

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PROJET DE 15 CRÉDITS PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN TECHNOLOGIE DE L'INFORMATION
M.Sc.

PAR
SERGE BRIÈRE

RÉINGÉNIERIE DU LOGICIEL : REVUE DU DOMAINE ET APPLICATION

MONTRÉAL, LE 12 OCTOBRE 2006

© Tous droits réservés par Serge Brière

CE PROJET A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE :

Dr. François Coallier, président du jury
Département de génie logiciel et des technologies de l'information

Dr. Alain April, directeur de projet
Département de génie logiciel et des technologies de l'information

IL A FAIT L'OBJET D'UNE PRÉSENTATION DEVANT JURY ET PUBLIC
LE 12 OCTOBRE 2006
À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

RÉINGÉNIÉRIE DU LOGICIEL : REVUE DU DOMAINE ET APPLICATION

Serge Brière

SOMMAIRE

De nos jours, les systèmes d'information se transforment rapidement en systèmes patrimoniaux et la nécessité de les adapter exerce une pression accrue sur les informaticiens parce que la nature des entreprises évolue sans cesse et les technologies de l'information croissent également à une vitesse fulgurante. Le défi de faire évoluer le système d'information à la vitesse de ces changements dans l'entreprise doit être supporté par une approche intégrée du processus d'affaires et de la réingénierie du logiciel.

Dans un premier temps, ce projet se veut une revue de la littérature sur le domaine de la réingénierie du logiciel. Cette revue couvre les aspects de l'historique, de la taxonomie, des modèles, des cycles de vie, de la maintenance et des risques. Dans un deuxième temps, cette revue de la théorie sur la réingénierie du logiciel est préparatoire pour son application à un projet. Dans ce projet, on utilise la réingénierie du logiciel, mais en s'appuyant sur celle de réingénierie du processus d'affaires. Pour y arriver, le cadre d'analyse des systèmes d'information centré sur le système de travail et ses composants : client, produit et service, les participants, l'information et la technologie, est utilisé. Cette approche ne fait que rendre conforme le travail de réingénierie du logiciel qui se fait généralement en combinaison avec la réingénierie du processus d'affaires qu'il soutient.

SOFTWARE RE-ENGINEERING: DOMAIN REVIEW AND APPLICATION

Serge Brière

ABSTRACT

Because of the fast change of nature of the companies and growth of information technologies, the information systems quickly become patrimonial systems for which the need for adapting them causes an additional pressure on the programmers. The challenge to make evolve the information system at the speed of these changes in the company must be supported by an integrated approach of the business process and the software reengineering.

This project starts initially as a review of the literature on the field of the software reengineering. It covers the aspects of the history, the taxonomy, the models, cycles of life, the maintenance and the risks. This review of the theory on the software reengineering is preparatory for its application to a project. In this project, we use the software reengineering, but supported by the business process. To make this possible, we use the analysis of the information systems centred on the work system framework and his components: customers, products and services, business process, participants, information, technology, context and infrastructure. This approach makes in conformity the work of software reengineering which is generally done in combination to the business process that it supports.

REMERCIEMENTS

Entreprendre ce travail de recherche alors que je vis éloigné des grands centres urbains demande de la part des gens qui m'entourent une très grande flexibilité.

J'aimerais d'abord remercier M. Alain April, mon directeur du projet, pour son aide précieuse et pour toute la souplesse dont il a fait preuve. Merci pour ces rencontres qui me donnaient un second souffle. Je tiens à remercier Cascades pour m'avoir permis de faire cette recherche dans le cadre de mon horaire de travail et pour m'avoir donné accès à toute l'information nécessaire. Un grand merci à M^{me} Isabelle Côté, adjointe aux ressources humaines, pour avoir répondu avec générosité à toutes mes questions. Je ne voudrais pas oublier de remercier mon atout le plus précieux, merci à Hélène, ma compagne, merci de partager ma vie, de tous ces moments d'efforts malgré ton handicap et merci pour ton humilité : tu m'encourages au quotidien, tu es un modèle de persévérance et je me trouve bien petit à tes côtés. Merci à ma fille, Marianne, qui a pris le relais et s'est occupée de sa mère pendant mes absences ces dernières années et qui malgré son jeune âge est souvent plus sage que moi... merci ma grande. Finalement, merci à mes beaux-parents Claire et Jérémie, et à ma belle-sœur, Sylvie, merci pour l'aide généreusement offerte et à mes parents, André et Solange, qui m'avez donné toutes ces valeurs et qui m'avez jamais rien demandé en retour, sauf de les retransmettre.

TABLE DES MATIÈRES

	Page
SOMMAIRE.....	i
ABSTRACT.....	ii
REMERCIEMENTS.....	iii
TABLE DES MATIÈRES.....	iv
LISTE DES TABLEAUX.....	viii
LISTE DES FIGURES.....	ix
LISTE DES ABRÉVIATIONS ET SIGLES.....	x
INTRODUCTION.....	1
CHAPITRE 3 REVUE DE LA LITTÉRATURE.....	3
1.1 Historique.....	3
1.1.1 Systèmes patrimoniaux.....	4
1.2 Taxonomie sur la réingénierie.....	5
1.3 Modèle de la réingénierie du logiciel.....	10
1.3.1 Le versant gauche (l'ascension).....	12
1.3.1.1 La rétro-ingénierie.....	12
1.3.2 Le sommet.....	15
1.3.3 Le versant droit (la descente).....	16
1.3.3.1 Ingénierie du logiciel.....	16
1.4 Modèles de cycle de vie du logiciel.....	17
1.4.1 Modèle de cycle de vie en cascade.....	18
1.4.2 Modèle de cycle de vie en V.....	19
1.4.3 Modèle de cycle de vie par extensions successives.....	20
1.4.4 Modèle de cycle de vie incrémental.....	21
1.4.5 Modèle de cycle de vie par prototype.....	22
1.4.6 Modèle de cycle de vie en spirale de Boehm.....	23
1.4.7 Modèle de cycle de vie par objet.....	24
1.5 Maintenance logicielle.....	25
1.5.1 Catégories de maintenance.....	26
1.5.2 Classification de la maintenance logicielle par critère.....	26
1.5.3 Mini cycle de la maintenance.....	27
1.5.4 Critères de maintenabilité.....	28
1.5.5 Qualité du logiciel.....	30
1.6 Les risques liés à la réingénierie.....	34
1.7 Brève perspective.....	38
1.8 Conclusion du chapitre 1.....	39

Serge Brière

Comment:

CHAPITRE 2	RÉINGÉNIERIE APPLIQUÉE À UN PROJET	41
2.0	Présentation du processus	41
2.1	Approche méthodologique.....	44
2.2	Réingénierie du processus d'affaires	45
2.2.1	Description de l'entreprise.....	45
2.2.2	Contexte	46
2.2.3	Description simplifiée du processus	46
2.2.4	Réingénierie de la gestion des cartes de temps.....	47
2.2.4.1	Explication du problème.....	48
2.2.4.2	Évaluation des compétences	49
2.2.4.3	Autres facteurs reliés au problème	51
2.2.4.4	Responsabilités du département informatique.....	53
2.2.5	Solution.....	53
2.2.5.1	Buts.....	54
2.2.5.2	Évaluation des activités	55
2.2.5.3	Détails de la solution proposée.....	56
2.2.5.4	Investissement et les économies réalisées	59
2.2.6	Risques reliés au projet.....	59
2.2.7	Impacts d'un échec	60
2.3	Rétro-ingénierie du logiciel des gestions des cartes de temps.....	60
2.3.1	Méthodologie de la rétro-ingénierie	61
2.3.2	Phase de la rétro-ingénierie	61
2.3.2.1	Outils disponibles pour le recouvrement de la conception	62
2.3.2.2	Activités de la méthodologie de recouvrement de la conception	63
2.3.2.3	Observations sur le code source	63
2.3.2.4	Observations sur les imprimés d'écrans	65
2.3.2.5	Observations sur le recensement des rapports.....	65
2.3.2.6	Observations des objets contenus dans la librairie d'exploitation.....	66
2.3.2.7	Reconception du diagramme de flux de données	66
2.3.2.8	Reconception du diagramme du modèle relationnel de données.....	66
2.3.2.9	Reconception du diagramme du modèle conceptuel de données	67
2.3.2.10	Analyse de la carte de temps	67
2.3.2.11	Analyse de la structure des fichiers	68
2.3.2.12	Analyse du code source sur les programmes.....	70
2.3.3	Sommaire de la rétro-ingénierie	70
2.3.3.1	Niveau de gestion des cartes de temps	71
2.4	Modèle des cas d'utilisation	73
2.5	La technologie	76
2.5.1	Choix technologique.....	76
2.5.2	La conception.....	78
2.5.2.1	Liste des biens livrables pour la conception détaillée	78
2.5.2.2	Observations à la suite à la conception détaillée	78
2.6	Descriptions des écrans.....	79

2.6.1	Description de la fonction des écrans	80
2.6.2	Impression des rapports	81
2.7	Conclusion du chapitre 2	81
CHAPITRE 3	DESCRIPTION DE L'EXPÉRIMENTATION	83
3.1	Sommaire du projet	83
3.1.1	Ordre de réalisation des tâches pour le projet.....	84
3.2	Critique de la théorie de la rétro-ingénierie	86
3.2.2	Ce qui a été utile pour la rétro-ingénierie	86
3.2.2.1	Les activités de la rétro-ingénierie à valeur ajoutée	86
3.2.2.2	Techniques utiles pour la phase d'implémentation	87
3.2.3	Les techniques de rétro-ingénierie les moins utiles.....	87
3.2.3.1	Phase d'analyse.....	87
3.2.3.2	Phase d'implémentation.....	88
3.2.4	Ce qui aurait été utile pour la rétro-ingénierie.....	89
3.2.4.1	Ce qui aurait été utile pour la phase d'analyse	89
3.2.4.2	Ce qui aurait été utile pour la phase d'implémentation	89
3.3	Comparaison des éléments techniques	90
3.4	Questionnaire et évaluation des résultats.....	91
3.5	Améliorations futures	92
3.5.1	Amélioration du modèle de réingénierie	93
3.5.2	Amélioration de l'application WEB des CT.....	93
3.6	Conclusion du chapitre 3	94
CONCLUSION	96
ANNEXE 1	Description de la carte de temps	97
ANNEXE 2	Écran de saisie d'une carte de temps	100
ANNEXE 3	Description des phases du processus sur les cartes de temps	102
ANNEXE 4	Instantanée du système de travail des cartes de temps	105
ANNEXE 5	Logigramme sur le système de travail des cartes de temps	105
ANNEXE 6	Remue-ménages des anomalies.....	109
ANNEXE 7	Fiche des symptômes.....	117
ANNEXE 8	Tableaux des coûts.....	119
ANNEXE 9	Tableau du plan de développement et des coûts liés	121
ANNEXE 10	Tableaux du retour sur l'investissement.....	123
ANNEXE 11	Tableau du cycle de la carte de temps	127
ANNEXE 12	Logigramme de la solution proposée.....	129
ANNEXE 13	Recensement des sources.....	131
ANNEXE 14	Recensement des imprimés d'écran.....	137
ANNEXE 15	Recensement des rapports.....	141
ANNEXE 16	Recensement des objets en exploitation	147
ANNEXE 17	Diagramme de flux de données de l'existant (DFD).....	152
ANNEXE 18	Modèle conceptuel de données de haut niveau de l'existant.....	154
ANNEXE 19	Diagramme de classes de la solution	156

ANNEXE 20	Diagramme de flux de données de la solution.....	158
ANNEXE 21	Modèle relationnel de données de la solution.....	160
ANNEXE 22	Dictionnaire de données de la solution.....	162
ANNEXE 23	Description des cas d'utilisation.....	179
ANNEXE 24	Plan de navigation et principaux dessins d'écrans	205
ANNEXE 25	Questionnaire sur l'évaluation de l'application WEB des CT.....	210
BIBLIOGRAPHIE	212

LISTE DES TABLEAUX

	Page
Tableau I Catégories de maintenance dans ISO 14764.....	26
Tableau II Facteurs influençant la maintenabilité.....	29
Tableau III Caractéristiques sur la qualité du logiciel ISO/IEC FCD 9126-1991.....	30
Tableau IV Sous caractéristiques et définitions ISO/IEC FCD 9126-1991.....	31
Tableau V Évaluation des parties prenantes en causes.....	50
Tableau VI Activités à conserver, à remanier et à éliminer.....	55
Tableau VII Zones d'information nécessaires sur la carte de temps.....	67
Tableau VIII Champs obligatoires pour la carte de temps.....	69
Tableau IX Nombre d'heures travaillées.....	84
Tableau X Comparaison des éléments techniques.....	91
Tableau XI Résultat suite au questionnaire.....	92

LISTE DES FIGURES

	Page
Figure 1	Relation entre les différents termes sur la réingénierie..... 9
Figure 2	Modèle général de la réingénierie du logiciel..... 11
Figure 3	Vue commune de la réingénierie du logiciel 12
Figure 4	Les différentes méthodologies de la réingénierie..... 14
Figure 5	Modèle de vie en cascade..... 18
Figure 6	Modèle de vie en V..... 19
Figure 7	Modèle de vie par extensions successives..... 20
Figure 8	Modèle de vie incrémental..... 21
Figure 9	Modèle de vie par prototype..... 22
Figure 10	Modèle de vie en spirale de Boehm..... 23
Figure 11	Modèle de vie par objet de Nerson 24
Figure 12	Enchaînement des phases du cycle de vie du logiciel 27
Figure 13	Les sous-caractéristiques d'ISO/IEC 9126 – 1991..... 31
Figure 14	Les sous-caractéristiques, évolution d'ISO/IEC 9126–1:2001..... 33
Figure 15	Qualité du produit dans le cycle de vie d'ISO/IEC 9126..... 34
Figure 16	Modèle de réingénierie proposé..... 43
Figure 17	Cycle de vie des cartes de temps..... 47
Figure 18	Niveau de gestion de la carte de temps..... 72
Figure 19	Plan de navigation..... 79
Figure 20	Description de la fonction des écrans 80

LISTE DES ABRÉVIATIONS ET SIGLES

CT	– Carte de temps
DFD	– Diagramme de flux de données
MCD	– Modèle conceptuel de données
MRD	– Modèle relationnel de données
PDF	– Portable Document Format
PY	– Comptes payables
RE	– Comptes recevables
RH	– Ressources humaines
RHU	– Ressources humaines (réf. au système de paye)
RPA	– Réingénierie du processus d'affaires
RL	– Réingénierie du logiciel
RPG	– Report program generator
SI	– Système d'information
TI	– Technologie de l'information
TF	– Temps facturé

INTRODUCTION

Avant même de définir la réingénierie du logiciel, nous nous sommes demandé avec quoi ou à quel domaine moins abstrait pourrait-il bien être comparé. Au cours des nos lectures, nous sommes tombés par hasard sur le site du gouvernement du Nouveau-Brunswick dans la section des beaux-arts, sur un article écrit par Kim Muir, Debra Daly Hartin et Peter Vogel de l'Institut canadien de conservation sur la restauration de l'une des œuvres du peintre anglais James Barry.

Voici un extrait de l'article¹ :

« Après l'arrivée de l'œuvre de James Barry « La mort du général Wolfe » à l'Institut canadien de conservation (ICC), nous avons procédé à un examen technique approfondi afin de déterminer la structure, les matériaux et la condition de la peinture. Le tableau s'est révélé essentiellement en très bon état et sa structure n'avait besoin que d'un léger traitement. Cependant, l'accumulation de saletés et les couches jaunies de vernis et de coloration appliquées lors des restaurations antérieures avaient défiguré l'apparence de la peinture. La toile avait été nettoyée lors des restaurations antérieures; certaines parties avaient été nettoyées plus que d'autres. Un tel « nettoyage artistique » n'est pas inhabituel. L'étendue du nettoyage de sections spécifiques peut dépendre de l'importance, de la tonalité et de la condition de ces sections. La présence d'un vernis original uniforme n'était pas visible sur l'ensemble de la surface de la peinture, qui montrait, cependant, des couches discontinues de résine et de glacis. Des résidus de vernis jauni, produits d'un nettoyage antérieur, ont également été décelés directement par-dessus la peinture originale... » «...L'établissement d'une stratégie de nettoyage a nécessité des recherches et une collaboration poussées. En plus de l'examen technique effectué par les restaurateurs, les scientifiques en conservation et les technologues en documentation de l'ICC, des restaurateurs et des conservateurs d'autres établissements ont également été consultés. »

Bien que toute analogie comporte ses faiblesses, il y a plusieurs parallèles à faire entre la réingénierie du logiciel et la restauration d'œuvres d'art. Par exemple, avant d'entreprendre la restauration d'une toile, il faut un engagement, obtenir du financement, trouver un restaurateur de calibre, apprendre l'histoire de l'œuvre, de l'artiste, sa

¹ Réf. : <http://www.gnb.ca/0130/Wolfe/ConservationFR.html>
<http://www.amnb.nb.ca/pdf/AMNB%20Fall%2003%20Fre.pdf>

technique, les matériaux utilisés à l'époque, les matériaux modernes à utiliser pour la restauration (produits de nettoyage, vernis, etc.), faire appel aux techniques modernes (radiographie, lumière ultraviolette, microscopie), etc.

Loin de vouloir faire un exposé sur la restauration d'œuvres d'art, ce que nous voulons illustrer en premier lieu, c'est l'ampleur insoupçonnée que sont les projets de restauration des systèmes informatisés tout comme sont ceux des œuvres d'art. Deuxièmement, quel que soit le domaine d'activité, les réingénieries passent à travers des étapes très similaires, dont le domaine du logiciel à tout avantage à s'inspirer. Lorsqu'on entreprend la réingénierie d'un système informatisé, celui-ci est déjà dans un mauvais état. Prioriser le projet, obtenir l'autorisation de la haute direction et les ressources nécessaires, l'étudier, le comprendre, le conceptualiser, le planifier, le programmer de nouveau, le tester, l'implanter, etc., tout ceci est un travail de longue haleine comportant plusieurs obstacles. Bref, il faut déterminer les méthodes qui conviennent le mieux au projet ainsi que toutes les techniques qui s'y rattachent. Peut-être la réingénierie du logiciel est-elle belle et bien une forme de restauration d'œuvre d'art avec plus au moins le talent artistique nécessaire pour y parvenir avec succès.

Le but de ce projet est donc d'intégrer certaines des phases de la réingénierie du logiciel dans un modèle en y incluant la phase la plus souvent négligée : la réingénierie du processus lui-même. Trois chapitres composent ce document. Le premier chapitre passe en revue la littérature du domaine : l'historique, les définitions, les concepts, les risques, etc. Au deuxième chapitre, nous appliquons certaines théories au processus de gestion des cartes de temps utilisé à Cascades inc. Finalement, le dernier chapitre rend compte de l'expérimentation.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

INTRODUCTION

Ce chapitre passe en revue la littérature sur la réingénierie du logiciel. Un bref historique de la réingénierie du logiciel y est présenté; suivent les principales définitions et le modèle général de la réingénierie du logiciel. J'aborde également quelques thèmes associés à l'ingénierie du logiciel dont la gestion de projet, le modèle du cycle de vie, la qualité du logiciel, les risques et une brève perspective. Ce survol de la littérature sur le domaine de la réingénierie permet de consolider une approche en vue de la réalisation d'un projet de réingénierie du logiciel.

1.1 Historique

Une des façons pour connaître l'époque approximative de l'arrivée d'un nouveau phénomène est d'observer les dates d'édition des ouvrages de référence sur le sujet en question. On a vu au début des années 90 apparaître une multitude de publications sur le concept de la réingénierie du logiciel. Le portefeuille de systèmes informatiques ayant atteint une masse critique en terme de quantité et leur vie utile tirait à leur fin. Les entreprises ont dû remettre en question l'exploitation de ces systèmes sur des ordinateurs centraux ou les convertir sur des architectures plus légères. Par la maintenance, ces systèmes informatiques ont évolué; or la mise à jour de la documentation n'ayant pas suivie, ce qui entraîne une perte de connaissance de leur fonctionnement dans l'entreprise.

Avant même qu'apparaisse la réingénierie, il y a d'abord eu l'ingénierie. Le terme *software engineering* est apparu dès 1968 lors d'une conférence de l'OTAN en Allemagne de l'Ouest à Garmisch [Nau68]. Dans les années 70, le nombre croissant de

logiciels en exploitation nous oblige à reconnaître le fait qu'un logiciel ne meurt pas facilement. C'est à cette époque que la maintenance logicielle acquiert sa reconnaissance. En outre, cette activité de plus en plus demandée en raison du volume de logiciels à traiter occasionne de nombreux retards. Dans les années 80, de nouvelles architectures émergent menaçant de plus en plus ces « vieux » logiciels et rendent leur maintenance contraignante. Dans les années 90, on assiste à des changements et à des améliorations remarquables tant dans le domaine du logiciel que celui de l'infrastructure. L'arrivée d'Internet et de la fibre optique transforme la planète en un vaste réseau compliquant davantage la problématique qu'est la maintenance logicielle. C'est à cette époque que le concept de la réingénierie du logiciel devient un sujet d'étude répandu. Le bogue de l'an 2000 a lancé véritablement l'amélioration des méthodes de réingénierie du logiciel. Or la problématique de reconception des systèmes patrimoniaux demeure entière. On parle désormais de *software evolution* [Yan03] principe qui introduit la notion de réingénierie permanente en raison de changements perpétuels des processus d'affaires et des technologies de l'information.

1.1.1 Systèmes patrimoniaux

L'expression « systèmes patrimoniaux » s'est largement répandue pour désigner les systèmes informatiques issus d'une génération précédente et nécessitant une mise à niveau. Dans [Yan03], on indique que la maintenance de ces systèmes est devenue la phase la plus coûteuse de leur cycle de vie. [Ben95] indique qu'on peut définir informellement les systèmes patrimoniaux comme de grands systèmes logiciels que nous ne savons pas comment leur faire face, mais qu'ils sont vitaux pour l'organisation. Sans documentation à jour, leur maintenance est souvent faite à l'aide du code comme source d'information. On a beau vouloir prolonger la vie de ces systèmes informatiques, arrive un moment où il faut bien se rendre à l'évidence : leur coût d'exploitation est exorbitant sans oublier qu'ils ne répondent plus aux besoins de l'entreprise en terme de fonctionnalités et d'efficacité.

Refaire ces systèmes patrimoniaux exige beaucoup de ressources tant humaines, financières que matérielles qui ne sont malheureusement pas toujours disponibles. [Bro87] nous dresse une liste des principales raisons qui sont autant de freins à la mise à niveau de systèmes patrimoniaux : la complexité, la conformité, les changements et l'invisibilité. La nature même du logiciel cause cette crise. Le processus du génie logiciel ne travaille qu'avec des représentations simplifiées et se conforme à des besoins arbitraires de nature humaine et organisationnelle (ex. : le système de paye qui doit correspondre aux différentes conventions collectives d'une même entreprise). Le logiciel est soumis à de nombreuses demandes de changement au cours de son cycle de vie en plus d'opérer sur une technologie tout aussi évolutive au fil du temps. Finalement, le processus de fabrication semble invisible et difficilement mesurable, le chef de projet doit s'en remettre à la documentation pour mesurer le progrès. D'autres auteurs, comme [Lub92], sont non seulement d'accord avec les raisons précédemment énoncées, mais ils en énoncent d'autres :

- a. des besoins définis par le client ou par quelqu'un d'autre en son nom et l'interprétation qui en est faite;
- b. des besoins inventés pour le projet sans être requis par le client;
- c. des besoins qui se rajoutent au fur à mesure du développement;
- d. des développements liés à l'évolution au lieu de repartir à zéro;
- e. un manque d'interaction entre la gestion du projet et sa planification.

1.2 Taxonomie sur la réingénierie

Afin de mieux comprendre le domaine de la réingénierie du logiciel, voici les principaux termes utilisés et leur définition. Les définitions suivantes sont extraites de l'article « Reverse Engineering and Design Recovery » et couvrent une partie importante des principaux termes liés à la réingénierie du logiciel.

- **Recouvrement de la conception** (*design recovery*) est un sous-ensemble de la rétro-ingénierie faisant appel à la connaissance du domaine, à l'information externe et au raisonnement flou, y sont rajoutées les observations significatives de haut niveau d'abstraction sur le système au-delà de celles obtenues directement en examinant le système lui-même. Le recouvrement de la conception recrée des abstractions de conception venant d'une combinaison de code, de documentation existante sur la conception (si disponible), d'expériences personnelles, et de connaissances générales sur un problème ou sur un domaine d'application. [Chi90]
- **Ingénierie** (*forward engineering*) est le processus traditionnel de passer d'un haut niveau d'abstraction et de conception détaillée vers le développement physique du système. [Chi90]
- **Redocumentation** (*redocumentation*) est la création ou la révision d'une représentation sémantiquement équivalente dans le même niveau relatif d'abstraction. Les formes de représentation résultantes sont habituellement considérées des vues alternatives (par exemple, flux de données, structures de données, et flux d'informations) destinées à une compréhension humaine. La redocumentation est la forme la plus simple et la plus ancienne de rétro-ingénierie, et peut être considérée une forme non envahissante et à faible niveau de restructuration. [Chi90]
- **Réingénierie du logiciel** (*software reengineering*) est l'examen et l'altération d'un système afin de le reconstituer sous une nouvelle forme suivit de son implantation. [Chi90]
- **Restructuration** (*restructuring*) est la transformation à partir d'une forme de représentation vers une autre forme ayant le même niveau d'abstraction tout en

préservant le comportement externe du logiciel (fonctionnalité et sémantique). [Chi90]

- **Rétro-ingénierie** (*reverse engineering*) est le processus d'analyser un système en vue d'identifier ses composantes, leurs corrélations et de créer des représentations du système sous une autre forme ou d'un niveau plus élevé d'abstraction. [Chi90]

La définition suivante est issue IEEE Std 1219 :

- **Maintenance logicielle** (*software maintenance*) est la modification d'un produit logiciel après la livraison pour corriger les défauts, pour améliorer l'exécution ou tout autre attribut, ou encore adapter le produit à un environnement changé. [Swe04]

La définition suivante est issue du document de Peter H. Aiken :

- **Rétro-ingénierie des données** (*data reverse engineering*) est l'utilisation de techniques structurées pour reconstituer les schémas du modèle de la base de données d'un système existant. [Aik96]

Les définitions ci-après sont extraites des références du volume « Successful evolution of software systems » et du site THE ACM, et les complètent.

- **Compréhension du logiciel** (*program understanding or program comprehension*) est reliée au domaine de la rétro-ingénierie. La compréhension du logiciel implique toujours qu'elle commence par le code source alors que la rétro-ingénierie du logiciel peut commencer par la forme binaire et exécutable du système ou aux descriptions de niveau élevé de la conception. Le domaine de la

compréhension du logiciel inclut l'aspect cognitif des processus de raisonnement humain dans la compréhension du logiciel. Elle peut être réalisée d'une façon ad hoc et sans produire une nouvelle documentation contrairement à la rétro-ingénierie qui elle est une approche systématique pour développer une représentation externe du système. La compréhension du logiciel est comparable au recouvrement de la conception, car les deux débutent au niveau du code source. [Yan03]

- **Recoder** (*recode*) implique de changer les caractéristiques du code source. Changement du langage de programmation et de la structure sont des changements au niveau du code source. D'autres changements possible incluent une normalisation du code, une amélioration de la lisibilité de code, et de renommer certaines fonctions ou procédures. [Yan03]
- **Reconceptualiser** (*redesign*) implique de changer les caractéristiques de conception. Les changements possibles incluent de restructurer l'architecture de la conception, de restructurer le modèle des données ou de la base de données, et d'améliorer l'algorithme. [Yan03]
- **Réusiner** (*refactoring*) est une discipline technique pour restructurer le code existant, pour changer la structure interne sans changer son comportement externe. [Fow02]
- **Respécifier** (*respecify*) implique de changer les caractéristiques des exigences. Ce type de changement peut se rapporter à changer seulement la forme des conditions existantes (c.-à-d., prendre des exigences floues et les traduire en un langage formel). Ce type de changement peut également se rapporter à des conditions changeantes du système, telles que l'addition de nouvelles exigences, ou la suppression et le changement de conditions existantes. [Yan03]

- **Rétrospécifier (reverse specification)** est une forme de rétro-ingénierie où des spécifications sont extraites du code source ou de la conception. Les spécifications dans ce contexte représentent une vue d'ensemble de ce qu'est le logiciel. Dans le développement habituel, les spécifications nous indiquent ce que le logiciel doit faire. Mais cette information n'est pas incluse dans le code source. Seulement dans de rares cas, elle peut être récupérée des commentaires venant code source et des personnes impliquées jadis dans le développement original du système. [Yan03]

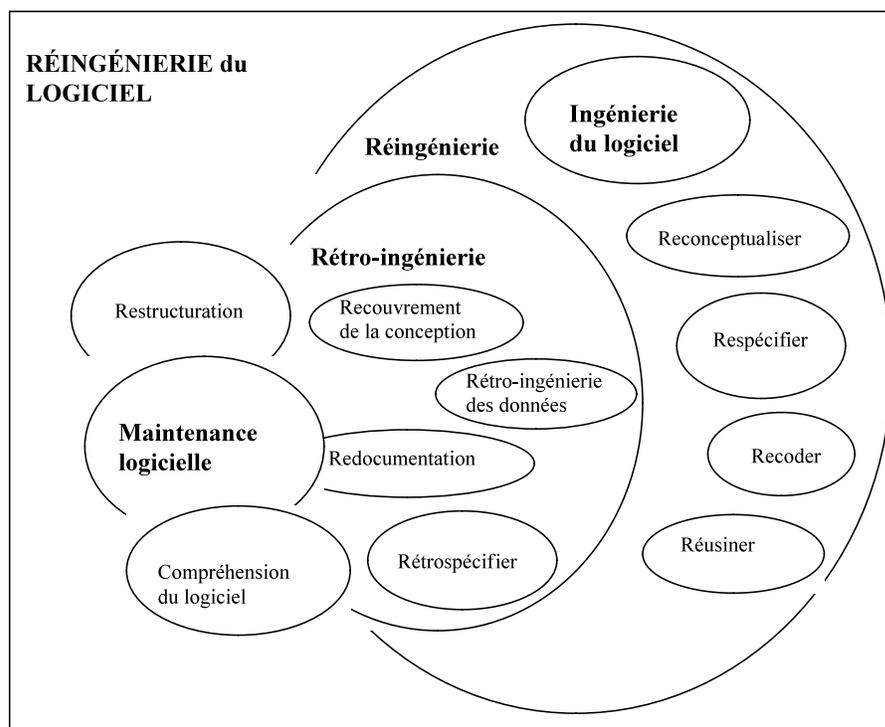


Figure 1 Relation entre les différents termes sur la réingénierie

La figure 1 illustre les différents termes utilisés en réingénierie et permet de comprendre à quel grand domaine ils se rapportent.

1.3 Modèle de la réingénierie du logiciel

De manière générale et après plusieurs années d'exploitation, arrive le moment où le logiciel (ou sa plateforme) devient désuet. Il ne possède plus les qualités internes et externes² suffisantes pour continuer d'être exploité dans sa forme actuelle, ou bien, la plateforme n'est plus supportée par le fabricant. Cependant, il a toujours son importance au sein de l'entreprise. Une réflexion sérieuse s'impose pour déterminer de quelle façon le logiciel sera mis à niveau par un rajeunissement, une modernisation, etc. Des alternatives se présentent : faire l'acquisition d'un produit équivalent ou entreprendre un processus de réingénierie du logiciel. Ce projet de maîtrise décrit la réingénierie appliquée à un logiciel de gestion des cartes de temps.

La réingénierie du logiciel consiste à examiner et à altérer le logiciel pour le reconstituer sous une nouvelle forme et le reprogrammer [Chi90]. Typiquement, il s'agit d'un changement radical et la plus dispendieuse des formes d'altération de logiciel selon les références citées dans [Swe04]. La réingénierie du logiciel est un concept composé principalement de deux activités importantes : *rétro-ingénierie* (*reverse engineering*) et *ingénierie* (*forward engineering*). L'union de ces deux activités en un seul cycle de vie constitue la réingénierie du logiciel. Si on se réfère à la figure 2 ci-après, la partie gauche du modèle de réingénierie (ascension) a pour but de reconstituer les exigences d'origine du logiciel. Le sommet décrit la période de redéfinition des exigences et finalement la partie de droite (descente) est la transformation des exigences en un nouveau logiciel, une nouvelle base de données et un nouvel ensemble matériel (implémentation).

² Voir la section 1.5.5 Qualité du logiciel

Pour mieux comprendre ce modèle, nous nous sommes inspirés de la démarche nécessaire à l'ascension d'une montagne. À la base de la montagne, on se trouve tout près des besoins d'affaires de l'entreprise. Plus on monte, plus loin et plus clair on voit bien qu'on s'éloigne du détail. Sur le sommet, un recul est possible, tout comme il est possible à ce moment-là de la réingénierie d'évaluer les nouvelles exigences et d'en profiter pour retrouver toute la motivation nécessaire vers la descente, vers la réalisation d'un nouveau logiciel. C'est là qu'il faut établir le plan, définir des méthodes, élaborer les techniques et trouver les ressources nécessaires pour réussir le projet de réingénierie.

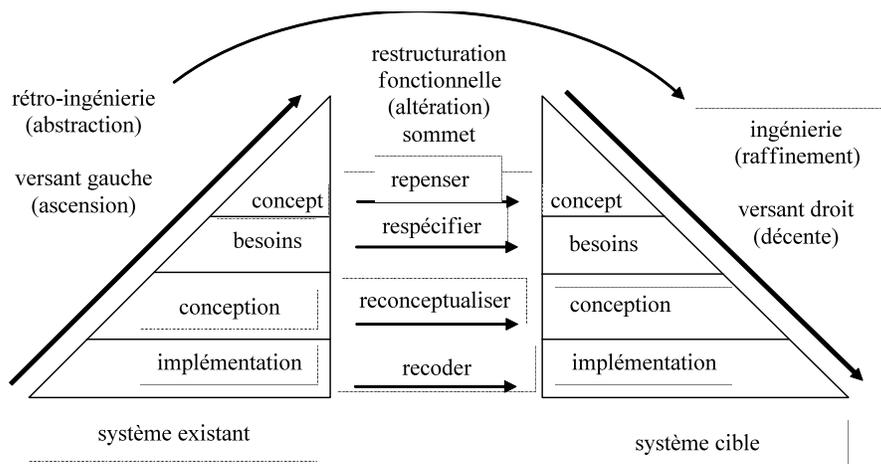


Figure 2 Modèle général de la réingénierie du logiciel [Byr92]

Dans le processus de réingénierie, il est tout de même possible de passer directement du côté de la rétro-ingénierie à celui de l'ingénierie, sans passer par la restructuration fonctionnelle, si les besoins ou la conception demeurent les mêmes. Par exemple, il est possible de « reconceptualiser » directement sans passer la redéfinition des besoins (respécifier) lorsque ceux-ci n'ont pas changé.

Le modèle de [Fei93], à la figure 3, illustre en détail, qu'il est possible de rester à un même niveau et passer de la rétro-ingénierie à l'ingénierie directement sans passer par tous les étapes de la réingénierie pour réimplanter la solution.

1.3.1 Le versant gauche (l'ascension)

1.3.1.1 La rétro-ingénierie

La rétro-ingénierie est le processus d'analyse du système logiciel en exploitation pour identifier ses composantes et leurs interrelations. Ceci en vue de créer une représentation du système dans une autre forme ou à un haut niveau d'abstraction [Chi90].

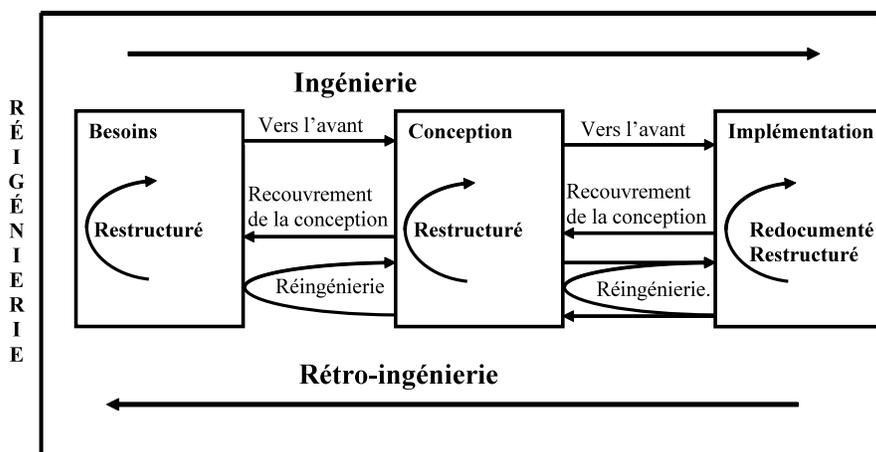


Figure 3 Vue commune de la réingénierie du logiciel [Fei93]

En résumé, il s'agit de documenter les spécifications techniques et conceptuelles. Le point de départ en rétro-ingénierie est la finalité de l'ingénierie, c'est-à-dire le système actuel en exploitation.

L'objectif primaire de la rétro-ingénierie est de reproduire la documentation technique et fonctionnelle à partir d'une analyse systématique de multiples sources d'information : le code source pour les structures de données et les programmes, les maquettes écrans, les formulaires de saisie, les rapports, la documentation existante, l'instanciation des données et finalement les usagers. Ainsi, on peut obtenir les éléments suivants :

- a. les composantes de la structure du système et leurs relations ;
- b. les fonctionnalités et la base sur laquelle les composantes opèrent;
- c. une compréhension simplifiée du système, car elle exclut les détails techniques.

Plusieurs autres objectifs sont visés par la rétro-ingénierie : subdiviser un système complexe pour mieux le comprendre, appréhender les impacts possibles des changements sur les autres systèmes, étudier les normes de programmation et découvrir des règles d'affaires, etc. Le but de la rétro-ingénierie du logiciel n'est pas le changement, mais intégré dans le cadre de la réingénierie du logiciel, la rétro-ingénierie est le point de départ du processus qui lui nous mènera au changement nécessaire pour refaire le système existant.

Comme les techniques et les méthodes de la rétro-ingénierie ne sont pas systématisées, [Yan03] stipule six précautions à prendre avant d'en commencer le processus :

- a. le code doit être propre au système et non générique;
- b. le code peut comporter des erreurs et induire en erreur;
- c. la rétro-ingénierie peut en soi générer des erreurs, il faut revalider;
- d. la rétro-ingénierie peut être très onéreuse et le retour sur l'investissement n'est pas garanti. Une analyse coût-bénéfice est requise;
- e. il n'y a pas de normes et de méthodologies normalisées;
- f. il n'y a aucune mesure bien établie.

Différentes pratiques permettent la réussite de la rétro-ingénierie et de retrouver en grande partie l'information d'origine qui a servi à la construction du logiciel ou fera en sorte de mieux comprendre le fonctionnement ce dernier :

- La compréhension du logiciel s'agit de son étude à partir du code qui peut être de manière « had doc » sans produire de bien livrable proprement dit.
- Le recouvrement de la conception doit reproduire toute l'information requise pour qu'une personne puisse comprendre pleinement ce que fait le logiciel, comment il le fait et pourquoi il le fait. Il renferme une vaste information qui à son tour se trouve

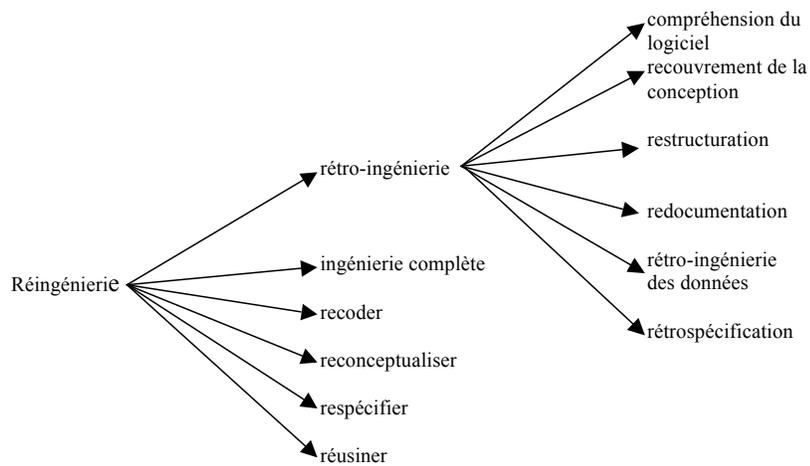


Figure 4 Les différentes méthodologies de la réingénierie

dans l'ingénierie traditionnelle du logiciel. MCD, MRD, DFD et autres modèles sont des livrables pouvant résulter du recouvrement de la conception.

- La restructuration est axée sur l'amélioration du logiciel sans changer sa fonctionnalité. Il améliore la compréhension du logiciel. C'est une forme de maintenance préventive dont le principal livrable est le code source lui-même.
- La redocumentation vise la mise à jour des documents. Les livrables résultants sont des représentations graphiques amenant des vues du logiciel sous différents angles comme des DFD, MCD et MRD et des diagrammes de flux de contrôle, etc.
- La rétro-ingénierie des données vise à reconstruire la documentation sur la base des données existantes. Le MCD, le MRD et autres modèles sont des livrables possibles et résultant de la rétro-ingénierie des données.
- Le rétrospécification est une description synthétisée de ce que fait le logiciel à partir de l'étude du code. Un document explicatif est l'un des biens livrables de cette pratique.

1.3.2 Le sommet

Le sommet de la réingénierie du logiciel (figure 2) est atteint au moment de la période de redéfinition des besoins. La restructuration fonctionnelle est réalisée en effectuant les changements conceptuels nécessaires pour parvenir à répondre à de nouveaux besoins. Cette nouvelle architecture sert de point de départ à l'ingénierie du logiciel. Cette période est déterminante dans le choix des technologies et méthodologies à utiliser pour effectuer les changements désirés.

1.3.3 Le versant droit (la descente)

1.3.3.1 Ingénierie du logiciel

Lorsque les nouveaux besoins sont établis, il existe plusieurs avenues pour entreprendre la construction du nouveau système. Cinq pratiques d'ingénierie du logiciel sont proposées selon le degré d'importance de la réingénierie à entreprendre :

- L'ingénierie complète du logiciel est le processus traditionnel qui consiste à passer à un haut niveau d'abstraction logique et de conception vers l'implémentation physique d'un système [Chi90]. Dans le domaine de la réingénierie, elle constitue la contrepartie de la rétro-ingénierie ayant le plus d'impacts. L'ingénierie du logiciel consiste à développer le système selon les nouvelles exigences. C'est l'approche la plus radicale du changement. Contrairement aux autres approches, elle affecte directement et à divers degrés les aspects humains, matériels et logiciels associés au processus d'affaires et supportés par les TI. Il est possible que la reconstruction du logiciel nécessite le changement du langage de programmation, de la base de données, l'ajout de nouveaux équipements, etc.

Les approches suivantes bien que faisant partie de la réingénierie ne touchent que l'un ou quelques-uns des éléments de l'ingénierie énumérés ci-après : les programmes, le langage, les structures de la base de données et autres éléments de nature logicielle. L'infrastructure matérielle et le processus d'affaires supporté sont rarement remplacés ou modifiés par ces éléments. Ces approches n'apportent pas de modifications aussi radicales au logiciel que peut être l'approche de la réingénierie au sens strict du terme.

- Recoder implique de changer le code source ou même à l'occasion de changer le langage de programmation. Recoder peut être nécessaire pour se conformer à de

nouvelles normes, pour améliorer la compréhension du programme ou pour respecter une nouvelle nomenclature.

- Reconceptualiser implique le changement des caractéristiques de conception. Ceci peut supposer la restructuration de l'architecture logicielle, des algorithmes, de la base de données ou bien de l'ensemble logiciel constituant le système.
- Respécifier implique le changement des exigences. Ce type de changement fait référence à la formulation des exigences transposées en un langage formel (rédigé). Il peut également se transposer en ajout, en retrait ou en altération de fonctionnalités incluses dans le logiciel.
- Réusiner est simplement un ajustement de l'aspect du code sans changer la fonction qu'il effectue. Aucune nouvelle fonctionnalité n'y est ajoutée.

1.4 Modèles de cycle de vie du logiciel

Le cycle de vie du logiciel représente la séquence de production de ce dernier. Chaque phase désigne une activité qui elle-même se subdivise en tâches pour assurer le suivi et la gestion du projet. Lorsque le temps est venu de passer à l'étape de l'ingénierie après la restructuration fonctionnelle, même si on décide de procéder par l'approche Big Bang [Fei93] qui consiste à développer séparément du système patrimonial, il existe plusieurs modèles de cycles de vie acceptables. Il faut choisir celui qui se prête le mieux au contexte en fonction des critères de complexité, de temps, de coûts, de compétences des développeurs, etc.

Voici les principaux modèles de cycles de vie d'un logiciel :

1.4.1 Modèle de cycle de vie en cascade

Le modèle historiquement le plus populaire, car les phases d'élaboration sont effectuées successivement. Il est adapté au développement de logiciel comportant peu de problèmes de faisabilité. Chaque phase se termine par une vérification et une validation pour éliminer le plus grand nombre d'anomalies à l'intérieur même de la phase. Les retours sur les phases antérieures ne se limitent qu'aux phases immédiatement adjacentes. Mal maîtrisé, ce modèle peut occasionner des erreurs en avalanches et entraîner des coûts d'énormes selon le nombre de phases à traiter à reculons.

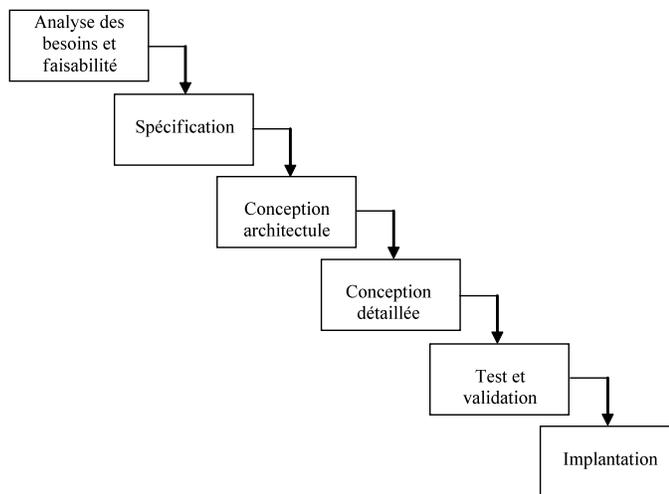


Figure 5 Modèle de vie en cascade

1.4.2 Modèle de cycle de vie en V

Couramment utilisé pour le logiciel du domaine de la gestion, ce modèle est une amélioration du modèle en cascade qui permet, en cas d'anomalie, de limiter un retour aux étapes précédentes. Il offre une meilleure visualisation du rôle des tests par rapport aux spécifications. À chaque phase du développement correspond une phase de validation. Ce modèle rend explicite le fait que les premières étapes préparent les dernières par les vérifications et les validations. Afin d'améliorer le logiciel, les phases de la partie montante doivent renvoyer de l'information aux phases en vis-à-vis lorsque des défauts sont détectés.

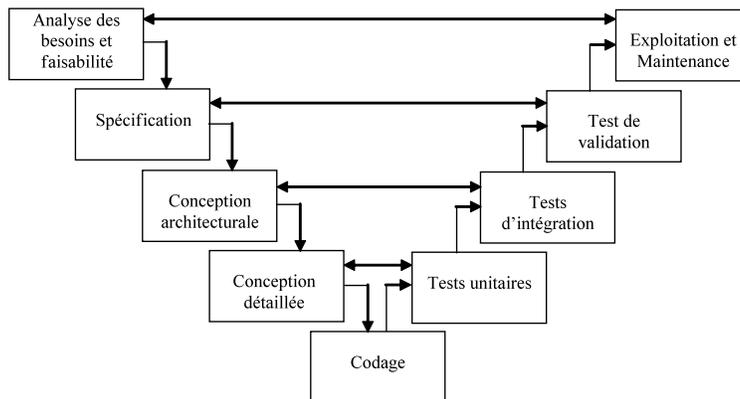


Figure 6 Modèle de vie en V

1.4.3 Modèle de cycle de vie par extensions successives

Son intérêt consiste à développer le logiciel en étendant ses fonctionnalités progressivement et de façon modulaire. Les fonctionnalités du nouveau système sont ainsi développées et introduites de manière graduelle à l'intérieur du système existant. Il y a autant évolution de l'architecture que des composantes logicielles. Il faut s'assurer que les extensions ne viennent pas déstabiliser le logiciel en place. Ce modèle permet de tester les extensions plus facilement et simplifie l'adaptation des utilisateurs.

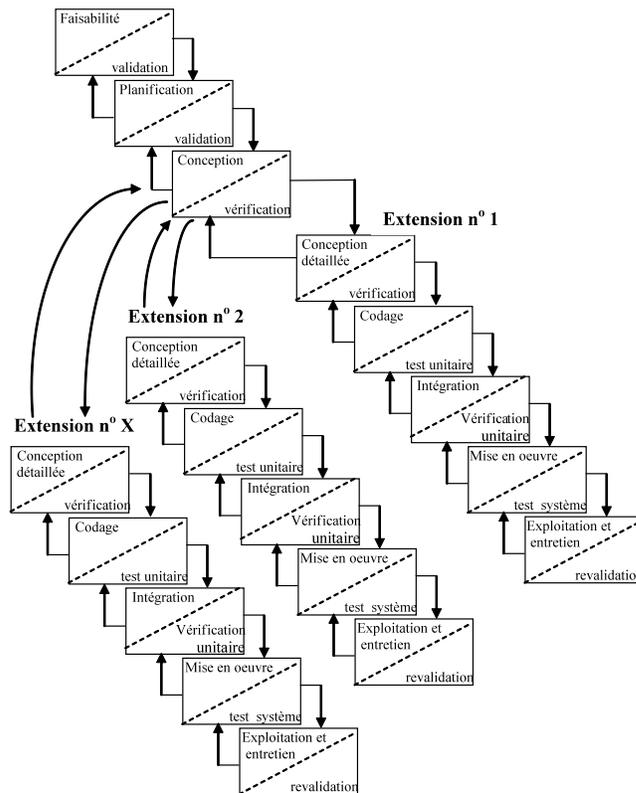


Figure 7 Modèle de vie par extensions successives

1.4.4 Modèle de cycle de vie incrémental

Ce modèle est utile pour la livraison de sous-ensembles, par exemple pour la livraison de la première phase d'un projet. Il utilise de façon systématique le développement par extensions successives. C'est une succession de cycles en cascade, dans laquelle les activités de chacune des phases doivent constituer une suite logique des phases précédentes. De plus, chaque phase vient incrémenter les résultats des travaux d'une phase de même nature d'un cycle parallèle. Bien qu'identifier le noyau et les incréments ne soit pas chose facile, ses avantages résident dans sa souplesse et dans l'absence de blocage en cas de spécification incomplète. Cependant, l'accroissement des coûts, les difficultés quant à la gestion du projet, des configurations et de contrôle de la qualité sont des inconvénients qu'il ne faut pas négliger lors du choix de ce modèle.

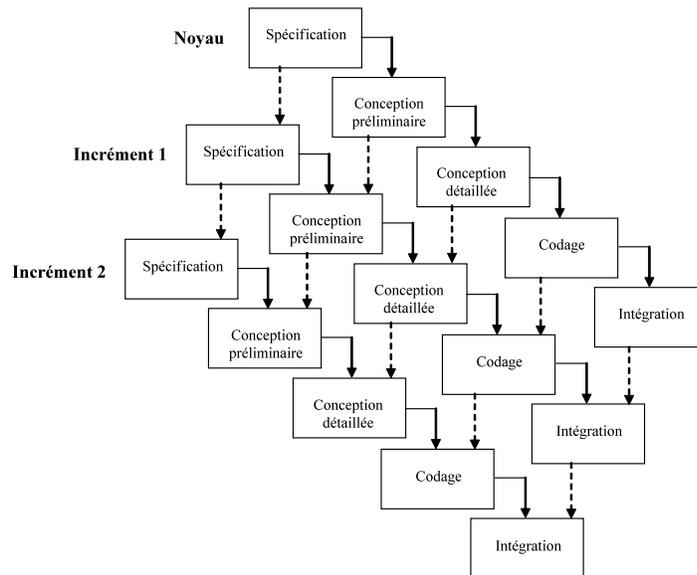


Figure 8 Modèle de vie incrémental

1.4.5 Modèle de cycle de vie par prototype

Ce modèle vise à explorer rapidement et à moindre coût les fonctions, l'architecture et la convivialité d'un logiciel. Il convient aux projets ayant des besoins mal définis. Il permet de clarifier, de compléter les spécifications en plus de déterminer les caractéristiques souhaitées. Il peut également servir à développer un logiciel pour un besoin ponctuel non réutilisable ou dans le cas d'un logiciel évolutif qui n'est pas encore stabilisé en ce qui concerne ses exigences. Il faut garder à l'esprit le moment où il faut passer à un modèle de développement plus structuré avant que le prototype devienne non rentable.

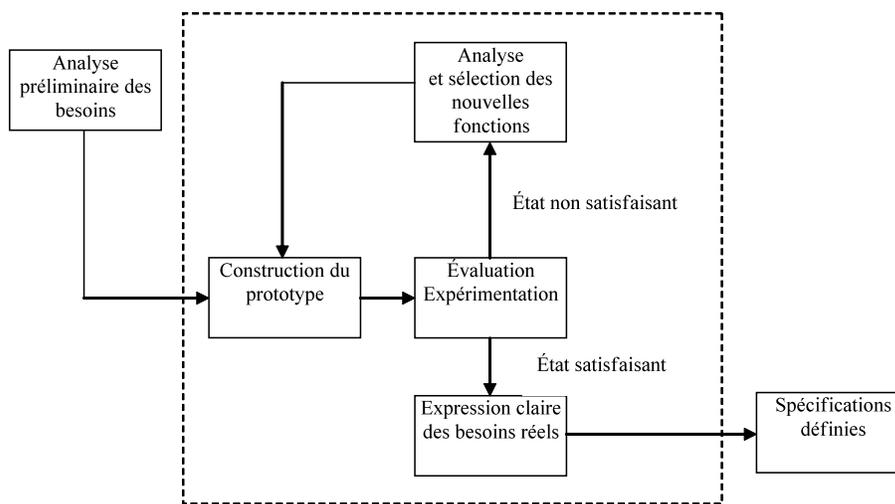


Figure 9 Modèle de vie par prototype

1.4.6 Modèle de cycle de vie en spirale de Boehm

Ce modèle convient pour la mise en œuvre de projets innovants, de très gros projets de développement et de rénovation. Il s'accommode des modèles existants en intégrant la gestion du risque. Plus complet et plus complexe, il introduit l'utilisation systématique de prototypes pour s'assurer de la bonne compréhension des besoins. Il est plus général que les précédents et peut même les inclure.

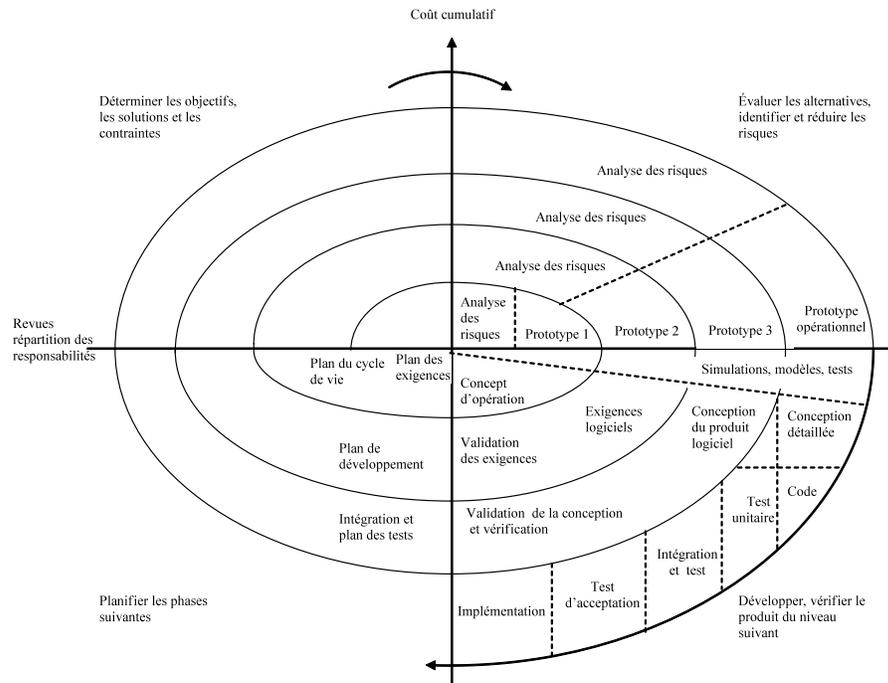


Figure 10 Modèle de vie en spirale de Boehm

1.4.7 Modèle de cycle de vie par objet

Ce modèle utilise des objets déjà définis du domaine et qui sont insérés dans une librairie de composantes réutilisables après chaque projet réalisé. Il faut les amener à un niveau de généralisation suffisant pour permettre leur réutilisation. Cet effort réduit le temps de développement et repère les erreurs. Ainsi, il sera possible de construire le modèle descriptif de la nouvelle application en faisant appel à des composantes, à des modules ou à des classes déjà présents en librairie.

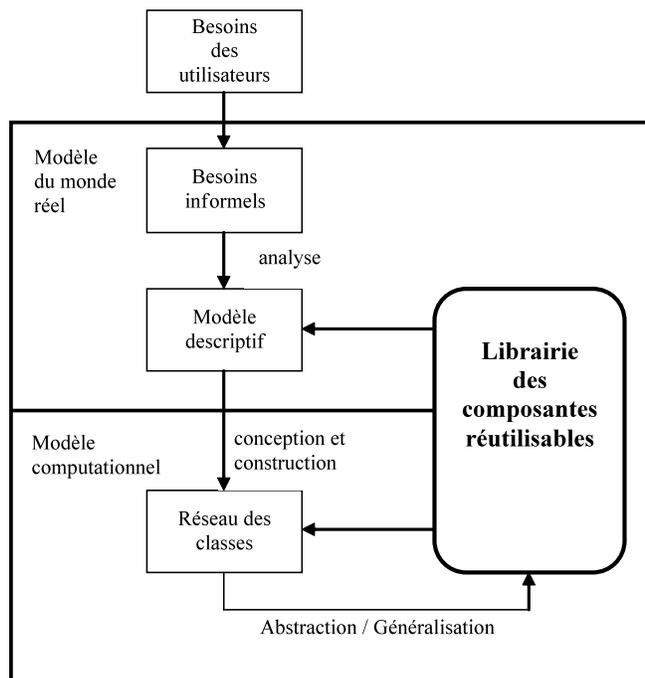


Figure 11 Modèle de vie par objet de Nerson [Ner92]

Dans la pratique quotidienne, le problème comporte souvent des caractéristiques qui ne concordent pas complètement avec celle d'un modèle unique du cycle de vie du logiciel. Il peut s'avérer nécessaire de composer avec plus d'un modèle pour un même développement d'une application comportant plusieurs processus d'affaires. Souvent, l'utilisation du modèle en V et du modèle incrémentale en raison de la décomposition en sous-ensembles sera préférable.

1.5 Maintenance logicielle

Une fois le logiciel en production, il entre dans une nouvelle étape ; celui de sa maintenance. La maintenance a pour but de conditionner la pérennité optimale du logiciel en corrigeant les erreurs de programmation tout en permettant son évolution pour correspondre aux nouvelles fonctionnalités exigées par les usagers et les pratiques d'affaires. La première loi de Lehman [Leh80] sur l'évolution du logiciel stipule qu'un logiciel doit nécessairement évoluer sinon il devient progressivement inutile. Au chapitre 6 du « Guide to the Software Engineering Body of Knowledge » [Swe04] on indique qu'historiquement la maintenance logicielle n'a pas reçu la même attention que le développement dans le passé. C'est probablement l'une des raisons des retards accumulés qui touche la mise à niveau des systèmes patrimoniaux. Mais cet état est en train de changer à cause, en autres, du problème qu'a suscité le bogue de l'an 2000 et le phénomène de passage vers l'Open Source.

La maintenance comporte quatre catégories selon les travaux de Lientz et Swanson [Ben00] dont ISO reprend les définitions [Swe04]. Plus de 80 % de l'effort de maintenance [Swe04] se situe dans les deux premières catégories : la maintenance adaptative et la maintenance perfective.

1.5.1 Catégories de maintenance

- Maintenance adaptative : Modification d'un logiciel exécutée après sa mise en production qui vise à maintenir le logiciel utilisable dans un environnement changé ou en cours d'évolution ;
- Maintenance perfective : Modification d'un logiciel après sa mise en production pour en améliorer l'exécution ou la maintenance (ex. : nouveaux besoins) ;
- Maintenance corrective : Modification réactive d'un logiciel exécutée après sa mise en production pour corriger des problèmes découverts ;
- Maintenance préventive : Modification d'un logiciel après sa mise en production pour détecter et corriger des défauts latents avant qu'ils ne tournent en défaillances.

1.5.2 Classification de la maintenance logicielle par critère

Les catégories suivantes de la norme ISO14764:2005 ont été classées selon deux critères : proactif et réactif. Ces critères renforcent l'aspect de l'intention et du temporel des catégories de maintenance en fonction des changements environnementaux comme les changements matériels, les processus d'affaires, etc.

Critère 1 : le travail consiste en une correction ou en une amélioration du logiciel.

Critère 2 : le travail est proactif ou réactif.

Tableau I
Catégories de maintenance dans ISO14764

		Aspect intentionnel	
		Correction	Amélioration
Aspect temporel	Proactif	Préventif	Perfectif
	Réactif	Correctif	Adaptatif

La maintenance logicielle (Figure 12) est la phase la plus longue du cycle de vie du logiciel, mais également la plus dispendieuse selon ce que rapporte plusieurs auteurs dont [Ben04] et [Yan03]. Ils stipule que depuis les années 70, les coûts de maintenance sont passés de 30 % à plus de 80 % du budget TI des entreprises.

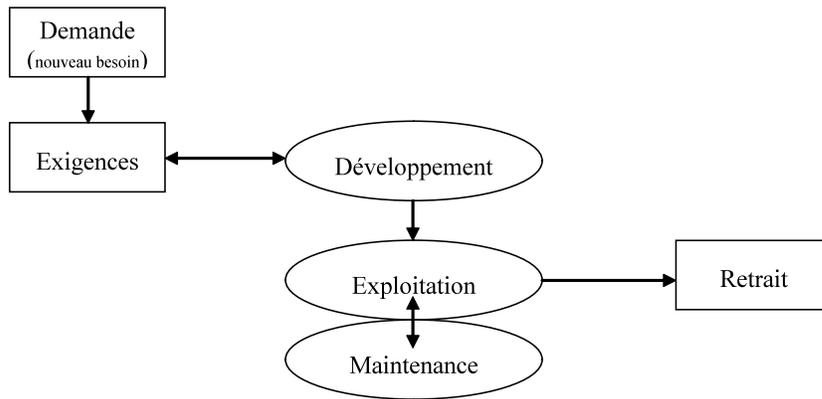


Figure 12 Enchaînement des phases du cycle de vie du logiciel [Pin02]

À l'intérieur de ce cycle de vie, la phase de la maintenance possède son propre cycle en fonction des nouvelles exigences, des erreurs de programmation et des changements apportés à l'environnement du logiciel. Dans [Ben00], on retrouve ce mini cycle souvent initié par une demande de l'utilisateur.

1.5.3 Mini cycle de la maintenance

- Requête de changement ;
- Phase de planification ;
 - Compréhension du logiciel ;
 - Analyse de l'impact du changement ;
- Codification du changement ;
 - Restructuration pour le changement ;
 - Propagation du changement ;
- Vérification et validation ;
- Redocumentation.

La compréhension du logiciel exige l'effort le plus important de ce mini cycle. Il permet d'entreprendre l'ensemble des activités. Le succès de cette étape est lié à la qualité de la maintenance que subit le logiciel au fil du temps. L'un des phénomènes particuliers au mini cycle de la maintenance est la propagation du changement. Il s'agit de changements déclenchés autour de la demande initiale pour permettre sa réalisation qui peuvent eux-mêmes mener à d'autres problèmes s'ils ne sont pas bien contrôlés. De plus, la qualité de la programmation, la normalisation du langage, le professionnalisme des programmeurs, les efforts mis sur la mise à jour de la documentation, etc., sont autant de facteurs qui influencent la facilité de compréhension comme la maintenabilité. La maintenabilité selon la définition de Pfleeger [Swe04] est la facilité avec laquelle un système logiciel ou un composant logiciel peuvent être modifiés en vue de corriger des erreurs, d'améliorer la performance ou autres attributs ou bien de l'adapter à un environnement modifié.

1.5.4 Critères de maintenabilité

[Hor05] ont catégorisé les critères de maintenabilité de la façon suivante :

- les propriétés du produit logiciel :
 - propriétés au niveau spécification (ex. : l'exactitude ou la taille fonctionnelle) ;
 - propriétés au niveau conception (ex. : la modularité) ;
 - propriétés au niveau codification (ex. : nombre d'instructions, complexité, maturité, affaiblissement de la structure, redondance).
- propriétés du processus de maintenance :
 - fréquence des changements ;
 - procédures d'entretien ;
 - processus d'essai (ex.: étendue des essais).
- propriétés des ressources TI :
 - qualité de gestion de l'équipe (ex. : taux d'activité, structure des communications, rotation du personnel) ;
 - qualité des membres de l'équipe (ex.: compétence, expertise des systèmes) ;

- infrastructure d'entretien (ex. : environnement de développement, outils).
- propriétés des changements :
 - changement des propriétés fonctionnelles (ex. : taille fonctionnelle, type de changement, complexité fonctionnelle, la sensibilité d'exécution) ;
 - programmation (ex. : ampleur et impacts des changements, probabilité d'erreurs) ;

On peut mesurer la maintenabilité selon l'ensemble des métriques suivants, constitué par Gilb, que [Yan03] a repris dans son ouvrage :

- a. temps de reconnaissance du problème ;
- b. temps de gestion ;
- c. temps de collection des outils de maintenance ;
- d. temps d'analyse ;
- e. temps d'élaboration des spécifications ;
- f. temps de correction ;
- g. temps des essais locaux ;
- h. temps des essais globaux ;
- i. temps des révisions de maintenance ;
- j. temps de mise en opération.

Il y a plusieurs facteurs qui influencent la maintenabilité. Parmi ceux établis par Sneed [Yan03], on retrouve la modularité, la flexibilité, la portabilité et la complexité.

Tableau II

Facteurs influençant la maintenabilité

Modularité	mesure relative à l'étendue sur laquelle un logiciel peut être fragmenté en plusieurs sections indépendantes de l'ensemble.
Flexibilité	mesure relative à l'indépendance du logiciel face aux spécificités de l'application.
Portabilité	mesure relative à l'indépendance du logiciel de son environnement technique.
Complexité	mesure relative du logiciel de son agrégation et de sa composition

1.5.5 Qualité du logiciel

Cependant, selon l'ancien standard ISO/IEC 9126 : 1991, la maintenabilité est l'une des six caractéristiques sur la qualité du logiciel.

Tableau III

Caractéristiques sur la qualité du logiciel ISO/IEC FCD 9126 – 1991

Capacité fonctionnelle	Attributs du logiciel portant sur l'existence d'un ensemble de fonctions et de propriétés données. Les fonctions sont celles qui satisfont aux besoins exprimés ou implicites des utilisateurs.
Fiabilité	Attributs du logiciel portant sur l'aptitude à maintenir son niveau de service dans des conditions précises et pendant une période déterminée.
Facilité d'utilisation	Attributs du logiciel portant sur l'effort nécessaire pour l'utilisation et sur l'évaluation individuelle de cette utilisation pour un ensemble défini ou implicite d'utilisateurs.
Rendement	Attributs portant sur le rapport existant entre le niveau de service d'un logiciel et la quantité de ressources utilisées dans des conditions déterminées.
Maintenabilité	Attributs portant sur l'effort nécessaire pour faire des modifications données.
Portabilité	Attributs portant sur l'aptitude du logiciel à être transféré d'un environnement à un autre.

Chaque caractéristique sur la qualité du logiciel selon le standard ISO/IEC 9126 : 1991 comporte des sous-caractéristiques. L'ensemble des caractéristiques forme la qualité interne (qualité de la conception et aptitude à évoluer) et externe (comportement lorsqu'il est utilisé) du logiciel.

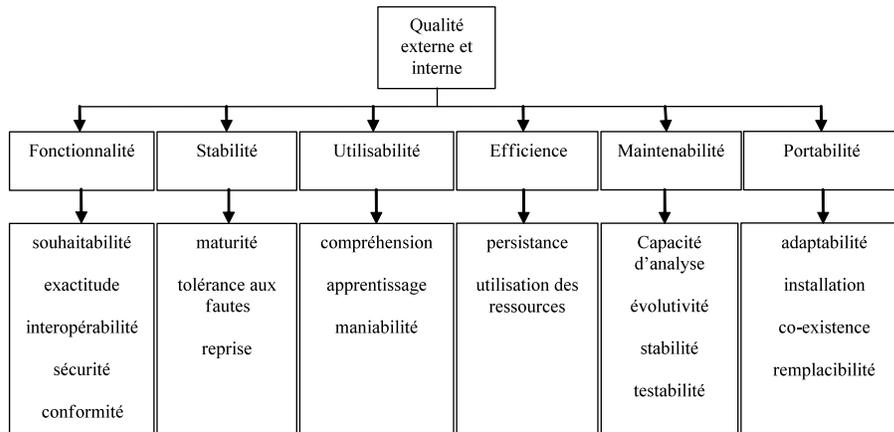


Figure 13 Sous-caractéristiques d'ISO/IEC 9126 – 1991

Tableau IV

Sous-caractéristiques et définitions ISO/IEC FCD 9126 –1991

Caractéristiques	Sous-caractéristiques	Définitions
Fonctionnalité	souhaitabilité	Attributs du logiciel relatifs à la présence et à la convenance d'un ensemble de fonctions pour des tâches indiquées.
	exactitude	Attributs du logiciel relatifs à la production de résultats ou d'effets justes et convenus.
	interopérabilité	Attributs du logiciel relatifs à sa capacité d'interagir avec d'autres systèmes.
	sécurité	Attributs du logiciel relatifs à sa capacité d'empêcher l'accès non autorisé, accidentel ou délibéré aux programmes ou aux données.
	conformité	Attributs du logiciel le liant aux normes, aux conventions ou aux règlements dans des lois et des prescriptions similaires.
Stabilité	maturité	Attributs du logiciel relatifs à la fréquence de l'échec par des défauts dans le logiciel.
	tolérance aux fautes	Attributs du logiciel relatifs à sa capacité de maintenir un niveau indiqué d'exécution en cas de défaut du logiciel ou de son interface.
	reprise	Attributs du logiciel relatifs aux possibilités pour rétablir son niveau d'exécution et pour récupérer en cas d'échec les données directement affectées au moment et l'effort.

Utilisabilité	compréhension	Attributs du logiciel relatifs à l'effort des utilisateurs pour identifier la logique de fonctionnement et de son applicabilité.
	apprentissage	Attributs du logiciel relatifs à l'effort requis pour apprendre l'application.
	maniabilité	Attributs du logiciel qui concernent l'effort pour l'opération et le contrôle.
Effcience	persistance	Attributs du logiciel relatifs à la durée du temps de réponse, de traitement et aux taux de performance.
	utilisation des ressources	Attributs du logiciel relatifs à la quantité de ressources utilisées et à la durée d'une telle utilisation en exploitation.
Maintenabilité	capacité d'analyse	Attributs du logiciel relatifs à l'effort de diagnostic d'erreurs ou de causes, ou pour l'identification des sections à être modifiées.
	évolutivité	Attributs du logiciel relatifs à l'effort pour la modification, à la correction ou au changement environnemental.
	stabilité	Attributs du logiciel relatifs au risque d'effets inattendus après des modifications.
	testabilité	Attributs du logiciel relatifs à l'effort pour le valider une fois modifié.
portabilité	adaptabilité	Attributs du logiciel relatifs à l'opportunité pour son adaptation à différents environnements sans appliquer d'autres actions ou moyens que ceux fournis à cette fin par le logiciel considéré.
	installation	Attributs du logiciel relatifs à l'effort pour l'installer dans un environnement indiqué.
	coexistence	Attributs qui font que le logiciel adhère aux normes ou aux conventions de portabilité.
	remplaçabilité	Attributs du logiciel relatifs à l'opportunité et à l'effort nécessaire pour intégrer d'autres environnements.

La norme ISO/IEC 9126 a évolué depuis 1991. Les travaux de révision ont rajouté la sous-caractéristique **attractivité** à la caractéristique qualité « Utilisabilité ». De plus, chaque caractéristique qualité a la sous-caractéristique **conformité** [Pin02].

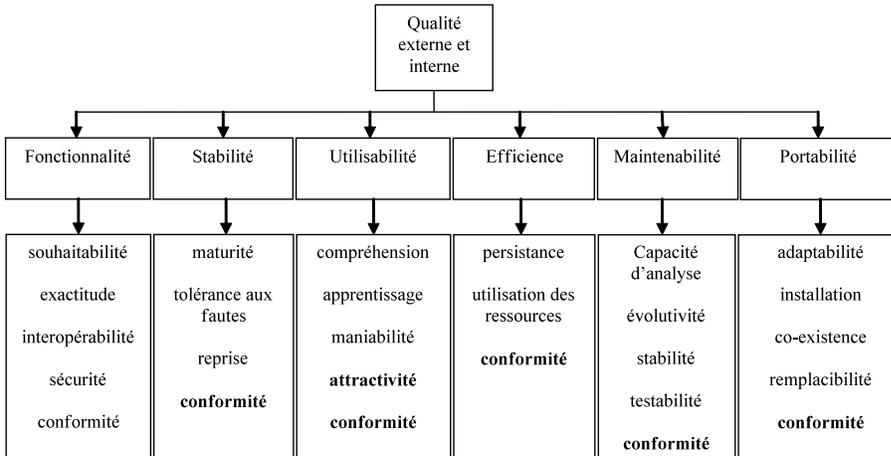


Figure 14. Sous-caractéristiques, évolution d'ISO/IEC 9126-1:2001

La nouvelle norme en révision comporterait également un ensemble normatif en quatre parties [Pin02] :

- a. 9126-1 Modèle qualité ;
- b. 9126-2 Métriques externes ;
- c. 9126-3 Métriques internes ;
- d. 9126-4 Qualité en utilisation.

La qualité du logiciel est fonction des attentes des acheteurs, des développeurs, des mainteneurs, des exploitants et des responsables de la qualité. De sorte que les attributs de qualité n'ont pas la même importance à chaque stade du cycle de vie du logiciel. Pour l'utilisateur, seules les qualités d'efficacité, de productivité, de sécurité et de satisfaction sont prises en compte.

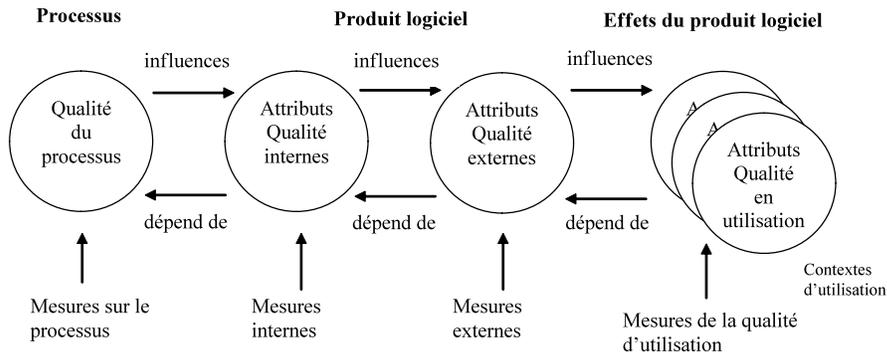


Figure 15 Qualité du produit dans le cycle de vie d'ISO/IEC 9126 en révision [Pin02]

1.6 Les risques reliés à la réingénierie

Le choix du modèle de cycle de vie³ peut être conditionné par plusieurs facteurs de risques :

- a. les propriétés désirables ou indésirables du système existant ;
- b. la capacité à éliminer ou à réduire les propriétés indésirables du système existant ;
- c. la maturité des technologies insérées dans le système existant ;
- d. l'introduction de nouvelles technologies à supporter par les programmeurs ;
- e. l'impact de l'introduction de nouvelles parties au système existant ;
- f. l'impact des changements sur la performance et la robustesse du système existant ;
- g. les coûts et la durée de la réingénierie.

³ Voir section 1.4 Modèles de cycle de vie du logiciel

Voici quelques exemples de risques⁴ avec et quelques contre-mesures :

- *défaillance du personnel*: embauche d'un personnel de haut niveau, adéquation entre profil et fonction, esprit d'équipe, formation mutuelle, personnes clés ;
- *calendrier et budget irréalistes*: estimation des coûts et du calendrier, développement incrémental, réutilisation, élagage des besoins ;
- *fonctions inappropriées*: analyse de l'organisation, analyse de la mission, formulation des concepts opérationnels, revues d'utilisateurs, manuel d'utilisation précoce ;
- *interfaces utilisateurs inappropriées*: scénarios et revues d'utilisateurs, analyse des tâches;
- *sur qualité*: élagage des besoins, analyse des coûts et bénéfices, conception tenant compte des coûts, maquettage ;
- *volatilité des besoins*: développement incrémental avec les derniers incréments les plus changeants ;
- *composants externes manquants*: inspections, essais et mesures, analyse de compatibilité ;
- *tâches externes défaillantes* : audit, contrat avec bonus, revues fréquentes ;
- *problème de performance*: simulation, modélisation, essais et mesures, prototypage ;
- *technologie inadéquate*: analyses techniques de faisabilité, maquettage.

Dans le rapport technique du Software Engineering Institute [Ber99], on répertorie selon leurs études de cas, les 10 principales causes d'échecs de la réingénierie dont voici les résumés :

⁴ réf. : <http://www.infeig.unige.ch/support/se/lect/gl/models/web.html>

1. L'organisation adopte par distraction une mauvaise stratégie ou une stratégie incomplète de réingénierie.

Un exemple de mauvaise stratégie consiste à « remplacer » plutôt qu'à « réparer » un sous-système important qui par le fait même fait disparaître une partie de la connaissance du processus d'affaires intégrée dans le système patrimonial. Ou encore, une approche « Big Bang » qui ignore comment déployer et faire transiter le système vers une utilisation opérationnelle.

2. L'organisation fait l'utilisation inadéquate des consultants et des entrepreneurs extérieurs.

Elle laisse la gestion du projet au consultant qui ne connaît pas assez le processus d'affaires. De plus, l'organisation veut minimiser les coûts alors que les consultants veulent maximiser leurs revenus.

3. La main-d'œuvre est attachée à de vieilles technologies avec des programmes de formation insatisfaisants.

L'introduction de nouvelles technologies force les changements dans les méthodes de travail ; ce qui peut provoquer de la réticence aux changements chez certains employés et même leur départ en même temps que leur expertise.

4. L'organisation n'a pas son système patrimonial sous contrôle.

Mal documenté, un processus de mise à jour inadéquat, etc. fait en sorte que le système actuel est difficile à comprendre, à gérer et à faire évoluer. Ce qui complique sa réingénierie, car il n'y a pas de bases solides pour comprendre ses fonctionnalités et les implémenter dans le nouveau système.

5. Il y a trop peu d'acquisition et de validation des exigences.

Les sources d'information sur les exigences de base sont absentes du système patrimonial. Il faut reprendre le processus de validation des exigences auprès de toutes les parties prenantes et les mettre à jour.

6. L'architecture du système n'est pas une considération primaire de la réingénierie.

Une évaluation est nécessaire pour déterminer si l'architecture du système patrimonial est viable à sa base pour tout nouveau développement ou refaire l'architecture repartir à zéro.

7. Il n'y a aucune notion d'un procédé séparé et distinct de la réingénierie.

L'approche doit être différente uniquement par le fait que la réingénierie commence par le système patrimonial. Typiquement, ceci implique qu'il faut utiliser l'approche de rétro-ingénierie pour obtenir une compréhension du système patrimonial qui préparera le nouveau développement.

8. Il y a une planification inadéquate ou une décomposition insatisfaisante pour suivre le plan.

Les responsables de la gestion du projet en soi sont trop préoccupés par des problèmes d'implémentation, cette dernière s'en trouve négligée. Si le développement nécessite l'intervention des gens de plusieurs disciplines, une planification adéquate est nécessaire pour bien coordonner le travail. Si le plan reste dans la tête de peu de gens et comme le temps altère la mémoire, le plan devient forcément incomplet.

9. La direction manque d'engagement à long terme.

La direction se doit de soutenir et de soigneusement surveiller le projet et, s'il le faut, le ramener sur ses rails. Si la direction se concentre sur d'autres activités, elle ne sera pas au courant des problèmes.

10. La direction prédétermine des décisions techniques.

Les décisions quant à l'approche technique, le calendrier, les coûts et la considération de performance sont prises par la haute direction sans consultation suffisante de l'équipe impliquée et sans suivi de recommandations de cette dernière ce qui mène souvent à un échec.

Ce qu'il faut retenir de ces dix raisons d'échec, c'est qu'aucune ne concerne directement la technologie, mais plutôt les comportements humains face aux projets de réingénierie. Le facteur humain semble être le facteur déterminant de la réussite d'un projet de réingénierie. Par facteur humain, on entend : l'engagement de la direction, l'expertise du personnel en gestion de projet ainsi que de sa connaissance du système patrimonial, l'implication de ce dernier, etc. sont d'autant de coefficients humains qui justifient la coordination de l'équipe pluridisciplinaire engagée dans un projet de réingénierie.

1.7 Brève perspective

Les systèmes patrimoniaux développés dans des langages tels que COBOL, RPG, etc. aujourd'hui considérés surannés ont pourtant été vus comme avant-gardistes à une autre époque. Il en ira de même pour les systèmes récents développés en langage orienté objet écrit en C++, Java, etc. : ils deviendront eux aussi des systèmes patrimoniaux. Et à leur

tour, ils s'ajoutent à la liste des systèmes qui nécessiteront une réingénierie. Alors qu'on peut comprendre les fonctionnalités et les processus qu'ils supportent en lisant le code des langages procéduraux, il en va autrement des systèmes orientés objets, d'où la nécessité de faire appel à d'autres concepts. La diversité des langages, le manque de formation adaptée, la rotation du personnel chargé de l'évolution de ces logiciels et l'utilisation de langages hybrides qui mènent à des systèmes diversifiés et de plus en plus complexes apporteront d'autres défis au domaine de la réingénierie du logiciel.

1.8 Conclusion du chapitre 1

Ceci termine le premier chapitre sur la revue générale de la littérature du domaine de la réingénierie du logiciel. Suite aux recherches effectuées, trois aspects importants de la réingénierie du logiciel se dégagent : les exigences, la gestion de projet et l'engagement. Les exigences sont la somme des besoins détectés dans la phase de rétro-ingénierie et les exigences découvertes par l'identification de nouveaux besoins. Des exigences connues et précises garantiront la réussite du projet tout en offrant un produit répondant aux besoins d'affaires. La gestion de projet concerne l'approche de gestion utilisée et la rigueur du suivi pour respecter l'échéancier, les coûts et l'avancement des travaux. Finalement, non seulement l'engagement des membres de l'équipe impliquée dans le projet est primordial, mais également l'engagement de la direction et de son implication dans le suivi pour effectuer les réalignements nécessaires. Une autre constatation, personnelle cette fois, concerne le réalisme. Il est illusoire de penser corriger des systèmes patrimoniaux de grande envergure en quelques mois, ou même, en quelques années lorsqu'ils ont évolué sur 10 ou 15 ans parfois même plus. Il faut un certain temps pour reproduire toutes les fonctionnalités existantes, en rajouter de nouvelles et redonner toute la stabilité au produit final.

Au chapitre 2, nous tenterons d'appliquer certains principes de la réingénierie du logiciel acquis par le biais de la revue théorique de la Section 1 en nous servant des méthodes et techniques étudiées pendant les cours du présent projet de maîtrise. En choisissant un système qui cadre bien à l'approche de réingénierie du logiciel, nous produirons les biens livrables jusqu'à l'étape de la réalisation du prototype opérationnel. La conception détaillée, c'est-à-dire la codification complète, les tests, l'implantation et la formation pourront être poursuivis hors cadre de ce projet.

CHAPITRE 2

RÉINGÉNIERIE APPLIQUÉE À UN PROJET

INTRODUCTION

Au chapitre 1, nous avons passé en revue le domaine de la réingénierie du logiciel et de ses principaux concepts : description du système patrimonial, taxonomie, la maintenance, la qualité du logiciel, les modèles de cycles de vie, etc. Dans le deuxième chapitre, nous procéderons à l'application de certaines approches de la réingénierie du logiciel en plus d'utiliser des approches de la réingénierie des processus d'affaires. Cette approche combinée devrait nous permettre de beaucoup mieux comprendre l'environnement de l'application et ses fonctionnalités, de mettre à jour plus rapidement les nouvelles exigences, de diminuer le temps consacré au processus de réingénierie du logiciel. Ceci dans le but de ne pas uniquement moderniser l'application sur de nouvelles technologies et de l'automatiser, mais également de profiter de la réingénierie du logiciel pour optimiser l'ensemble du fonctionnement du processus du système de gestion des cartes de temps.

2.0 Présentation du processus

Si on observe dans notre environnement professionnel, étonnamment, plusieurs processus d'affaires sont faiblement supportés par les TI. De plus, les TI qui y sont impliqués ne sont pas toujours adéquatement utilisés. Pour appliquer certaines approches jumelées de la réingénierie du logiciel (RL) et de la réingénierie du processus d'affaires (RPA), un des processus qui répond le mieux aux critères suivants a été retenu :

- a. l'importance du processus dans l'entreprise ;
- b. le niveau de structuration du processus ;
- c. les TI supportant le processus ;

- d. la taille de l'application logicielle supportant le processus ;
- e. l'âge de l'application.

Le système de gestion des cartes de temps est appelé système Temps Facturé (TF). Il est l'exemple parfait d'un système patrimonial, mais à petite échelle. Le processus de gestion des cartes de temps est au cœur même de la philosophie de gestion de l'entreprise qui prône le partage des ressources entre ses unités administratives. Le processus est peu structuré et repose sur des TI faiblement liés entre eux : feuilles électroniques sur Excel, courriel, numériseurs, logiciels en langage RPG, etc. Le système du TF a plus de 15 ans dans sa version actuelle et les développeurs occupent d'autres fonctions ou ne sont plus employés par l'entreprise.

Le projet est fondé sur l'affirmation qu'il n'y a pas qu'un seul chemin pour parvenir à la mise à jour d'une application logicielle. Avec le modèle de la figure 16, l'hypothèse que pour obtenir un résultat optimum, on doit débiter par la RPA et que celle-ci nous conduira à la réingénierie du logiciel selon deux chemins possibles en fonction du système existant est formulée. Si le processus existant n'est pas ou peu supporté par les TI, il est possible de passer de la RPA directement à l'ingénierie du logiciel (parcours A). Alors que si le processus est moyennement ou fortement supporté par les TI, le deuxième chemin consiste à travailler sur l'amélioration du processus d'affaires et de passer par la suite aux activités usuelles de la réingénierie du logiciel pour répondre à tous les besoins y compris les besoins non explicites (parcours B). Une troisième possibilité s'offre à nous, elle n'est pas vraiment de la réingénierie et elle fait plutôt figure de syllogisme, elle consiste à réorganiser le processus d'affaires sans modifier le système logiciel (parcours C). Il s'agit de travailler, à titre d'exemple, sur l'environnement externe du système logiciel en éliminant des procédures manuelles, en changeant des méthodes de travail et certaines activités dans le temps.

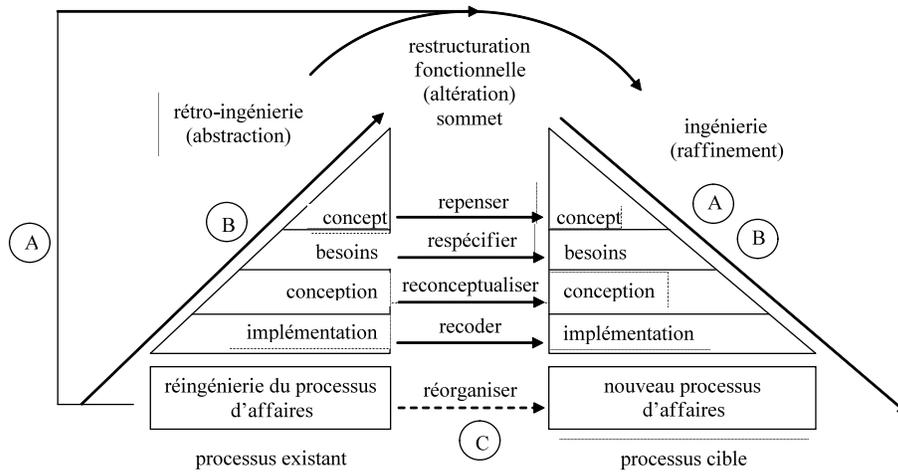


Figure 16 Modèle de réingénierie proposé

Il est indiqué dans plusieurs travaux, dont celui de [Chi90], que la réingénierie du logiciel est l'occasion pour une entreprise de réévaluer généralement le processus d'affaires et d'entreprendre les modifications pour rendre le système d'information conforme à ces changements dans l'entreprise. De plus, il est rare que la réingénierie d'un système d'information se fasse sans l'ajout de nouvelles fonctionnalités. [Gra02] indique que les TI ne peuvent pas à elles seules soutenir les changements organisationnels et d'un autre côté, il faut avoir une vision plus large de la réingénierie des processus pour également inclure les clients, les outils, techniques, méthodes, les gens, la réglementation, etc. Dans les pages suivantes, nous nous sommes appliqués à analyser largement le processus de gestion des CT basé sur l'ensemble du système de travail [Sta01].

2.1 Approche méthodologique

L'approche méthodologique pour la réalisation du modèle de la figure 16, s'appuie sur l'utilisation des techniques utilisées dans le domaine de la RPA et de celles de la RL. En utilisant une approche combinée, nous croyons possible de réaliser une économie de temps sur l'ensemble du projet par rapport à d'autres approches de réalisation. Cette affirmation est basée sur le fait, que dans la situation bien précise de la gestion des CT, les pratiques d'affaires se passent plus souvent dans la tête des utilisateurs que dans le code. À la limite, la quantité d'informations dans la tête des utilisateurs par rapport à celle décrite dans la documentation pourrait bien servir de métrique quant à l'indice d'utilisation de cette méthode.

La méthodologie de réingénierie possède une approche en 3 phases :

- 1^{re} phase : Réingénierie du processus d'affaires par l'analyse du système de travail de la gestion des CT. Il s'agit de regrouper l'information sur le processus (procédures, conventions, culture, les clients, etc.).
- 2^e phase : Rétro-ingénierie par le recouvrement de la conception du système de gestion des CT. Découvrir des exigences fonctionnelles additionnelles contenues dans le code en prévision du passage à la conception et à la réalisation
- 3^e phase : Ingénierie du système de gestion des CT. La méthode de développement par prototype a été choisie à cause de l'absence des exigences définies directement par les usagers.

2.2 Réingénierie du processus d'affaires

Pour effectuer la réingénierie du processus d'affaires du système de gestion de CT selon le modèle de réingénierie de la figure 16, l'approche est basée sur le modèle de [Byr92] et du cadre de l'analyse par le système de travail⁵ [Alt99] [Alt01]. L'analyse du système de travail des CT consiste à l'étude plus large du processus selon les six éléments suivants :

- a. les clients internes ou externes du système de travail;
- b. les produits et/ou services;
- c. le processus d'affaires;
- d. les participants au processus;
- e. l'information dans le processus;
- f. les technologies de l'information.
- g. le contexte
- h. l'infrastructure

L'analyse du système de travail a permis d'obtenir l'instantanée [Alt02] contenant les six éléments précédemment énumérés que l'on retrouve à l'annexe 4.

2.2.1 Description de l'entreprise

Cascades inc. est implantée principalement en Amérique du Nord et en Europe. Il y a plus de 15 600 employés répartis dans 140 unités opérationnelles. Cascades recycle annuellement plus de deux millions de tonnes de papier et carton, comblant ainsi une grande partie de ses besoins en fibres.

⁵ Un système de travail est un système dans lequel des participants et/ou des machines réalisent une ou plusieurs activités d'un processus en utilisant de l'information, de la technologie et d'autres ressources pour produire des produits et services extrants aux clients internes ou externes. [Alt99]

Elle est un chef de file en matière de fabrication, de transformation et de commercialisation de produits d'emballage : carton plat, cartons caisses, produits d'emballage spécialisés, papiers fins et de papiers tissus.

2.2.2 Contexte

Le siège social de Cascades inc. joue un rôle de centralisateur pour une dizaine d'usines du groupe situées à Kingsey Falls. Dans le souci de minimiser les coûts de gestion de ses usines, le siège social veille à offrir certains services en fournissant et en gérant les ressources en main-d'œuvre nécessaires en fonction des demandes.

Les services offerts aux divisions sont réalisés par des employés temporaires et des étudiants. Ces derniers doivent remplir des cartes de temps (CT) pour les travaux effectués. Ils sont rémunérés par Cascades inc. (siège social) en fonction de cette CT. Les heures travaillées dans les usines sont refacturées à ces mêmes usines par Cascades inc. Du point de vue financier elles sont autonomes, c'est-à-dire, qu'elles doivent payer pour les services procurés. Les services offerts vont de l'entretien paysagé, de l'entretien ménagé, du déneigement, du remplacement de commis et de réceptionnistes, etc.

2.2.3 Description simplifiée du processus

Le processus de gestion⁶ des CT est un des sous-processus inhérents au processus global du système de paye. Il s'adresse principalement aux employés saisonniers, à temps partiel et aux étudiants. Les employés remplissent chaque semaine une CT indiquant les heures travaillées selon le jour de la semaine et à quelle usine.

⁶ Annexe 3

La CT existe sur deux supports. Le support papier est un formulaire à remplir et le support électronique est une feuille Microsoft Excel identique au format papier. Une fois que les employés ont complété l'un ou l'autre des formulaires, ils peuvent l'acheminer à l'adjointe aux RH par trois canaux différents. Le premier canal : un casier prévu à cette fin et situé à l'arrière du siège social. Le deuxième canal : la remise en mains propres du formulaire à l'adjointe aux RH. Le dernier canal : l'envoi de la CT à une adresse électronique prévue à cet effet.

Le lundi, les CT sont regroupées, classées par ordre de n° d'employé et saisies à l'ordinateur sur le système TF par l'adjointe aux RH. Après la saisie, les renseignements alimenteront le système de paye et le système des comptes recevables qui sont hors des limites du projet.

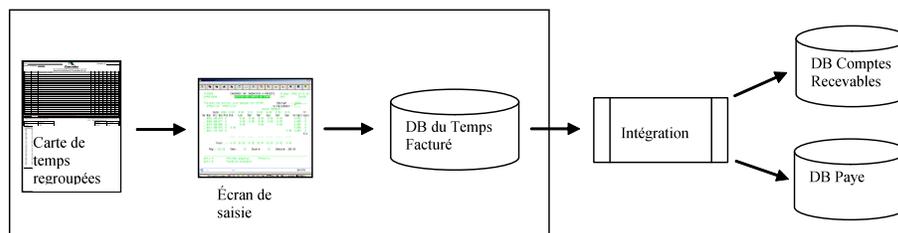


Figure 17 Cycle de vie des cartes de temps

2.2.4 Réingénierie de la gestion des cartes de temps

Le système de gestion des CT, « informatiquement » appelé Temps Facturé, fait partie de ces activités [Por05] qui soutiennent les activités primaires de l'entreprise, soit celui de la rémunération des employés. Il est une des composantes de ces nombreux systèmes qui en coût d'opération prennent 85 % [Ben04] du budget informatique et qui ne sont nullement priorités. Paradoxalement, dans le cas qui nous occupe, les coûts sont très faibles en TI mais élevés en ressources humaines.

2.2.4.1 Explication du problème

Le processus de gestion des CT est un processus **sans propriétaire**⁷. Cette absence de leadership l'a empêché d'évoluer, car le processus répond aux attentes, mais au prix d'une perte de temps colossale doublée d'inefficacité. La perte de temps étant répartie entre plusieurs intervenants sans autorité, aucun ne peut remettre en question son fonctionnement. On peut segmenter le problème en trois volets : la logistique⁸ de la carte, le cycle du processus de la CT et l'utilisation des TI.

A) Logistique de la carte de temps

Ajoutons que les multiples canaux de transmission des CT, les formats de support, le processus repose sur une méthode encore jamais actualisée. Malgré l'utilisation d'ordinateurs, du chiffrier Excel et de l'interconnexion en réseau des différents systèmes, l'information n'est pas partagée ni communiquée efficacement ou rapidement entre les usines et le siège social.

B) Cycle du processus de la carte de temps

Prise individuellement, chacune des tâches accomplies par chacun des employés dans ce processus n'a que peu d'importance comparativement aux autres tâches et responsabilités hebdomadaires.

Pour l'ensemble des employés, le processus n'est pas lié à des fonctions primaires de leur travail quotidien. Le processus de traitement des CT est morcelé, divisé et dilué à

⁷ Annexe 7

⁸ Relatif aux activités de transport, de gestion des stocks ou aux différents moyens d'organisation d'un processus qui ont pour objet d'optimiser les flux matériels traités par une production (source : www.granddictionnaire.com) .

travers différentes organisations. Il n'est pas un sous-processus systématiquement intégré aux autres processus plus complets tels que la paye et la facturation. Le processus n'a pas de propriétaire pour assumer son développement.

Par contre, le processus livre les résultats. Il fait ce qu'il doit faire, c'est-à-dire que l'employé est payé et l'usine est facturée.

C) Utilisation des TI

Le processus n'a jamais évolué dans le temps. Il procède de la même manière qu'il y a 20 ans. Les outils technologiques ont évolué, mais n'ont jamais été incorporés au processus à titre de levier.

2.2.4.2 Évaluation des compétences

Nous avons synthétisé dans le tableau V, les différentes données pour qu'apparaissent clairement les points d'intérêts des acteurs relativement au processus des CT.

Voici les différents constats :

- a. Pour l'ensemble des acteurs, le processus de gestion des CT n'est pas un processus relié à leurs fonctions primaires de travail, sauf le paie maître car ce processus fait partie des intrants de base pour ses fonctions. Paradoxalement, il ne participe pas au processus, mais en reçoit les données ;
- b. La plupart des acteurs clés ont plusieurs années d'expérience. Sauf le contrôleur adjoint qui vient tout juste d'être admis au sein du processus et effectue une fonction importante de contrôle, mais de très courte durée ;

Tableau V

Évaluation des parties prenantes en causes

ACTEURS	FONCTIONS	Ancienneté	Tâche Secondaire ou Principale	Connaissance du processus	Engagement	Motivation	Satisfaction
Employés saisonniers	Produisent les cartes de temps	variable	S	-	+	-	+
Employés temporaires	Produisent les cartes de temps	variable	S	-	+	-	+
Étudiants	Produisent les cartes de temps	variable	S	-	+	-	+
Adjointe aux RH	Saisie des cartes de temps	+ 5 ans	S	-	+	-	-
Commis aux comptes recevables	Incorpore les cartes de temps à la paye et aux comptes recevables	+ 5 ans	S	+	-	-	-
Contrôleur adjoint	Vérifie si les heures payées correspondent à la carte de temps	- 1 an	S	-	-	-	-
Commis adjoint aux comptes recevables	Numérise les cartes de temps	+ 20 ans	S	-	-	-	-
Commis à la paye	Produit la paye	+ 6 ans	P	+	+	-	-
Commis aux comptes payables des usines	Vérifie la facturation reçue pour leur usine	variable	S	-	+	-	-

- c. Les employés qui remplissent les CT ne sont motivés que par leur paye et n'ont pas intérêt au processus ;
- d. L'adjointe aux RH respecte son engagement à saisir les CT dans un délai raisonnable. C'est un travail routinier enchâssé dans son cycle de tâches hebdomadaires. Elle aimerait bien s'en débarrasser ou du moins en voir le volume diminué ;

- e. Le commis aux comptes recevables connaît bien le processus, mais n'en voit que les résultats sur les factures produites et expédiées aux usines et divisions externes. Étant en lien avec d'autres acteurs, il reconnaît certaines tensions à découvrir la source de problèmes de facturation reliés à ce processus ;
- f. Le commis adjoint aux comptes recevables prendra sa retraite d'ici six mois, sa motivation ne cesse de décroître vis-à-vis ce processus ;
- g. Les commis aux comptes payables des usines sont insatisfaits des outils de vérification pour valider les factures reçues par le siège social. Les fichiers en format PDF contenant les CT numérisées sont incomplets et parfois même inutilisables. Certaines factures ne peuvent être vérifiées avec le dernier fichier PDF reçu. Les commis doivent s'en référer à d'anciens fichiers PDF ;

On peut conclure qu'il y a peu de motivation face à ce processus pour l'ensemble des opérants, mais tout ce monde respecte leurs engagements et concourt à le rendre opérationnel. Le leadership est à peu près inexistant. Le style de leadership face à ce processus en serait un de gestion par l'exception. C'est à la contrôleur du siège social à qui revient par défaut la propriété du processus. C'est donc à elle d'en exercer le leadership.

2.2.4.3 Autres facteurs reliés au problème

Croissance : Le nombre d'usines a considérablement augmenté ces dix dernières années. La division de l'horticulture embauche beaucoup d'employées temporaires et d'étudiants. L'horticulture offre ses services aux différentes usines et divisions externes d'aménagement paysager. Le volume des CT à gérer fluctue au courant d'une année en fonction des saisons. La charge de travail demeure peu connue des gestionnaires. Ce phénomène accentue l'impression qu'il s'agit d'un processus de second ordre sans grande influence sur la charge de travail existante.

Degré de structuration : semi-structuré, certaines règles de fonctionnement dans le processus manquent de précision. Il n'y a pas de manuel de procédures.

Implication : trop de personnes sont engagées. Certaines activités du processus peuvent être affectées à une même personne vu la logique des séquences de ces activités et les outils utilisés pour les exécuter. Par exemple, l'activité incorporation des CT dans le système de paye et dans le système de facturation consiste en une option simple du menu au niveau de l'application du TF. Cette activité est affectée au commis des comptes recevables alors que l'adjointe aux RH peut aussi bien effectuer cette activité à la fin de la saisie des CT.

Intégration : le processus manque d'intégration. Il est supporté par plusieurs systèmes d'information (logiciels : Temps Facturé, Paye, Facturation, Messagerie Internet, Acrobat Reader). Il est confié à plusieurs intervenants provenant de services différents. La coordination de l'emploi de l'information sur les CT dans ces conditions est faible (les CT sont triées plusieurs fois selon le système utilisé, elles sont numérisées plusieurs fois selon leur destination et pertinence...)

Degré de dépendance envers les machines : Le processus livre ce que l'on attend de lui à savoir la paye des employés temporaires et les fichiers PDF destinés aux commis aux payables des divisions externes. Cependant, ces derniers dépendent entièrement du matériel utilisé qui numérise les CT et les sauvegarde sur CD ou les envoie par courriel. La capacité de la boîte de messagerie utilisée limite le nombre de CT pouvant être réunies dans un même fichier PDF; il y a aussi la capacité du numériseur (scanner) en nombre de copies papier pouvant être insérées à la fois dans le bac du numériseur.

L'attention portée aux exceptions, aux erreurs et au mal fonctionnement : en observant le déroulement du processus⁹, nous avons constaté qu'il y a arrêt de l'exécution de certaines activités dès qu'un scanner ne fonctionne plus ou que l'employé responsable de cette activité est absent. De ce fait, plusieurs activités s'accumulent.

2.2.4.4 Responsabilités du département informatique

Le système des CT aurait pu à l'origine n'être qu'un sous-module du système de paye. Le système de paye de l'époque appartenait à une firme informatique externe. Les gestionnaires du département d'informatique ont préféré en faire un système parallèle en fonction de leurs besoins et ont développé des options de transfert de l'information vers le système de paye pour conserver son indépendance. La responsabilité d'entretien du logiciel s'est vue soumise à un ballottage entre différents départements informatiques au cours de ses vingt années d'existence. Ce qui a aussi nuit à sa modernisation et moins de 40 heures/année, soit moins de 3200 \$/an y est investi¹⁰.

Pour conclure, le processus de gestion des CT profiterait grandement de l'apport d'une technologie moderne pour simplifier son exploitation.

2.2.5 Solution

L'objectif principal vise à permettre à ceux qui génèrent l'information de la saisir dans le système et libérer ainsi l'adjointe aux RH et l'adjointe aux comptes recevables qui s'occuperont de tâches d'une plus grande importance. Notons qu'il se sera question que du processus de gestion de la CT et non au système Temps Facturé dans son ensemble.

⁹ Annexe 6

¹⁰ Annexe 8

2.2.5.1 Buts

Objectifs reliés au processus :

- a. disposer d'un dépôt unique ;
- b. éliminer les manipulations des CT (regroupement, numérisation, classification) ;
- c. décentraliser la saisie des cartes temps ;
- d. décentraliser la consultation ;
- e. centraliser l'information sur une seule base de données ;
- f. maintenir un historique des transactions ;
- g. simplifier l'accès à l'information ;
- h. augmenter les contrôles (validation, autorisation, confirmation, etc.).

Objectifs reliés à technologie :

- i. optimiser l'infrastructure en place ;
- j. aucun investissement matériel ;

Objectifs reliés à l'apprentissage :

- k. modéliser selon l'analyse orientée objet ;
- l. conceptualiser en fonction d'une interface graphique WEB sur l'intranet de Cascades inc.

Suivent les démarches complètes pour atteindre ces objectifs.

2.2.5.2 Évaluation des activités

Tableau VI

Activités à conserver, à remanier ou à éliminer

Participants	Activités chronologie	STATUT		
		Conserver	Remanier	Éliminer
Employés	Remplir les cartes de temps		X	
	Transmettre les cartes de temps aux Ressources Humaines		X	
Adjointe aux Ressources Humaines	Imprimer les cartes de temps (reçues par courriel)			X
	Récupérer les cartes de temps (du casier)		X	
	Saisir les cartes de temps dans le logiciel Temps Facturé		X	
	Transmettre les cartes de temps au commis adjoint à la gestion			X
	Numériser les cartes de temps en format PDF			X
	Archiver sur disque les fichiers PDF des cartes de temps			X
	Informé le commis aux comptes recevables de la saisie complétée		X	
Commis aux comptes recevables	Incorporer les données des cartes de temps aux comptes recevables		X	
	Incorporer les données des cartes de temps au système de paye		X	
	Production des factures	X		
Commis adjoint aux comptes recevables	Trier pour mettre de coté les cartes de temps non facturables			X
	Numériser les cartes de temps en format PDF			X
	Transmettre les fichiers PDF aux comptes payables des divisions externes			X
	Regrouper et trier les cartes de temps par numéro d'employés			X
Paie maître	Produire la paye	X		
Commis payables des usines	Vérifier les factures		X	

Dans un premier temps, il a fallu déterminer les activités du processus des CT à conserver, à remanier ou à éliminer. L'analyse des dysfonctionnements représentée au tableau VI indique le statut de l'activité obtenu suite à un remue-méninges¹¹, le. On constate que tout ce qui touche la gestion des CT est à remanier ou à éliminer. Or, nous savons que certaines fonctionnalités présentes devront être conservées pendant un certain temps. À titre d'exemple, il y aura toujours des CT papier et remplies à la main. Pourquoi ? Bien que ce puisse être une solution de relève suite à une défaillance, il y a environ 10 % des employés d'entretien¹² qui n'a pas accès à un ordinateur ou peu de connaissance informatique.

¹¹ Annexe 6

¹² Suite à un entretien avec la responsable de l'entretien ménager

2.2.5.3 Détails de la solution proposée

La solution : concevoir un écran de saisie WEB accessible sur l'intranet de l'entreprise et à toutes les usines.

Cette solution est centrée uniquement sur la CT et ne remet pas en question tout le fonctionnement du système de travail. Cette solution passe principalement par une amélioration de l'utilisation des TI, permettant ainsi de régler le problème des dysfonctionnements. Néanmoins, tout le processus des CT est révisé, mais en se concentrant uniquement sur la CT, on délimite nettement le processus de réingénierie. En clair, on ne touche pas au système de paye ni à celui des comptes recevables.

a) Éléments de solution à l'information

Le concept d'une CT est désuet. Un média différent pour communiquer l'information doit être conçu. Une interface graphique interactive de format Web a été retenue. Par contre, notons que l'information présente sur l'actuelle carte est relativement complète. Un numéro d'employé et un mot de passe devraient être utilisés pour permettre un contrôle d'accès à l'information.

b) Éléments de solution à la technologie

Elle consiste à développer une interface permettant aux employés de saisir les renseignements relatifs à leur CT. L'interface développée en HTML devra communiquer les informations au système d'information du TF déjà existantes. Toutes les fonctions de validation permettront de diminuer les erreurs relatives aux dates, nombre d'heures, aux usines à facturer et aux cartes illisibles.

c) Éléments de solution aux systèmes d'information

Une base de données (BD) relationnelle doit être créée. Cette BD permettra de conserver les renseignements et l'historique relatif aux CT. Cette BD pourrait être développée en SQL pour permettre une méthode actualisée du traitement des données. La numérisation des CT est ainsi éliminée du processus et l'information est accessible par tous les participants au système de travail.

d) Éléments de solution à l'égard des employés

L'employé devra saisir l'information directement sur un écran. L'information sera sauvegardée dans la base de données et une preuve de la saisie sera imprimée. Un avis de cette nouvelle CT sera également transmis au superviseur de l'employé requérant ce qui autorisera la CT par le biais d'une autorisation électronique. L'employé n'aura plus à chercher son superviseur pour faire signer sa CT pour ensuite la retourner.

e) Éléments de solution au produit

La notion de carte de temps demeure, mais elle entre dans le 21^e siècle. Le produit s'adapte parfaitement bien aux technologies de l'information.

f) Éléments de solution au processus

L'employé n'aura plus le choix de méthodes multiples pour acheminer les CT, les participants n'auront plus à les numériser. La sauvegarde des CT numérisées sur cédérom ne sera pas plus nécessaire. Par contre, des procédures pour la gestion et le support de la nouvelle interface client et la base de données s'avéreront indispensables. Le processus devra également avoir un propriétaire.

g) Éléments de solution aux participants

- L'adjointe aux RH doit saisir via le WEB les cas d'exemption. Elle autorisera le transfert des CT vers le système Temps Facturé ;
- La commis adjointe aux comptes recevables n'aura plus à numériser les CT ainsi que leur transmission par courriel aux différentes usines ;
- Les commis aux comptes payables des usines pourront voir les CT les concernant avant même de recevoir la facture en consultant par l'intranet les CT.

h) Éléments de solution quant à l'infrastructure partagée

Certains équipements tels que les numériseurs (photocopieurs multifonctions) et la station de brûlage des cédéroms ne seront plus requis.

i) Éléments de solution face au contexte d'affaires

Notre système de travail n'a pratiquement aucune influence sur le contexte d'affaires de Cascades inc. Les solutions proposées ne sont là que pour satisfaire les employés. En outre, ceci nous a permis d'identifier un processus sans propriétaire, mais qui s'avère absolument nécessaire puisque la paye d'employés en dépend.

j) Éléments de solution quant à la réduction du risque

L'élimination de certains équipements tels que les numériseurs (photocopieurs multifonctions) et la station de brûlage des cédéroms permet de diminuer les risques liés aux problèmes d'équipement.

En ce qui concerne les participants, le travail est considérablement diminué par l'élimination de tâches redondantes telles que la numérisation des CT ce qui diminue les risques d'erreurs dues aux nombreuses manipulations.

2.2.5.4 Investissement et les économies réalisées¹³

Ce changement de processus requiert un investissement de 68 000 \$¹⁴ et l'équivalent de 106 jours de travail de conception. Le retour sur l'investissement est de 69 % dès la première année. Ce qui indique la rentabilité de la solution en moins de deux ans¹⁵. Actuellement, le coût de gestion par carte de temps est de 9,62 \$, l'objectif prévu est de diminuer d'environ 50 % les frais de gestion par carte, soit de ramener les coûts à 4,49 \$ par carte de temps¹⁶. Il n'a aucuns frais pour du matériel supplémentaire puisqu'on utilise l'infrastructure existante.

2.2.6 Risques reliés au projet

- Risques financiers

Le coût total de 68 000\$ est basé sur des projections aux coûts réels de l'entreprise. Mais comme la majeure partie est développée dans le cadre du projet d'application de cette maîtrise en technologie de l'information de l'ETI, les frais directs seront nettement moindres. Les coûts difficilement quantifiables sont les coûts en temps reliés à la consultation après de collègues en TI et d'employés en vue d'une meilleure compréhension du système.

¹³ Les coûts sont basés sur ceux en vigueur dans l'entreprise

¹⁴ Annexe 9

¹⁵ Annexe 10

¹⁶ Annexe 8

- Risques techniques

Des équipements déjà existants dans l'entreprise serviront pour l'expérimentation. Si elle échoue ou ne remplit pas les exigences requises, le système des CT utilisé présentement demeure.

- Risques humains

Ce projet n'affecte pas en profondeur l'organisation du travail. Il accélère le processus de gestion des CT pour l'adjointe aux RH et élimine des tâches sans valeur exécutées par le commis adjoint à la gestion. Si certains des employés n'ont pas la formation et les aptitudes pour saisir leur carte de temps, la saisie peut être effectuée par l'adjointe aux RH en continuation avec le processus actuel. Cependant, nous ne voudrions pas que ce projet devienne un 4^e dépôt de la carte de temps pour les employés ce qui ne ferait que compliquer un problème déjà passablement compliqué.

2.2.7 Impacts d'un échec

Malgré tous les efforts consacrés au projet, s'il devait échouer, les impacts financiers, techniques et humains sont quasi nuls. Les connaissances acquises durant le projet offrent une contrepartie suffisamment intéressante pour évacuer tout potentiel d'échec. Comme tout projet, l'expertise acquise s'ajoute à l'expertise antérieure pour faire partie du bagage de connaissances qui à son tour sera réinvesti dans d'autres projets.

2.3 Rétro-ingénierie du logiciel des gestions des cartes de temps

Maintenant que nous avons une excellente idée de l'environnement de ce système de travail et que l'on possède des solutions d'amélioration pour l'ensemble du processus de gestion des CT, nous entreprendrons cette fois, l'étape de la rétro-ingénierie du logiciel

pour découvrir des éléments de connaissance que seule l'étude de l'architecture et du code peut nous fournir.

2.3.1 Méthodologie de la rétro-ingénierie

En faisant référence à la figure 16, suite à l'étape de la RPA, on doit entreprendre la rétro-ingénierie du système logiciel relié à la gestion des CT. C'est l'utilisation du recouvrement de la conception (design recovery) qui sera utilisé pour nous conduire à l'étape de la restructuration fonctionnelle correspondant à la solution proposée et finalement passer à l'étape d'ingénierie du logiciel pour supporter le nouveau processus d'affaires.

2.3.2 Phase de la rétro-ingénierie

Cette phase consiste à étudier le système logiciel en place (programmes, fichiers de données, base de données, etc.) pour bien le comprendre, à colliger les connaissances contenues uniquement dans son code, mais également mettre au jour des contraintes fonctionnelles difficilement contournables. Mais avant même d'entreprendre le travail de rétro-ingénierie, on se doit d'évaluer les actifs informationnels [Aik96] qui nous permettent d'accomplir cette tâche. L'idée est de dresser la liste des sources d'information avant même d'entreprendre le travail. Cette étape est d'autant plus importante qu'en identifiant ces actifs on identifie forcément les actifs manquants.

Quels sont les actifs ?

- a. un accès au code source et aux structures de fichier ;
- b. une connaissance du processus ;
- c. une connaissance de l'entreprise ;
- d. un réseau de contacts à l'intérieur de l'entreprise.

Quels sont les actifs manquants ?

- e. une documentation du système des CT ;
- f. un dictionnaire de données ;
- g. les procédures usagers ;
- h. un accès aux concepteurs d'origine.

Suite à l'évaluation des actifs, considérerons les nouveaux actifs nécessaires à la réalisation de la solution :

Quels actifs devrions-nous acquérir ?

- a. les principes de base de la conception orientée objet ;
- b. une expertise de l'environnement WEB ;
 - a. Serveur HTML ;
 - b. Langage HTML ;
- c. acquisition des outils de développement, etc.

Quels biens livrables devront être produits pour permettre la réalisation technique ?

- a. diagramme de classes ;
- b. diagramme de flux de données ;
- c. modèle relationnel de données ;
- d. dictionnaire de données ;
- e. description des cas d'utilisation ;
- f. plan de navigation et des dessins d'écran.

En l'absence quasi totale d'information sur la conception du système de gestion des CT, l'utilisation du code, de la documentation, de l'expertise professionnelle sur le système et finalement de l'analyse du système de travail élaboré à la première partie de cette section permettra de retrouver l'abstraction du système de gestion des CT.

2.3.2.1 Outils disponibles pour le recouvrement de la conception

Logiciel Visio : En définissant une connexion ODBC, on peut extraire la structure des fichiers pour reproduire en partie le modèle relationnel.

Chiffrier Excel : En définissant une connexion ODBC, on peut extraire les instanciations des fichiers.

Debugger RPG : Permet de faire la trace durant l'exécution d'un programme et de visualiser le contenu des variables pour accélérer et mieux comprendre la logique d'exécution.

2.3.2.2 Activités de la méthodologie de recouvrement de la conception

Dans cette partie, un travail de cueillette et d'analyse d'information est fait. Pour chaque activité, des observations sont inscrites pour préciser certains constats en vue de produire le modèle de la solution.

Listes des activités :

- a. recensement des sources (annexe 13) :
 - programmes de menus (MNU) ;
 - programmes de commandes (CMD) ;
 - programmes de contrôles (CLP) ;
 - programmes (RPG) ;
 - définitions des fichiers et des index (DDS) ;
- b. recensement des imprimés d'écran (annexe 14) ;
- c. recensement des rapports (annexe 15) ;
- d. recensement des objets contenus dans les bibliothèques d'exploitation (annexe 16) ;
- e. reconception du diagramme de flux de données DFD (annexe 17) ;
- f. reconception du diagramme du modèle de relation de données MRD (annexe 18) ;
- g. reconception du diagramme du modèle conceptuel de données MCD (annexe 19) ;
- h. instanciation des données dans un chiffrier ;
- i. analyse de la carte de temps par zone de saisie (annexe 1) ;
- j. analyse de la structure des fichiers (annexe 14 pour TFTEMP) ;
- k. analyse du code source des programmes ;
- l. sommaire des observations par activités de la méthodologie.

2.3.2.3 Observations sur le code source¹⁷

Il existe trois menus principaux (MNU) : TFAOP1, TF001M et TF100M :

¹⁷ Annexe 13

TFAOP1 est le menu principal, même s'il est conçu pour l'environnement 36, il aiguille vers les programmes de la version en exploitation (RPG 400 et ILE) (voir l'annexe 14). Il est utilisé pour accéder au menu de la saisie des CT et la gestion de la fin de mois. Il n'est pas directement utile à la saisie des CT, mais intervient plutôt dans la partie financière.

TF001M n'est plus utilisé, mais est toujours accessible. Les programmes sous adjacents proviennent de l'ancienne version dans l'environnement 36, mais ne sont plus exploités.

TF100M contient l'option 1 qui sélectionne le principal programme de la gestion des CT. Les autres options concernent la gestion financière des données figurant sur les CT pour le transfert vers les différents systèmes satellites : la paye, le magasin et les comptes recevables. Les options de TFAOP1 pourraient migrer vers TF100M.

4 programmes de commande (CMD) :

Les programmes de commande permettent la saisie interactive des paramètres servant à l'appel des programmes RPG. Aucune des commandes ne concerne directement la saisie des CT.

10 programmes de contrôle (CLP) :

Le programme de contrôle prépare l'environnement nécessaire à l'exécution du programme RPG qui lui est lié. Le programme CLP lié au principal programme de saisie des CT ne lui transfère aucun paramètre.

43 programmes RPG :

Des 43 programmes, 1 seul concerne (TFU650R) l'intrant et la mise à jour des données contenues sur la CT. Il n'existe aucun programme de recherche sur l'historique des CT. Les programmes de l'ancien système existent toujours sur le serveur As400. Ils ne sont plus utilisés, mais sont toujours présents dans la librairie d'exploitation. Il y a plus de 200 programmes, menus, etc. de l'ancien système. Notez qu'ils ne figurent pas en annexe.

2.3.2.4 Observations sur les imprimés d'écrans¹⁸

C'est l'un des points de départ pour retrouver l'ensemble du fonctionnement du logiciel et établir les liens sous-adjacents. Le 1^{er} écran est relié au menu principal TFAOP1 et contient l'option 1. Menu Temps Facturé (mécanicien) correspond à l'ancienne version utilisée. TFAOP1 a été conçu pour l'environnement 36, il est compatible avec l'environnement 400. Les autres options restent utiles.

Option 1. Intrant et correction des CT du menu TF100M est le principal sujet d'étude, il mène au programme TFU650R. Le premier écran requiert la date du 1^{er} jour de la semaine et semble être une étape inutile, il pourrait donc être intégré à la fenêtre de saisie de la CT.

2.3.2.5 Observations sur le recensement des rapports¹⁹

L'ensemble des rapports concerne la gestion financière de la CT. Aucun rapport n'est relié à la vérification des CT par les supérieurs immédiats qui autorisent les CT. Non plus qu'aucun rapport n'est lié à la consultation des CT par les commis aux payables des

¹⁸ Annexe 14

¹⁹ Annexe 15

usines externes qui vérifient l'exactitude de la facture avec les travaux inscrits sur les CT des employés. Le rapport TFU675R Validation de la carte de temps est celui qui s'inspire le plus du format de la CT.

2.3.2.6 Observations des objets contenus dans la librairie d'exploitation²⁰

Il y a deux versions du système Temps Facturé : la version pour l'environnement 400 et la version pour l'environnement 36. Les sources des deux versions sont séparées dans des librairies distinctes soit : CCASF pour la version 400 et CAS pour la version 36. Cependant, les objets (programmes exécutables, fichiers, index, etc.) sont tous entremêlés avec d'autres objets. Ces librairies contiennent des dizaines d'autres systèmes comptables. Les objets de la version 36 ne sont pas mis en annexe étant donné que cette version n'est plus exploitée par l'entreprise.

2.3.2.7 Reconception du diagramme de flux de données²¹

Ce diagramme permet d'identifier les fichiers entrants et sortants, les limites du système et les systèmes satellites du système de gestion des CT. Cette section a permis de repérer le lien vers le système du magasin qui transite par le fichier TFFACP mais qui n'a pas d'impact direct sur la gestion des CT.

2.3.2.8 Reconception du diagramme du modèle relationnel de données²²

À l'aide de l'option rétro-conception du logiciel Visio, j'ai extrait l'information contenue sur les fichiers, mais seuls des fichiers du programme de saisie des CT (TFU650R) ont été extraits. Plusieurs fichiers s'ouvrent, mais n'ont aucune relation directe avec la CT.

²⁰ Annexe 16

²¹ Annexe 17

²² Annexe 18

2.3.2.9 Reconception du diagramme du modèle conceptuel de données²³

Ce modèle de haut niveau permet plusieurs observations. Tous les employés ne soumettent pas de carte de temps dans le cycle d'une semaine. La CT est utilisée pour payer un employé, pour facturer le travail d'un employé ou pour emprunter un équipement à une usine externe ou finalement pour transférer de l'information vers un système du magasin pour facturer un travail demandé par un ordre de travail (job). Une combinaison des trois occurrences est possible.

2.3.2.10 Analyse de la carte de temps²⁴

Il y a 18 zones d'information dont sept seulement devraient être saisies de façon manuelle, soit les zones :

Tableau VII

Zones d'information nécessaires sur la carte de temps

N°	Entête sur la carte
1	Nom
2	Numéro de l'employé
3	Semaine du
4	Division
5	Section
6	Description, détails
7	Dim, Lun, Mar, Mer, Jeu, Ven, Sam

Les zones de 8 à 18 peuvent toutes être déduites des zones de saisie précédentes. La liste des divisions – section occupe la moitié de la CT est incomplète. La rubrique « La semaine du » est de fait la semaine se terminant le, c'est-à-dire le samedi d'une semaine

²³ Annexe 19

²⁴ Annexe 1

donnée. Les erreurs sur les CT complétées sont nombreuses et doivent être corrigées par l'adjointe aux RH. Elles surviennent surtout durant les semaines ayant des jours fériés.

Des règles de fonctionnement peuvent être déduites de l'observation des CT complétées :

- a. les employés qui remplissent des CT ne sont pas payés pour les jours fériés non travaillés;
- b. les employés qui travaillent les jours fériés sont payés à temps régulier sauf les gardiens et les étudiants;
- c. les gardiens et les étudiants qui travaillent plus de 40 heures voient le temps excédent payé à temps et demi;
- d. les employés de bureau sont toujours payés à temps régulier.

2.3.2.11 Analyse de la structure des fichiers

Comme le DFD de l'annexe 17 démontre qu'uniquement le fichier TFTEMP (annexe 14) est en écriture durant la saisie des CT, seule l'analyse détaillée de celui-ci a été effectuée. Les autres fichiers ne sont que des références utiles pour compléter l'intrant manuel des CT. L'instanciation de tous les fichiers est compilée dans un chiffrier. L'analyse de TFTEMP démontre que 23 des 50 champs ne sont pas utiles à la saisie des CT. Ces champs ont été inclus dans le fichier TFTEMP pour satisfaire d'autres besoins ou des besoins qui n'existent plus. Le concept selon lequel chaque CT est une facture où figure le détail de la facture (rubrique et détail) n'existe pas, mais chaque ligne de la carte est une facture en soi acheminée vers une usine externe (client). Ainsi, chaque CT sert à facturer plusieurs usines à la fois selon le nombre d'heures travaillées pour chacune des usines.

Les champs suivants sont nécessaires à la gestion des CT :

Tableau VIII

Champs obligatoires pour la carte de temps

No.	Champ	Lon	Déc	Type	E/S	Description	Source de la donnée	Détails dans TFU650R (liaison)
1	TSEQU	5	0		S	NO SEQUENCE	TFGENA.TFSEQ	Prochain # seq disponible
2	TNOEM	10		A	E	NO EMPLOYE	carte de temps	#WSEQU sera réinitialisé à 0 chaque semaine. incrémenter dans le programme TFU650R.
3	TNODI	3		A	E	NO DIVISION	carte de temps	#EMPL0
4	TNOSE	3		A	E	NO SECTION	carte de temps	#WSECT (section facturé)
--	TGLFA	10	0			NO G/L FACTURATION	TFCOGP.CGLFA	recherche sur #WDIVI + #WSECT
--	TGLDE	10	0			G/L DEPENSE EQU	TFPOSP.CGLRE	recherche sur RHUM030P.CD@PST
6	TICODE	3	0		E/S	CODE DISTRIBUTION	carte de temps	#WGAIN valider dans TBMD01 (PARM01L)
7	TDTDI	8	0		S	DATE TRAV AAAAMMJJ DIM	carte de temps	JRS(1)
8	THRDI	4	2		E/S	NOMBRE HEURES DIM	carte de temps	#WDIMA
9	TDTLU	8	0		S	DATE TRAV AAAAMMJJ LUN	carte de temps	JRS(2)
10	THRLU	4	2		E/S	NOMBRE HEURES LUN	carte de temps	#WLUND
11	TDTMA	8	0		S	DATE TRAV AAAAMMJJ MAR	carte de temps	JRS(3)
12	THRMA	4	2		E/S	NOMBRE HEURES MAR	carte de temps	#WMARD
13	TDTME	8	0		S	DATE TRAV AAAAMMJJ MER	carte de temps	JRS(4)
14	THRME	4	2		E/S	NOMBRE HEURES MER	carte de temps	#WMERC
15	TDTJE	8	0		S	DATE TRAV AAAAMMJJ JEU	carte de temps	JRS(5)
16	THRJE	4	2		E/S	NOMBRE HEURES JEU	carte de temps	#WJEUJ
17	TDTVE	8	0		S	DATE TRAV AAAAMMJJ VEN	carte de temps	JRS(6)
18	THRVE	4	2		E/S	NOMBRE HEURES VEN	carte de temps	#WVEND
19	TDTSA	8	0		S	DATE TRAV AAAAMMJJ SAM	carte de temps	JRS(7)
20	THRSA	4	2		E/S	NOMBRE HEURES SAM	carte de temps	#WSAME
--	TNOCI	3	0			NO COMPAGNE	GECIEA.GENOCI	l'employeur qui paie
22	TPOST	8		A	E/S	POSTE EMPL GL PAIE	RHUM030P.CD@PST	#WPOST
23	TAUPA	7	4		E/S	Taux Horaire Paie	RHUM090P.T1@HOR	#WTAUPA Recherche no cie + no. employé
--	TMETI	3	0			FONCTION METIER	RHUM030P.NO@FNC	
--	TCLAS	3	0			CLASSE	RHUM030P.NO@CLS	
24	TAUFA	7	4		E/S	Taux Facturation	TBMD01P.C016	si l'employé ne travaille pas pour sa division, il y a un taux de refacturation
--	TCLIE	3	0			# CIE DU CLIENT	TFCOGP.CNOCI	cie qui est facturé
25	TDESC	44		A	E/S	DESCRIPTION TRAVAIL	carte de temps	
27	TSUPP	1	0		E/S	CODE SUPPL FACT	TBMD01P.T555SUP	si code de gain = 2 TSUPPF = 1

De cette liste de champs obligatoires, on peut déduire les fichiers de référence nécessaires pour compléter la CT.

Liste des fichiers obligatoires autres que TFTEMP :

- TFCOGP : fichier des divisions – sections;
- TFGENA : fichier qui contient le dernier n° de séquence pour avoir une clé unique;
- TFPOSP : fichier des numéros de grand-livre par titre de poste de travail (titre de l'employé);
- GECIEA : fichier qui contient le numéro de l'entreprise de l'employeur;
- RHUM030P : fichier des RH contenant l'historique des postes de l'employé;
- RHUM090P : fichier des RH contenant les données salariales.

Des 13 fichiers ouverts, seuls six sont réellement nécessaires à la saisie de la CT. Les fichiers historiques ne contiennent que des renseignements sommaires de la CT et ne

peuvent pas servir pour la consultation ultérieure. D'ailleurs, aucun programme ne permet de les consulter.

2.3.2.12 Analyse du code source sur les programmes

Le programme d'intrant et de mise à jour des CT se nomme programme TFU650R (annexe 14). Il comporte plus de 4 000 lignes de codes excluant le code relié aux écrans et existe depuis 11 ans dans sa 2^e version. Il est toujours considéré comme la nouvelle version comparativement à la version de l'environnement IBM S36. Pour accélérer sa compréhension, nous avons concentré nos efforts sur le module d'ajout d'une CT. Le « débogueur RPG » a permis de parcourir la logique de fonctionnement durant un ajout. Le code n'est pas structuré et les fonctions de validation sont copiées intégralement à plusieurs reprises dans le programme. La majorité du code est relié à d'autres fonctions que la saisie d'une CT d'un employé et il renvoie plutôt au code relié à l'utilisation d'équipement. Le module de validation de l'écran n° 3 (\$VE03), le module d'ajout (1^{re} partie du module \$MAJ03) et celui de modification (2^e partie du module \$MAJ03) sont presque tous identiques. Ensemble ils représentent 30 % du code.

L'analyse a permis également de déterminer que l'ensemble des autres programmes du TF sert principalement à la maintenance des fichiers de référence et à l'aspect financier. Par aspect financier, on entend certains programmes qui ne servent qu'à valider et transférer de l'information vers le système de paye, de facturation et le magasin.

2.3.3 Sommaire de la rétro-ingénierie

Le processus de rétro-ingénierie a permis d'identifier et de comprendre les éléments importants du système de gestion des CT. En nous concentrant d'abord sur le menu qui conduit au programme de saisie d'une CT en ajout, nous croyons avoir considérablement réduit le temps de recherche nécessaire à la compréhension du principal logiciel de

gestion des CT. L'analyse du processus d'affaires au préalable a permis d'acquérir des connaissances à la phase de rétro-ingénierie et une meilleure conception du système de gestion des CT dans sa globalité. Nous en arrivons au constat qu'il existe deux niveaux distincts de gestion dans ce système. Le premier niveau, celui de la gestion physique de la CT et le second est celui de la gestion financière de la CT.

2.3.3.1 Niveau de gestion des cartes de temps

C'est tout le processus autour des différents transferts vers les systèmes de paye et de facturation qui cause problème. Le cœur du processus financier de la CT fonctionne très bien et il est efficace. Or, les tâches du processus alimentant le niveau financier sont inefficaces : CT remplie à la main, nombreuses erreurs d'inscription, perte de l'intégrité des données reliées à la CT, aucun programme de consultation de l'historique pour vérifier la facturation.

La phase rétro-ingénierie a permis de consolider un certain nombre d'acquis sur le système de gestion des CT. La prochaine étape consiste maintenant à passer à l'ingénierie de la solution en fonction des trois éléments suivants :

- a. améliorations de la performance du processus contenu dans l'analyse du système de travail ;
- b. apport de nouveaux besoins fonctionnels ;
- c. contraintes issues de la rétro-ingénierie.

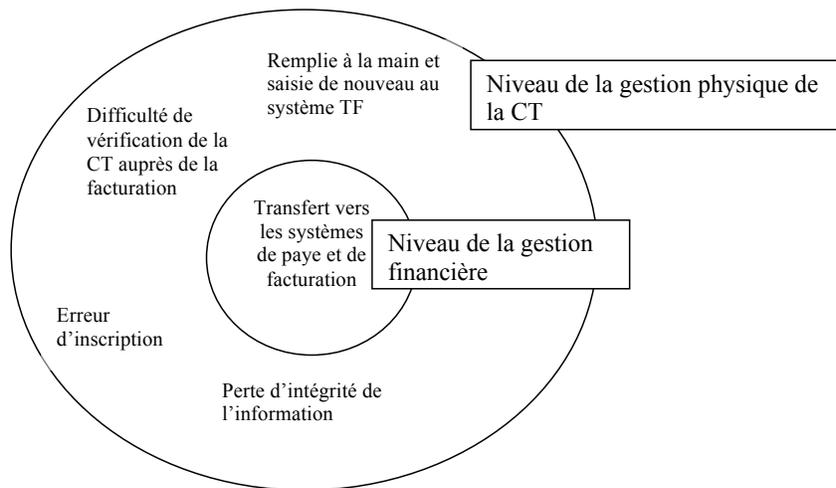


Figure 18 Niveaux de gestion de la CT

À la suite de l'analyse du système de travail des CT ainsi que de la rétro-ingénierie du système logiciel actuellement en exploitation (TF), la solution se dégage. Comme l'a confirmé la rétro-ingénierie, le problème concernant la gestion des CT ne se situe pas au niveau du noyau financier mais bien les activités gravitant autour de celui-ci. C'est pourquoi la solution s'occupe d'abord de ce premier écueil. Nous entreprenons la phase d'ingénierie par la conception d'une solution (voir section 2.2.3).

Pour arriver à optimiser le système des CT, il faut offrir aux employés la possibilité de saisir directement leur CT par l'intermédiaire d'un écran ressemblant à celui d'un guichet automatique accessible sur l'Intranet. Cet écran devra être simple dans sa présentation puisque nous aurons à composer avec un niveau de familiarité varié des employés avec les TI. Ce processus aboutira à une décentralisation de la saisie et à une centralisation des données ce qui libérera l'adjointe aux RH qui pourra vaquer à d'autres tâches cléricales. L'objectif à moyen terme est de saisir les CT par une seule porte d'entrée sur l'interface WEB.

Quatre grandes fonctionnalités doivent se trouver dans la solution :

- a. la gestion de l'intrant des CT ;
- b. la gestion de la sécurité et de l'incorporation vers la DB du TF ;
- c. la gestion des autorisations par le supérieur immédiat ;
- d. la gestion de la consultation par les usines externes.

Ceci oblige l'informatisation de deux nouvelles fonctionnalités : la gestion des autorisations et la gestion de la consultation. Elles remplaceront les méthodes manuelles. L'intrant des CT sera une mise à jour du processus actuel. La gestion de l'incorporation devient nécessaire pour permettre le transfert de la CT saisie via l'application de type WEB sur l'Intranet vers la DB du TF. La gestion de l'incorporation a pour but de conserver l'intégrité des données des CT pour permettre une consultation ultérieure.

Le développement est principalement basé sur le modèle de vie en extension (figure 7). Les fonctions 1 et 2 doivent être développées simultanément, mais les fonctions 3 et 4 pourront attendre des étapes subséquentes. Les quatre fonctions feront l'objet d'un détail afin qu'elles soient prêtes pour leur codification.

2.4 Modèle des cas d'utilisation

Les cas d'utilisation correspondent aux quatre grandes fonctionnalités de la solution. À chaque grande fonctionnalité correspond un actant différent. À l'annexe 24, nous retrouvons les cas d'utilisation détaillés.

- Adjointe aux RH : Gestion de la sécurité et incorporation des CT.

C'est l'administrateur du système. Son code d'accès lui donne l'autorité et les options pour la gestion de l'application. Elle a accès à toutes les fonctionnalités.

Liste des cas d'utilisation :

CU-A1 : S'authentifier.
 CU-A2 : Ajouter un employé ou un superviseur.
 CU-A3 : Modifier un employé ou un superviseur.
 CU-A4 : Désactiver un employé ou un superviseur.
 CU-A5 : Ajouter une CT.
 CU-A6 : Modifier une CT.
 CU-A7 : Annuler une CT.
 CU-A8 : Imprimer une carte de temps.
 CU-A9 : Incorporer les CT Intranet.
 CU-A10 : Imprimer liste sélective des CT.

Ces cas d'utilisation correspondent à ses responsabilités de gérer la base de données des employés qui ont accès au système de saisie des CT WEB, d'intervenir pour corriger et compléter certaines CT et de les incorporer au système du TF après approbation par les superviseurs.

- Employés: Gestion de la CT.

C'est l'employé qui effectue la saisie de ses cartes de temps. La mise à jour est toujours possible avant l'approbation par le superviseur. Les CT demeurent disponibles dans l'historique aux fins de consultation.

Liste des cas d'utilisation :

CU-E1 : S'authentifier.
 CU-E2 : Ajouter une CT.
 CU-E3 : Consulter une CT.
 CU-E4 : Modifier une CT.
 CU-E5 : Imprimer une CT.

Pour qu'un employé puisse utiliser l'application des CT WEB, il y a deux conditions :

- Être inscrit au système de paye;
 - Être inscrit comme utilisateur du guichet des bordereaux de paye ce qui lui permet d'obtenir un mot de passe.
- Superviseur : Autoriser la CT.

Le superviseur d'un employé est celui qui autorise l'inscription de la CT au système pour permettre le transfert des informations vers le système de paye et des comptes recevables.

Liste des cas d'utilisation :

CU-S1 : S'authentifier.
CU-S2 : Autoriser les CT.
CU-S3 : Imprimer une CT.
CU-S4 : Administration.

Le superviseur s'assure de la validité des informations saisies par les employés qu'ils supervisent. Il exige des corrections si nécessaires par l'employé et en dernier recours par l'administrateur du système des CT WEB.

- Commis aux comptes payables : Vérifier les factures avec les CT.

Les commis aux comptes payables s'assurent de la validité des informations contenues sur la facture reçue par le siège social en vérifiant avec la CT WEB saisie par l'employé.

Liste des cas d'utilisation :

CU-C1 : Vérifier facture avec CT.
CU-C2 : Imprimer CT.

Si des erreurs y sont détectées, les changements sont faits au système des recevables du siège social. Aucun changement ne peut se faire via le système des CT une fois que celles-ci sont transférées au système de paye et des comptes recevables.

2.5 La technologie

2.5.1 Choix technologique

La plupart des employés qui remplissent des CT ne sont pas assignés à des postes de travaux fixes et non pas toujours accès au même ordinateur personnel. L'application des CT ce doit être accessible à partir de tous les postes de travail. Pour contourner ce problème de déploiement, une application du type WEB est tout désignée. De plus, l'utilisation d'un navigateur WEB couvre une plus grande étendue d'utilisateurs avec des habilités variées qu'une application spécifique du type client-serveur. La CT est avant tout un formulaire à remplir avec peu de validation à faire durant la saisie. La performance de l'application n'est pas un critère très important en regard de la vitesse de saisie des utilisateurs et de la quantité des CT à traiter.

Sommaires des critères reliés au choix d'une application WEB :

- absence d'un poste de travail fixe pour l'employé;
- évite le déploiement de l'application sur chaque poste de travail;
- familiarité des utilisateurs pour le WEB;
- similitude de la CT avec un formulaire;
- faible volume des CT;
- peu de validation à faire durant la saisie des CT;
- réutilisation de la base de données actuelle.

C'est en décembre 2005 que Cascades inc. a décidé que l'outil pour le développement d'application WEB serait *Visual Studio 2005 WEB development* en utilisant le langage CSharp (C#). Ainsi, le choix de l'outil de développement est fait en fonction du choix technologique de l'entreprise. Visual Studio s'avère également un excellent outil pour le développement rapide d'application (RAD) avec son vaste choix d'éléments dans ses barres d'outils.

L'acquisition d'une application clé en main sur les CT ou d'un « package » quelconque n'a jamais été réellement étudiée pour plusieurs raisons. L'une des raisons principales,

autres que le désir d'expérimenter le développement d'applications WEB, c'est qu'il aurait fallu quand même développer des interfaces logicielles pour permettre l'intégration des informations vers les systèmes de paye et des comptes recevables. De plus, l'offre de logiciel pour l'utilisation directe de la base de données DB2 UDB for iSeries est relativement faible. Cependant, IBM a développé un ensemble de fonctions pour permettre à Visual Studio 2005 en combinaison avec le framework .NET 2.0 d'accéder DB2 UDB for iSeries plus simplement [Bed05]. L'entreprise préconise le développement de ses propres logiciels par les programmeurs de son centre des TI. Il faut garder en mémoire également que l'objectif n'est pas uniquement de moderniser l'application et de l'automatiser en faisant l'acquisition d'un produit disponible sur le marché, mais également de revoir tout le système de travail entourant la gestion des CT.

Comme le choix technologique se fait sur une base pragmatique, le modèle de développement ou du cycle de vie en est fait tout autant. On privilégie l'approche par prototypage pour les raisons suivantes :

- au même titre que le système de gestion des CT n'a pas propriétaire, la volonté de modéliser l'application n'est pas le résultat d'un désir de la direction. Il n'y a pas de vision exercée par un décideur;
- les exigences ne sont pas toutes établies avec certitude par un utilisateur expert;
- Il n'y a pas de modèles existants sur lesquels on pourrait s'appuyer pour comparer le fonctionnement de l'application;
- l'outil de développement se prête bien à l'approche de prototypage;
- explorer rapidement les fonctions à faibles coûts avant d'entreprendre le développement complet de l'application de CT.

Sommaire du choix technologique :

- application WEB;
- développé avec Visual Studio 2005 WEB Development;
- par prototypage.

2.5.2 La conception

2.5.2.1 Liste des biens livrables pour la conception détaillée

Comme l'analyse des besoins et de la solution a été clairement identifiée (section 2.2.3) et que nous connaissons l'architecture actuelle du système des CT, nous passons à l'étape de la conception architecturale de la solution. Voici la liste des biens livrables de la solution qui ont été réalisés la conception architecturale :

- a. diagramme de classes (annexe 20) ;
- b. diagramme de flux de données (annexe 21) ;
- c. modèle relationnel de données (annexe 22) ;
- d. dictionnaire de données (annexe 23).

2.5.2.2 Observations à la suite à la conception détaillée

À la suite à la conception détaillée, quelques problèmes ont dû être résolus :

Problème n° 1 : Parce que les fonctions financières ne sont pas migrées dans la partie WEB du système des CT, le transfert du fichier WEB dans le fichier actuel des CT (TFTEMP) est obligatoire. Une option incorporation permettra ce transfert.

Problème n° 2 : Le support papier n'existant plus, l'obligation de traçabilité des modifications nous oblige à conserver l'historique des changements. Pour respecter ceci, les champs contenant les heures saisies sont en double, soit les heures saisies par l'employé sont transférées dans d'autres champs (heures corrigées) après autorisation du superviseur pour être modifiés s'il y a lieu par l'adjointe au RH. Donc, les heures sont conservées selon deux états : avant autorisation et après autorisation.

Problème n° 3 : La version WEB des CT n'acceptera que des CT pour la rémunération d'employés. Les autres types de cartes comme celles reliées à l'utilisation d'équipement

doivent être gérés par d'autres applications ou directement par le système des comptes recevables (RE) ou par le système de facturation des services (FI).

Problème n° 4 : Un superviseur peut-être également un employé qui remplit lui-même une CT pour des travaux effectués dans une usine externe. Cependant, il ne peut autoriser ses propres CT. Ceci nous force à concevoir deux portes d'accès à la version WEB, une pour la saisie des CT et l'autre pour la supervision et l'administration.

2.6 Descriptions des écrans

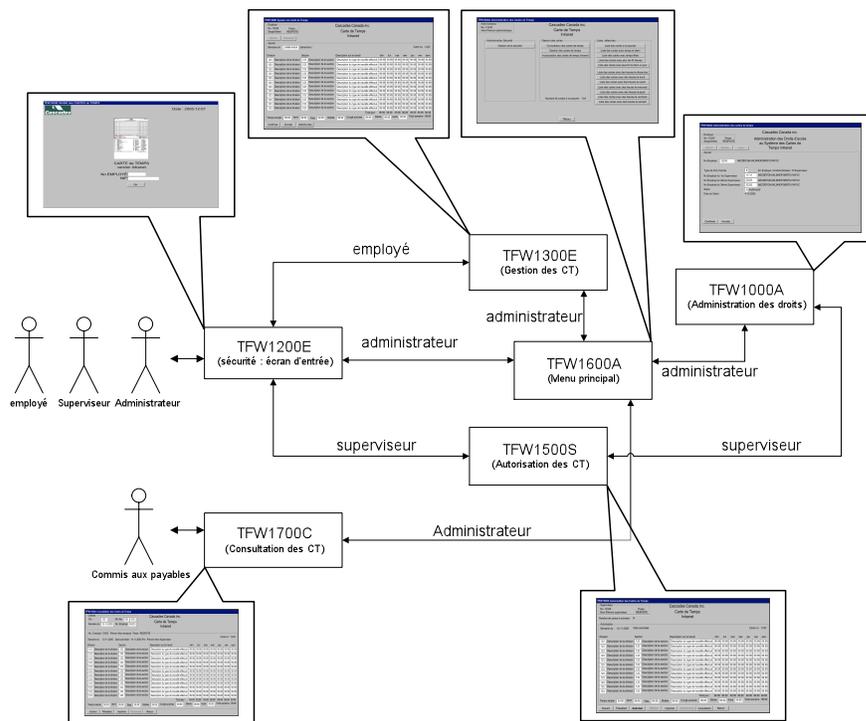


Figure 19 Plan de navigation

2.6.2 Impression des rapports

En ce qui concerne la production de rapports, aucune demande en ce sens n'a été exprimée pour l'application WEB. La production de rapports est toujours possible, mais elle se fait par le système du temps facturé dans lequel les informations CT sont transférées. Dans le fonctionnement présent, l'employé qui remet sa CT n'en conserve aucune copie. Dans la solution proposée, l'employé aura la possibilité d'effectuer l'impression de sa CT dans l'écran de gestion ainsi que l'administrateur et les superviseurs dans leur écran de consultation. L'imprimée de la CT sera identique à l'écran de saisie des CT est accessible par un bouton d'impression. Cependant, comme la philosophie de l'application est de diminuer la circulation de CT en format papier, nous n'encourageons pas l'impression des CT pour ne pas recréer l'un des problèmes que nous voulons résoudre.

2.7 Conclusion du chapitre 2

Dans ce chapitre, nous avons expérimenté le modèle proposé de réingénierie du système des CT. Ce modèle regroupe dans un même cycle la réingénierie du processus d'affaires des CT et celle du système d'information par la rétro-ingénierie (recouvrement de la conception). Ce modèle visait l'exploration de l'environnement humain, matériel et logiciel dans lesquels évolue le processus de gestion des CT. Le processus de redécouverte du schéma initial de ce système d'information est d'autant plus complexe qu'il est semé d'embûches comme la plupart des systèmes patrimoniaux : faiblesse de la documentation et des modèles fournis par les systèmes de gestion de bases de données, présence de structures non déclarées explicitement, code obsolètes et sans oublier que le processus des CT est sans propriétaire. À la suite de l'analyse, nous voulions moderniser l'application vers une solution WEB. Le passage de l'application des CT sous une forme plus moderne ne saurait pas être fait sans heurt. Le chapitre suivant, contient une description de

l'expérimentation. Ce chapitre décrit les étapes de l'ingénierie pour la réalisation de l'application WEB sur les CT. Le chapitre contient un critique sur la rétro-ingénierie des CT et fait la comparaison entre l'application actuelle en langage RPG et la nouvelle version développée pour le WEB. Finalement, l'utilisation d'un questionnaire, rempli par l'utilisateur principal, aide à identifier des améliorations à la méthode de réingénierie proposée ainsi qu'à l'application WEB résultante.

CHAPITRE 3

DESCRIPTION DE L'EXPÉRIMENTATION

INTRODUCTION

Ce chapitre fait le point sur l'expérimentation de la réingénierie des CT. Un retour est fait sur le modèle de réingénierie proposé. On retrouve ce qui a été utile et ce qui l'a moins été. Nous présentons également un tableau des résultats à la suite de l'analyse du questionnaire d'évaluation rempli par l'utilisateur principal des CT. Enfin, une liste d'améliorations est proposée.

3.1 Sommaire du projet

À la suite de la revue de la littérature sur la réingénierie du logiciel, nous avons proposé un modèle de réingénierie qui s'appuie sur la réingénierie des processus d'affaires par la méthode de l'analyse et par le système de travail, et de la rétro-ingénierie par le recouvrement de la conception. Comme le projet sur les CT a décrit, c'est à partir d'un processus peu documenté que nous avons travaillé, qui repose sur la bonne volonté des parties prenantes. Le but étant de retirer le maximum d'information sur le processus lui-même avant d'entreprendre le recouvrement de la conception. Le modèle visait à améliorer tout le fonctionnement du processus des CT plutôt que de simplement re-automatiser le processus avec un nouveau langage de programmation. Le tableau IX démontre que sur le nombre d'heures investies au projet, il y a eu environ 180 heures d'analyse consacrées au système de travail, alors que l'étape du recouvrement de la conception a exigé quelque 80 heures. La majeure partie de l'information, concernant le processus, a été extraite par l'analyse du système de travail, beaucoup d'information venant du recouvrement de la conception était rapidement classée comme inutile pour la clarification du processus des CT.

Tableau IX

Nombre d'heures travaillées

Tableau du nombre d'heures de travail (approximatif)			
		Type de travail	nombre d'heures
2004	novembre	Analyse du Système de travail	40
	décembre	Analyse du Système de travail	80
2005	juillet	Recherche documentaire	20
	août	Analyse du Système de travail	60
	septembre	Rédaction sur la réingénierie	80
	octobre	Rédaction sur la réingénierie	80
	novembre	Recouvrement de la conception	80
	décembre	Rédaction de la conception	120
2006	janvier	Rédaction de la conception	60
	février	Révision - correction - rapport	60
	mars	Apprentissage Visual Studio	120
	avril	Développement WEB	120
	mai	Développement WEB	120
	juin	Révision - correction - rapport	60
	juillet	Développement WEB	120
	août	Développement WEB	120
		Total	1340

L'analyse du système de travail, le recouvrement de la conception et la conception de la solution ont représenté environ 48 % de l'effort total, alors que le développement du prototype de la solution a représenté 52 % de l'effort. L'effort d'apprentissage des différentes composantes à maîtriser pour réaliser une application WEB est bien plus important que ce que nous avons anticipé au départ.

3.1.1 Ordre de réalisation des tâches pour le projet

Modèle proposé pour le projet (ordre des tâches) :

- Réingénierie du processus des CT :
 - analyse du système de travail ;
 - identification du problème;
 - cueillette des documents d'information;
 - entrevues avec les parties prenantes;

- description du processus;
 - description du problème;
 - identification de la solution ;
 - description de la solution;
 - détail sur l'investissement;
- Méthodologie de la réingénierie :
 - recouvrement de la conception ;
 - description des outils;
 - recensement des codes sources;
 - identification du code source important;
 - extraction des données dans un chiffrier;
 - analyse du code source;
 - analyse des données;
 - conception de la solution ;
 - cartographie de la solution (DFD, MRD, etc.);
 - création de la base de données;
 - implémentation de la solution ;
 - programmation du prototype;
 - validation par les utilisateurs.

Si l'approche de l'analyse par le système de travail propose une série de techniques : logigramme, entrevue, remue-méninges, instantanés, etc., il en va différemment du recouvrement de la conception. Néanmoins, l'approche du recouvrement de la conception utilisée dans le projet correspond à celle décrite dans [Dum97]. Cette approche est qualifiée de dépendante du domaine de l'application. Ce modèle utilise conjointement avec le code source procédural de l'application, les connaissances d'un expert pour définir l'application. Cette expertise a été grandement utilisée et intégrée dans la section 2.2 sur l'analyse par le système de travail. L'autre approche de recouvrement de la conception est dite indépendante du domaine de l'application. Sans expert, elle peut être automatisée en utilisant des outils de génie logiciel assisté par ordinateur (GLAO) et le code source. [Dum97] a recensé différentes approches de recouvrement de la conception : Gall, Shin, Yeh et Canfora. Parmi celles-ci, Gall est celle qui se rapproche le plus de celle utilisée dans ce projet. L'approche du projet était de recenser tout le code source relatif aux menus d'écrans, aux programmes, aux rapports, à la création des fichiers physiques et aux index. Ensuite, nous avons identifié

le code source susceptible de nous fournir le plus d'information et de l'exécuter sous contrôle. Les imprimés d'écrans concernant les menus et les écrans de mise à jour de la carte ont été les plus significatifs en information fournie. L'extraction des données contenues dans les fichiers physiques vers le chiffrier Excel s'est avérée également une source d'information très précieuse. En ayant les écrans de saisie des CT et l'instanciation des données dans les fichiers sous Excel, grâce à cette approche il était possible de déduire une bonne partie du comportement du logiciel de traitement des CT. En dernier lieu, l'utilisation du débogueur appliqué au programme de saisie a permis de consolider les connaissances sur l'utilisation des fichiers de références tels que ceux du système de paye et des recevables.

3.2 Critique de la théorie de la rétro-ingénierie

Dans cette section, on retrouve après l'expérimentation, les techniques qui ont aidé à la réalisation de la rétro-ingénierie et celles qui ont été moins utiles selon que nous étions dans la phase d'analyse ou d'implémentation. La dernière section contient quelques techniques qui auraient facilité la rétro-ingénierie si elles avaient été utilisées.

3.2.2 Ce qui a été utile pour la rétro-ingénierie

La phase de rétro-ingénierie pour l'analyse et l'implémentation a été possible grâce aux éléments suivants :

3.2.2.1 Les activités de la rétro-ingénierie à valeur ajoutée

Avant d'entreprendre la rétro-ingénierie par le recouvrement de la conception, la réingénierie du processus d'affaires par l'analyse du système de travail a permis de faire ressortir le principal problème : la gestion du support d'information de la CT. Les deux approches ont un lien commun qui s'avère utile de partager, soit un expert. Dans le système des CT, l'expertise est diluée à travers plusieurs intervenants, l'utilisation de

nombreuses techniques : le remue-méninges, le logigramme, la fiche des symptômes, etc., qu'on recommande d'utiliser dans l'analyse par un système de travail plus approprié que de ne s'en tenir qu'au recouvrement de la conception. Rapidement, ceci éliminait l'étude de tout le code source pour se concentrer qu'au code principal. Les sources reliées aux écrans, aux menus et aux programmes de mise à jour de la CT ont été les plus significatives en termes d'acquisition d'informations pertinentes.

3.2.2.2 Techniques utiles pour la phase d'implémentation

Le choix de développer une application WEB oblige d'exploiter un ensemble de techniques et d'outils de programmation : journalisation, schéma SQL, encryptions, vues logiques, etc., encore peu utilisés dans l'entreprise, mais qui connaissent une progression rapide.

L'utilisation du logiciel Visual Studio Web Development 2005 a aidé grandement à simplifier le développement de l'application. Même s'il a fallu faire un effort d'apprentissage important, son environnement intégré pour la gestion du code source séparé de l'interface graphique nous a aidés à maîtriser les différentes composantes d'une application WEB.

3.2.3 Les techniques de rétro-ingénierie les moins utiles

Voici les éléments peu utiles lors de la phase de rétro-ingénierie pour l'analyse et l'implémentation :

3.2.3.1 Phase d'analyse

Le recensement de tout le code source n'a pas été d'une grande utilité. La lecture du code source et des commentaires ont révélé peu d'informations nouvelles. Force nous a été de constater qu'il y avait beaucoup de code redondant ou mort (inexploité) dans le logiciel. Le système du temps facturé n'est qu'une passerelle d'information entre la CT,

les systèmes des comptes à recevoir et le système de paye. C'est pourquoi le recouvrement de la conception n'a été que partiel et que nous nous sommes limités à ce qui touchait la carte de temps.

L'étude des rapports a également été peu informative, pourtant un rapport aurait pu servir d'outil de vérification pour la facturation²⁵ (TFU675R - Validation des cartes de Temps) bien que ce dernier ne soit pas disponible pour les utilisateurs. Sa conception est orientée en fonction de son utilisation dans le système des comptes recevables et sert à des fins de contrôle interne au commis. Il n'est d'aucune utilité aux commis aux comptes payables des usines externes pour valider la facturation reçue du siège social.

L'extraction de la base de données par le logiciel Visio a donné des résultats peu concluants. Il n'est pas bien adapté à l'environnement de l'iSeries d'IBM. Aucun autre outil de rétro-ingénierie étant disponible, nous avons dû nous résigner à faire manuellement une grande partie du travail.

3.2.3.2 Phase d'implémentation

Dans la solution initiale, il était prévu de produire différents rapports, mais seule l'impression de la carte sur le modèle à l'écran s'est avérée utile.

L'utilisation du guichet des relevés paye comme guichet de saisie des cartes de temps ne sera pas utilisée. L'exploitation sur un même guichet de ces deux applications s'avère impossible. Il faudrait mettre à la disposition de l'employé l'utilisation d'une souris. Ce qui n'est pas possible pour deux raisons; premièrement, il faudrait refaire le meuble qui contient la console, car il n'y a pas de surface de travail pour manipuler une souris. Et deuxièmement, l'utilisation d'une souris permet d'atteindre l'environnement de la

²⁵ Annexe 15

configuration de la console. Après la phase de test avec le prototype, une solution de rechange sera proposée.

3.2.4 Ce qui aurait été utile pour la rétro-ingénierie

Voici ce qui aurait été utile dans la phase de rétro-ingénierie pour l'analyse et l'implémentation :

3.2.4.1 Ce qui aurait été utile pour la phase d'analyse

Une implication d'une instance décisionnelle comme le responsable des ressources humaines au projet aurait aidé pour obtenir la collaboration des employés. Les employés qui nous ont offert leur collaboration l'on fait sur une base volontaire à l'image de ce processus où tout le monde collabore pour le faire fonctionner sans leadership. Mais obtenir de l'information de la part des autres divisions : commis aux comptes payables et des superviseurs des employés s'est avéré compliqué. Mais avec un prototype à présenter, les commis, superviseurs et employés deviennent plus réceptifs au projet.

3.2.4.2 Ce qui aurait été utile pour la phase d'implémentation

Comme l'organisation n'a pris que récemment la décision que l'outil officiel de développement des applications WEB serait Visual Studio WEB Development 2005, les efforts d'apprentissage auraient dû se faire plus tôt dans la réalisation du projet.

L'ajout d'un développeur WEB aurait été nettement profitable. Mais comme cette ressource n'était pas disponible dans l'organisation, nous avons fait le prototype nous-mêmes. Ce qui ne nous a pas laissé assez de temps pour valider chaque étape du développement du prototype par les utilisateurs impliqués dans le processus de gestion des cartes de temps.

3.3 Comparaison des éléments techniques

Bien que le système de temps facturé qui soutient la gestion des informations contenue sur la carte de temps tourne sur un seul langage, un seul serveur et un logiciel qui émule un terminal, il en va différemment pour l'application WEB. Pour arriver à simplifier le processus des CT, il faut augmenter la complexité technique pour le soutenir.

Voici la comparaison entre l'application actuelle et l'application WEB :

- Du côté matériel, si le code de l'application actuelle tourne sur un seul ordinateur iSeries pour arriver à soutenir l'exploitation de l'application WEB de la CT, il faut que le code s'exécute sur trois ordinateurs différents : la station de travail de l'utilisateur (PC), le serveur WEB (IIS) et le serveur de la base de données (iSeries).
- Du côté logiciel, l'application actuelle des CT utilise uniquement le langage RPG alors que la solution WEB fait appel à une multitude de langages : HTML pour l'affichage des composantes WEB, CSharp pour la logique d'affaires et la validation, JavaScript pour le contrôle local de l'interface WINDOWS et SQL pour l'accès à la base de données sous ADO.NET.

Finalement, bien que l'application WEB nécessite l'ajout de nouveaux fichiers de données, les deux applications partagent la même base de données.

Comme le démontre le tableau X, il y a deux fois plus d'éléments techniques dans la solution WEB que dans l'application actuelle. Le développement de l'application WEB fait appel à une mosaïque de langages pour réaliser l'ensemble des fonctionnalités nécessaires à l'application WEB des CT. D'autres éléments techniques sont également à considérer comme l'aspect du graphisme et le fonctionnement d'une application WEB

Tableau X
Comparaison des éléments techniques

	Application actuelle	Application WEB
Langages	RPG	Csharp
		JScript
		SQL
		HTML
		DB2 UDB for Iseries
		ADO .NET
Ordinateurs	Serveur Iseries (BD)	Server WEB IIS
	PC (émulation Terminal)	Serveur Iseries (BD)
		PC (Fureteur)
Base de données	DB2 UDB for Iseries	DB2 UDB for Iseries

où les pages une fois générées sont effacées du serveur WEB et régénérées à chaque action. Ce qui nous oblige à considérer l'aspect performance avec beaucoup plus de rigueur. Par performance, on entend, en autres, la vitesse d'affichage des pages WEB et du traitement du flot d'information.

L'application actuelle des CT nécessite peu d'intervenants alors que la solution WEB fera appel à plusieurs spécialistes TI pour permettre sa maintenance : spécialistes de serveurs WEB, de l'iSeries, de la sécurité informatique, des programmeurs WEB, etc.

3.4 Questionnaire et évaluation des résultats

Le questionnaire²⁶ a permis, à la suite d'une démonstration de son fonctionnement, de valider la réceptivité du principal utilisateur face au prototype. Le questionnaire visait à mesurer quatre thèmes quant à la qualité du logiciel : l'apparence, la fonctionnalité, la conformité et la performance. Tous ces thèmes ont reçu des notes positives.

²⁶ annexe 25

Tableau XI
Résultat suite au questionnaire

	Niveau d'amélioration	Notes
Apparence	+	Identique à la carte de temps (format papier) alors que l'application du TF n'est pas en format graphique.
Fonctionnalité	+	Plus de rapports de contrôle disponibles
Conformité	+	Conforme aux nouvelles exigences de sécurité et de conservation des informations.
Performance	-	La saisie est plus lente, mais cette responsabilité est transférée vers l'utilisateur de la carte de temps.

Voici les principaux commentaires de l'adjointe aux ressources humaines retenus durant la présentation :

- Il permet la saisie des minutes sous les formes : .00, .25, .50, .75;
- Il reste toujours une faible partie des utilisateurs qui n'auront pas accès à un ordinateur et qui devront remettre une carte de temps;
- Il conserve une grande flexibilité quant aux correctifs. Il y a beaucoup de cas spéciaux qu'il serait inutile d'intégrer dans le logiciel sous forme de validation. Elle préfère permettre aux utilisateurs de saisir leur information et d'assurer d'un meilleur contrôle à travers des rapports d'exceptions;
- L'amélioration de la qualité de l'information par la validation et par le contrôle des superviseurs permettra une réduction des appels au soutien reliés à la surfacturation de certaines divisions;
- Une courte formation devra être nécessaire suivant les habiletés du type d'utilisateur.

3.5 Améliorations futures

Comme le modèle unit deux types de réingénierie complémentaires : celle du processus d'affaires et celle du logiciel, les améliorations doivent couvrir ces deux domaines. En plus des améliorations au modèle d'ingénierie lui-même, il y a les améliorations

possibles à l'application des CT qui ont servi comme processus d'affaires pour tester le modèle de réingénierie.

3.5.1 Amélioration du modèle de réingénierie

Lier dans un même cycle la réingénierie du processus d'affaires et de celui de la réingénierie du logiciel nous apparaît pragmatique. Chacune de ces réingénieries comporte un ensemble de techniques. Structurer l'approche et recenser les différentes techniques propres à chaque étape permettrait d'établir les premières assises d'une méthodologie. Il faudrait expérimenter également le modèle sur quelques réingénieries pour en mesurer les gains basés sur différents critères : précision de l'information, acquisition de connaissances, gain de productivité et de temps de réalisation, etc. De plus, il faudrait évaluer à quel type de processus il s'applique le mieux.

Sommaire des améliorations :

- structurer l'approche ;
- identifier les techniques les plus appropriées pour chaque étape du processus d'ingénierie ;
- proposer à quel type de processus il s'applique ; etc.

3.5.2 Amélioration de l'application WEB des CT

Certaines améliorations touchent l'aspect externe du logiciel alors que d'autres concernent l'aspect interne.

Du côté externe et comme l'application est encore au stade de prototype, plusieurs fonctionnalités pourront être implémentées dans la version finale. Par exemple, la saisie de la CT hors du réseau de l'entreprise apporterait un gain substantiel en productivité tout en offrant plus de flexibilité à l'employé. L'utilisation du courrier électronique dans le processus des CT WEB pour l'envoi de la CT par courriel à l'employé ainsi que la

confirmation de l'autorisation de sa CT par son superviseur ne sont là que quelques fonctionnalités qui augmenteraient l'intérêt de l'utilisation de l'application WEB. Rendre l'application multilingue, bien qu'il n'y ait pas de besoins exprimés actuellement, ceci devrait être la norme de tout développement.

Du côté interne, structurer le logiciel selon les couches d'architecture 3-tiers : la couche présentation, la couche logique d'affaires et la couche d'accès aux données permettra une indépendance entre l'aspect logiciel et matériel pour simplifier la maintenabilité et la portabilité de l'application WEB. La gestion de la sécurité est à améliorer si on veut rendre l'application WEB accessible sur le réseau Internet.

Finalement, il faudra retirer totalement le côté de l'application encore en langage RPG pour l'incorporer dans la version WEB de l'application. La partie en langage RPG de l'application des CT gère l'intégration vers le système de paye et des comptes recevables. Cette partie devrait être beaucoup plus simple à convertir en application WEB du fait que peu d'intervenants sont en cause et de l'expertise acquise sur la version WEB.

3.6 Conclusion du chapitre 3

Le modèle de réingénierie expérimenté ne se veut pas qu'un modèle qui permet de comprendre en profondeur un système d'information; il se veut un modèle se dotant en plus d'un ensemble de techniques pour y arriver. Le modèle de réingénierie n'a utilisé qu'un faible échantillon de techniques possibles qui pourraient être échantillonnées, structurées et identifiées pour être appliquées à des étapes bien précises de la méthodologie. Ce seul projet ne permet pas d'évaluer l'économie de temps réalisé par le modèle proposé. Néanmoins, la méthodologie appliquée aux systèmes patrimoniaux devrait permettre d'extraire plus rapidement l'information d'où elle provient. C'est-à-

dire baignée dans la mémoire de plusieurs intervenants ainsi que dans le logiciel supportant le processus.

En ce qui concerne l'application résultant de la méthodologie appliquée au système des CT, le cycle de développement par prototypage aura permis de mesurer la réceptivité de l'application WEB des CT auprès des employés avant de passer au cycle suivant, soit celui du développement d'une solution logicielle permanente. Nous prévoyons une adaptation progressive de la version WEB de la CT. De ce fait, le retrait progressif de la CT format papier est également attendu. C'est pourquoi une approche par prototypage auprès de quelques utilisateurs bien ciblés devrait nous donner une assurance de succès. Nous comptons sur l'effet d'entraînement qu'offre la décentralisation de la CT pour gagner de plus en plus d'adeptes en attendant que soit obligatoire l'utilisation de la version WEB clairement demandée par la direction des ressources humaines.

CONCLUSION

Si la littérature est nombreuse et accessible pour l'ingénierie du logiciel, il en va autrement pour la réingénierie. Pourtant, la réingénierie est devenue un incontournable dans le domaine du logiciel. À l'intérieur de la réingénierie, le niveau d'automatisation des différentes approches de la rétro-ingénierie ne permet pas à lui seul de moderniser un système d'information. L'apport humain avec sa méthode heuristique reste une part importante de ce processus. Pour augmenter la maturité de l'ensemble de la réingénierie, normaliser le tout dans un référentiel de bonne pratique et rendu accessible à l'industrie du logiciel pourrait permettre des gains substantiels en économie de temps et d'argent au même titre que les référentiels pour la bonne gouvernance ou la gestion de projet informatique font figure de proue. Il n'est pas toujours facile de comprendre les différentes approches de la réingénierie et de les appliquer correctement selon les différents problèmes que nous voulons résoudre. Nous croyons avoir d'avantage besoin d'une méthodologie que d'une simple mécanique qui traduit les applications dans un autre langage. Nos connaissances sur les systèmes d'information patrimoniaux, nous proviennent grandement des gens qui y sont impliqués à le faire fonctionner. Et comme la réingénierie du logiciel se fait rarement sans revoir le fonctionnement complet du système d'information, il est impératif d'analyser tous les éléments autour du système d'information sous l'approche de l'analyse par le système de travail. Dans l'entreprise, l'accès aux sources n'est pas un problème, c'est sa compréhension qui en est un. En combinant ces deux méthodes dans un même cycle sous une méthodologie structurée incluant les outils les mieux adaptés à chaque étape, on pourrait, du moins l'espérons-nous, faire progresser le domaine de la réingénierie des processus d'affaires supporté par les TI tout autant que la réingénierie du logiciel. Déjà, avec la rapidité des changements que l'on vit dans le domaine du logiciel, on se lance déjà dans la réingénierie des applications WEB et avec quel impact sur les systèmes patrimoniaux ?

ANNEXE 1

Description de la carte de temps

- Analyse de la carte de temps par entête de saisie

No.	Entête sur la carte	Description	Améliorations / Remarques
1	Nom	Prénom et Nom de l'employé	avoir deux zones de saisie distinctes : nom et prénom et ajouter le poste.
2	Numéro de l'employé	de 1 à 5 chiffres	Il doit absolument le connaître.
3	Semaine du :	date du dernier jour de la semaine correspondant au samedi	habituellement "semaine du" correspond au 1er jour de la semaine soit le dimanche.
4	Division	Correspond à un lieu physique de travail (usines, bâtiments, etc.)	pas toujours bien connue par l'employé. Permettre d'imprimer les cartes de temps par filtre sur la division.
5	Section	Correspond à un département ou une activité relié à la division	pas toujours bien connue par l'employé. Permettre d'imprimer les cartes de temps par filtre sur la division.
6	Description, détails	Explication du travail effectué	devrait être obligatoire et significatif.
7	Dim, Lun, Mar, Mer, Jeu, Ven, Sam	heures travaillées par division-section et par jour	1h15min correspond à 1.25 (centième) Ajouter total de la semaine.
8	Total :	total du jour	
9	Congé férié	zone d'information uniquement	Ceux qui complètent des carte de temps, n'ont pas à inscrire d'heures les jours fériés s'il ne les ont pas travaillé.
10	Semaine de vacances	saisie le nombre d'heures	
11	Rég. :	nombre d'heures payé en temps simple	avoir une zone de correction et par qui. Option de filtrer ceux qui rentre des heures les jours fériés et ceux qui dépasse 40 heures semaines.
12	Demi :	nombre d'heures payé en temps et demi	avoir une zone de correction et par qui. Seul les gardiens (division-section 143-100) qui travaillent les jours fériés ont du temps et demi.
13	Double :	nombre d'heures payé en temps double	avoir une zone de correction et par qui. (jamais utilisé)
14	Fête :	nombre d'heures payé pour la fête	avoir une zone de correction et par qui. Si travaillé uniquement.
15	Mobile :	nombre d'heures payé pris en congé mobile	
16	C. Autorisé	nombre d'heures payé pour un congé autorisé	
17	Décès	nombre d'heures payé pour absence relié à un décès	
18	Autre :	nombre d'heures payé pour tout autre raison	devrait avoir une zone explicative.

ANNEXE 2

Écran de saisie d'une carte de temps

Écran de saisie d'une carte de temps

TFU650R CASCADES INC INGENIERIE & PROJETS 18 Sept. 2002 14:52:25
 QPADEV0008 GESTION DES CARTES DE TEMPS INFO CAS038

Indiquez vos options, puis appuyez sur ENTRÉE.
 2=Modifier 4=Détruire

Employé : _____
 ou Équipement : _____

Op	Séq	Div	Sec	Ate	Dim	Lun	Mar	Mer	Jeu	Ven	Sam	8/31	Tr.
-	0001	JOB	ATO	0		8.00	8.00	8.00	8.00	8.00			G-005 0
-	0002	JOB	ATY	0		8.00	8.00	8.00	8.00	8.00			G-005 0
-	0374	JOB	ANK	0		8.00		4.00		2.00			G-005 0
-	0611	JOB	ATO	0	8.00								G-006 0
-	0641	118	132	0							4.00		G-005 0

Total . . :					8.00	24.00	16.00	20.00	16.00	18.00	4.00		
Rég. . :					106.00		Demi: .00	Double: .00		Semaine: 106.00			

Fin

Défil-H F2=Créer séquence F3=Sortir
 Défil-B F12=Écran précédent

MA d MW 04/070

ANNEXE 3

Description des phases du processus sur les cartes de temps

Description des phases du processus sur les cartes de temps

Voici les principales phases du processus des CT. Notez que l'application supportant la gestion des CT s'appelle le système du temps facturé (TF). De plus, la notion temporelle entre chaque phase n'est pas indiquée.

Phase 1) Saisir les cartes de temps

Chaque lundi matin, l'employé¹ transmet sa carte de temps complétée aux RH du siège social de Cascades inc. Il la dépose dans le casier prévu à cette fin pour celle en format papier ou en la transmettant par courriel à une adresse électronique prévue à cette fin pour les autres en format chiffré électronique (Excel). Ce sont les deux méthodes officielles de remise. Il est important de noter que certains employés déposent directement leur carte de temps sur le bureau de l'adjointe aux RH, qu'elle soit présente ou non.

L'adjointe aux RH accède la boîte de courriel et imprime toutes les cartes temps reçues. Elle les classe par ordre du numéro d'employé. Habituellement, l'ordre des numéros d'employé correspond au classement des CT par ordre alphabétique des noms de famille des employés dans 95% des cas.

L'adjointe aux RH saisit les CT dans le système du temps facturé (TF). Il faut environ 4 heures pour regrouper, classer et saisir les 200 CT reçues en moyenne par semaine. Ce travail est habituellement complété pour le mardi matin. Les CT sont par la suite déposées sur le bureau de la commis adjointe à la gestion.

Phase 2) Incorporation des CT aux systèmes des recevables et de la paye

Il s'agit de transférer l'information de la base de données du TF vers:

- le système des comptes recevables en vue de produire la facturation
- le système de paye pour produire la paye des employés

Le mardi matin, le commis aux comptes recevables, via des options du système du TF, incorpore les informations de la base de données du TF vers ces deux systèmes. Cette phase ne dure que quelques secondes. Pour chaque système, un rapport de vérification est produit. Ensuite, le traitement de l'information est pris charge par chacun des commis responsables de ces applications.

¹ Employé ayant travaillé dans la semaine finissant le dimanche quel que soit le nombre d'heures

Phase 3) Numériser les cartes de temps

Les mardis ou les mercredis pendant 2 à 3 heures, cette tâche est effectuée. Il s'agit dans un premier temps de séparer les CT « facturables » de celles qui ne le sont pas. Une CT non facturable est une CT complétée par un employé qui a effectué un ordre de travail pour le siège social lui-même et non pas pour l'une des usines externes.

Par la suite, il faut les numériser sur un photocopieur qui génère automatiquement un fichier PDF. Ce fichier est transmis à une liste de distribution (fonctionnalité du serveur de courriel) contenant les adresses de courriel des commis aux comptes payables des usines facturées. Notons que les CT doivent être séparées en deux paquets pour ne pas dépasser la taille maximum des fichiers transmissibles sur le système de messagerie. Dès lors, il faut reclasser toutes les cartes (facturables et non facturables) par numéro d'employé. La commis adjointe aux RE doit remettre les CT à l'adjointe aux RH pour qu'à son tour, elle puisse numériser de nouveau les CT en vue d'être conservé sur CD ROM pour des requêtes futures.

Phase 4) Produire la paye et la facturation

Le mardi ou mercredi matin au plus tard, la paie maître produit la paye. La paie maître prend pour acquis que les données provenant du système du TF sont intégrées.

Le jeudi, la commis aux RE imprime les factures. L'impression des factures peut contenir des informations provenant de différentes applications. Elle n'est pas dédiée uniquement aux informations provenant du système du TF. Environ 6 systèmes, autres que le TF, alimentent le système des RE.

Ces deux tâches sont situées à l'extérieur du système de gestion des CT. Mais si un problème à la paye ou aux comptes recevables survient touchant les données en provenance des CT, nous devons revenir au système de gestion des CT pour comprendre et corriger la situation.

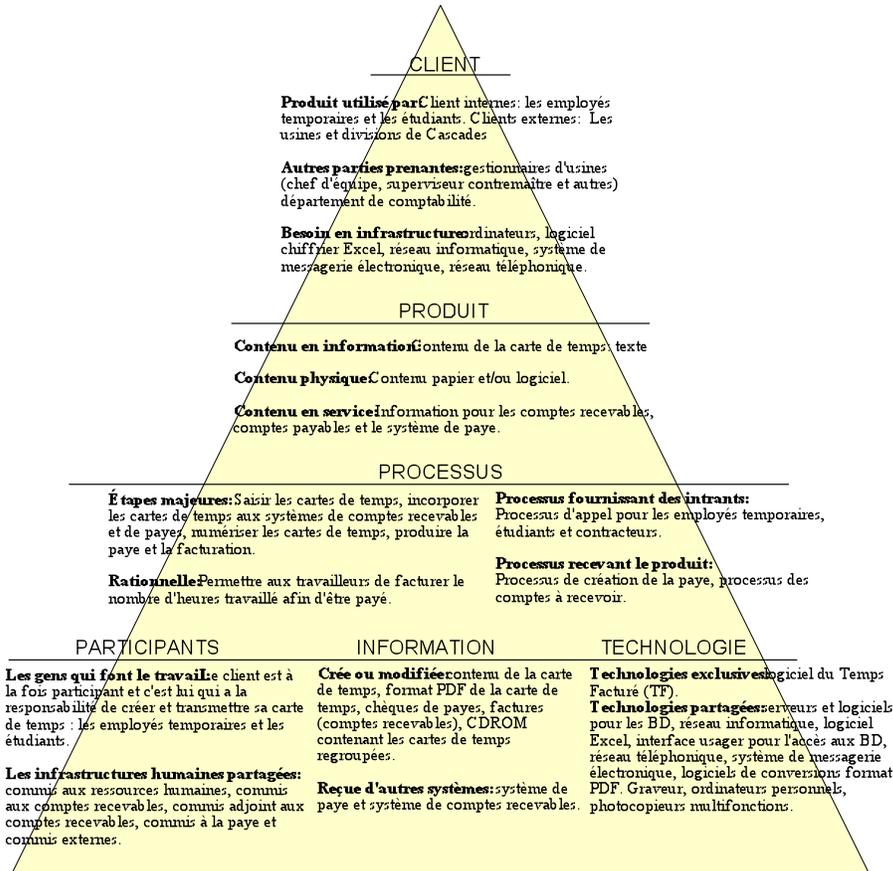
Phase 5) Vérification externe

Chaque semaine, les commis aux PY de chacune des usines facturées vérifient la correspondance entre les factures et les CT qu'il a reçues par courriel sous la forme de fichiers PDF. S'il n'y a pas d'erreurs, ce processus devrait permettre de payer le siège social dans un délai raisonnable et sans erreur.

ANNEXE 4

Instantanée du système de travail des cartes de temps

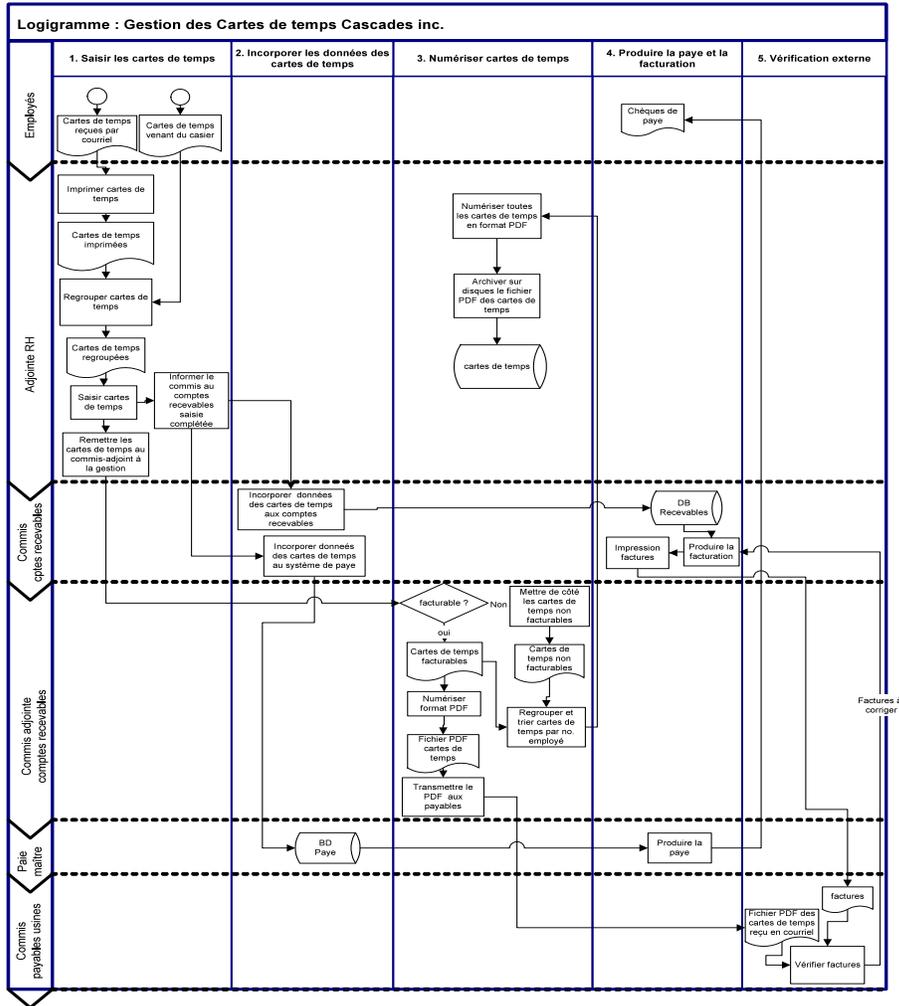
Instantanée du système de travail des cartes de temps



ANNEXE 5

Logigramme sur le système de travail des cartes de temps

Logigramme sur le système de travail des cartes de temps



ANNEXE 6

Remue-méninges des anomalies

Remue-ménages des anomalies

PHASE 1. Saisir les cartes de temps

Activité 1.1 : Transmettre leur carte de temps

Acteurs : Employés

Une fois la semaine de travail complétée, les employés transmettent leur CT de deux manières formelles :

- Ils déposent la CT dans un casier prévu à cette fin;
- Ils transmettent la CT par courriel.

Occasionnellement, certains déposent leur CT directement sur le bureau de l'adjointe aux RH.

Problème(s) : Certaines CT sont égarées et même perdues.

Cause(s) :

- C'est dû à la présence de trois sites de dépôt des CT. Le bureau de la l'adjointe aux RH est un lieu de dépôt qui n'est pas dans le processus formel;
- Les employés qui déposent leur carte de temps sur le bureau de l'adjointe aux RH, s'assurent-ils de sa présence au travail ?
- Si l'adjointe aux RH est absente, il n'est pas certain que la personne qui la remplace va récupérer les CT laissées sur son bureau.

Impact (s) :

- Une autre carte de temps devra être produite et autorisée par son superviseur.
- L'employé ne sera pas payé dans des délais raisonnables.

Activité 1.2 : Imprimer les CT reçues par courriel

Acteur : Adjointe aux RH

Elle accède à la boîte de courriel et imprime toutes les CT reçues. Elle vide ensuite la boîte de courriel des cartes âgées de plus deux semaines.

Problème(s) : Des CT ne sont pas imprimées.

Cause(s) :

- Pas de mécanisme de confirmation de la réception de la carte de temps à l'employé qui l'a fait parvenir courriel.

Impact(s) :

- L'employé devra refaire parvenir sa carte de temps.
- Ces employés ne seront pas payés dans des délais raisonnables.

Activité 1.3 : Regrouper les CT venant des deux sources de dépôt
Acteur : Adjointe aux RH

Elle classe les CT par ordre du n° d'employé. L'ordre des n° d'employés correspond au classement des CT par ordre alphabétique des noms de famille dans 95 % des cas. C'est à dire, les n° d'employés de 1 à 99 sont ceux que leur nom de famille commence par la lettre A, de 100 à 199 sont ceux que leur nom de famille commence par la lettre B, etc. C'est le mécanisme de recherche pour retrouver une carte de temps dans la pile.

Problème(s) : ▪ Difficulté de retrouver une CT dans la pile.

Cause(s) : ▪ Certains employés oubliant leur n° d'employé, la recherche devient difficile à cause de l'ordre des n° d'employés n'est pas fidèle à 100 % à l'ordre alphabétique.

Impact(s) : ▪ L'employé devra refaire parvenir sa carte de temps.
▪ Ces employés ne seront pas payés dans des délais raisonnables.

Activité 1.4 : Saisir les CT
Acteur : Adjointe aux RH

Les CT sont saisies dans le système du TF. Il faut environ 4 heures pour saisir les 200 CT.

Problème(s) : ▪ Certaines CT ne peuvent être saisies à temps à cause de retards, de lisibilités, de manque de précisions, etc.

Cause(s) : ▪ Les CT doivent être saisies au plus tard pour le mardi. On ne peut pas retarder le processus pour les retards. La paye doit être traitée pour le mercredi matin.

Impact(s) : ▪ Des traitements spéciaux doivent être appliqués :

Des options au système du temps facturé permettent *de payer l'employé sans facturer l'usine* en attendant la carte de temps (la preuve du travail accompli). Plus tard, une fois la carte reçue (ex. : une semaine plus tard), on facture l'usine sans payer l'employé.

Activité 1.5 : Informer le commis aux comptes recevables de la saisie complétée.
Acteur : Adjointe aux RH

Habituellement, la saisie est complétée pour le mardi matin.

Problème(s) : ▪ Il arrive que le commis aux comptes recevables incorpore les données des CT à son système de facturation mais que le travail ne soit pas complété.

Cause(s) : ▪ Il n'y pas de mécanisme formel qui informe le commis aux comptes recevables que la saisie est complétée. Il prend habituellement pour acquis que rendu au mercredi le travail de saisie des CT est complété.

Impact(s) : ▪ Des employés ne seront pas payés dans des délais raisonnables;
 ▪ Délais dans la facturation des divisions;
 ▪ Une restauration des fichiers venant « backup » peut s'avérer nécessaire pour reprendre le travail au risque de plusieurs problèmes en aval du processus comme une double facturation des usines et des employés payés en double.

Activité 1.6 : Remettre les CT à la commis adjointe aux RE.

Acteur : Adjointe aux RH

Toutes les CT saisies sont remises par l'adjointe RH à la commis adjointe aux RE. Elles sont déposées dans son bureau.

Problème(s) : ▪ Les CT ne sont pas toujours numérisées à temps.

Cause(s) : ▪ Si la commis adjointe à la gestion est absente, les cartes ne seront pas numérisées en format PDF et transmises aux usines comme pièce justificative de facturation.

Impact(s) : ▪ Les commis aux comptes payables des usines ne pourront effectuer leur rôle de vérification de la facturation transmise par le siège social.

PHASES 2. Incorporer les données des cartes de temps

Activité 2.1 : Incorporer les CT au système des comptes recevables

Acteur : Commis aux comptes recevables

Problème(s) : ▪ aucun

Activité 2.2 : Incorporer les CT au système de paye.

Acteur : Commis aux RE.

Problème(s) : ▪ aucun

PHASES 3. Numériser cartes de temps

Activité 3.1 : Mettre de côté les CT non facturables.
Acteur : Commis adjointe aux comptes recevables.

Elle retire de la pile les CT qui ne sont pas facturées aux usines.

Problème(s) : ▪ Le tri comporte occasionnellement des erreurs.

Cause(s) : ▪ Il n'y a pas de références écrites qui indiquent les CT facturables aux usines de celles qui ne le sont pas. La commis adjointe à la gestion connaît par cœur les usines qui sont facturées;
▪ Elle pourrait numériser toutes les CT, mais le « scanner » n'est pas assez performant et le fichier PDF résultant serait trop volumineux pour être transmis par le système de messagerie électronique.

Impact(s) : ▪ Demande du temps de travail avec peu de valeur rajoutée.

Activité 3.2 : Numériser les CT en format PDF.
Acteur : Commis adjointe aux comptes recevables.

Les CT sont numérisées sur un photocopieur et le fichier PDF est transmis à son adresse de courriel. Mêmes triées, elle doit souvent « scanner » les CT en deux paquets pour ne pas dépasser la taille maximum des fichiers transmissibles sur le système de messagerie.

Problème(s) : ▪ Les CT ne peuvent pas toujours être numérisées.

Cause(s) : ▪ Le « scanner » est occasionnellement défectueux et n'a pas été conçu pour ce volume de page.

Impact(s) : ▪ Les commis aux PY des usines n'ont pas les CT numérisées pour valider la facturation.

Activité 3.3 : Regrouper et trier CT par no employé.
Acteur(s) : Commis adjointe aux RE du siège social.

Reclasser toutes les CT (facturables ou non) par no d'employé. C'est l'inverse du travail fait à l'activité 3.1.

Problème(s) : ▪ Des erreurs de classification par n° d'employé se produisent.

Cause(s) : ▪ C'est la troisième fois que les CT sont à reclasser par ordre de n° d'employé.

Impact(s) : ▪ Rend les recherches ultérieures plus difficiles à cause du manque de rigueur de la classification par n° d'employé.

Activité 3.4 : Expédier le fichier PDF aux commis des PY des usines.

Acteur : Commis adjointe aux RE.

Une fois le fichier PDF reçu dans sa boîte de courriel, elle redirige ce fichier vers le groupe de distribution.

Problème(s) : ▪ Si un commis est absent ou quitte l'entreprise, le nouveau commis ne recevra pas les CT.

Cause(s) : ▪ Il faut maintenir un groupe de distribution (fonctionnalité du logiciel de messagerie électronique) à jour contenant la liste des commis aux comptes payables des usines.

Impact(s) : ▪ Si l'un des commis aux comptes payables des usines est absent, souvent l'usine va demander un nouvel envoi vers la personne qui la remplace dans l'usine.

Activité 3.5 : Numériser toutes les CT en format PDF.

Acteur : Adjointe aux RH

La commis adjointe au RE remet les CT à l'adjointe aux RH pour qu'à son tour elle numérise toutes les CT en vue d'être conservées sur CDROM pour des requêtes futures.

Problème(s) : ▪ Le travail n'est pas fait systématiquement.

Cause(s) : ▪ Retour à la case départ. Ce travail n'est présentement pas fait pour des raisons d'organisation du travail.

Impact(s) : ▪ Du travail qui s'accumule et qui demandera un rattrapage.

Activité 3.6 : Archiver sur disque le fichier PDF des CT

Acteur : Adjointe aux RH

Il s'agit de graver tous les fichiers PDF des CT par année sur CDROM.

Problème : ▪ Le travail n'est pas fait systématiquement.

Cause(s) : ▪ Ce travail n'est présentement pas fait pour des raisons d'organisation du travail.

Impact(s) : ▪ Du travail qui s'accumule, qui demandera du rattrapage.

PHASES 4. Produire la paye et la facturation

Activité 4.1 : Produire la facturation

Acteur : Commis aux comptes recevables

Le commis imprime les factures issues du système de gestion des CT.

Ce travail est situé à l'extérieur du système de gestion des CT. Mais si un problème de facturation survient, nous devons revenir au système de gestion des CT pour corriger la situation.

Problème(s) : ▪ Elle manque d'information pour répondre aux requêtes des employés désirants des informations sur la paye versée.

Cause(s) : ▪ Elle n'a pas toutes les CT en main pour répondre aux questions.

Impact (s): ▪ Délais pour répondre aux questions des employés

Activité 4.2 : Produire la paye

Acteur : Paie maître

Le commis produit la paye issue du système de gestion des CT.

Ce travail est situé à l'extérieur du système de gestion des CT. Mais si un problème de paye survient, nous devons revenir au système de gestion des CT pour corriger la situation.

Problème(s) : ▪ Elle manque d'information pour répondre aux requêtes des employés désirants des informations sur la paye versée

Cause(s) : ▪ Elle n'a pas toutes les CT en main pour répondre aux questions.

Impact (s): ▪ Délais pour répondre aux questions des employés

PHASES 5. Vérification externe

Activité 5.1 : Vérifier les factures par les commis des autres divisions

Actants : Commis aux PY des usines

Les commis aux PY des usines reçoivent dans leur boîte de courriel les factures à payer du siège social et deux fichiers PDF contenant les CT. Un commis vérifiera si la facture correspond aux CT facturées.

Problème(s) : ▪ Les recherches dans les fichiers PDF sont longues et difficiles.

Cause(s) : ▪ Il n'y a pas possibilité d'utiliser la fonction recherche d'Acrobat Reader.

▪ Ils doivent faire une recherche visuelle dans deux fichiers PDF.

▪ Les fichiers PDF contiennent toutes les CT facturables et non seules celles concernant leur usine. Aucun mécanisme de recherche rapide ne fonctionne dans un fichier PDF.

Impact (s): ▪ Quelques fois les CT sont peu lisibles (les CT numérisées ont été remplies auparavant à la main avec un crayon, un stylo, un feutre, etc.)

ANNEXE 7

Fiche des symptômes