuelle du projet, nous développerons un questionnaire pour mesurer si les améliorations apportées par la formalisation des processus ont porté fruit.

2.5 Conclusion

TSA est une entreprise en pleine croissance caractérisée par de petites équipes dynamiques et motivées. TSA possède une certification ISO 9001:2000 pour son processus de gestion de projet. Cependant au niveau développement logiciel, les processus sont toujours à l'état embryonnaire. En effet, son processus de gestion est un processus générique et incomplet par rapport au processus de développement logiciel offert par la méthode SCRUM, DSDM, ou ISO 12207. TSA est ouverte aux suggestions d'amélioration, d'où une opportunité pour nous de proposer un processus complet de développement logiciel. Dans ce projet notre solution

s'appliquera à une seule équipe. Un succès du projet sera récompensé par une diminution des remises en fabrication (reworks), une amélioration de la communication et des délais de livraison, une amélioration de la formation des nouveaux employés. Le succès sera mesuré de façon intangible à l'aide d'un formulaire qui sera conçu pour permettre d'apprécier les changements apportés par notre projet.

CHAPITRE 3

PROCESSUS APPLICABLE ET DÉMARCHE DU PROJET

3.1 Introduction

Dans le chapitre précédent, nous avons montré qu'il n'existe aucun formalisme pour les processus de développement au sein de TSA. Il nous est donc possible de s'inspirer des processus existants pour bâtir le nôtre. Ainsi la revue de littérature réalisée dans le chapitre I montre que la tendance actuelle des entreprises de développement logiciel web s'oriente de plus en plus vers les méthodologies agiles particulièrement la méthode Scrum. Cette méthode n'offre pas un processus complet de développement et doit se combiner à XP par exemple pour couvrir le processus de gestion de projet, de développement et les techniques liées au développement. Bien que la combinaison Scrum - XP ait déjà connu un succès en entreprise, il faudrait l'adapter dans le contexte de TSA. Pour cette adaptation nous nous sommes inspirés de l'approche stratégique décrite en annexe (annexe I). La première étape consiste:

- À analyser le processus Scrum/XP dans le contexte de TSA. Cette analyse permettra d'identifier les processus applicables à TSA.
- À cartographier les processus identifiés au niveau 1 de la pyramide Qualigram.
- À faire une cartographie au niveau 2 de la pyramide Qualigram.

3.2 Analyse des processus Scrum et XP dans le contexte TSA

Lors de notre étude, nous avons constaté que la méthode Scrum est plus utilisée surtout en Amérique du nord. Il existe des certifications relatives à la méthode, ce qui constitue une forte valeur ajoutée. Combiné avec XP, Scrum offre un processus complet tant au niveau des processus de gestion de projet, qu'au niveau des processus et des techniques de développement logiciel. Un sondage réalisé au sein de TSA laisse apparaître des craintes sur la technologie XP:

- Selon certains, la méthode XP est plus considérée comme une bonne pratique pour les entreprises de jeux vidéo ou de logiciel très critique, pas pour les fournisseurs de logiciel web.
- Le directeur chargé des technologies de l'équipe de AÉROPLAN au sein de TSA affirme que XP reste une bonne pratique mais lors de l'intégration finale du produit, beaucoup de problèmes majeurs surviennent. Par conséquent, il n'est donc pas prêt à approuver une telle technologie dans son équipe actuelle avec le nombre de ressources actuelles.
- Les techniques du « pair programming » seront difficiles à adopter vu que nos délais sont courts et que le principe de mettre un développeur et un testeur ensemble est perçu comme un rallongement du temps de développement.

Ainsi nous avons choisi d'implémenter quelques processus de gestion de projet de Scrum qui peuvent s'incorporer au sein d'une équipe technique. Le processus de développement de Scrum sera complété avec les savoirs faire actuel de l'équipe. La norme ISO 12207 nous servira à cet effet. De plus ISO 12207 nous permettra de formaliser les termes à utilisés dans le processus.

Nous avons évalué qu'il existe des risques liés à l'intégration de Scrum au sein du groupe de TI bien que ceux-ci soient motivés. En effet Scrum est un nouveau concept pour l'équipe qui diffère de notre approche actuelle. Pour augmenter nos chances de succès nous allons adopter une démarche itérative et participative. Les membres de l'équipe seront consultés à travers de petites rencontres formelles ou pas. Les nouveaux processus devraient contenir plus de similarité avec les façons actuelles de travail de l'équipe. Des ajustements seront apportés au fil du temps pour arriver à une solution parfaite.

3.3 Solution de cartographie de processus de développement logiciel

Pour cartographier les processus, nous avons adopté la démarche stratégique proposée par Qualigram. Cette démarche de cartographie des processus commence par le niveau 1 de la pyramide Qualigram. À ce premier niveau le travail principal consiste à déterminer les

processus principaux de l'équipe de développement. Une fois tous les processus principaux identifiés, chaque processus pourra être détaillé au niveau 2 de la pyramide Qualigram. Au niveau 2, toutes les procédures, les rôles, les activités et les entrées/sorties devront être identifiés.

3.3.1 Étape 1: Niveau 1 Qualigram

La cartographie des processus au niveau 1 de la pyramide Qualigram, permet de mettre en évidence ce qui doit être fait et pourquoi. Par une technique de **collecte de donnée**, qui s'est réalisée soit par :

- des séances d'interviews de 30 maximums avec les membres de l'équipe de TI.
- l'étude de la documentation existante.

Nous avons identifié des processus de base parmi des activités fournies par Scrum, qui s'applique à l'équipe de développement. Cependant la qualité des données dépend de l'expérience de l'intervenant.

Des changements majeurs sont à noter au niveau de la fréquence d'exécution de certains processus Scrum. En effet certaines activités planifiées par jour de travail, ont été révisées pour une fréquence d'une semaine. Ce choix est surtout pour ne pas apporter des changements brusques sur les habitudes de l'équipe. Quand aux autres activités, elles seront intégrées graduellement au fil du temps.

La figure 3.1 décrit les processus de développement du logiciel inspiré de Scrum, et de nos pratiques actuelles.