

ERIC DARRAS

PLANS DE MIGRATION DE SYSTÈMES PATRIMONIAUX VERS DES ERP

Mémoire présenté
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de maîtrise en informatique
pour l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DE SCIENCES ET GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

SEPTEMBRE, 2004

Résumé

Les progiciels de gestion intégrés (ERP) sont des applications fonctionnant à travers une organisation qui, grâce à leur intégration, automatisent les processus d'affaires de l'organisation. Les ERP se sont rapidement imposés dans l'industrie pour remplacer des applications patrimoniales. La pratique démontre que la majorité des déploiements de ERP dépassent leurs budgets et leurs échéances, par conséquent la recherche s'est mise à étudier les projets de migration pour fournir un cadre de déploiement optimal. Peu d'exemples de déploiement dans de grandes organisations sont documentés rigoureusement. Les applications ERP sont divisées en modules qui doivent être connectés à l'environnement patrimonial de l'organisation pour fournir des avantages concurrentiels. L'intégration des applications de l'entreprise (EAI) est définie comme le processus d'intégration d'applications avec l'environnement informatique. Le déploiement d'applications ERP peut être aisé lorsque les organisations possèdent une structure hiérarchique simple et réalisent des opérations dans un ou quelques lieux géographiques. Mais lorsque les organisations ont une structure hiérarchique complexe et sont géographiquement dispersés, le déploiement implique des choix, des défis techniques et de gestion uniques.

Ce mémoire résume les stratégies de migration, les meilleures pratiques de réingénierie des processus d'affaires, l'analyse des applications patrimoniales et l'influence de la centralisation du pouvoir sur les processus d'une organisation.

Abstract

Enterprise Resource Planning (ERP) systems are enterprise wide systems which, because of their integration, automate many business processes of a company. They have rapidly become the de facto industry standard for replacement of legacy systems. Because there is evidence that the overwhelming majority of ERP implementations exceed their budget and their time allocations, researchers have begun to analyse ERP implementation in case studies in order to provide an implementation framework which maximises efficiencies. Few examples of large organisation implementations are thoroughly documented. ERP applications are divided into modules that must be connected to an organisation's software environment to provide leverage. Enterprise Application Integration (EAI) is defined as the process of integrating enterprise systems with existing applications. Implementing ERP systems can be straightforward when organizations are simply structured and operate in one or a few locations. But when organizations are structurally complex and geographically dispersed, implementing ERP systems involves difficult, possibly unique, technical and managerial choices and challenges.

This research summarises implementation strategies, best practices to reengineer business processes, analyse legacy software and the influence of centralisation of power on organisational processes.

Avant-Propos

Je tiens à remercier mes directeurs M. Belkhiter et M. Badri qui m'ont assisté de leurs conseils tout au long de mes efforts. Les styles de direction et de soutien de mes deux directeurs se complètent et m'ont permis de mesurer le travail à accomplir. Je tiens à exprimer ma gratitude à l'université Laval, et aux trois professeurs qui m'ont rédigé une lettre de référence pour que je sois admis à la maîtrise : M Alikacem, M Caillibot et M Debbabi.

Je tiens aussi à remercier ma famille qui m'a beaucoup soutenu et encouragé, en particulier ma mère et mon frère Xavier. Je remercie Liliane de m'avoir laissé la liberté de faire une maîtrise. Mon travail s'est aussi réalisé grâce aux encouragements et à l'amitié de membres de la communauté universitaire. Je remercie tout particulièrement Denis Poirier, Daniel Bienvenu, et bien d'autres étudiants que j'ai pu côtoyer pendant mes études.

Divers intervenants m'ont aidé dans mes recherches en répondant aimablement à mes demandes d'entrevues notamment : M André Armstrong, M Louis Bastarache, M Jean Bédard, M Hugues Bélanger M Dominique Bérubé, M Richard Pinault et M Guy Teasdale. Qu'ils trouvent ici l'expression de mes sincères remerciements et de ma profonde gratitude.

Ce mémoire vient mettre un terme à un chapitre de ma vie et je me rends compte que cet avant propos est sans doute la dernière opportunité pour moi de dire au revoir à tous ceux que je pu croiser à l'université Laval pendant mon baccalauréat ou ma maîtrise. Pendant ma maîtrise, j'ai pu constater le talent et l'enthousiasme d'étudiants du second et troisième cycles ainsi que ceux des nouveaux professeurs qui se sont ajoutés au personnel, qui dynamisent le département d'informatique et de génie logiciel de l'université Laval.

Je dédie mon mémoire à mon père

Table des matières

Résumé.....	i
Abstract.....	ii
Avant-Propos	iii
Acronymes	ix
Introduction.....	1
Chapitre 1 Analyse des systèmes patrimoniaux	7
1.1 Vues technique du système patrimonial	7
1.1.1 Modélisation de la maintenance logicielle.....	7
1.1.2 Définition d'un système patrimonial	9
1.1.3 La gestion des configurations	9
1.1.4 L'architecture selon des vues par niveaux logiciels	12
1.1.5 Le manque de documentation des systèmes patrimoniaux	14
1.2 Système patrimonial et fonctionnement de l'organisation	16
1.2.1 Systèmes patrimoniaux et nature des opérations	16
1.2.2 Distinction des règles d'affaires dans le système patrimonial.....	21
1.2.3 Méthodes CMM pour mesurer les tâches des employés.....	22
1.2.4 Vérifier si les usagers réalisent des transactions autorisées.....	24
1.3 Identifier les problèmes du système patrimonial dans son environnement	25
1.3.1 Les problèmes techniques posés par le système patrimonial.....	25
1.3.2 Analyser l'état de l'entreprise par questionnaire	26
1.4 Conclusion	28
Chapitre 2 Intégration et maintenance évolutive.....	29
2.1 Changements théoriques dans l'environnement patrimonial.....	29
2.2 EAI : L'intégration d'applications de l'entreprise	31
2.2.1 La modélisation organisationnelle de l'EAI	32
2.2.2 La modélisation technique de l'EAI	34
2.2.3 Modèle d'évaluation d'outils EAI	38
2.3 Améliorer une application patrimoniale	39
2.3.1 Supprimer les anomalies du code source.....	40
2.3.2 Améliorer la conception du code source.....	41
2.3.3 Une méthode spécifique : la migration orientée objet.....	44
2.4 Conclusion	47
Chapitre 3 Identification des objectifs.....	48
3.1 Le succès ou l'échec du projet.....	49
3.1.1 L'objectif technologique.....	50
3.1.2 Objectifs pendant le projet.....	50
3.2 Réingénierie des processus d'affaires.....	51
3.2.1 Définitions	52
3.2.2 Analyser l'organigramme d'une organisation	59
3.3 Motiver les changements	62
3.3.1 Convergence des processus d'affaires et des systèmes d'information	62
3.3.2 Coûts des projets de migration.....	63
3.3.3 Le retour sur investissement : une science inexacte	68

3.4	Conclusion	69
Chapitre 4	Gestion du projet de migration	70
4.1	Établir un système ouvert	70
4.1.1	Les systèmes ouverts et fermés.....	70
4.1.2	Les standards et spécifications.....	72
4.2	Gestion des facteurs tactiques.....	74
4.2.1	Stratégie et tactique.....	75
4.2.2	Relation entre le fournisseur et l'acquéreur.....	76
4.2.3	Adoption et diffusion de l'innovation.....	80
4.2.4	La formation de la main d'œuvre	85
4.2.5	Négocier les changements aux processus d'affaires.....	88
4.3	Gestion des facteurs stratégiques.....	89
4.3.1	Soutien de la direction	90
4.3.2	Stratégie ERP	92
4.4	Conclusion	95
Chapitre 5	Plan de migration	96
5.1	Imposer une centralisation	96
5.1.1	Concurrence entre applications.....	96
5.1.2	Approche Multi site	97
5.2	Planifier plusieurs projets de migration.....	98
5.2.1	Éviter l'approche « Big Bang »	98
5.2.2	Déploiement par phases	100
5.2.3	Durée de l'analyse et du déploiement.....	102
5.2.4	Étude de cas	103
5.3	Produits livrables	105
5.3.1	Documents livrables	105
5.3.2	Composants logiciels	109
5.3.3	Convertisseur de données	110
5.4	Conclusion	115
	Conclusion générale.....	116
	Bibliographie	119
	Bibliographie complémentaire.....	123
	Annexe : Conversion du franc à l'euro.....	124
	Index	126

Liste des tableaux

Tableau 1 : Coûts de maintenance par catégorie d'activités.....	8
Tableau 2 : Modèle de référence par niveaux de Ivar Jacobson.....	13
Tableau 3 : Les processus de cycle de vie	17
Tableau 4 : Exemples d'outils informatiques associés au cycle de vie de soutien.....	20
Tableau 5 : Problèmes au niveau des traitements.....	25
Tableau 6 : Problèmes au niveau des données.....	26
Tableau 7 : Type de changement par niveau logiciel de [JACO 1997].....	31
Tableau 8 : Type de projet de maintenance évolutive et migration, selon [ULRI 2002]	31
Tableau 9 : Modèle organisationnel de l'EAI.....	33
Tableau 10 : Nomenclature des transactions d'un système EAI	36
Tableau 11 : Description des états finaux d'un processus distribué.....	38
Tableau 12 : Critères de performance pour l'évaluation par niveaux de [THEM 2003].....	39
Tableau 13 : Sept attributs correspondant au bonheur orienté objet de [MEYE 1988].....	45
Tableau 14 : Les causes des problèmes des processus d'affaires	53
Tableau 15 : Coûts du projet Gires en millions de dollars en mars 2003.....	67
Tableau 16 : Réévaluations des coûts totaux du projet Gires en millions de dollars [LYRE 2003].....	67
Tableau 17 : Facteurs de succès de projets de migration de [HOLL 1999-a]	75
Tableau 18 : Facteurs influençant la perception des usagers, [CHIA 2001]	81
Tableau 19 : Description des cours et manuels de formation [TCHO 2002]	87
Tableau 20 : Itérations du processus ASAP.....	89
Tableau 21 : Plan de migration suivi par l'université étudiée par [YAKO 2001]	104
Tableau 22 : Calendrier de migration	105
Tableau 23 : Plan de migration	106
Tableau 24 : Plan de retraite logicielle	106
Tableau 25 : Plan de gestion du changement des processus d'affaires	107
Tableau 26 : Document d'évaluation des itérations (après chaque itération).....	108
Tableau 27 : Plan de relations avec le fournisseur / vendeur.....	109
Tableau 28 : Types de livrables développés pour les projets de migration [LANG 1998]	109
Tableau 29 : Processus de conversion de données	111
Tableau 30 : Effet d'escalier de l'arrondi de la conversion Franc – Euro – Franc.....	125

Liste des figures

Figure 1 : Modélisation de la maintenance logicielle de [BENN 2000].....	8
Figure 2 : Modèle des processus d'affaires distribués de [KWAK 2002]	37
Figure 3 : Diagramme de transition d'états des processus et de tâches distribués	37
Figure 4 : Diagramme de graphes d'appels de fonctions.....	42
Figure 5 : Évolution du modèle d'affaires et du système d'information de [KELL 1999]....	63
Figure 6 : Rapport coût / vitesse du changement d'un système logiciel de [MEYE 2001] ..	64
Figure 7 : Processus générique de gestion de plaintes du IEEE 1044.1-1995	79

Acronymes

Acronyme	Définition
APCS	Assembly Process for COTS-Based Systems
API	Application Public Interface
ARIS	Architecture of Integrated Information Systems
ASAP	Accelerated SAP
CAO	Conception Assisted Design
CASE	Computer-Aided Software Engineering
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off The Shelf
DM	Deutschmark
EAI	Enterprise Application Integration
ERP	Enterprise Ressource Planning
FAR	Federal Acquisition Regulation
GUI	Graphical User Interface
IDL	Interface Description Language
IEC	International Electrotechnical Commission
IEC/ISO	Standard commun entre IEC et ISO
IEEE	Institute of Electrical and Electronics Engineers, Inc.
ISO	International Standards Organisation
LDAP	Lightweight Directory Access Protocol
MARC	MAchine-Readable Cataloguing
OMG	Object Management Group
OSI	Open Systems Interconnection
PCI	Perceived Characteristics of the Innovation
RPC	Remote Procedure Call
SEI	Software Engineering Institute
SPR	Software Problem Report
UCITA	Uniform Computer Information Transactions Act
UML	Unified Modeling Language
XML	Extensible Markup Language

Introduction

Ce mémoire comporte un ensemble de recherches qui s'inscrivent dans le cadre d'une problématique portant sur le changement d'un système informatique d'envergure à l'intérieur d'une organisation. D'une part, il établit les fondements théoriques d'une migration technique de systèmes informatiques. D'autre part, il rassemble les meilleures pratiques et démarches pour répartir les tâches entre intervenants pour s'attaquer aux autres problèmes de gestion qui peuvent se greffer au problème informatique.

Les entreprises et gouvernements subissent des pressions croissantes pour fournir des services intégrés et permettre des transactions sécuritaires, qui s'adaptent au débit disponible ou aux capacités de la machine, fiables et disponibles. Le développement de tels outils est un exercice difficile. La modification de ces services pour s'adapter à un changement dans les besoins l'est encore plus.

Les interdépendances entre les applications avec l'organisation qui les utilise et la complexité des transactions font que l'impact d'un petit changement dans un environnement informatique peut être difficile à anticiper. Pour cette raison les changements d'un système informatique doivent suivre une démarche rigoureuse. Des changements qui corrigent des erreurs, ou adaptent une application à son nouvel environnement entrent dans le cadre de la maintenance corrective. Des changements qui ont pour objectif d'ajouter des fonctionnalités complètement nouvelles à une application patrimoniale sont soit une maintenance évolutive soit un nouveau système qui remplace le système patrimonial.

Les échecs des projets informatiques de migration ne se sont pas seulement dus à des problèmes informatiques. Ces problèmes peuvent être, entre autres, dus à la mauvaise formation des usagers du système ou aux changements inadéquats apportés aux processus d'affaires de l'organisation ou encore à l'obsolescence du système en place qui ne répond plus aux besoins. C'est sur ces problèmes que porte notre réflexion.

L'informatique est une science qui s'est développée depuis les années 50. Si initialement, les premières applications informatiques étaient installées dans des organisations sans actifs

informatiques, aujourd'hui, l'informatique a pénétré tous les niveaux de l'entreprise. De nos jours, il est très rare pour une organisation de voir une nouvelle application informatique remplacer un processus qui n'y faisait pas appel du tout. De la même manière, il y a rarement des entreprises qui soient entièrement équipées d'outils sur mesure ou entièrement équipées d'outils achetés sur le marché et utilisés sans modification.

Les technologies de l'information que les entreprises ont intérêt à maintenir sont souvent des bases de données qui maintiennent les informations de l'entreprise et qui peuvent être consultées ou manipulées par un grand nombre d'employés en réseau conformément aux processus d'affaires. Par conséquent, historiquement, l'un des premiers systèmes informatiques en réseau dans les grandes organisations a été le « main-frame ». Les applications « main-frame » sont des applications qui effectuent toutes les tâches sur un ordinateur central. Le poste de l'utilisateur n'est alors qu'un terminal qui peut examiner les traitements de l'ordinateur central et transmettre des instructions par une interface clavier. Tous les traitements sont alors effectués sur l'ordinateur central. Dans ce contexte, les tâches des employés de soutien technique étaient le plus souvent limitées à des corrections sur le serveur « main-frame ». À cette époque, les techniciens avaient un contrôle étroit sur les modifications qui étaient apportées sur le système. La gestion de l'informatique était donc hautement centralisée.

Ensuite sont apparus les applications client serveur. Les applications client serveur se répartissent la charge des opérations entre le client et le serveur car les postes clients sont capables d'effectuer une partie des opérations au profit du serveur. Par ailleurs, l'apparition et la démocratisation de l'ordinateur personnel, dans les années 80, ont permis à des petites organisations, avec des budgets modestes, d'installer des systèmes informatiques performants et d'engager quelques employés de soutien technique. Cet essor informatique a donné les moyens aux sous unités d'une organisation, d'installer sur des postes client des applications réseau sans nécessairement consulter la haute direction de l'informatique de l'organisation. Lorsque ce progrès a été reconnu par la direction des organisations, de nombreuses organisations ont ainsi accordé un système de gestion décentralisée de l'informatique.

À cause de l'évolution des standards et des innovations technologiques, un investissement engagé dans une organisation n'assure un avantage compétitif que s'il est maintenu et amélioré. Ainsi, de nombreux systèmes, dits patrimoniaux, doivent être maintenus ou remplacés par des applications plus modernes selon l'évolution des besoins des organisations et celle du domaine. L'absence de normes dans le domaine de la migration informatique donne lieu et à une mauvaise estimation des enjeux, des ressources nécessaires, des compétences à rechercher et des objectifs à atteindre. L'absence de règles et de standards reconnus empêche les responsables de ces projets d'expliquer correctement leurs besoins et de se concentrer sur les objectifs prioritaires. Ces lacunes peuvent conduire à des échecs retentissants.

Une analyse du Standish Group [STAN 1999] déclarait qu'en 1998, dans le domaine des technologies de l'information, seuls 25 % des projets parviennent à être menés à terme avec succès, que 50 % des projets ne sont menés à terme qu'au prix d'une révision des objectifs et des budgets et que 25 % des projets échouent. Par ailleurs, les révisions des budgets d'un projet informatique augmentent avec la taille des entreprises. Ainsi, si la taille de l'entreprise est grande, on estime que les augmentations de budget peuvent dépasser de 65 % le budget initial. Pour essayer de faire face à ces difficultés, certaines organisations rédigent des contrats de services avec des fournisseurs pour leur imputer toute responsabilité de l'échec de l'implantation de nouveaux systèmes informatiques. Cette tendance, regrettable, donne l'image d'un domaine en manque de maturité.

Selon [ERPR 2004], le terme ERP désigne le système de gestion intégrée de l'entreprise, reposant sur un "progiciel intégré" paramétrable. L'acronyme ERP (Enterprise Resources Planning) dérive de la méthode MRP (Manufacturing Resources Planning), célèbre méthode utilisée en gestion industrielle, dans des domaines comme la gestion de production. Cette intégration signifie notamment que les données utiles sont stockées en un seul endroit : il existe par exemple un seul registre des clients pour les fonctions commerciales et de facturation, garantissant l'absence d'incohérences et supprimant les saisies redondantes.

Les logiciels de gestion intégrée constituent le système majeur d'intégration informatique de l'entreprise. Un consultant ERP requiert autant une connaissance des logiciels qu'une

capacité de compréhension globale de l'entreprise et plus largement des changements que vit l'entreprise avec Internet. L'installation d'un ERP répond souvent à des objectifs de valeur ajoutée. L'objectif peut être économique : chercher à réduire des coûts de maintenance ou à améliorer un processus. Le processus est souvent humain et touche alors des aspects variés, comme l'activité et la performance individuelle des personnels concernés.

Enfin, les objectifs organisationnels concernent par exemple la réduction des délais de traitement ou d'acheminement des flux d'information dans l'entreprise. L'ERP représente un défi organisationnel qui doit être soigneusement préparé. La définition des objectifs est un moment important, où il s'agit d'allumer le phare qui pendant toute la durée du projet va servir de guide à tous les acteurs concernés.

La plupart des gouvernements et entreprises, s'efforcent de remplacer et d'intégrer des applications datant parfois de l'apparition des systèmes informatiques par des logiciels professionnels récents et compétitifs. Cet ensemble d'applications a un impact sur toute l'organisation. Ces choix technologiques doivent suivre des orientations. Soit l'architecture informatique se conforme à l'organisation, soit c'est l'organisation qui se conforme à l'architecture informatique. Lorsqu'une organisation cliente commande à un fournisseur un système sur mesure pour son client, l'architecture informatique suit le fonctionnement de l'organisation. L'inconvénient est que la maintenance de ce produit acheté par un seul client entraîne des coûts exorbitants. Ce choix reste valide dans certains contextes, par exemple dans le contexte militaire (car on requiert un système ultrasecret, sécuritaire et très difficile à reproduire). L'autre alternative pour l'organisation est d'acheter un produit logiciel commercial disponible sur le marché (telles les applications ERP) maintenu en permanence par le fournisseur qui suit l'évolution du domaine. Dans ce cas, c'est l'organisation qui s'adapte à ce produit. Parfois au contraire, les organisations décident que les systèmes patrimoniaux doivent être maintenus et que les seuls changements à apporter sont d'intégrer les systèmes patrimoniaux entre eux. Il existe plusieurs techniques pour y parvenir.

Une attention particulière doit donc être apportée aux problèmes d'assurance qualité lors de la modernisation d'applications d'affaires ou administratives. Les problèmes d'assurance

qualité, lors de la phase de modernisation et d'implantation d'une application, vont se poser avec d'autant plus de force que la modernisation et l'interopérabilité des applications vont devenir nécessaires. Les efforts de maintenance font l'objet de beaucoup de recherches. Cependant, les efforts de modernisation sont souvent négligés dans de nombreux modèles représentant le cycle de vie logiciel. Pourtant, nous pensons qu'il s'agit là d'un domaine crucial qui est appelé à se développer.

Les organisations ont un intérêt à être rigoureuses dans la gestion des configurations de leurs systèmes patrimoniaux tout en mettant l'accent sur la modernisation entre systèmes patrimoniaux et nouveaux systèmes. Jusqu'à aujourd'hui, l'entretien, l'évolution et la réingénierie des systèmes patrimoniaux ont été gérés avec moins de rigueur que le développement de nouveaux systèmes. Cependant, en raison de l'importance des investissements dans les systèmes patrimoniaux et de la complexité de ces derniers, il peut être essentiel de faire preuve de plus de rigueur méthodologique lors de la modernisation de systèmes patrimoniaux que dans le développement de nouveaux systèmes.

L'implantation de solutions technologiques retenues par les entreprises peut se solder par un échec si la formation des employés est mal organisée ou lorsque le changement technologique s'ajoute à des changements au niveau des processus d'affaires de l'organisation. Les applications ERP constituent une technologie sophistiquée, imposée par l'organisation à ses membres, qui requièrent beaucoup de formation et de coordination dans les opérations des usagers et une formation détaillée. Ce mémoire examine les problèmes de formation du personnel lors du déploiement d'un nouveau système informatique dans une organisation.

L'objectif de ce projet de maîtrise sera donc d'analyser les méthodes de développement de plans de migration qui doivent assurer une transition stable d'un ou plusieurs systèmes patrimoniaux vers un nouveau système dont le fonctionnement est plus efficace. C'est dans ce contexte et dans un objectif d'assurance qualité, en mettant l'emphase sur les plans de migration de systèmes patrimoniaux, que s'insère ce projet de maîtrise. Le plan du mémoire est le suivant :

1. État de l'art pour l'analyse d'applications patrimoniales qui vont disparaître.

2. Revue des meilleurs moyens pour améliorer l'environnement informatique pour l'intégrer avec l'application cible.
3. Identification des objectifs du projet de migration en abordant la question des coûts et la réingénierie des processus d'affaires.
4. Établissement des priorités pour la gestion du projet en distinguant les facteurs stratégiques et tactiques.
5. Organiser les projets de migration entre eux et résumer les méthodes, modèles et processus reconnus, caractériser les principales étapes qui devraient jaloner le plan migratoire, à l'aide d'études de cas.

Les systèmes patrimoniaux existeront toujours. Les organisations ont rarement des environnement complètement sur mesure ou complètement issu du marché sans modification. La maintenance n'est pas un projet unique mais un effort continu. Au contraire la migration est une étape précise, un effort épisodique qui s'inscrit dans une vision plus large de modernisation de l'organisation.

Chapitre 1 Analyse des systèmes patrimoniaux

Une analyse des systèmes patrimoniaux d'une organisation doit pouvoir répondre à quelques questions de base. À quoi correspond le système patrimonial, comment fonctionne-t-il et à quoi sert-il dans l'organisation ? À l'aide de quels modèles peut-on examiner le système patrimonial ? Après avoir établi ces définitions et fourni plusieurs moyens d'appréhender un système patrimonial, il sera possible d'analyser et de circonscrire les problèmes posés par des systèmes patrimoniaux, notamment par questionnaire.

1.1 Vues technique du système patrimonial

Le système patrimonial peut être analysé ou appréhendé de plusieurs manières différentes. Ces vues du système patrimonial sont groupées en vues du système à l'intérieur de son environnement et en vues du système individuellement.

L'analyse passera en revue la perspective du système patrimonial à l'intérieur de son environnement. Soit du point de vue de la maintenance logicielle, du point de vue de définitions formelles, du point de vue de l'inventaire de l'infrastructure patrimoniale, du point de vue de la nature des opérations de l'organisation et enfin du point de vue des procédures des employés ou usagers de l'organisation.

L'analyse d'un système patrimonial individuellement essaie de distinguer, à l'intérieur du code source, les règles d'affaires sur lesquelles le système est basé. On peut comprendre le système patrimonial selon les transactions qui y sont effectuées. On peut enfin analyser le système patrimonial à l'aide de la documentation qui l'accompagne.

1.1.1 Modélisation de la maintenance logicielle

Pour comprendre les enjeux de la modernisation de systèmes patrimoniaux, il faut pouvoir les distinguer des enjeux de la maintenance. La maintenance des produits logiciels du point de vue du développement de logiciels correspond aux efforts d'entretien d'un système en place. Même si la maintenance permet de répondre à certains nouveaux besoins des usagers, elle est censée avoir une portée limitée alors que la migration inclut les efforts pour intégrer ou remplacer des systèmes.

La recherche sur la maintenance logicielle fournit des modèles pour situer une application dans son environnement (incluant une analyse de l'équipe de maintenance) et par rapport au fournisseur qui a livré l'application. Le modèle de Bennett [BENN 2000] présenté dans la figure 1 illustre à quel point un système est maintenable. L'étape à laquelle se trouve le produit influence la facilité des opérations de maintenance qui peuvent être réalisées.

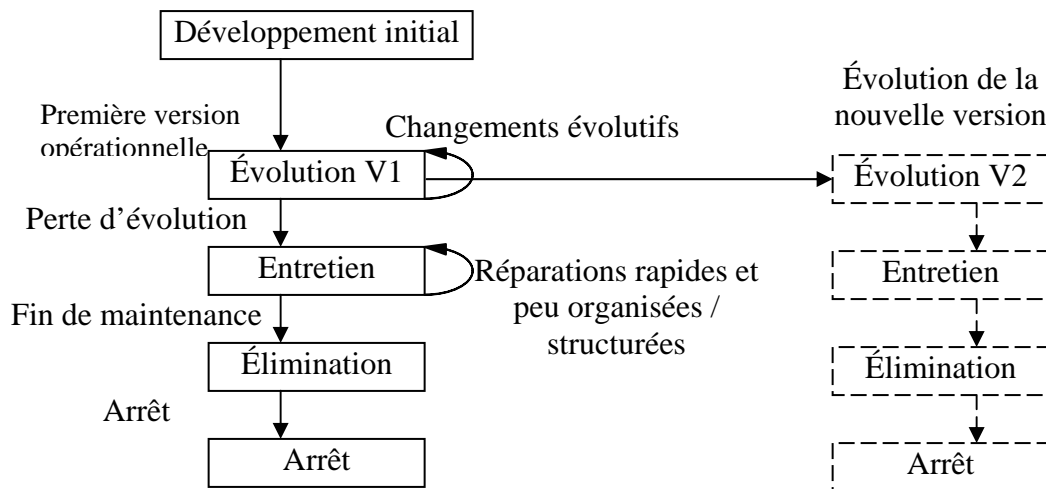


Figure 1 : Modélisation de la maintenance logicielle de [BENN 2000]

L'article de référence réfère à d'autres recherches, plus anciennes qui établissent 4 étapes d'activités de maintenance : soit l'adaptation, le perfectionnement, la correction et la prévention. Un produit logiciel suit les 4 étapes qui sont le développement initial, l'évolution, l'entretien, la mise hors service, le stockage.

Un sondage cité dans l'article [BENN 2000] fixe des pourcentages des coûts de maintenance par catégorie.

Tableau 1 : Coûts de maintenance par catégorie d'activités

%	Activités	Description
75	Adaptation	Effort pour réagir à des changements dans l'environnement logiciel
	Perfectionnement	Répondre à des nouvelles spécifications des clients et usagers
21	Correction	Résoudre des erreurs du logiciel
4	Prévention	Empêcher des erreurs de survenir dans l'avenir

Si des changements pouvaient être anticipés dès la conception du logiciel, ces changements pourraient être inclus dans le code source sous la forme de paramètres. Cependant,

beaucoup de changements qui surviennent sont ceux que les concepteurs ne pouvaient même pas imaginer.

Les résultats du sondage de la figure 1 semblent ne présenter qu'un seul aspect de la maintenance informatique. Il faudrait tenir compte du contexte d'utilisation des systèmes. L'affirmation selon laquelle la prévention ne représente qu'une petite partie des efforts de maintenance logicielle apparaît inquiétante. La mise à jour d'un anti-virus, devrait être spontanée et pas uniquement après que des erreurs logicielles ne soient apparues.

1.1.2 Définition d'un système patrimonial

De nombreux articles expliquent comment gérer un système patrimonial, mais établissons une définition pour le cadre de l'étude. Alan O'Callaghan [O'CA 1999] définit un système patrimonial comme un bien patrimonial : un bien tangible ou intangible obtenue d'un prédécesseur, cela peut aussi inclure l'effet dans le long terme d'un événement ou un processus. Dans le domaine de l'informatique, le terme 'système patrimonial' semble utilisé à tort et à travers pour correspondre à tout équipement informatique qui a été installé dans une organisation et qui aurait le moindre problème.

Plus raisonnablement, on peut parler d'un système patrimonial pour un système informatique, qui fournit une valeur d'affaires significative à une entreprise, grâce à un investissement, qui peut avoir eu lieu il y a plusieurs années. Un système patrimonial est donc quelque un outil qui fonctionne et qui a correctement fonctionné dans le passé.

1.1.3 La gestion des configurations

Dans les organisations, il peut y avoir une adoption de produits informatiques standards ou l'achat de produits informatiques faits sur mesure pour l'organisation. Dans chaque cas, c'est la gestion des configurations qui s'assure que chaque version utilisée par les usagers est contrôlée, approuvée et fonctionne de la même manière pour tous les usagers selon les besoins exprimés par les différents intervenants.

Pour moderniser des équipements informatiques dans une organisation, il faut avoir un inventaire du parc informatique. Cette gestion fait partie des efforts de gestion des configurations.

Selon le [CMMI V1.1, 2002], niveau 2, un processus de gestion des configurations précis des produits logiciels installés devrait idéalement inclure les pratiques suivantes :

- Établir des versions de référence (aussi appelés référentiels)
 - Identifier les éléments de configuration
 - Établir un système de gestion des configurations
 - Créer ou distribuer des versions de référence
- Identifier et contrôler les changements
 - Identifier les demandes de changements
 - Contrôler les éléments de configuration
- Assurer l'intégrité
 - Établir des enregistrements de gestion de la configuration
 - Réaliser des audits de configuration

Les équipements en place influencent les nouveaux logiciels que l'on peut acheter, choisir et installer. Le parc informatique requiert des employés compétents dans une équipe de soutien technique pour maintenir ces équipements. La gestion des configurations dans une organisation devrait être réalisée par l'équipe de soutien technique et contrôlée ou auditée par l'équipe d'assurance qualité.

Lorsqu'un système informatique doit être utilisé par les usagers d'une organisation alors le système doit pouvoir fonctionner sur le parc informatique de l'organisation. Si le parc informatique est partiellement incapable d'accueillir le nouveau système alors soit des altérations sont nécessaires sur certains postes du parc informatique, soit au niveau du nouveau système informatique. L'altération du nouveau système informatique pour s'adapter au parc informatique doit être découragée.

Ulrich [ULRI 2002], se concentre sur le problème des outils redondants dans une organisation. En effet, la coexistence de technologies concurrentes (comme .NET et J2EE) peut obliger les développeurs à réécrire des parties d'une application dans le système concurrent, donc créer de la programmation redondante et des systèmes multiples qui

effectuent les mêmes opérations. L'hétérogénéité peut créer de la confusion car il est plus difficile de trouver toute la logique d'affaire dans une seule lecture (.NET ou J2EE).

1.1.3.1 Les applications à ne pas oublier

En bases de données on classe les éléments d'un système en procédures, formulaires et rapports. Pour résumer, les procédures sont des modules qui effectuent des opérations sur un ordinateur sans fournir des résultats à un usager, les formulaires sont des modules utilisés par l'utilisateur pour consulter et modifier les données d'un système, les rapports sont des modules qui se contentent de lire des données d'un système et qui les présentent sous forme détaillée ou synthétique.

Des rapports, qui n'effectuent que des accès en lecture aux bases de données, peuvent se greffer de manière invisible à une application principale, lorsque des droits de lecture sont accordés à des usagers et à leurs applications client. Des usagers peuvent ainsi ajouter des rapports qui consultent des bases de données disponibles sur le réseau sans que le soutien technique n'en soit averti. Ces rapports devraient être enregistrés dans la gestion de configuration. Malgré tout, la gestion des configurations ne devrait considérer que les éléments logiciels essentiels les éléments accessoires devraient être délaissés.

1.1.3.2 Les serveurs d'applications

Pour pallier aux contraintes de la technologie client serveur, les serveurs d'applications offrent une solution innovatrice. Un logiciel client très simple (tel un navigateur) fait appel à des applications qui se trouvent sur un serveur (tel un serveur Web), qui lui est relié aux bases de données. Un serveur d'application contrôle l'environnement d'exécution des applications sur le serveur. Il prend en charge l'ensemble des fonctionnalités qui permettent à plusieurs clients d'utiliser une même application : la gestion de la session utilisateur, en conservant des contextes propres à chaque utilisateur, et tolérer les montées en charge et reprises sur incident.

Les serveurs d'applications ont l'avantage, du point de vue de la gestion des configurations de réunir en un seul lieu les applications interactives utilisées par les clients. Ils séparent les machines qui contiennent les données, les applications qui y accèdent, et les usagers qui

font appel au serveur d'application. Ces applications peuvent donc être mises à jour plus rapidement et mettre en commun les accès à plusieurs bases de données distinctes d'une manière harmonisée.

1.1.4 L'architecture selon des vues par niveaux logiciels

L'architecture peut être vue comme une somme d'éléments logiciels fonctionnant ensemble et agissant à plusieurs niveaux différents. Une bonne architecture est un système modulaire, qui permet aux éléments logiciels de ne pas être dépendants les uns des autres. Ainsi, les changements d'un composant n'influencent pas le fonctionnement ou le comportement des autres. En associant chaque élément logiciel à un niveau logiciel simple et intuitif on réalise une architecture dans laquelle il est facile de retrouver, analyser, maintenir, modifier et réutiliser les composants.

Il existe plusieurs niveaux logiciels pour des domaines ou disciplines spécifiques de l'informatique à l'aide desquels on peut examiner un système. On va d'abord considérer l'architecture logicielle du point de vue de la téléinformatique, de la réutilisation logicielle et selon d'autres points de vue. Enfin, le concept des patrons logiciels est apparu pour expliquer comment catégoriser des modèles de solutions à des problèmes spécifiques.

1.1.4.1 Niveaux en couches logiciels OSI

En téléinformatique un modèle de couches logicielles répartit les opérations à réaliser entre des composants. Ces opérations sont effectuées par différentes couches logicielles. La finalité de chaque couche est de fournir des services à la couche située immédiatement au-dessus sans l'encombrer sur la manière de réaliser le service, la couche supérieure étant l'utilisatrice du service. Pour que deux couches échangent des informations, il doit exister un ensemble de règles admises en ce qui concerne l'interface.

Le modèle OSI (Open Systems Interconnection) de ISO (l'organisation internationale de normalisation) fut proposé dans les années 80 et se base sur trois concepts centraux :

- Les services
- Les interfaces
- Les protocoles

Le modèle OSI en 7 couches (physique, liaison de données, réseau, transport, session, présentation, application) a fait l'objet de beaucoup de recherches, mais est un modèle théorique adopté par des chercheurs et peu par l'industrie. Cette décomposition n'est pas la seule possible ou imaginable, mais elle demeure une décomposition utile des composants logiciels en téléinformatique.

1.1.4.2 Autres niveaux en couches logiciels

Ivar Jacobson [JACO 1997], établissait des niveaux logiciels du point de vue de la réutilisation de composants logiciels.

Tableau 2 : Modèle de référence par niveaux de Ivar Jacobson

Niveaux logiciels	Description et exemple
Applications	C'est ce niveau que l'on réutilise les composants des niveaux subséquents <ul style="list-style-type: none"> • Système de guichet automatique, Système de facturation, système de paiement.
Composantes d'affaires	Composants qui satisfont des besoins d'affaires <ul style="list-style-type: none"> • Gestion de clients, gestion des comptes, Gestion des avoirs bancaires
Composantes intergicielles	Composants soutenant les systèmes informatiques répartis <ul style="list-style-type: none"> • Java, CORBA, OLE/COM, ActiveX
Composantes systèmes	Composants du système d'exploitation <ul style="list-style-type: none"> • Windows NT, TCP/IP

[ULRI 2002], établissait des niveaux logiciels du point de vue des transactions pour gérer toute modification ou transformation indépendamment les unes des autres.

1.1.4.3 Les patrons

Les patrons logiciels sont un inventaire des meilleures architectures logicielles employées par l'industrie. Tous les modèles de patrons reposent sur le même principe : la décomposition du problème en modules indépendants les uns des autres. Pour autant une unanimité dans la décomposition est ardue. [O'CA 1998] et [O'CA 1999], explique que, généralement les entreprises qui développent du logiciel pour satisfaire leurs propres besoins organisent leurs systèmes informatiques à leur image. Donc, chaque entreprise a tendance à développer une architecture en fonction de l'organisation.

Les patrons logiciels représentent un ensemble de solutions informatiques à des problèmes génériques et qui ont une structure commune. L'auteur [OCA 1998] a mis au point le système Adaptor pour enregistrer et lier les différents patrons. Dans le système Adaptor, trois types de patrons (systèmes informatiques, procédures et hiérarchies) interagissent entre eux. Tout un changement enregistré sur un patron se répercute sur les autres qui dépendent de lui. En cas de conflit entre les différents patrons, le système propose des solutions à haut niveau.

La recherche de O'Callaghan semble très fine. En effet, sa recherche explique les différentes architectures d'autres auteurs. Utiliser les patrons pour trouver des solutions rodées et adaptées aux problèmes de gestion des systèmes patrimoniaux semble tout à fait pertinent. Malgré tout, si on voulait critiquer sa recherche on résumerait le logiciel Adaptor à une matrice de traçabilité très détaillée avec des alternatives prédéfinies.

1.1.5 Le manque de documentation des systèmes patrimoniaux

Le manque de documentation pertinente et récente accompagnant un code source est un problème qui existe dans de nombreuses organisations. Dans les organisations, il arrive souvent, lorsqu'on maintient un système, que l'on néglige l'entretien des documents de conception, pour ne maintenir à jour que le code source. Ainsi une entreprise peut se retrouver avec un système qui fonctionne mais qui n'est pas correctement documenté.

1.1.5.1 La documentation automatique par l'exécution du code

Pour pallier à ces lacunes, les recherches de Alok Mehta [MEHT 2001-a], [MEHT 2002], [MEHT 2001-b], se sont efforcées de faciliter l'analyse automatique et dynamique des systèmes patrimoniaux à l'aide de tests de régression. Par cette technique, on parvient à générer de la documentation valide et à jour même s'il n'y a que les modules exécutables des applications patrimoniales qui sont disponibles. Les tests de régression sont des ressources peu exploitées du point de vue de l'évolution d'applications.

Les tests de régression sont des tests qui, le plus souvent, vérifient et valident un besoin d'affaire. Ainsi l'analyse d'un système patrimonial avec un outil d'analyse du code statique n'a pas la même portée qu'une analyse du système patrimonial avec un outil d'analyse du

code qui est dynamique. Une fois que l'on sait quelles opérations l'application est censée effectuer, avec des outils de couverture de code, on peut calculer quelle partie du code source accomplit ces opérations. Le code source qui se rapporte à un test de régression se rapporte aussi à un groupe de spécifications ou de besoins spécifiques.

La recherche de Mehta demeure particulièrement spécialisée et ne semble pertinente que dans de rares cas, c'est-à-dire lorsque la documentation disponible est de faible qualité. Si une documentation faite sur mesure et correcte est disponible, cette documentation pourrait être plus utile que la documentation générée à partir de tests de régression.

1.1.5.2 La documentation automatique par compilation du code source

Certains logiciels modernes permettent de compiler des codes sources et de produire de la documentation de conception de bas niveau automatiquement en recueillant les noms des variables, des traitements et les commentaires dans le code.

Les outils de développement sont aidés par des utilitaires de documentation innovateurs. En Java, le compilateur est secondé par un outil de génération automatique de documentation appelé Javadoc. C'est un compilateur qui explique la structure du code source. Cet outil rend la mise à jour de la documentation plus facile.

1.1.5.3 Limites

La documentation automatique est le plus souvent pertinente lorsque le langage est de très haut niveau et qu'il existe une panoplie d'instructions spécialisées pour effectuer des opérations précises.

Si on s'efforce de documenter un langage de bas niveau, la documentation automatique sera de faible qualité. La documentation de l'assembleur sera moins explicite que la documentation d'un programme orienté objet. En effet, en assembleur, la récurrence doit être programmée pour être utilisée. Dans un langage de haut niveau ces opérations existent déjà dans le langage ; ils n'ont pas besoin d'être réécrits et peuvent être correctement interprétés. Malgré certaines limites dans les langages de bas niveau, on ne peut qu'espérer que l'usage des générateurs de documentation automatique se généralise pour les langages

de haut niveau ainsi, la documentation (qui doit être réalisée en fin de projet) ne sera pas sacrifiée lorsque des dépassements de budgets auront lieu.

Le cadre de l'article de Quilici [QUIL 1995] est celui de l'ingénierie inverse pour extraire toutes les spécifications d'un système automatiquement. L'auteur étudie des outils polyvalents qui pourraient analyser des codes sources dans plusieurs langages de programmation. [QUIL 1995] conclut que la documentation automatique a ses limites et doit être complété par le travail de spécialistes informatiques et de spécialistes du domaine. Il existe une barrière sémantique que des outils de d'analyse automatique ne peuvent pas dépasser. Ces outils s'adaptent rarement bien à la taille des codes sources analysés. Les outils d'analyse de code et de documentation automatique ne se comportent pas de la même manière avec un code source de 1000 lignes de code ou de 100 000 lignes de code lui est fourni en entrée.

1.2 Système patrimonial et fonctionnement de l'organisation

Plusieurs facteurs dans une organisation ont une influence sur les systèmes patrimoniaux. Notamment la nature des opérations effectuées par le système patrimonial. Lorsqu'une application soutient les opérations d'une organisation, les règles d'affaires et les contraintes de l'organisation sont souvent programmées dans le code source de l'application. Les règles et contraintes programmées dans une application, devraient assurer que les employés les respectent. Pourtant, il peut arriver que des usagers n'emploient pas les applications à leur disposition ou encore l'emploient à des fins autres que ceux prévues par l'organisation. Un contrôle des transactions des usagers permet de vérifier le fonctionnement de l'application dans l'organisation. Notons que le modèle de maturité des entreprises proposé par le SEI (Software Engineering Institute) recommande aux organisations de rédiger les procédures d'affaires et d'inclure dans ces procédures les applications nécessaires pour les réaliser.

1.2.1 Systèmes patrimoniaux et nature des opérations

Les opérations et activités d'une organisation peuvent se rapporter principalement à sa mission ou à l'administration de l'organisation. Cette distinction s'efforce de suivre les

orientations existant dans les standards IEEE et vise à identifier des cas concrets. La nature des opérations réalisées par un système informatique dans une organisation peut influencer la stratégie de migration du système.

Le standard [ISO/IEC 12207 1998] 12207 adopté conjointement par ISO et IEC en 1995 et mis à jour en 1998 identifie les trois différents types de cycles de vie dans une organisation identifiés dans le tableau 3.

Tableau 3 : Les processus de cycle de vie

Cycle de vie primaire	Cycle de vie organisationnel	Cycle de vie de soutien
Acquisition Développement Opération Maintenance	Formation Amélioration Gestion Infrastructure	Documentation Gestion de la configuration Assurance qualité Vérification & Validation Audit Gestion des plaintes et problèmes

Les outils utilisés dans une organisation sont influencés par chacun de ces cycles de vie comme semble le confirmer la recherche de [LOSA 2002], qui sera examinée plus en détail dans la section 2.2.1.

Explication :

Dans une organisation de fabrication des véhicules, la mission principale de cette entreprise est de fabriquer des véhicules. C'est-à-dire qu'un processus de cycle de vie primaire est de développer des véhicules.

- Pour réaliser la mission, l'organisation peut utiliser des outils de CAO (Conception assistée par ordinateur)
- Pour réaliser l'administration de l'entreprise, cette organisation peut aussi avoir un département de paie qui utilise un système informatique de paie pour payer les employés

Ce département de paie avec son système informatique ne se rapporte qu'indirectement à la mission de l'entreprise. Ce département de paie effectue des tâches de gestion, c'est-à-dire qu'une tâche du processus de cycle de vie organisationnel est de rémunérer les employés.

Tout autre système ne se rapporte que indirectement à la mission de l'entreprise. Ainsi, l'intégrité du fonctionnement du système de courrier électronique n'est pas dépendante des

opérations de tous les autres usagers. Si les employés décident de ne pas utiliser cette application d'une manière cohérente, ces transactions ne portent pas atteinte à l'intégrité du système. Ces outils soutiennent tous les processus de cycle de vie de l'organisation.

1.2.1.1 Outils soutenant les processus de cycle de vie primaires

Les organisations ont besoin de produits performants et à la pointe de la technologie pour les systèmes soutenant les processus de cycle de vie primaires. Ces systèmes doivent être modernes, ciblés pour un ou plusieurs départements et entretenus pour fournir un avantage compétitif à l'organisation.

Il est dans l'intérêt des gestionnaires locaux d'une organisation d'investir des ressources dans le cycle de vie primaire. L'avantage relatif d'une application, pour le cycle de vie primaire, s'il est exposé clairement à un gestionnaire ou aux membres d'une organisation est de nature à susciter l'adhésion.

Le plus souvent ces outils peuvent être installés localement; un gestionnaire d'unité sera prêt à financer ces outils à même son budget régulier, car il en voit l'importance et l'intérêt direct et continu pour son unité. Un délai ou dépassement de budget dans le projet de déploiement d'un outil soutenant le cycle de vie primaire, pourra être tolérée, à cause de l'avantage relatif que procurerait ce nouvel outil.

1.2.1.2 Outils soutenant les processus de cycle de vie organisationnel

Les activités d'administration correspondent aux activités, décidées par les gestionnaires, qui permettent aux activités de mission de s'accomplir. Le système de paie, dans une organisation, doit payer les salaires des employés, mais n'a qu'un lien indirect avec la mission de l'entreprise. Les activités d'administration correspondent à des tâches associées au processus de cycle de vie organisationnel.

Les organisations peuvent donc se permettre d'avoir un produit archaïque soutenant les processus de cycle de vie organisationnels. Un produit obsolète n'empêchera pas l'organisation d'atteindre ses objectifs principaux. Des systèmes de gestion, tant qu'ils ne font pas d'erreurs, peuvent être mal entretenus. Le système de gestion de la paie peut

consommer beaucoup de temps ou de mémoire que nécessaire pour réaliser des calculs. Cette inefficacité ne rend pas le système inutile, car elle n'est pas un facteur de différenciation des concurrents du domaine.

Les systèmes qui soutiennent les processus de cycle de vie organisationnel sont utilisés par un groupe restreint : les gestionnaires. Les outils soutenant le cycle de vie organisationnel impliquent un degré de centralisation qui peut se répercuter au niveau du soutien technique. Cette centralisation du pouvoir et du contrôle peut se révéler un obstacle au déploiement de ces systèmes, puisque tout avantage relatif du système ne se rapporte pas au cycle de vie primaire et ne bénéficiera qu'à la haute direction.

La distribution des coûts de déploiement de ces systèmes entre tous les départements de l'organisation est de nature à susciter de la résistance puisqu'ils grèvent les budgets locaux. Seul, l'octroi de budgets spécifiques, de la haute direction, pour ce type de projets permet de contourner ce problème. Il est très difficile pour un gestionnaire d'hypothéquer toutes les initiatives ou nouveaux projets d'amélioration dans son unité pour permettre l'implantation d'un outil qui ne devrait qu'augmenter la productivité de la direction centrale [LYRE 2003].

1.2.1.3 Les outils de communication

Cette catégorie est proposée pour regrouper les systèmes informatiques ou protocoles de très bas niveau tels que décrits au tableau 2 et tableau 5, ainsi que les logiciels de communication (courrier électronique, répondeurs automatiques, téléavertisseurs etc.). Ces composants ne soutiennent pas directement des processus de l'organisation. En effet, il est difficile d'associer précisément ou exclusivement un composant logiciel de bas niveau, tel que le système d'exploitation, à un processus que suit un groupe de l'entreprise. Ces systèmes qui ne soutiennent qu'indirectement les activités d'administration et de mission sont plutôt faciles à mettre en place. La résistance à ces changements est moindre, car ces outils peuvent soutenir tous les processus d'une organisation.

Une difficulté (retard ou dépassement de budget) qui aurait pour effet d'hypothéquer les initiatives ou nouveaux projets d'amélioration d'une unité locale pourrait certes

compromettre le projet. Cependant, un des avantages des outils de communication est que l'objectif principal du projet est l'implantation de la technologie et non pas de respecter les besoins et les processus des usagers. Par conséquent l'outil de communication peut être utilisé à d'autres fins que celles prévues par la direction.

1.2.1.4 Outils soutenant les processus de cycle de vie de soutien

Selon le standard ISO/IEC 12207 [ISO/IEC 12207 1998], les processus de cycle de vie de soutien sont des processus qui permettent de gérer la documentation, la gestion des configurations, d'effectuer le travail de vérification et de validation, de suivre les audits et les tâches d'assurance qualité. Il existe une panoplie d'outils informatiques spécialisés qui permettent de réaliser ces opérations. Pour illustrer les opérations et les outils informatiques les soutenant, nous mentionnerons simplement un exemple d'outil par activité.

Tableau 4 : Exemples d'outils informatiques associés au cycle de vie de soutien

Activité	Outils de soutien informatiques
Documentation	Bases de données identifiant chaque document d'une manière unique
Gestion de la configuration	Des logiciels de gestion de versions de fichiers et des logiciels conservant des copies de secours
Assurance qualité	Outils de surveillance et de contrôle utilisés par le groupe d'assurance qualité ou des auditeurs externes
Vérification & validation	Outils de tests automatiques et les outils de négociation et de communication avec les clients
Audit	Outils de planification et d'organisation d'audits utilisés par le groupe d'assurance qualité ou auditeurs externes
Gestion des plaintes et problèmes	Bases de données identifiant chaque erreur découverte, son diagnostic et la solution associée d'une manière unique

Les outils décrits sont autant un service pour les usagers qu'un contrôle ou une contrainte pour les usagers. Le cycle de vie de soutien exerce un contrôle sur les membres d'une organisation et les processus qu'utilisent ces derniers. Cependant, les membres sont ainsi assurés du fonctionnement fiable et correct des outils employés et documents de référence.

Souvent, les outils soutenant les cycles de vie de soutien sont des modules ou parties d'un autre outil soutenant le cycle de vie primaire ou le cycle de vie organisationnel. Ainsi, sauf

dans le cas de logiciels très spécialisés, il peut être difficile de distinguer les outils soutenant les processus de cycle de vie de soutien, des autres outils employés pour un cycle de vie primaire ou organisationnel dans une entreprise. Les logiciels qui mesurent la productivité, qui vérifient et surveillent les transactions effectuées par les usagers seront facilement déployés dans une organisation dans la mesure où ils donnent des résultats fiables et sont transparents pour les usagers. Des logiciels d'audit des employés, qui emploient une interface difficile à manipuler, sont à bannir.

Seuls des outils très spécialisés (tels des outils d'audit) sont exclusivement utilisés par le groupe responsable de l'assurance qualité. Dans ces situations, l'adoption de ces équipements n'aura pas un grand impact sur le fonctionnement de l'entreprise, mais plutôt sur le fonctionnement de groupe d'assurance qualité, s'il existe. Les outils de soutien utilisés par le seul groupe d'assurance qualité, ou de quelques rares usagers n'est pas difficile à gérer et ne mérite pas plus d'explications.

1.2.2 Distinction des règles d'affaires dans le système patrimonial

Le paragraphe 1.1.3 considère les systèmes patrimoniaux dans leur ensemble dans leur environnement technologique, mais il faut aussi pouvoir considérer les systèmes patrimoniaux individuellement. Il est difficile de remplacer un système patrimonial parce qu'on ne parvient pas toujours à reproduire des aspects utiles du système patrimonial. [ULRI 2002] identifie les règles d'affaires dans le système patrimonial comme étant les éléments de valeur qu'il faut capturer.

Ulrich distingue le code qui contient des règles d'affaires, et celui qui exécute les opérations. Le code, qui effectue les opérations techniques (manipulent les fichiers, transmettent les messages, etc.), a moins de valeur que le code qui contient la logique d'affaires de l'entreprise. Ces règles d'affaires sont incluses dans le code source. Cependant, il n'est pas facile de distinguer les instructions qui correspondent aux règles d'affaires, du code technique. Ce sont cependant ces règles d'affaires qui reflètent le fonctionnement de l'entreprise et que l'on pourrait vouloir déplacer vers un logiciel qui remplacerait le système patrimonial.

Il faut lier les règles d'affaires aux processus d'affaires de l'entreprise car les règles d'affaires vont probablement changer lorsque les processus d'affaires de l'entreprise changent aussi. D'une manière générale, la taille des règles d'affaires dans le code source représente 20 à 30 % du code source. C'est cette partie du code qui a le plus de valeur.

Les règles d'affaires peuvent être comprises de deux manières différentes : les déclarations de politiques ou conditions que doit satisfaire l'entreprise et les règles qui gouvernent la manière dont une entreprise fait des affaires. Pour ne pas tout inclure dans les règles d'affaires, [ULRI 2002] limite dans limite les règles d'affaires aux règles qui modifient des données d'affaires. Par conséquent des règles qui modifient des variables temporaires sont exclues.

Les règles d'affaires incluent :

- Les règles de présentation (validation dans les interfaces)
- Les conditions ou actions de déclenchement
- Les appels à d'autres parties du programme
- Les droits d'accès (Lecture et écriture)

Exemple

Soit un système informatique d'assurance médicale pour les employés d'une entreprise. Des règles d'affaires pourraient être.

Si l'employé fait une demande d'indemnisation
 Si l'employé est assuré et l'employé a toujours payé ses cotisations
 Alors la demande d'indemnisation est valide
 Ajouter la demande à la liste pour appréciation
 Changer le statut de la demande à "en attente d'approbation"

1.2.3 Méthodes CMM pour mesurer les tâches des employés

Les membres d'une organisation réalisent des tâches à l'aide des systèmes patrimoniaux. Le SEI (Software Engineering Institute), propose de modéliser le fonctionnement des organisations par des procédures. Une procédure peut réunir plusieurs tâches. Pour extraire des mesures sur les tâches des employés, les procédures de l'organisation doivent être écrites, identifiées, connues des employés, approuvées par l'entreprise, auditées et que les

outils informatiques employés pour réaliser ces procédures soient connus. Chaque groupe ou département de l'entreprise gère ses propres tâches.

Le SEI fournit un modèle aux entreprises pour mesurer leur niveau de maturité CMM (Capability Maturity Model) et déterminer si elles sont capables de gérer le développement, l'acquisition, et l'entretien de produits et services. Initialement, en 1991, il existait plusieurs modèles CMM distincts pour la gestion du génie logiciel, les ressources humaines, l'acquisition de logiciels et le développement intégré des processus et produits. En 2001, le SEI publia le CMMI (Capability Maturity Model Integration) [CMMI V1.1, 2002], qui fixe un seul modèle pour la gestion de tous les domaines d'activité examinés dans des modèles précédents. Le CMMI préconise la rédaction officielle des procédures de l'organisation, et qu'un groupe existe pour vérifier leur mise en oeuvre. Le CMMI fournit des règles à suivre pour gérer l'intégration de nouveaux produits dans l'entreprise qu'à partir du niveau 3. Par conséquent il paraît prématuré pour une entreprise de se lancer dans une migration de ses applications informatiques si elle n'est pas de niveau CMM 2, ou si il n'existe aucun contrôle de la qualité dans l'organisation. Dans la section 1.2.4 nous énumérons quelques approches pour pallier à l'absence de procédures écrites.

1.2.3.1 Le contrôle des processus de l'organisation

Il est très utile de savoir comment les usagers se servent réellement des applications dans une organisation, mais il n'est pas toujours possible de recueillir ces données. Parfois les diverses sous unités, d'une organisation, emploient des outils distincts pour effectuer des tâches analogues ou encore parfois, les sous unités emploient les mêmes outils mais pour des objectifs différents. Le groupe d'assurance a la responsabilité de d'uniformiser les procédures à travers une organisation. Ce groupe est censé dépendre de la haute direction de l'organisation plutôt que d'une unité ou équipe locale. Le groupe d'assurance qualité doit s'assurer que lorsque des plaintes ou erreurs soulevées, des ressources sont disponibles pour les résoudre, et qu'une solution est effectivement apportée.

Si on se trouve dans une entreprise sans groupe d'assurance qualité officiel, le plus souvent, il y a quelques membres qui effectuent des vérifications de manière intermittente ou de

manière non officielle. En outre, on peut essayer d'adopter des logiciels ou méthodes pour accélérer ou faciliter les vérifications.

1.2.4 Vérifier si les usagers réalisent des transactions autorisées

Sans processus documenté ni groupe de contrôle de la qualité, seul les transactions électroniques sont un indice pour déterminer les tâches des employés, contrôler leur travail et détecter des fautes. Les employés peuvent suivre des procédures parallèles, officieuses, ou tolérées dans une organisation. Avec une application patrimoniale, il est de bon augure de vérifier que seul les transactions autorisées sont effectuées. Un audit de la sécurité des transactions devrait faire apparaître tout manquement aux droits d'accès en fournissant un portrait plus fidèle du comportement des usagers. Les droits d'accès et de contrôle des employés dans l'organisation doivent respecter la hiérarchie dans l'entreprise et les processus de l'entreprise.

1.2.4.1 Intégrer les droits d'accès de plusieurs systèmes patrimoniaux

Jusqu'à maintenant nous avons expliqué comment et pourquoi analyser les droits d'accès des usagers pour comprendre les transactions qu'ils effectuent et les procédures qu'ils suivent. Il faut en outre considérer le cas où le projet modernise les droits d'accès des usagers. Le projet de migration ou de modernisation peut être un projet concernant les droits d'accès aux applications. Avec la multiplication des plates-formes et des systèmes, les usagers, dans une même organisation, peuvent être amenés à avoir de nombreux identifiants d'authentification. Les projets EAI (Enterprise Application Integration) peuvent lier des systèmes différents, cependant il faudrait accompagner le lien des opérations, d'une intégration au niveau de l'authentification. Parfois, le nombre des noms d'utilisateur et des mots de passe est si grand qu'ils constituent un obstacle à l'automatisation d'une opération puisque l'utilisateur devrait s'authentifier plusieurs fois avant de réaliser une transaction. La multiplication des systèmes d'authentification, complexifie les modifications cohérentes sur les droits d'accès aux usagers à travers toute une organisation. Des difficultés, qui surviennent, deviennent systémiques lorsque la gestion des comptes d'utilisateur et des droits d'accès ne suit pas un processus intègre.

Exemple

Lorsqu'un membre de l'organisation prend sa retraite, son compte usager n'est pas nécessairement supprimé pour toutes les applications informatiques. Les ressources humaines ne communiquent pas à tous les administrateurs réseaux le départ à la retraite du membre de l'organisation.

Une solution est de n'avoir qu'un seul nom d'utilisateur et mot de passe pour toutes les applications de l'organisation, et que toutes les authentifications étaient centralisées, intégrées ou identiques.

1.3 Identifier les problèmes du système patrimonial dans son environnement

Après avoir examiné le système patrimonial sous toutes ses coutures il faut être capable de résumer les problèmes posés par le système patrimonial. Les problèmes posés peuvent être techniques, mais ils peuvent être dus au fait que le système patrimonial n'est pas adapté à l'organisation dans laquelle il est utilisé.

1.3.1 Les problèmes techniques posés par le système patrimonial

L'identification du ou des problèmes permet de circonscrire les alternatives de solution possibles. Selon [ULRI 2002], il existe essentiellement 2 catégories de problèmes dans les systèmes patrimoniaux : des problèmes au niveau des données et au niveau des traitements.

Tableau 5 : Problèmes au niveau des traitements

Problème	Description
Fragmentation	Une opération logique est effectuée en faisant appel à d'autres fonctions à travers divers systèmes distincts
Redondance	La même opération se retrouve plus d'une fois à travers un ou plusieurs systèmes
Diversité Technique	Des applications qui sont censés fonctionner ensemble sont sur des plates formes différentes, écrit dans des langages différents.
Diversité de conception	L'architecture d'un système ne suit pas une conception unique ou cohérente
Taille & Temps	L'application est trop volumineuse ou le temps d'exécution est trop long pour offrir des services adéquats.

Tableau 6 : Problèmes au niveau des données

Problème	Description
Fragmentation	Des données liées sont distribuées à travers divers systèmes distincts
Redondance	Les mêmes données se retrouvent plus d'une fois à travers un ou plusieurs systèmes
Intégrité	Des informations contradictoires se retrouvent à travers un ou plusieurs systèmes
Sémantique	Des données homonymes (données ayant la même description) identifient des informations différentes Des données synonymes (données ayant une description distincte) identifient des informations identiques
Accessibilité	Les données ne sont pas accessibles aux usagers qui pourraient vouloir les manipuler
Flexibilité	Le système informatique ne peut pas être aisément modifié pour s'adapter aux besoins, changeants, des usagers dans un contexte d'affaires
Sécurité	Les données ne sont pas protégées contre une manipulation interdite ou erronée : lecture, ajout, modification, suppression. Les données ne sont pas encryptées pour empêcher de l'espionnage.

1.3.2 Analyser l'état de l'entreprise par questionnaire

Certains auteurs proposent, par des questionnaires, de déterminer si une organisation est prête à réaliser une migration. Ces questionnaires peuvent s'appliquer pour des organisations qui doivent faire du développement logiciel ou simplement acheter une nouvelle application qui doit en remplacer une autre.

1.3.2.1 Questionnaire de [BERG 1997]

Le questionnaire très détaillé de J. Bergey [BERG 1997], analyse le système patrimonial, le système cible, le projet de modernisation et l'organisation qui veut effectuer la modernisation.

Le système patrimonial : le questionnaire examine le système patrimonial, l'environnement patrimonial et les systèmes de soutien du système patrimonial. Par ce questionnaire on devrait notamment être en mesure de déterminer :

- Les algorithmes critiques du système,
- Les efforts de maintenance qui ont porté atteinte à l'architecture du système,
- L'inventaire des pannes et des recouvrements d'erreur pour analyser la fréquence et l'importance des erreurs qui sont survenues.

- les liens entre le système patrimonial et les intervenants externes
- les outils d'administration de développement, de tests, de logistique et l'adéquation de leur intégration

Le système cible : le questionnaire examine le système cible, son environnement, ainsi que son environnement de soutien pour lequel il faudra faire des achats.

L'organisation : le questionnaire examine le fonctionnement de l'organisation pour s'assurer qu'elle a des critères mesurables de succès pour l'atteinte d'objectifs. On s'assure que les employés concernés par des changements sont identifiés, que les objectifs sont correctement communiqués aux employés et que les problèmes soulevés par l'organisation suivent un processus qui doit aboutir à le résoudre.

Le projet : le questionnaire examine le plan de modernisation, l'évaluation des risques, la reformulation des spécifications et des besoins.

Les processus de développement logiciel : le questionnaire examine l'existence et le suivi des étapes formelles pour comparer les avantages de différentes stratégies, l'existence de processus d'évaluation des risques leur fréquence et leur résolution. Si des processus existent pour changer de langage de programmation ou la plate-forme, est-ce que ces processus sont respectés ? Comment gère t'on les changements logiciels aux projets en cours.

L'évolution technologique : la technologie évolue et il faut la suivre pour profiter des innovations technologiques et proposer des modernisations utiles. Il y a des questions pour passer rapidement en revue les innovations technologiques, et des questions plus précises pour sélectionner un produit en particulier lorsqu'une technologie spécifique est considérée comme pertinente.

Les paragraphes précédents doivent être interprétés dans une approche globale au niveau de l'entreprise, donc il faut s'assurer que les objectifs du projet de modernisation soient correctement communiqués aux intéressés, et que les rôles et responsabilités soient clairement définis. Il s'ensuit que si certains produits, processus ou activités, identifiés dans le questionnaire ne sont pas mis en œuvre il faut chercher la raison de cette absence (erreur ou choix délibéré, et si oui, pourquoi). Sans évolution unifiée d'un système on ignore

comment résoudre des difficultés globales, comment établir des priorités globales, comment s'assurer d'une coordination à travers l'entreprise. L'analyse par le questionnaire du fonctionnement de l'organisation et du projet peut se faire en parallèle à l'analyse du système patrimonial. Ensuite le questionnaire peut se concentrer d'avantage sur le système cible (l'avenir) ou la portée des nouvelles technologies.

1.4 Conclusion

La migration d'une application patrimoniale dans une grande organisation passe par l'analyse du système patrimonial. C'est-à-dire en examinant l'application informatique du point de vue de la gestion des configurations, de la maintenance logicielle, du fonctionnement de l'entreprise et des documents disponibles. Les membres d'une organisation sont affectés par la nature des processus de cycle de vie, que soutient une application. Les membres d'une organisation perçoivent différemment un changement à une application d'après le processus de cycle de vie soutenu. Cette perception se répercute au niveau de la responsabilité de financer ou de réaliser le changement.

Chapitre 2 Intégration et maintenance évolutive

Généralement, lorsqu'un projet de migration a été décidé, la préoccupation essentielle n'est pas de corriger ou d'améliorer l'application qui va disparaître mais de s'en débarrasser. Pourtant une nouvelle application doit s'intégrer dans l'environnement informatique de l'organisation. Cet effort d'intégration d'une nouvelle application requiert souvent des améliorations de l'environnement informatique dans lequel il s'intègre et qui va demeurer présent. Il existe plusieurs manières de construire le système cible à partir du système patrimonial. Le système cible peut être construit en ayant une profonde compréhension des mécanismes du système patrimonial. Il est aussi possible de construire le système cible à partir du système patrimonial ou sans porter atteinte au fonctionnement interne du système patrimonial. Au minimum, pour modéliser cette construction il faut pouvoir identifier et garder la trace des nouveaux éléments logiciels qui vont apparaître, de ceux qui seront modifiés et de ceux qui seront supprimés.

En particulier, cette section va s'attarder sur le cas de l'EAI (l'intégration d'applications de l'entreprise). Le terme EAI est relativement récent et est un concept qui a suivi l'essor des applications ERP et qui requiert une liaison entre les applications ERP et les systèmes patrimoniaux. Dans la recherche, la définition du terme EAI demeure floue, et réunit un ensemble de notions différentes. Pour certains, l'EAI correspond à tout effort d'intégration que ce soit à l'intérieur ou à l'extérieur de l'entreprise. Pour d'autres auteurs l'EAI correspond à un projet qui accompagne une réingénierie des processus d'affaires. Cette section examinera les divers efforts de modèles qui ont été proposés pour soutenir un projet EAI.

2.1 Changements théoriques dans l'environnement patrimonial

Plusieurs auteurs ont établi des catégories des processus de modernisation d'un système patrimonial. Un processus se base sur la compréhension du code source modifié. On peut moderniser le code source en analysant le code source ou sans l'analyser.

On utilise la modélisation par niveaux logiciels où se situent les modifications pour déterminer le type de changement requis. La traçabilité des les changements dépend d'une gestion des configurations rigoureuse tel que mentionné à la section 1.1.3.

D'après Robert Seacord [SEAC 2000], il existe 2 approches, l'approche « white box » et « black box ». Nous les mentionnons ici pour mémoire.

Modernisation du système patrimonial par l'approche boîte blanche

C'est un effort de modernisation en analysant et comprenant le code source ou « white box ». La compréhension du fonctionnement interne de l'application par une modélisation pointue est la base de l'effort de modernisation. Cela implique de produire ou analyser une modélisation de l'application et du domaine pour comprendre le code source.

Modernisation du système patrimonial par l'approche boîte noire

Modernisation sans analyser ou comprendre le code source ou « black box ». L'analyse se concentre sur la compréhension des interfaces d'un l'élément logiciel. La modernisation de l'élément logiciel correspond à une forme d'encapsulation. L'analyse ignore le code source et le fonctionnement interne de l'élément logiciel qui n'est pas modifié. Cette modernisation comporte des limites.

La maintenance corrective se fait souvent en boîte noire, sans comprendre le système patrimonial, [ULRI 2002]. Sans le savoir, du code peut être ajouté pour réaliser une fonction déjà programmée dans l'application. Cette nouvelle fonction constitue une redondance dans le système. Avec le temps, ces redondances peuvent s'accumuler. L'objet de la migration structurée est d'effectuer des changements contrôlés.

On peut donc considérer qu'il existe un type de changements selon le niveau logiciel de Jacobson [JACO 1997], présenté dans la section 1.1.4.2.

Tableau 7 : Type de changement par niveau logiciel de [JACO 1997]

Niveau du changement	Type de changement
Applications	Encapsulation de l'application. Connexion des interfaces des applications Lecture et transmission ou interprétation des impressions d'écran sous un autre format. Émulation de l'application à distance
Composantes d'affaires	Intégration d'applications d'affaires distinctes qui sont censés fonctionner ensembles. Réingénierie des applications d'affaires
Composantes intergicielles	Augmenter la taille du réseau. Rendre l'application disponible sur un réseau informatique réparti.
Composantes systèmes	Porter vers nouveau système d'exploitation. Changer un équipement matériel (carte graphique)

Tableau 8 : Type de projet de maintenance évolutive et migration, selon [ULRI 2002]

Type de projet	Description du projet
Consolidation d'applications	Réorganiser et améliorer l'architecture d'un système informatique pour lui ajouter des qualités qu'elle requiert.
Intégration de plusieurs applications	Prendre des applications séparées et les faire fonctionner ensemble comme un tout. Exemple : Système de gestion de la paie, des assurances médicales et des retraites pour les employés d'une organisation.
EAI : (Intégration des applications de l'entreprise)	On ne modifie rien, on intègre tout. Ce type de changement ne modifie pas le fonctionnement interne des divers systèmes qu'il ne fait que regrouper.
Analyse de package et déploiement	Remplacer un système patrimonial par un système informatique complet
Changement de plate-forme	Déployer sur une autre plate-forme avec un autre système d'exploitation. Cela implique souvent de changer recompiler le code source avec un nouveau compilateur
Re-conception des données	Réorganiser et améliorer l'architecture des données d'un système informatique

2.2 EAI : L'intégration d'applications de l'entreprise

Le terme EAI est relativement récent et est un concept qui a suivi l'essor des applications ERP. En effet, les ERP se proposent de gérer les ressources de l'entreprise, mais encore faut-il les faire fonctionner correctement avec les systèmes déjà en place.

Le EAI correspond à une discipline de génie de l'intégration des applications. Les projets EAI sont des projets qui s'efforcent de tirer profit des divers systèmes patrimoniaux, de les réunir pour obtenir un nouveau produit. Le concept a été résumé par certains auteurs par la formule « On ne modifie rien, on intègre tout ». Dans un projet EAI, l'essentiel du travail

visée à gérer des groupes restreints de tâches qui regroupent des sous-tâches qui s'exécutent en parallèle sur les systèmes patrimoniaux.

Le choix de recourir à l'EAI est valable lorsque les systèmes patrimoniaux sont :

- fiables,
- responsables de systèmes critiques,
- efficaces et l'altération de ces systèmes est risqué et coûte cher,
- capables d'offrir des fonctionnalités très spécialisées et sont difficiles à remplacer.

Essentiellement, lorsqu'on met en œuvre une solution EAI, il n'y a pas de justification du point de vue temps et argent de remplacer le système patrimonial. Pour autant il est incorrect de considérer les efforts EAI « Engineering Application Integration » comme un travail généralement rapide et peu structuré.

Plusieurs technologies, notamment Corba (Common Object Request Broker Architecture) de l'OMG (Object Management Group) ont permis de réaliser beaucoup de progrès pour l'EAI en encapsulant les systèmes patrimoniaux dans des objets et en les rendant disponibles sur un réseau par des RPC (Remote Procedure Call : appels de procédure à distance).

Pour étudier l'EAI, dans un premier temps, il est nécessaire d'examiner la modélisation organisationnelle nécessaire pour l'EAI, la modélisation technique de l'EAI et examinerons enfin un modèle d'évaluation des outils EAI.

2.2.1 La modélisation organisationnelle de l'EAI

La puissance et l'indépendance des applications développées au cours de la dernière décennie rendent ces applications de plus en plus isolées. Pour réaliser l'intégration de ces applications dans les organisations, plusieurs auteurs ont proposé des modèles pour catégoriser et cibler le travail d'intégration à accomplir.

Dans leur recherche, Losavio et al. [LOSA 2002], étendent un modèle conceptuel d'intégration d'outils CASE pour avoir une conception unique et cohérente au niveau des *processus*, *services* et *mécanismes* dans une organisation. Ce qui est essentiel dans cette recherche est le mot *intégration*. Les auteurs établissent des catégories d'intégration :

1. Intégration montante « Upward » : correspond à l'intégration des processus, services ou mécanismes se rapportant à la vision hiérarchique dans l'organisation.
2. Intégration arrière « Backward » : correspond à l'intégration des processus, services ou mécanismes se rapportant à la gestion interne de l'organisation.
3. Intégration frontale « Forward » : correspond à l'intégration des processus, services ou mécanismes se rapportant à la gestion des rapports que l'organisation entretient avec l'extérieur.

Tableau 9 : Modèle organisationnel de l'EAI

	Intégration montante « Upward »	Intégration arrière « Backward »	Intégration frontale « Forward »
Processus			
Exemple : Gestion de l'approvisionnement	Planification de l'approvisionnement de la haute direction	Approvisionnement pour les producteurs	Approvisionnement avec les fournisseurs, et vendeurs
Services	Systèmes d'aide à la décision Systèmes à base de connaissances Systèmes d'audit ou d'assurance qualité	Outils case Systèmes patrimoniaux ERP	Systèmes de gestion des données client Systèmes de gestion des données du fournisseur Système de e-commerce entre entreprises Système de e-commerce entre le vendeur et le client
Mécanismes			
Architecture	Styles architecturaux <ul style="list-style-type: none"> • Invocation par évènements • Niveaux en couche • Dépôt d'archives 	Patrons <ul style="list-style-type: none"> • Courtier • Réflexion • Façade d'encapsulation • Intercepteur 	
Technologie	Matériel Logiciel BD Technologie de communication		

Selon [LOSA 2002], la distinction entre Intégration montante, Intégration arrière, Intégration frontale, ne s'applique pas lorsqu'on considère les mécanismes. Les mécanismes peuvent soutenir les 3 vues différentes de l'intégration. Les mécanismes correspondent à ce qui était examiné sous le titre d'*outils de communication* examinés dans la section 1.1.4.

Les mécanismes peuvent se diviser en deux niveaux : l'architecture et la technologie.

L'architecture fournit une description syntaxique et sémantique qui permet la conception et l'analyse de systèmes créés à partir de composants. Des connecteurs organisent l'interaction entre les composants. Les composants et les connecteurs communiquent à travers des interfaces.

L'auteur examine différents styles architecturaux et patrons architecturaux et leur pertinence vis-à-vis du standard [ISO/IEC 9126-1 1998] sur l'évaluation de la qualité des applications logicielles. Les six critères d'évaluation de ce standard sont la capacité, la fonctionnalité, la fiabilité, la facilité d'utilisation, le rendement, la maintenabilité, la portabilité.

L'analyse de [LOSA 2002] est très fine, mais est très succincte, laconique et emploie un vocabulaire peu répandu. Ce modèle répond aux mêmes préoccupations de définition de l'effort EAI selon le contexte des processus qu'il soutient, et de catégorisation des composants informatiques selon les patrons employés ou niveaux de logiciel. Ces éléments ont été examinés dans ce mémoire à la section 1.1.4, L'architecture selon des vues par niveaux logiciels, enfin les réflexions sur les mécanismes peuvent être comparés à ce qui a été écrit à la section 1.2.1, Systèmes patrimoniaux et nature des opérations. Mais l'article précité suit une démarche rigoureuse va plus loin que ce mémoire en comparant la modélisation de l'EAI selon le standard [ISO/IEC 9126-1 1998] d'évaluation logicielle.

Le vocabulaire utilisé est sans doute bien défini, mais leurs recherches ne semblent pas avoir été largement reconnues ou reprises par la majorité des spécialistes du domaine. Comme exemple, Losavio utilise le terme « intégration » dans un sens très large qui inclut la « réingénierie des processus d'affaires » et intégration technique alors que d'autres auteurs du domaine établissent clairement une distinction entre ces deux concepts.

2.2.2 La modélisation technique de l'EAI

En 2001 Oba [OBA 2001] a proposé une modélisation technique de l'EAI en se fondant sur des outils de l'OMG (Object Management Group) : soit Corba pour gérer les interactions entre les systèmes informatiques répartis, et le formalisme de JointFlow pour modéliser les processus d'affaires de l'organisation. Selon Oba, les outils disponibles ne satisfont pas

tous les besoins EAI. Les besoins technologiques indispensables pour mettre en place l'EAI sont :

- Premièrement : Un système ouvert sur un réseau qui permet la liaison entre de nombreux systèmes externes
- Deuxièmement : Des moyens pour adapter et modifier les flux de travail.
- Troisièmement : Des moyens pour décrire et exprimer des processus d'affaires.

En 2002 Kwak [KWAK 2002], reprend et complète la recherche de Oba en raffinant certains concepts et en utilisant les standards XML pour modéliser les processus d'affaires. En effet, pour accroître l'interopérabilité de systèmes hétérogènes des vendeurs de systèmes de gestion de flux de processus « Workflow Engine » ont mis au point un nouveau standard qui s'ajoute à XML appelé Wf-XML (Workflow-XML).

Un système de gestion de flux de processus attribue à chaque processus des pré-conditions de déclenchement et des post-conditions qui définissent les sous tâches qui peuvent être déclenchées ensuite. Le système de gestion de flux de processus peut démarrer des processus sur d'autres ordinateurs dans un sous réseau pour réaliser une tâche globale ou confier une partie de l'exécution à un autre « Workflow Engine ».

[KWAK 2002] considère plutôt l'EAI entre une organisation et d'autres organisations externes alors que l'approche de [OBA 2001] concerne plutôt l'intérieur de l'organisation.

C'est pourquoi il définit les sous processus internes et les sous processus externes, qui peuvent fonctionner sur les machines d'une autre organisation. Ce point de vue implique un moyen de partager la gestion des processus d'affaires entre les organisations. On y parvient avec un moteur de flux de processus, d'un adaptateur et d'un inventaire d'interface de service (« Service Interface Repository »). Le « Workflow Engine » a une liste des sous processus dont il a le contrôle : les sous processus internes, et d'une liste de sous processus dont un autre « Workflow Engine » a le contrôle : les sous processus externes.

Le « Workflow Engine » peut être comparé à une machine enregistrant l'état des divers sous processus en réponse à des événements extérieurs (par exemple lorsqu'un sous processus se termine) et en démarrant les nouveaux sous processus sur le poste voulu.

Une division de l'EAI proposée par Oba proposait une distinction entre les tâches structurées et ordonnées, dont on connaît la séquence de travail, et les tâches non structurées dont on ne connaît pas la séquence de travail. La nomenclature de Kwak ressemble à celle de Oba qui distingue l'EAI statique et l'EAI dynamique.

EAI statique :

Le concepteur du processus d'affaires définit à l'avance et dans le détail toutes les connexions entre les applications de l'entreprise qui peuvent être liées ensemble. Dans ce cas l'exécution des processus de travail est censée suivre des chemins prédéfinis. Il est alors difficile d'ajouter ou de modifier un processus ou changer une configuration lors de l'exécution du processus. On pourrait citer ici l'exemple de traitements séquentiels en lot peu sophistiqués. Le EAI statique est plus simple à mettre en place mais plus long et coûteux à modifier.

EAI dynamique :

La liaison des services des applications cibles de l'organisation est réalisée lors de l'exécution selon des règles de liaison. Dans ce cas ce type d'intégration permet d'identifier des sous processus internes et externes. Cette nomenclature est efficace pour contrôler des processus distribués parmi différentes applications de l'entreprise.

Pour partager les opérations entre les divers systèmes à intégrer, Oba propose la nomenclature suivante :

Tableau 10 : Nomenclature des transactions d'un système EAI

Tâches	Description
Processus d'affaires	Un processus d'affaire possède des attributs lui permettant d'identifier et de déclencher des instances de processus qui se rapportent à un domaine d'opération de l'organisation
Instance de processus	Instance concrète d'un processus d'affaire.
Élément de travail	Tâche qui doit être effectuée par des personnes ou ordinateurs à l'intérieur d'une instance de processus. Les éléments de travail peuvent s'exécuter en parallèle.

Le modèle qui soutient l'EAI précise les flux et le type de transitions entre les éléments de travail. On distingue 4 types de nœuds de contrôle :

- Partage OU,
- Partage ET,
- Fusion OU,
- Fusion ET.

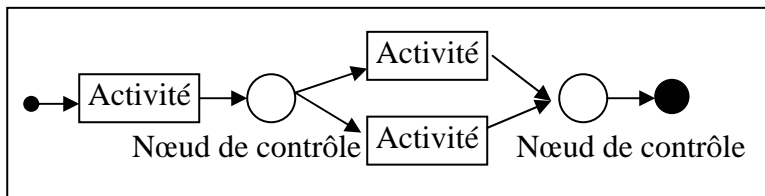


Figure 2 : Modèle des processus d'affaires distribués de [KWAK 2002]

Le modèle des processus d'affaires de la figure 2 illustre le contrôle des processus inter systèmes au niveau des nœuds de contrôle. La recherche de Kwak présente une vision plus raffinée au niveau des transitions d'états des éléments de travail pour qu'ils soient contrôlés par les processus distribués.

Pour contrôler les processus inter systèmes on propose d'employer une technologie qui associera un état aux instances des procédures à distance. Cet état sera enregistré et modifié dans une base de données. Les sous opérations sur plusieurs systèmes indépendants qui composent le processus d'affaire auront toutes un état associé dans la base de donnée. Ces états tels que définis par Kwak, sont illustrés par la figure 3.

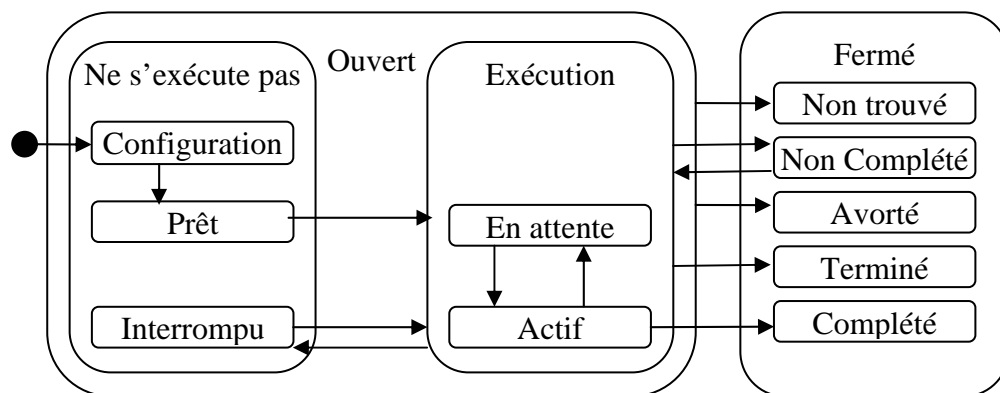


Figure 3 : Diagramme de transition d'états des processus et de tâches distribués

Ce diagramme de transition d'états est plus détaillé dans la recherche de Kwak que dans celle de Oba. Des explications sont fournies, en ce qui concerne les états finaux d'un processus distribué, pour tenir compte des divers problèmes et interruptions qui peuvent le corrompre.

Tableau 11 : Description des états finaux d'un processus distribué

Statut	Description
Non Trouvé	Le service est introuvable
Avorté	Le fournisseur décide que le service ne peut être rendu car il a été remplacé par un autre ou d'autres problèmes surviennent
Terminé	Les services rendus par le processus sont identiques mais des étapes sont enlevés à la suite d'une réingénierie des processus d'affaires
Non Complété	Le système cible met fin au processus le processus. Cependant cette fermeture n'est pas une décision du fournisseur de service. Elle peut résulter d'une panne du système du fournisseur. Dans ce cas, il faut recommencer le processus et le compléter entièrement.
Complété	Le processus distribué se termine normalement lorsque toutes les opérations sont terminées

Les modèles EAI tels qu'identifiés à la figure 2 se suivent et ont tendance à se ressembler. Ces modèles ont des ressemblances avec les diagrammes de transitions d'état ou des diagrammes d'activité selon le formalisme UML.

2.2.3 Modèle d'évaluation d'outils EAI

L'EAI a pour objectif de réaliser une intégration des applications de l'entreprise, sur mesure. Des outils informatiques sont disponibles pour soutenir cette intégration. Or, d'après [THEM 2003], les systèmes ERP sont complexes et non flexibles. Souvent les ERP ne sont pas conçus pour collaborer avec d'autres applications. Les systèmes ERP échouent parfois dans l'objectif d'intégration de tous les systèmes de l'entreprise. Pour évaluer un produit qui doit soutenir l'effort d'EAI selon Themistocleous [THEM 2003], il y a deux étapes :

- l'évaluation des technologies d'intégration
- l'évaluation d'outils EAI spécifiques

Toutes les recherches en EAI se concentrent sur des niveaux d'intégration pour mesurer l'efficacité des outils EAI. En effet, les nombreux standards et tendances du marché ont

contribué à créer des confusions technologiques. Pour aider les organisations à choisir un produit qui répond à leurs besoins, les auteurs proposent d'évaluer la performance des applications à différents niveaux :

- Niveau de connectivité (définit le type d'intergiciel employé)
- Niveau de transport (définit le transport des données d'un système à un autre)
- Niveau de traduction (définit la traduction des données d'un format en un autre)
- Niveau d'automatisation des processus (définit la modélisation des processus d'affaires)

Chaque couche est évaluée à l'aide des critères du tableau 12.

Tableau 12 : Critères¹ de performance pour l'évaluation par niveaux de [THEM 2003]

Critère	Description
Outils de développement	Des outils de développement permettent aux usagers de manipuler et d'intégrer les applications sans avoir à en comprendre le fonctionnement interne, mais aussi de développer des adaptateurs ou interfaces sur mesure pour des applications patrimoniales maison.
Communication asynchrone	La communication asynchrone est un dispositif de communication différé dans le temps – asynchrone - entre plusieurs machines
Communication synchrone	Les transactions synchrones sont des opérations qui interrompent leur exécution et attendent une réponse d'une autre machine avant de poursuivre leur exécution.
Intégration de systèmes sur mesure	Le produit permet l'intégration de systèmes sur-mesure
« Package »	Le produit se limite à l'intégration de systèmes standard ou génériques
Intégration intra organisationnelle	Le produit permet l'intégration des processus entre les organisations
Intégration inter organisationnelle	Le produit permet l'intégration des processus à l'intérieur de l'organisation

2.3 Améliorer une application patrimoniale

Cette section examine les moyens intrusifs de modifier et corriger les erreurs qui existent dans une application patrimoniale. Ces moyens correspondent à une analyse statique du code source. Dans son livre [ULRI 2002] présente un inventaire des anomalies des données

¹ [THEM 2003] propose huit critères de performance pour l'évaluation par niveaux, nous reprenons les sept critères principaux. Le dernier critère « Integrated » apparaissant redondant.

et traitements qui peuvent être détectées et corrigées. Nous présentons les catégories d'anomalies répertoriées et les pistes de solutions pour corriger des erreurs dans cette section. La difficulté principale au niveau de l'intégration des données est le problème de réconciliation sémantique des données homonymes ou synonymes, de plusieurs applications distinctes. La recherche de [SHET 1990] fait autorité dans le domaine de la gestion de bases de données fédérées.

Il existe aussi des moyens pour restructurer un système patrimonial en l'encapsulant dans un objet. Il est ainsi possible d'étendre les fonctionnalités du système patrimonial. De nombreux outils dans l'industrie et de nombreuses recherches dans le domaine examinent comment transformer une application patrimoniale en un objet. Cette solution n'est pourtant pas une panacée.

2.3.1 Supprimer les anomalies du code source

Même si un code source compile sans générer de message d'avertissement, des anomalies peuvent s'y trouver. [ULRI 2002] dans son livre passe en revue les anomalies et les solutions qu'on peut employer pour analyser le code d'une manière statique pour détecter plus facilement les anomalies dans un code source.

2.3.1.1 Liste d'anomalies

Corruption de la logique du code :

Lorsqu'on exécute des opérations en utilisant des instructions développées à d'autres fins on s'attaque à la cohérence du code. Exemple : avoir une instruction dans un code source dont le but est de générer une erreur. L'erreur générée permet de sortir d'une boucle « while ».

Boucles infinies :

Les boucles infinies sont une erreur qui n'empêche pas un programme d'être compilé correctement. Cependant, si le système rentre dans une boucle infinie, il ne pourra plus en ressortir. Pour sortir de la boucle, il faut arrêter l'exécution du programme pour le redémarrer.

Code mort :

Le code mort est du code dans une application qui n'est jamais exécutable quelle que soit la configuration de l'application. On peut supprimer le code mort dans un code source sans que cela ait une incidence sur l'application.

Récurrence :

La récurrence a un comportement différent selon les compilateurs et les systèmes d'exploitation. Il faut donc prêter une attention particulière aux opérations qui emploient la récurrence et remplacer ces instructions si le besoin s'en fait sentir.

2.3.2 Améliorer la conception du code source

Certains efforts d'intégration de systèmes patrimoniaux ne sont pas toujours correctement communiqués aux cadres, en particulier lorsqu'il s'agit de modifications rapides. Il peut en résulter une conception confuse. Pour structurer le code il importe d'avoir un code source modulaire et simple. Généralement les modules devraient avoir la même quantité de points de fonctions, être facile à lire et les règles d'affaires, tels que définis dans la section 1.2.2, soient lisibles dans le code source.

2.3.2.1 Données figées dans le code

Il faut extraire des données figées dans le code source (en anglais « hard coded »). Cette extraction peut se révéler coûteuse. Le plus souvent les développeurs d'applications patrimoniales, gérées correctement, ont généralement conscience du problème de données figées dans le code source. Cependant on ne retrouve pas toujours la même attention pour les données figées dans le code source des tests automatiques qui vérifient et valident le fonctionnement d'un système patrimonial.

Exemple de données figées dans une application :

Soit une application qui enregistre des achats. Plusieurs articles peuvent être enregistrés en une transaction. Le logiciel doit calculer la somme des prix des articles et calculer le prix des taxes gouvernementales. Si les valeurs des taxes sont spécifiées dans le code source, il faut les modifier dans le code source, recompiler

l'application et l'installer dans toutes les caisses enregistreuses lorsque la valeur des taxes change. Il serait mieux que l'on puisse modifier la valeur des taxes en passant un paramètre à l'application dans un champ spécifique.

2.3.2.2 Structurer le code source

L'article de Markosian [MARK 1994] n'est pas récent mais demeure quand même pertinent. Il explique que pour transformer des systèmes patrimoniaux il faut des outils appropriés pour rendre les applications rédigées en un seul code source modulaire. Un outil, étudié par Markosian, analyse un code source Cobol rédigée en un seul long bloc de code. L'outil réorganise le code source en transformant les accès aux variables globales en appels de fonctions avec des paramètres. L'outil enlève des variables globales, apanage des programmes procéduraux, et crée un système plus fonctionnel par de l'encapsulation. L'outil permet de calculer les paramètres des fonctions sans erreurs, plus efficacement que ne le ferait un humain.

2.3.2.3 Modélisation des dépendances du code source

L'article de Robert Seacord [SEAC 2001-b] expose les techniques mises au point lors de la migration d'une application écrite en Cobol en une application programmée en Java.

On peut grouper des ensembles de fonctions selon des transactions liés logiquement ou selon les transactions liés par le système patrimonial. Seacord propose de modéliser la migration de code à l'aide de graphes d'appels de fonctions. Ces graphes regroupent les fonctions selon les appels qui sont envoyés à d'autres fonctions.

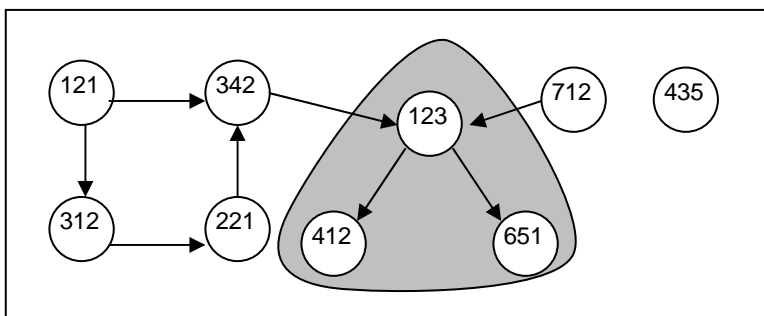


Figure 4 : Diagramme de graphes d'appels de fonctions

L'article définit quatre types de fonctions

- Fonctions racine (comme le 712)

- Fonctions nœud (comme le 123)
- Fonctions feuilles (comme le 412)
- Fonctions isolées (comme le 435) sans lien avec d'autres

Pendant un projet de migration incrémentale, de nouveaux modules de l'application cible remplacent des modules de l'application patrimoniale. Pour éviter des erreurs, le transfert du contrôle entre l'ancien et le nouveau système pour une même tâche doit être réduit au minimum. Ainsi, chaque système effectue le maximum d'opérations avant de transférer entièrement le contrôle à l'autre système. Selon le formalisme de [SEAC 2001-b], les fonctions les plus difficiles à modifier sont les fonctions nœud puisqu'elles exigent le développement d'interfaces avec le système patrimonial. Les fonctions racine sont plus faciles à modifier que les fonctions nœud. Les fonctions feuilles sont les plus faciles à modifier car ces fonctions sont isolées, mais sont aussi très rares.

Les fonctions 123, 412 et 651 de la figure 4 peuvent être remplacées lors d'une seule étape, car les deux feuilles sont appelées par une fonction nœud 123. Le nouveau composant qui remplacera ces fonctions doit détecter les appels envoyés à la fonction 123 et effectuer les traitements que réalisaient les trois fonctions, c'est-à-dire avoir une interface avec les fonctions 342 et 712.

Cette modélisation est intéressante, mais dans le cas d'un grand système, on peut se retrouver avec des graphes d'appels contenant tellement d'éléments et de relations qu'ils deviennent illisibles. Pour remédier à cette complexité Seacord a proposé les simplifications suivantes :

- Produire un graphe d'appels de fonctions séparé pour chaque élément racine
- Éliminer les appels de fonctions lorsque la fonction s'appelle elle-même
- Ne pas considérer les appels de fonctions qui lisent, mettent à jour, ou suppriment des informations dans des bases de données

Les graphes résultants, ainsi simplifiés, deviennent moins complexes à manipuler et à comprendre. Cette modélisation peut être utilisée en complément de la modélisation UML et soutenue par des outils CASE ou avec un compilateur spécialisé et correspond à une technique d'analyse statique et de collecte de métriques du logiciel.

Les graphes d'appel de fonctions constituent une technique d'analyse de bas niveau qui ne regarde des systèmes que du point de vue fonctionnel. Cette analyse comporte donc des limites et n'est utile que pour des migrations lorsque le code source du système patrimonial et du système cible sont disponibles. Lorsque la modernisation fait l'objet de nombreux incréments, cette démarche a l'inconvénient de faire ressembler le système cible au système patrimonial puisque on le remplace morceau par morceau.

2.3.3 Une méthode spécifique : la migration orientée objet

Le paradigme objet est un paradigme relativement récent qui a influencé le génie logiciel en simplifiant beaucoup de difficultés liées à l'encapsulation, l'entretien, la modification et la réutilisation de code, tout en assurant une plus grande sécurité. Par conséquent, beaucoup d'auteurs ont expliqué les avantages de ce nouveau paradigme pour les systèmes patrimoniaux. Il faut cependant noter que l'encapsulation d'un système patrimonial dans une technologie orientée objet n'est pas une panacée et que certaines technologies considérées comme orienté objet ne confèrent pas tous les attributs du paradigme objet.

2.3.3.1 Établir une architecture objet distribuée

La recherche de Seacord [SEAC 2000] passe en revue toutes les approches de modernisation. Une architecture objet distribuée est un premier pas pour l'intégration des systèmes distribués. Les intergiciels, tel que Corba, permettent à des systèmes d'information orientés objet d'être disponibles sur des réseaux par des appels de procédures à distance. Cependant, la conversion d'un code source en langage procédural à un langage orienté objet n'est pas une tâche aisée. Par ailleurs, l'intégration de 2 intergiciels (Corba et .NET) peut être la source de bien des complications.

2.3.3.2 Encapsuler les systèmes patrimoniaux en objets

La recherche de [O'CA 1999], illustre l'importance de la programmation orientée objet pour les systèmes patrimoniaux. Les objets encapsulent les données et les processus qui les manipulent. Le fonctionnement d'un système repose sur l'indépendance des objets. Le fonctionnement global du système se base sur l'échange de messages entre les objets. Cette structure permet de réduire la complexité des applications en permettant des modifications

par un raffinement successif des objets. Pour être justifiable, l'adoption de la technologie objet doit améliorer les bénéfices d'affaires et accroître la productivité et fournir de la flexibilité face aux changements dans le domaine.

2.3.3.3 Les attributs du paradigme Objet

Lorsque des auteurs proposent d'encapsuler un système patrimonial dans un objet, ce nouvel élément logiciel ne possédera pas nécessairement tous ces attributs du paradigme objet. En effet, le Visual Basic de Microsoft, avant la technologie .NET, ne contenait pas la notion d'héritage, de polymorphisme, ou de classes abstraites. Le C++ ne fait pas de gestion automatique de la mémoire. Bertrand Meyer [MEYE 1988], dans son livre « Object Oriented Software Construction », au chapitre « Seven steps to object based happiness », dénombre sept attributs correspondant à des niveaux que doivent respecter les applications pour respecter le paradigme orienté objet.

Tableau 13 : Sept attributs correspondant au bonheur orienté objet de [MEYE 1988]

Niveau	Caractéristique	Description
1	Des objets basés sur une structure modulaire	Les systèmes sont modulaires en considérant leurs structures de données
2	Abstraction des données	Les objets sont décrits comme des instanciations de types de données abstraites
3	Gestion automatique de la mémoire	La mémoire non utilisée doit être désaffectée automatiquement
4	Les classes	Tous les types de données non primitifs sont des modules et chaque module de haut niveau est un type
5	L'héritage	Une classe peut être définie comme une extension ou une restriction d'une autre
6	Le polymorphisme et la liaison dynamique	Les entités du programme peuvent référer à des objets de plus d'une classe et les opérations peuvent avoir différentes exécutions dans différentes classes
7	L'héritage multiple et répétée	Il est possible de déclarer une classe comme héritier de plus qu'une classe et hériter d'une même classe plusieurs fois.

2.3.3.4 Limites

Même si la migration orientée objet fait l'objet de recherches et fournit des outils qui permettent de réaliser la migration plus facilement, le paradigme objet n'est pas une

panacée. Cette technique demeure difficile à gérer lorsqu'on est en présence d'un très grand nombre de classes.

Selon Taivalaari [TAIV 1995], les techniques traditionnelles de gestion de bibliothèques de classes sont inadéquates lorsqu'il y a plus de 500 classes. Avec autant de classes, il devient difficile de visualiser les relations entre les objets dans leur hiérarchie. Les outils de conception graphique sur le marché ne s'adaptent pas encore bien aux projets avec un milliard de lignes de code, pourtant plus la taille du code grandit, plus on a besoin d'outils de visualisation.

La modularité dans les programmes orientés objet peut être autant un avantage qu'un inconvénient. La modularité permet d'éviter aux informaticiens d'effectuer des changements ciblés au code source. Lorsque le code source est bien conçu, l'informaticien profite de la modularité pour situer précisément ses changements et obtenir un système performant. Cependant, si le code est mal conçu cette mauvaise conception est pérennisée par la modularité. L'informaticien peut toujours profiter de la modularité, et reporter à l'avenir des changements à la conception qui seraient pourtant nécessaires. Ainsi, la programmation orientée objet a des avantages par rapport aux autres paradigmes de programmation, même si des programmes, respectant le paradigme objet, peuvent devenir obsolètes.

Des recommandations de [TAIV 1995] pour gérer les bibliothèques de classes de plus de 500 classes préconisaient :

1. La séparation des politiques et mécanismes dans une bibliothèque de classe ou composant
2. L'identification claire des politiques et mécanismes
3. Soutenir la coexistence pacifique des bibliothèques de classes ou composants dans un système
4. Une identification claire de quoi, où et comment étendre la bibliothèque de classes ou composants
5. De bien définir les exceptions sémantiques
6. De suivre les règles de conception

En outre, il faudrait avoir des critères objectifs d'obsolescence, pour dire qu'un élément logiciel est arrivé à son terme.

2.4 Conclusion

L'analyse de l'environnement patrimonial doit permettre de comprendre, et de corriger les défauts de l'environnement dans lequel l'application cible doit s'intégrer. L'analyse doit permettre de tirer profit de l'environnement patrimonial pour intégrer la nouvelle application. La technologie orienté objet, et en particulier la technologie Corba, a permis d'encapsuler des applications patrimoniales et de les rendre disponibles en réseau. Cette technologie a stimulé le développement d'une nouvelle discipline, encore imprécise, de génie de l'intégration d'applications : l'EAI. L'EAI peut être utilisé en complément d'un projet ERP ou indépendamment de toute nouvelle application.

Chapitre 3 Identification des objectifs

Dans un projet informatique il faut spécifier les problèmes à résoudre. Ce n'est qu'ainsi qu'on peut fixer des objectifs réalistes et établir des livrables spécifiques qui sont réalisables et mesurables.

Une stratégie complète devrait soigneusement tester toutes les hypothèses avant d'émettre des objectifs généraux ou spéciaux pour s'assurer qu'on se concentre sur le bon problème. Le système patrimonial est-il réparable ou faut-il complètement le remplacer ? L'objectif est-il purement technique ou correspond-il à une réingénierie des processus d'affaires ? Lorsque l'objectif principal est purement technique, alors l'adoption de la technologie est une fin en soi. Par ailleurs, lorsque le changement technologique est l'objectif de la migration la rentabilité du nouveau système peut être considérée comme un sous objectif.

Lorsque l'objectif du projet est la réingénierie des processus d'affaires, alors l'informatique est un moyen et non pas une fin. Il ne faut pas chercher à déployer un système à tout prix, car l'objectif principal est de modifier le comportement des membres de l'organisation. Dans un projet de réingénierie des processus d'affaires, la rentabilité du système informatique ne devrait pas être un sous objectif, c'est plutôt la modification du processus d'affaires qui doit être recherchée. Il est facile d'identifier le non respect d'une spécification du système ou encore d'identifier les vices cachés dans un logiciel lorsqu'on les découvre. Cependant, l'absence de gains espérés est difficilement attribuable exclusivement à un logiciel. Notons que même les tribunaux ne reconnaissent pas les dommages de la part d'un fournisseur de logiciel, lorsque les logiciels ne donnent pas les gains espérés, comme en fait état le livre de [RAE 1994]. La responsabilité légale n'est engagée que par des vices cachés dans le logiciel. En outre, [RAE 1994] explique que l'absence de profits ou le manque de productivité des usagers d'un logiciel acheté ne sont pas des motifs de remboursement reconnus par les tribunaux.

L'objectif du projet de migration se distingue des objectifs pendant la période de transition. En effet, les objectifs pendant la période de migration sont d'avoir un niveau déterminé de qualité, pendant et juste après la migration. Un niveau de qualité devrait être mesuré

pendant la période d'ajustement ou de rodage. Cette qualité examine plutôt la stabilité de l'application que la stabilité de la productivité des usagers.

En ayant une vision claire des objectifs et sous objectifs on peut identifier les ressources nécessaires pour le projet et expliquer les besoins aux gestionnaires pour diriger le projet. Les coûts du projet doivent pouvoir être justifiés et l'intérêt du projet exposé clairement aux gestionnaires.

3.1 Le succès ou l'échec du projet

Établissons d'abord un vocabulaire de référence. Un échec complet est un projet qui est annulé; où aucun livrable n'est accepté par le commanditaire ou produit par le fournisseur. Un échec moyen est un projet où toutes les fonctionnalités demandées ont été livrées, mais le projet dépasse son budget en coûts ou en temps. Un demi échec est un projet où des budgets en coûts et en temps ont été respectés mais on déploie et livre un système qui ne satisfait pas tous les besoins demandés. Pour obtenir un succès pendant un projet de migration logicielle les spécifications du logiciel doivent être cernées correctement. Notons que les organisations gardent souvent le secret des échecs de projets de migration informatique.

Les résultats du projet sont souvent une fonction de la nature des opérations du système patrimonial telle qu'examinée dans la section 1.2.1, ou dans la section 2.2.1. Les responsables de la migration doivent faire face à des difficultés différentes en fonction de la nature des opérations du système patrimonial. Ces difficultés doivent être connues pour pouvoir fixer des objectifs pertinents pour le succès ou l'échec du projet.

Pour reprendre l'analyse de O'Callaghan [O'CA 1998], il ne faut pas se dire

- Comment fait on pour maintenir le système patrimonial le plus longtemps possible mais plutôt
- Quel est le meilleur soutien à fournir aux nouveaux produits ou services vendus et quelle est la configuration optimum des technologies de l'information pour atteindre cet objectif ?

3.1.1 L'objectif technologique

Pour réussir la modernisation d'un système patrimonial, tous les auteurs s'accordent à expliquer qu'il faut fixer clairement des objectifs à atteindre. De plus, la technologie est le moyen et non pas le moteur du changement. Les processus d'affaires des membres de l'organisation sont corrects mais la technologie ne parvient pas à satisfaire tous les critères de l'organisation. Les efforts de maintenance corrective et évolutive satisfont des objectifs technologiques. La maintenance corrective englobe la correction d'erreurs détectées ou anticipées. La maintenance évolutive englobe les efforts pour s'adapter à de nouveaux besoins informatiques, par exemple le passage, en 2002 en France, à la monnaie unique : l'euro.

Lorsque l'objectif de la migration informatique est essentiellement technologique, alors il est beaucoup plus facile d'en mesurer le succès. C'est aussi le pire objectif à fixer pour le déploiement d'un ERP. L'objectif d'un ERP doit être de modifier les processus d'affaires.

3.1.2 Objectifs pendant le projet

L'objectif du projet est distinct des objectifs pendant le projet. L'objectif pendant la modernisation est de réussir une migration sans affecter les membres de l'organisation, les organisations partenaires ou les clients et usagers externes. Les objectifs pendant le projet concernent surtout des décisions tactiques qui consistent à jouer sur les moyens, à modifier les actions spécifiques pour s'adapter aux incidents de parcours et continuer à maintenir un niveau de fonctionnement pour les différents groupes d'utilisateurs et des applications. L'ensemble des tactiques s'inscrit dans la stratégie et sont examinés dans le détail dans le chapitre 4.

Pour un projet de migration, les objectifs de fonctionnement pendant et après la migration devraient être chiffrés. Souvent une période d'ajustement existe pendant laquelle la productivité des utilisateurs de l'organisation est moindre. Une estimation de la période d'ajustements et du niveau de qualité acceptable devrait être proposée et comparée aux résultats réels.

Des mesures du fonctionnement de l'application incluraient notamment :

- Le nombre de pannes (interruptions du ou des serveurs) en fonction du temps
- La durée moyenne des pannes
- Le nombre de transactions refusées (le serveur n'est pas en panne mais ne réalise pas la transaction demandée)
- Le nombre de transactions retardées (le serveur n'est pas en panne et effectue la transaction espérée mais pas dans des délais réguliers / dépassement du temps maximal)
- Le nombre d'arrêts planifiés en fonction du temps
- La durée moyenne des arrêts planifiés

Les arrêts planifiés correspondent aux réindexations ou réorganisations des bases de données, la mise à jour des applications, les opérations de maintenance préventive et autres opérations de perfectionnement.

3.2 Réingénierie des processus d'affaires

Rajouter la technologie sur de mauvais processus d'affaires est inefficace. Si elles sont mal utilisées les technologies de l'information permettent de prendre des mauvaises décisions plus vite. La migration ou modernisation informatique et la réingénierie des processus d'affaires sont liées. La réingénierie des processus d'affaires et l'informatique ont une relation de symbiose. Sans réingénierie des processus d'affaires, l'informatique seule, n'offre presque aucun avantage. Sans système informatique en réseau très peu de réingénierie n'est possible.

La réingénierie ne peut exister qu'au niveau global d'une organisation. En effet, parfois en réalisant des économies dans une unité de l'organisation on peut amener de grandes dépenses dans d'autres unités. Pour réaliser des économies qui profiteront à toute l'organisation il faut examiner les processus d'affaires à un niveau global.

La réingénierie des processus d'affaires ne peuvent exister que dans une organisation. La structure de l'organisation a un impact sur le fonctionnement de l'organisation. Un examen de la répartition des pouvoirs au sein des organisations permettra de comprendre les enjeux informatiques liés à la réingénierie des processus. Les enjeux informatiques essentiels sont la gestion des droits d'accès qui assurent le respect de la hiérarchie de l'organisation et des processus d'affaires.

3.2.1 Définitions

La définition fournie par Hammer et Champy [HAMM 2001], pour un processus est une « collection d'activités qui prennent un ou plusieurs types d'éléments en entrée et crée un élément en sortie qui a de la valeur pour le client ». Un processus est différent d'une tâche.

Même dans une grande organisation, on peut résumer les activités de l'organisation à moins d'une dizaine de processus clés. Ces processus clés peuvent ensuite être divisés en sous processus et ainsi de suite. Normalement, il faut quelques semaines pour identifier et rédiger une liste des processus. Dans leur ouvrage, les auteurs examinaient les processus de Texas Instruments. Les activités de cette grande entreprise étaient décrites au moyen de moins d'une dizaine de processus clés qui fournissaient tous de la valeur pour les clients de l'entreprise.

Pour reprendre la définition de Hammer et Champy, la réingénierie des processus d'affaires correspond au "ré examen fondamental, re-définition radicale des processus de l'entreprise pour avoir des améliorations substantielles en terme de coût, qualité, service et de temps". La réingénierie des processus d'affaires doit améliorer l'efficacité des processus dans une organisation.

Exemple de processus inefficace :

Des articles sont achetés dans un grand magasin par un client qui n'est pas satisfait de la marchandise et retourne le produit pour obtenir un remboursement. Ce retour affecte plusieurs départements du magasin.

- La réception (ou service à la clientèle) reçoit les biens et rembourse le client
- Des employés ramènent les articles dans l'entrepôt et les remettent dans le stock
- Les gestionnaires d'inventaire modifient les bases de données pour noter la transaction de retour de marchandise
- Les responsables des promotions (ou marketing) déterminent le prix auquel l'article était vendu pour le déduire des revenus. En effet, le prix du produit peut avoir changé depuis le moment de la vente.
- Le responsable des ventes réduit la commission du vendeur qui a vendu le produit selon le cas, et consigne le remboursement du client.

Ainsi, cette simple transaction peut accumuler plusieurs erreurs dans chaque département. Ces erreurs pourraient se répercuter sur les salaires des employés, la quantité et la localisation des produits en inventaire et les revenus de l'organisation.

Dans l'exemple ci-dessus, aucun département ne contrôle complètement la transaction de retour d'articles et aucun département ne se soucie suffisamment de la transaction pour en améliorer le processus. Toute proposition d'amélioration dans un seul département serait peine perdue car les erreurs des autres départements les rendraient inutiles.

La liste des problèmes au niveau des processus de l'organisation inclut des problèmes de performance, de disponibilité, de « Scalability », de fiabilité ou de Sécurité. Ces problèmes peuvent avoir plusieurs causes, et sont résumés dans le tableau 14.

Tableau 14 : Les causes des problèmes des processus d'affaires

Problème	Cause
Performance, Disponibilité	La fragmentation d'un processus entre plusieurs unités La redondance d'une tâche entre plusieurs unités
« Scalability » (Adaptation des capacités de la machine au débit demandé)	Le manque d'efficacité des tâches pour réaliser le processus et obtenir des résultats utiles.
Fiabilité, Sécurité	Le manque d'intégrité des résultats d'un processus

Pour [HAMM 2001], tous les problèmes des processus peuvent être réglés en réduisant la fragmentation des processus.

3.2.1.1 La fragmentation des processus d'affaires entre plusieurs groupes

Selon [HAMM 2001], le problème essentiel qui peut affecter les processus d'affaires d'une organisation est la fragmentation des processus. Quand on a un processus distribué entre plusieurs unités, bien souvent il n'y a jamais une seule personne responsable du processus au complet et du résultat de ce processus face au client. Beaucoup de personnes participent à sa mise en œuvre, mais ce n'est pas le travail d'une seule unité de l'organisation.

De nombreuses erreurs peuvent donc survenir lorsqu'un processus est morcelé entre les unités. Ces erreurs peuvent découler d'exceptions aux situations normales. Dans cette

situation, les personnes à l'extérieur de l'organisation ne sont pas nécessairement au courant des erreurs. Ainsi, une fois enclenchée, le processus est quasiment perdu dans l'organisation, jusqu'à ce que le processus soit terminé et qu'un produit final soit livré ou un résultat déterminé.

L'amélioration d'un processus, lorsqu'il est fragmenté entre plusieurs groupes, est souvent difficile car l'amélioration devrait remonter et ensuite redescendre la voie hiérarchique. Donc il y a beaucoup d'occasions pour une amélioration de tomber à l'eau, si elle ne provient pas de la haute direction. Selon [HAMM 2001], la réingénierie correspond à la réinvention des façons de faire des affaires. Cela ne correspond pas à la simple modification ou à l'amélioration des processus d'affaires. Lorsqu'on recherche des améliorations de l'ordre de 10% on peut employer la modification ou l'amélioration. La réingénierie est un changement plus radical qui devrait fournir des augmentations de productivité de l'ordre de 50 à 100%. Ainsi la réingénierie est révolutionnaire et non pas évolutionnaire.

Les organisations sont divisées en unités. Elles ne sont pas divisées selon leurs processus d'affaires. Dans un grand nombre d'organisations il existe un département des finances, mais il n'existe pas de processus des finances. Par contre : *Facturer le client et obtenir le paiement*, est un processus. Il ne faut pas voir les processus d'affaires comme une invention théorique, mais une réalité.

Dans leur ouvrage Hammer et Champy expliquent qu'il faut avoir des processus simples, même s'il faut accepter des tâches plus ardues. Cette vision s'oppose à celle de l'économiste anglais du 18^{ème} siècle, Adam Smith selon lequel il faut avoir des tâches simples même si cela implique des processus complexes pour lier toutes ces tâches ensembles. Smith voyait dans la spécialisation et la simplification des tâches un moyen de décupler la productivité. Contrairement à Adam Smith, Hammer et Champy considèrent qu'une organisation doit avoir des processus simples même si cela implique des tâches complexes et un personnel très qualifié.

Pour Hammer et Champy : la vérification, la réconciliation, les temps d'attente, la surveillance et le contrôle sont du travail improductif qui existe à cause des barrières ou

frontières dans une organisation. Ce travail est nécessaire pour compenser la fragmentation des processus et serait éliminé par une réingénierie efficace. Ceci permet ainsi aux gens de se concentrer sur le travail vraiment productif.

La réingénierie des processus d'affaires s'effectue sur des processus qui sont inefficaces. On peut choisir les processus à réformer selon les critères suivants :

- Choisir les processus qui ne fonctionnent pas correctement
- Choisir les processus qui ont le plus gros impact sur le client
- Choisir les processus qui peuvent être modifiés le plus facilement

L'essentiel n'est pas de comprendre les processus improductifs pour les réparer, mais de les détruire et les reconstruire. En effet, il y a une limite à la productivité des réparations. Cette valeur marginale est évaluée à 10 % par les auteurs. Il vaut mieux organiser le travail d'après les résultats des processus et non pas en fonction de tâches. La réingénierie des processus d'affaires se fait grâce à une plus grande déconcentration des pouvoirs dans une organisation. Pour plus de détails sur la déconcentration des pouvoirs consulter la section 3.2.2. Notons que pour pallier au problème de manque de responsabilité globale, certaines organisations ont donné le contrôle des processus à des comités. Cependant, si on est redevable devant un comité qui se réunit que trois fois par an, il y a peu d'opportunités d'interactions ou d'améliorations de ces processus.

3.2.1.2 Sept solutions possibles

La résolution de la fragmentation vise à accroître la performance des membres d'une organisation, sans que ces membres consomment nécessairement plus de ressources de l'organisation.

[HAMM 2001] identifient sept types de changement qui sont censés résoudre la fragmentation :

1. L'intégration de plusieurs tâches à travers l'organisation pour qu'elles soient réalisées par un seul groupe (ou personne).
2. La déconcentration des pouvoirs de décision auprès des employés. Une décision est prise rapidement sans avoir à remonter la hiérarchie de l'organisation.

3. Les étapes du processus suivent un ordre naturel. Des étapes séquentielles ne sont pas nécessairement suivies pour arriver au résultat.
4. Le travail est effectué à l'endroit le plus censé et raisonnable
5. On réduit le nombre de contrôles et de vérifications jalonnant le processus
6. Mettre au point plusieurs versions d'un processus :
Des processus sur mesure pour un groupe sont développés pour un ensemble de cas qui peuvent se présenter dans ce groupe.
Exemple : Unité 1 emploie un processus simplifié pour les cas simples, unité 2 emploie un processus avancé pour régler le même type de problème.
7. Réduire la quantité d'information à réconcilier.
Exemple : Soit un produit acheté et lorsqu'il est livré, le prix a déjà changé. Si le produit livré est censé lors de la réception, être associé à celui qui a été acheté, alors on peut se retrouver avec plusieurs documents différents identifiant la même commande avec plusieurs prix différents. Alors des questions de réconciliation du prix et du produit surgissent.

Ces solutions peuvent être mises en œuvre si on peut déléguer ou modifier les rôles, droits et privilèges des participants. Si ces solutions sont censées être soutenues par des outils informatiques alors la question des droits d'accès est essentielle. La réingénierie est liée à la possibilité de modifier rapidement et efficacement les droits, rôles ou privilèges d'accès.

3.2.1.3 Limites

Malgré l'apport de [HAMM 2001], certaines prises de position semblent trop radicales. [HAMM 2001] déclare notamment que « Reengineering must feel disruptive not comfortable » : c'est-à-dire que les changements doivent être inconfortables. Par ailleurs, l'affirmation des auteurs selon laquelle, dans toutes les organisations, toutes les tâches d'audit, de réconciliation ou de vérification constituent du travail improductif semble trop radicale et contredit l'approche de gestion de la qualité du CMM. Enfin, les auteurs avouent que certains chiffres présentés pour soutenir leurs arguments n'ont pas été recueillis d'une manière rigoureuse.

La réingénierie des processus d'affaires n'est pas une panacée. Pour [HAMM 2001] elle implique un travail rigoureux et des sacrifices. Les responsables de la réingénierie affrontent souvent de la résistance chez les employés. Cette résistance est normale. Certaines personnes peuvent ou vont perdre leur emploi à cause d'une réorganisation radicale du travail. Il ne faut pas essayer de réaliser la réingénierie sans mécontenter certains. C'est une erreur. Satisfaire tout le monde va soit dévaluer l'efficacité du

programme de réingénierie soit retarder sa mise en oeuvre dans le temps. Il est inexact de dire qu'un projet de réingénierie a échoué parce que les employés ont résisté au changement. Un projet échoue parce que le niveau de résistance n'a pas été correctement géré.

Enfin, les auteurs ne détaillent pas les difficultés qui peuvent surgir du fait qu'une organisation peut ne pas avoir la maturité nécessaire pour effectuer correctement des opérations requises et choisir correctement des processus optimaux. Une entreprise doit avoir la capacité de mesurer l'efficacité de ses processus et de sa gestion. Cela implique de la part de l'organisation, un haut niveau de maturité CMM. Selon le SEI (Software Engineering Institute) il est hasardeux d'entreprendre des projets qui requièrent un niveau CMM supérieur à celui que maîtrise l'organisation. Il existe cinq niveaux CMM. Généralement, passer d'un niveau à un autre dure un an et ce n'est qu'au niveau cinq qu'une organisation a la maturité nécessaire pour réussir une « amélioration » de ses processus. Une étude [JUNG 2003] du SEI vient confirmer cette thèse avec des preuves rigoureuses et chiffrées. Dans son étude [JUNG 2003] démontre un lien entre efficacité de la gestion de la maintenance logicielle et maturité CMM de l'organisation.

3.2.1.4 L'intégrité des processus

[HAMM 2001] insiste beaucoup sur les problèmes de la fragmentation des processus d'affaires, mais d'autres auteurs considèrent l'importance de l'intégrité des processus. Selon André Tchokogué [TCHO 2002], si les processus d'affaires d'une organisation sont réorganisés, alors il faut impérativement vérifier l'intégrité des processus avant de les mettre en vigueur. Cette étape de vérification de l'intégrité des processus d'affaires est explicitement prévue dans la méthode « Fasttrack 4 SAP » pour les ERP de SAP.

L'intégrité des processus vérifie :

- les risques posés par les nouveaux processus
- le contrôle des risques posés par les nouveaux processus
- les nouveaux rôles et responsabilités des usagers
- les nouveaux droits d'accès des usagers

[HAMM 2001] établissait dans son ouvrage la notion de propriétaire de processus et distinguait les personnes « responsable » et « accountable ».

- « Responsable » : Une personne responsable de la tâche. Un chef qui s'assure que le travail quotidien s'effectue chaque jour.
- « Accountable » : Une personne envers qui on est redevable devant qui il faut se justifier : c'est-à-dire un chef logique qui est propriétaire de processus au complet même si les opérations quotidiennes sont réalisées par d'autres à travers plusieurs unités.

C'est-à-dire qu'il peut exister une personne responsable des résultats d'un processus même si d'autres gèrent les tâches qui font parti de ce processus. Cependant Hammer et Champy se concentrent uniquement sur la productivité de l'entreprise face à sa clientèle. D'autres enjeux devraient influencer les processus d'affaires.

[TCHO 2002] spécifie que la vérification de l'intégrité des nouveaux processus se termine par l'acceptation formelle « Sign-off » et écrite par les responsables de processus, et des vérificateurs avant d'officialiser ces processus. Lorsqu'un logiciel professionnel commercial est acheté par une organisation, c'est l'organisation qui doit s'adapter au système et non l'inverse. Dans ce contexte une étape formelle de validation de test et d'acceptation des nouveaux processus par leurs propriétaires prend tout son sens. Cette étape d'acceptation prépare l'entreprise à l'arrivée de nouveaux systèmes informatiques.

Quelques leçons tirées de l'expérience de Pratt et Whitney [TCHO 2002] méritent d'être mentionnées.

Il faut gérer la quantité de réingénierie. Cela signifie qu'il faut maximiser la concordance avec les processus d'affaires du système informatique. Pour autant, il ne faut pas chercher une concordance parfaite et absolue avec les processus d'affaires du système informatique. Si une solution est jugée satisfaisante elle n'a pas besoin d'être optimale. En effet, si trop d'exceptions sont analysés, ces exceptions peuvent compromettre le projet. Il faut éviter de consacrer toutes ses ressources à des exceptions marginales.

L'acceptation des nouveaux processus par les propriétaires de processus devrait être une étape formelle et écrite. Les propriétaires devraient signer un document affirmant que les nouveaux processus sont intègres et qu'ils peuvent et seront suivis.

3.2.2 Analyser l'organigramme d'une organisation

Le fonctionnement de l'entreprise doit être compris par les responsables de la migration. La réingénierie des processus d'affaires essaie de faire des optimisations à travers l'organisation, donc de détruire des barrières entre les départements de l'organisation mais ces barrières peuvent être instituées par des règlements ou statuts de l'organisation.

L'une des distinctions importantes à établir est le lien hiérarchique entre les employés et les relations de l'organisation avec ses sous unités. Ce mémoire analyse dans le détail, la hiérarchie d'une organisation à l'aide du droit administratif [LEDR 2004]. Le droit administratif permet de comprendre l'impact de l'organigramme d'une organisation sur les droits d'accès, les contrôles informatiques ainsi que sur la manière de modifier les processus d'affaires de l'organisation.

Une ville est une personne morale unitaire, c'est-à-dire que dans une ville, sur le territoire de la municipalité, il n'existe qu'une entité qui possède sur le territoire municipal le nom de ville; parmi les personnes morales publiques, il n'y en a qu'une qui dispose des prérogatives que l'on reconnaît à la ville. L'organisation unitaire se caractérise par une triple unité : une seule organisation, un seul pouvoir souverain, un seul décideur. Les organisations fortement centralisées peuvent être tempérées par la déconcentration. Il s'agit pour nous de développer ici l'organisation centralisée.

Le concept d'organisation unitaire et centralisé s'oppose à celui d'organisation composé de sous unités décentralisées. La décentralisation correspond à une délégation formelle de l'autorité à des niveaux hiérarchiques inférieurs. Avec la décentralisation il existe des sous unités ou départements autonomes qui disposent de certaines compétences. Dans cette situation, la haute direction de l'organisation ne peut pas annuler une transaction réalisée par un représentant d'une unité décentralisée. Il ne peut effectuer que d'autres transactions qui pourraient en limiter les effets, sauf dans le cas de la tutelle.

3.2.2.1 Définition de l'organisation centralisée

Juridiquement, l'organisation centralisée correspond à une organisation au sein de laquelle n'existe qu'une seule personne morale. Celle-ci a la charge de l'ensemble des attributions de

gestion : il n'y a pas d'autres sous unités autonomes. D'un point de vue plus concret, la centralisation signifie que tous les membres de l'organisation sont des agents de l'organisation, insérés dans une hiérarchie unique dominée par les organes centraux de l'organisation. Tout le pouvoir est concentré au sommet de l'organisation.

Ce système est renforcé par le pouvoir hiérarchique qui s'exerce du supérieur hiérarchique vers le subordonné. Ce pouvoir comporte trois éléments : le pouvoir d'instruction (faculté dont dispose le supérieur de fixer à l'avance, pour le subordonné, la manière dont il devra agir), le pouvoir de réformation (consiste dans les faits, pour le supérieur, de modifier éventuellement les décisions prises par le subordonné), le pouvoir disciplinaire (pouvoir de noter les subordonnés et de les sanctionner en conséquence).

Sur un plan pratique, les organisations centralisées ont peu de chances de fonctionner sur un territoire étendu. Tout en restant dans la centralisation, on peut persister dans l'idée qu'il faut maintenir une centralisation très forte (concentration) ou l'atténuer par la déconcentration.

3.2.2.2 L'organisation centralisée concentrée

La centralisation des décisions de gestion vise une concentration géographique. C'est-à-dire que toutes les décisions sont prises par des agents de l'organisation, la haute direction. L'organisation seule décide de tout au nom de l'intérêt général. Il peut y avoir quelques représentants locaux, mais ces agents sont subordonnés et dépourvus de tout pouvoir de décision : ils transmettent les décisions. Ce système a un avantage : il renforce l'unité de l'organisation. Deux conditions principales sont nécessaires pour que la centralisation puisse se maintenir durablement : un nombre réduit d'affaires à traiter et d'autre part une zone géographique d'étendue limitée. Sans ces deux conditions, les gouvernants de l'organisation ne pourront plus matériellement prendre toutes les décisions à l'intérieur de l'organisation. La haute direction risque donc d'être surchargée. Ce système aboutit à l'apoplexie au centre et à la paralysie aux extrémités. Aujourd'hui, les organisations unitaires adoptent des mécanismes qui tempèrent la centralisation très forte : la déconcentration.

3.2.2.3 L'organisation centralisée déconcentrée

La déconcentration apparaît lorsqu'il y a un découpage du territoire en circonscriptions administratives, et qu'à l'intérieur de ces circonscriptions qui correspondent à une division du travail (et non pas à une division du pouvoir comme dans la décentralisation), il existe des représentants de l'organisation qui se voient accorder des compétences et des pouvoirs au nom de l'organisation. Par exemple, certains représentants d'une organisation peuvent être mandatés pour prendre des décisions pour leur organisation. La déconcentration devient nécessaire dans une organisation où l'interventionnisme s'accroît. Il n'est plus possible de faire monter tous les dossiers, et les autorités déconcentrées se voient investies de compétences croissantes.

Ces représentants n'auront pas besoin d'attendre le feu vert de la haute direction, car ils disposent d'une compétence discrétionnaire d'appréciation. L'avantage de ce découpage est une meilleure adaptation de la décision à la situation que l'on veut régler, car le responsable local peut avoir une connaissance plus concrète et plus précise que celle que peut avoir la haute direction. En d'autres termes « on peut gouverner de loin mais on n'administre correctement que de près. » Les responsables locaux reçoivent toujours des instructions de leur supérieur et peuvent être sanctionnés (car placés sous l'autorité hiérarchique) mais ils disposent d'une marge de manoeuvre intéressante.

3.2.2.4 Impact de l'organigramme sur les systèmes informatiques

Le pouvoir hiérarchique doit être soutenu par des outils informatiques de gestion des privilèges, des rôles et des droits d'accès pour les membres d'une organisation. La décentralisation des droits en informatique attribue aux membres d'une organisation des droits d'accès fixes. La déconcentration du pouvoir en informatique implique que des droits et responsabilités appartenant à un supérieur hiérarchique puissent être délégués à des subordonnés en fonction d'un découpage territorial (ou autre) et aussi que ces droits d'accès puissent être repris à tout moment. Le langage de spécification des droits d'accès Ponder [NICO 1995] reprend les préoccupations de déconcentration des droits en accordant des délégations de droits à des usagers. La hiérarchie des classes orientées objet dans LDAP (Lightweight Directory Access Protocol) contrôle les droits d'accès aux membres d'une organisation. Le protocole LDAP considère comme sujets de droits : les individus,

les groupes et les unités organisationnelles, et aussi les machines qui dans la hiérarchie LDAP sont une sous classe d'individus.

3.3 Motiver les changements

Lorsque des changements au système patrimonial peuvent devenir nécessaires pour des raisons technologiques mais aussi à cause de changements dans le domaine, il est important de pouvoir communiquer ces nouveaux besoins aux gestionnaires pour s'assurer de leur soutien pour les tâches à accomplir. Il existe un mythe, qui peut influencer les décisions des gestionnaires, selon lequel les systèmes patrimoniaux fournissent peu ou pas de valeur d'affaires à l'entreprise. Il faut expliquer au gestionnaire qu'un système patrimonial peut être tout à fait valide et exécuter des opérations tout à fait utiles.

Pour motiver les changements il faut pouvoir démontrer l'importance du changement informatique en rapport avec les processus d'affaires, et donc expliquer l'insuffisance des systèmes patrimoniaux. En pratique, les premières décisions que devra motiver le responsable d'un projet sont des coûts du projet. Ce mémoire reprend des modèles d'estimation des coûts de migration de systèmes informatiques. Enfin, pour justifier les coûts du projet il faut pouvoir démontrer un avantage d'affaires.

3.3.1 Convergence des processus d'affaires et des systèmes d'information

Le système d'information dans une entreprise est le lieu de convergence de :

- l'environnement de l'industrie;
- l'environnement informatique (innovations)
- du fonctionnement de l'entreprise
- du fonctionnement des systèmes patrimoniaux

Selon Sue Kelly [KELL 1999], le système patrimonial doit être changé si les règles d'affaires ne sont pas soutenues par le système informatique. Cet écart illustré par la figure 5 de [KELL 1999] compare le système d'information patrimonial au modèle d'affaires dans le temps. Avec le temps les règles d'affaires ont toutes tendance à changer.

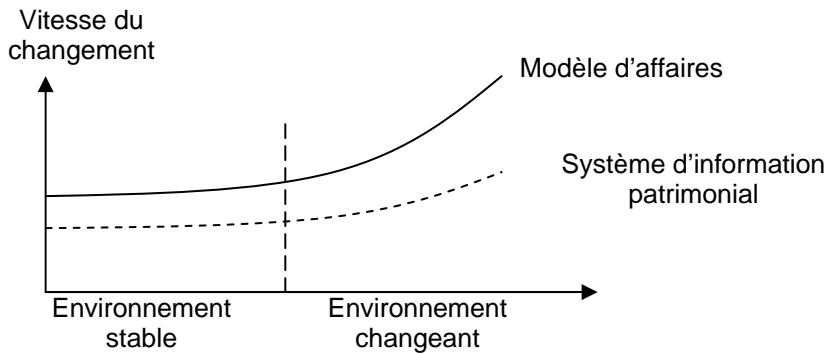


Figure 5 : Évolution du modèle d'affaires et du système d'information de [KELL 1999]

Pour faire de la réingénierie de processus d'affaires ou simplement pour analyser ce que fait l'entreprise, il faut pouvoir voir ce que fait toute l'entreprise. Les systèmes patrimoniaux peuvent donner des vues incohérentes ou irréconciliables des entités.

- Donc c'est un obstacle à l'intégration.
- C'est aussi un motif d'intégration car il faut réduire le dédoublement.

La décision de modifier un équipement devrait tenir compte de l'impact de l'application sur l'organisation, son adéquation aux besoins d'affaires et de son adéquation face à l'environnement technologique. Selon le cas, le système devrait être supprimé, remplacé, amélioré ou simplement entretenu.

3.3.2 Coûts des projets de migration

La migration des applications de gestion d'une organisation est peu fréquente. Le gestionnaire se retrouve avec peu de références ou expérience à sa disposition pour évaluer et comparer les coûts de développement à l'interne avec les coûts d'acquisition et de migration, car ces coûts sont difficilement quantifiables. Quelques recherches sur des modèles d'estimation des coûts de migration de systèmes informatiques semblent émerger. Les modèles très approximatifs proposés pour estimer les coûts de migration des ERP et l'estimation des coûts de migration à l'euro pour la France dans l'Europe communautaire seront abordés. Les deux modèles d'estimation des coûts seront ensuite comparés en examinant les coûts du projet Gires du gouvernement du Québec, Canada.

[MEYE 2001] considère dans son livre les coûts associés aux changements au système patrimonial. Ce dernier parle d'une vitesse optimale du changement. C'est-à-dire que le coût des changements d'un système diminue en fonction de l'âge de l'application jusqu'à ce que le système commence à devenir obsolète, et alors les coûts commencent à augmenter. La courbe de la figure 6 provenant de [MEYE 2001] illustre ce rapport. Un parallèle pourrait être dressé avec la le modèle de [BENN 2000] présenté dans la figure 1 qui illustre à quel point un système est maintenable.

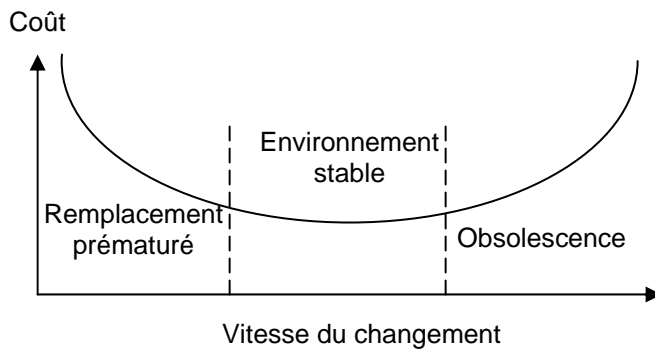


Figure 6 : Rapport coût / vitesse du changement d'un système logiciel de [MEYE 2001]

3.3.2.1 Coût de déploiement des ERP

L'article de Scheer [SCHE 2000], explique que les ERP requièrent beaucoup de ressources notamment en espace mémoire, les besoins de réseautique, et en coûts de formation. Ces coûts sont fixes et prévisibles. Par contre, la réingénierie et le déploiement du ERP peuvent provoquer de l'insatisfaction chez les usagers. L'analyse démontre que les coûts de la réingénierie des processus d'affaires et du déploiement de l'ERP coûtent plus cher que le prix de la licence logicielle de l'ERP. Dans le domaine des ERP, quatre entreprises se partagent le marché en position de monopole, soit : Oracle, PeopleSoft, SAP et Baan. Baan, Peoplesoft et SAP ont calculé que les clients des ERP dépensent entre trois et sept fois le prix de la licence logicielle en coûts de déploiement et de services associés. L'étude de Scheer [SCHE 2000], valide le chiffre selon lequel les clients ERP dépensent cinq fois le prix de la licence logicielle en efforts de déploiement. [ANDE 2003] mentionne que les coûts non informatiques liés au déploiement sont quatre plus grands que fois les coûts informatiques. En outre, [SCHE 2000] pense que puisque les prix des équipements

matériels et logiciels vont en décroissant, le rapport entre les coûts de déploiement et coûts de licence risque d'augmenter dans l'avenir.

3.3.2.2 Coût de la migration à l'Euro

Le coût de la réingénierie des processus d'affaires selon le calendrier du projet n'est pas facilement prévisible. Le livre de Marc Langlois [LANG 1998], fournit un modèle d'estimation des coûts de projets liés au passage à la monnaie unique européenne entre 1999 et 2002. Ces modifications liées à l'euro affectaient les organisations dans tous les domaines d'affaires et avaient un impact, à des degrés divers, sur tous les membres de ces organisations. À ce titre, le coût de migration à l'euro et les coûts de migration d'un ERP apparaissent comparables.

D'après [LANG 1998], le coût du projet euro était estimé à :

- 2 % des coûts d'exploitation pendant 3 ans pour les banques
- 1,5 % des primes d'assurance pour les compagnies d'assurance
- 0,5 % du chiffre d'affaires pour les entreprises de distribution.
- 1 % du chiffre d'affaires pour toute autre entreprise normale.

Plus précisément, les coûts du projet Euro pour les banques, estimés à 2 % des coûts d'exploitation pendant 3 ans, étaient répartis avec les pourcentages suivants :

Description	Coûts
Systèmes d'information	40 %
Formation	15 %
Marketing	15 %
Travaux de réorganisation	30 %

Il est intéressant de noter que le projet de changement des systèmes informatiques pour le passage à l'euro était particulier car il avait un impact sur l'intérieur et l'extérieur des entreprises. Le changement à l'euro affectait les transactions comptables et budgétaires et affectait la stratégie de marketing et de publicité des organisations. Les applications internes de l'entreprise, tels que des applications comptables ou applications de marketing, ne sont pas connues du grand public. Mais, pendant la migration à l'euro, les clients de ces organisations étaient affectés puisque les prix des articles changeaient. Le changement informatique pour s'adapter à la monnaie européenne affectait donc toutes les unités de l'organisation.

Il faut bien entendu nuancer le modèle et les chiffres présentés. Tout d'abord ils ne sont pas justifiés par des recherches rigoureuses, ni confirmées dans la pratique par l'auteur. En effet, le livre date de 1998, et le passage à l'euro a eu lieu en 2002.

3.3.2.3 Coût de la migration du projet Gires

Le projet Gires et d'autres illustrent à quel point il est nécessaire de faire des efforts pour estimer rigoureusement les coûts de déploiement d'une nouvelle technologie dans une organisation. Le projet Gires (Gestion Intégrée des Ressources) selon le rapport [LYRE 2003], lancé en 1998 par le gouvernement du Québec, visait à remplacer plus de mille systèmes informatiques qui étaient utilisés dans les différents ministères et organismes par un réseau central pour gérer les ressources humaines, matérielles et financières.

Le projet Gires devait remplacer deux systèmes patrimoniaux SAGIP et SYGBEC et implanter un logiciel professionnel à travers tous les ministères. Le budget du projet Gires a été initialement estimé à 84 millions de dollars canadiens. Cette première évaluation ressemble au modèle énoncé dans [SCHE 2000], soit 14 millions pour la licence, d'après [LYRE 2003] et 5 x 14 millions pour les coûts de déploiement, pour un total de 84 millions. Cependant le modèle de [SCHE 2000] ne tient pas compte de l'état du système patrimonial particulièrement archaïque du gouvernement du Québec. Le système patrimonial SAGIP et SYGBEC constituaient le point de départ du projet et influait sur les coûts du projet. Les systèmes SAGIP et SYGBEC étaient programmés en Cobol et assembleur, et les bases de données de ces systèmes reposaient sur des serveurs locaux VSAM et DB2. Les serveurs étaient des ordinateurs centraux IBM. SAGIP employait le SGBD M204. Cette technologie matérielle et logicielle était très ancienne et difficile à maintenir.

En 2003, le projet Gires fut interrompu avant d'être étendu à tous les ministères pour des raisons de dépassement important de budget. Dressons un parallèle entre l'estimation des coûts du projet euro du livre de Marc Langlois [LANG 1998], avec les coûts du projet Gires du gouvernement du Québec. L'estimation des coûts moyens du projet euro correspondait à 1% du chiffre d'affaires pour les entreprises. Selon le site officiel du ministère du trésor [TRES 2004], le budget annuel du gouvernement du Québec est de l'ordre de 50 milliards \$ pour le budget de dépenses 2003-2004. L'application de

l'estimation indique que le projet Gires aurait pu coûter 1 % de 50 milliards : soit 500 millions de dollars canadiens.

En 2003, les cinq sites pilotes du projet Gires avaient dépensé 104 millions, voir le tableau 15. En mars 2003, les coûts du projet Gires avaient déjà subi plusieurs réévaluations. À sa dernière évaluation, en septembre 2003, les coûts du projet totalisaient 345 millions de dollars, voir le tableau 16. Donc le chiffre de 500 millions calculé par le modèle d'estimation de [LANG 1998] ne semble pas radicalement incorrect. Bien entendu, il faut immédiatement nuancer nos conclusions car notre démarche n'est ni scientifique, ni rigoureuse car le modèle inspiré du livre de Marc Langlois n'est pas confirmé dans la pratique et demeure très superficiel.

Tableau 15 : Coûts du projet Gires en millions de dollars en mars 2003

Ministère	Coûts
Le ministère des Régions	2,5
La régie des rentes	7,6
La commission des normes du travail	8,7
Le ministère des ressources naturelles	15,2
Le ministère des transports	70
Total	104

Tableau 16 : Réévaluations des coûts totaux du projet Gires en millions de dollars [LYRE 2003]

Date d'estimation	Coûts
Juin 1999	84
Novembre 2000	275
Septembre 2003	345
Septembre 2003, tenant compte des coûts et frais d'intérêt capitalisés	398

En septembre 2003 un rapport indépendant rédigé à la demande du gouvernement du Québec [LYRE 2003] analysait l'état du projet et l'avancement des travaux en faisant des recommandations. Le rapport déterminait que :

- L'intégration devait être une vision et non un bien livrable associée à un produit unique
- Le déploiement du projet Gires a fait prendre du retard aux ministères et organismes publics et a hypothéqué les initiatives de ces ministères et organismes.

- Des licences acquises du ERP n'ont pu être utilisées à cause des retards accumulés. Les fonctionnalités du ERP continuaient à évoluer alors qu'elles n'étaient pas encore déployées.
- L'implantation de la technologie était devenue une fin et non plus un moyen

3.3.3 Le retour sur investissement : une science inexacte

Pour choisir entre plusieurs solutions possibles, il faut pouvoir les comparer, c'est-à-dire associer des qualités communes ou distinctes aux alternatives. Cependant, certaines alternatives sont moins quantifiables que d'autres. Pour certains, la valeur d'une application informatique patrimoniale ne peut être représentée que d'une manière très approximative, voire inexacte.

3.3.3.1 La valeur ajoutée

La valeur ajoutée par un changement requiert des calculs objectifs, mais souvent ces calculs se basent sur des hypothèses. Les responsables de la modernisation doivent définir la valeur qu'apportera le changement pour le motiver auprès du gestionnaire. Le gestionnaire doit rechercher un gain d'affaires pas un gain technologique, même si des changements d'affaires impliquent de grands changements technologiques. La valeur ajoutée peut être considérée du point de vue de la valeur économisée par une solution, de l'accroissement de la productivité, ou de l'augmentation des revenus.

[HAMM 2001] mesure la valeur ajoutée à l'entreprise par la réingénierie des processus d'affaires. Pour ces auteurs, tout processus et son amélioration ne doivent pas être mesurés d'après les tâches réalisées par l'utilisateur d'un système, ils ne peuvent être mesurés que par les bénéfices apportés au client. Une analyse des tâches d'un usager est très spécifique et ne donne qu'une information fragmentaire et qui ne peut être comprise que par l'organisation, pas par le client. Par exemple, réaliser une soudure est une tâche qui en soi n'a de valeur que pour l'organisation, alors que produire une voiture suit un processus qui crée de la valeur pour le client de l'organisation.

La vision de [HAMM 2001] se concentre exclusivement sur les clients de l'organisation. On pourrait la nuancer en indiquant qu'une migration logicielle qui améliorerait le fonctionnement des processus d'une organisation vis-à-vis d'une entité externe apporte de la valeur ajoutée : que ce soit un client, un fournisseur ou le gouvernement.

3.3.3.2 Le paradoxe de la productivité

[ANDE 2003] appelle l'impossibilité d'observer une plus grande productivité (plus d'articles produits ou services rendus à partir des ressources employées) avec des investissements informatiques, un paradoxe de la productivité. [POST 2000] s'est efforcé de vérifier et de valider le paradoxe de la productivité à l'aide de démarches empiriques d'analyse de données. Sur les huit hypothèses étudiées, une seule fut validée. L'hypothèse est que lorsqu'on adopte un ERP, on se retrouve avec moins d'employés, cette diminution entraîne une baisse des coûts. Pourtant les résultats démontrent aussi que le rapport entre coûts dépensés par employé par rapport au revenu augmente. L'analyse se basa sur l'étude de 50 entreprises dont certaines n'ont rendu disponibles que des données incomplètes.

3.4 Conclusion

La migration informatique ne se limite pas à des enjeux techniques, les enjeux les plus importants concernent les processus d'affaires. La migration informatique doit résulter en une convergence des processus d'affaires et des systèmes d'information. Les coûts des projets de migration dépassent de loin les coûts informatiques du projet, parce qu'ils modifient la manière dont les membres de l'organisation travaillent et interagissent. Une méthode de preuve d'un retour sur investissement dans ce domaine est encore une question en suspens.

Chapitre 4 Gestion du projet de migration

Pour gérer le projet de migration il faut choisir attentivement le système cible. Dans un premier temps, notre étude porte sur le choix de solutions vendues librement sur le marché; les produits COTS (Commercial Off The Shelf). Le choix du système cible dépend du domaine et de ses standards, et doit consacrer une relation de collaboration entre le fournisseur et l'acquéreur. La seconde section reprend la dichotomie de [HOLL 1999-a] qui distingue dans les projets de migration les aspects stratégiques et les aspects tactiques. Les aspects stratégiques : notamment le soutien de la direction et la stratégie ERP sont essentiels au succès du projet de migration alors que les aspects tactiques : notamment la gestion des plaintes, l'adoption, la diffusion de l'innovation et la formation de la main d'œuvre ne peuvent que contribuer au succès en s'inscrivant dans la stratégie de migration.

4.1 Établir un système ouvert

Une solution technologique fournit un avantage lorsqu'elle peut être remplacée aisément par d'autres produits plus performants, plus économiques et disponibles publiquement sur le marché. Des produits logiciels sont considérés comme interchangeables lorsqu'ils respectent les mêmes standards du domaine. Le terme « systèmes ouverts » défini dans les sections subséquentes inclut les produits commerciaux (COTS : commercial Off The Shelf), et les logiciels libres qui respectent les standards du domaine. Puisque les systèmes ouverts dépendent des standards, une section analysera les avantages et les inconvénients des standards, publics et privés. Un examen succinct exposera les inconvénients de standards technologiques uniques qui autrefois étaient fréquents au niveau des gouvernements.

4.1.1 Les systèmes ouverts et fermés

Le livre de [MEYE 2001] présente une analyse des enjeux des produits informatiques commerciaux publics. L'auteur distingue les systèmes ouverts et les systèmes fermés.

Il définit les systèmes ouverts comme étant une collection d'éléments logiciels et matériels et d'utilisateurs qui ont les propriétés suivantes :

- Les éléments logiciels et matériels interagissent dans le but de satisfaire des besoins déclarés
- Les éléments logiciels et matériels ont des interfaces qui sont complètement définies, disponibles publiquement et maintenues par le consensus d'un groupe.
- Dans ce système les composants logiciels sont conformes aux spécifications de l'interface.

Les interfaces des systèmes ouverts sont complètement définies lorsque les données échangées sont connues dans le détail (API : Application Public Interface). Les interfaces des systèmes ouverts sont disponibles publiquement lorsqu'il n'existe pas de restrictions sur l'usage de l'interface par le public. Les interfaces des systèmes ouverts sont maintenues par le consensus d'un groupe lorsque l'interface n'est pas décidée sans consultation. Si le consensus du groupe est large alors il se développera plusieurs produits logiciels qui respecteront le consensus. Les systèmes ouverts correspondent à des produits commerciaux COTS, ainsi que d'autres éléments logiciels notamment les logiciels libres.

Un produit commercial ou COTS (Commercial Off The Shelf) est un produit qui est

- Vendu, loué ou sous licence auprès du grand public
- Offert par un vendeur essayant d'en profiter
- Soutenu et mis à jour par le vendeur qui garde sur le produit des droits d'auteur
- Disponible en plusieurs exemplaires
- Employé sans modifications internes par un consommateur

Les liens essentiels entre produits COTS et systèmes ouverts, sont les standards publics. Les produits qui respectent des standards communs publics sont en principe équivalents. Lorsque des vendeurs offrent des produits interchangeables un marché apparaît et évolue pour ce produit logiciel. Dans un marché ouvert, les produits COTS peuvent être directement comparés et sont en théorie équivalents parce qu'ils respectent des standards communs.

La concurrence entre produits commerciaux amène des avantages. Les avantages des systèmes ouverts et COTS sont les suivants :

- Moindres coûts
- Moins de dépendances sur des solutions propriétaires
- Délai de développement réduit
- Produit mieux testé

- Portabilité accrue
- Interopérabilité accrue
- Insertion de technologie plus stable

On choisit un produit COTS en fonction :

- du segment du marché auquel il appartient
- de la technologie qu'il met en œuvre
- des standards qu'il respecte
- de la performance du produit
- de la disponibilité du produit et du fournisseur
- de la maturité du produit et du fournisseur

Les inconvénients du marché sont que des changements peuvent avoir lieu continuellement.

Ainsi, tous les 2 ou 3 ans il peut y avoir des changements substantiels comme :

- la disparition d'un produit
- l'arrivée d'un nouveau produit
- la disparition d'un fournisseur ou sa ruine

Pour ne pas simplement réagir passivement à ces événements, [MEYE 2001] recommande d'anticiper ce genre d'évènements.

4.1.2 Les standards et spécifications

Définition d'un standard selon [MEYE 2001] : « Un document disponible au public qui définit les spécifications pour les interfaces, les services, les processus, les protocoles ou les formats de données et qui est établi et maintenu par le consensus d'un groupe ». Pour [MEYE 2001], l'examen du produit devrait toujours être précédé d'un examen des standards auxquels il se conforme.

Cette définition s'oppose aux standards qui appartiennent à une entreprise en situation de monopole. On peut généralement retrouver le terme « standard », mais il ne s'agit en fait que de pratiques qui sont adoptées ou révoquées discrétionnairement à tout moment par l'entreprise en situation de monopole. [MEYE 2001] pose la définition d'une spécification : « Un document qui spécifie d'une manière complète précise et vérifiable les besoins, la conception, le comportement ou caractéristiques d'un système ou composants d'un système ».

4.1.2.1 Les standards uniques

Les entreprises privées peuvent adopter des standards technologiques uniques mais n'y trouvent que rarement un avantage concurrentiel. Par contre, il convient d'examiner les standards uniques pour le cas des gouvernements. En effet, dans les années 60, les gouvernements étaient les principaux acheteurs de technologie informatique en Amérique du nord. À cette époque, les gouvernements ont beaucoup développé des applications pour leurs propres besoins et ont inventé leurs propres standards technologiques uniques alors que ce n'était pas indispensable. Les gouvernements ont parfois justifié leur standards technologiques distincts en évoquant leur mission et objectifs différents des entreprises privées. Pourtant ce sont surtout dans les contextes militaires, qu'il est impératif d'employer une technologie à laquelle aucun adversaire n'a d'accès normal, mais dans des contextes civils cette précaution est superflue.

Aujourd'hui, la position de monopole, dans le marché des technologies de l'information, des gouvernements en Amérique du nord a considérablement diminué. Les standards technologiques uniques qui existent aux divers niveaux du gouvernement ont rendu le retour aux forces du marché difficile. Aux USA, dans les années 90, des directives gouvernementales de la FAR (Federal Acquisition Regulation), ont émis des règles pour contraindre le gouvernement américain à s'adapter au marché et non l'inverse.

D'après [MEYE 2001], la persistance au gouvernement d'une culture de développement au lieu d'une culture d'acquisition, était perceptible au niveau des documents de spécifications informatiques qui précisaient non seulement les besoins mais aussi comment les satisfaire. Le passage d'une culture de développement à une culture d'acquisition implique une perte de contrôle mais procure un plus grand éventail de choix.

4.1.2.2 Avantages des standards privés et inconvénients des standards publics

Les standards proposés ou imposés par des entreprises privées, peuvent présenter des ressemblances avec des standards publics. Cependant, ils sont modifiables par l'entreprise qui les a émis en tout temps, ne sont pas assujettis aux mêmes garanties de publicité, de consultation et de révision. Les entreprises qui développent ces standards peuvent ne laisser

qu'à leurs clients, parfois moyennant rétribution, le droit de connaître les standards utilisés ou de proposer des changements.

Malgré tout, certains standards publics sont peu connus dans des domaines hautement spécialisés (par exemple le standard MARC²). [MEYE 2001] souligne que l'inconvénient des standards publics comme ceux de l'IEEE est qu'ils sont très longs à développer, soumettre et adopter. Ce qui en réduit l'intérêt pratique. Les standards officiels ont tendance à confirmer les standards privés de l'entreprise privée qui réussit à dominer le marché. Les standards des entreprises peuvent être assez difficiles à distinguer des standards officiels. Enfin, des standards publics peuvent être développés qui vont à l'encontre des intérêts des usagers ou du grand public (par exemple le standard UCITA³). [MEYE 2001] conclut que la stabilité des standards publics ou privés détermine généralement la stabilité des produits qui les mettent en oeuvre.

4.2 Gestion des facteurs tactiques

La distinction de [HOLL 1999-a], entre aspects tactiques et stratégiques des projets de migration, présente des avantages. Encore faut-il s'accorder sur les définitions des termes « stratégique » et « tactique ». Pour faire suite aux standards publics, à la section 4.1, nous examinons les relations entre le fournisseur et l'acquéreur, que [HOLL 1999-a] inclut dans les facteurs tactiques. L'établissement d'un système ouvert, tel que défini à la section 4.1, et le choix du système cible doivent consacrer une relation de collaboration entre l'acquéreur et le fournisseur. Lors de la migration les tests constituent le moyen d'échange privilégié entre les responsables de la migration de l'organisation et les fournisseurs. L'acquéreur et le fournisseur doivent s'assurer d'avoir un robuste processus de gestion des plaintes des usagers. Enfin, un facteur tactique important est l'adoption et la diffusion de l'innovation par les usagers. Traditionnellement, la théorie de la diffusion d'une innovation, est basée sur des décisions individuelles d'adoption volontaire. Cependant, la culture d'une organisation ainsi que la manière dont l'innovation est imposée par la haute direction

² MARC : MACHine-Readable Cataloguing – Le standard informatique pour les bibliothèques est peu connu des informaticiens

³ UCITA : Uniform Computer Information Transactions Act – Ce standard propose d'ajouter des mouchards informatiques pour enlever des logiciels sur des postes clients, diminuer les garanties des acquéreurs, et rendre l'ingénierie inverse illégale pour les produits COTS.

peuvent influencer l'adoption et la diffusion de l'innovation. La formation de la main d'œuvre sera examinée par le résumé d'une étude de cas d'adoption d'un ERP dans une entreprise. Enfin, la négociation des changements des processus d'affaires de SAP sera examinée à l'aide d'une étude de cas.

4.2.1 Stratégie et tactique

Les recommandations de [HOLL 1999-a] proviennent d'une revue de la littérature et de l'examen d'organisations spécifiques au cours d'une étude. [HOLL 1999-a] distingue les facteurs de succès en facteurs stratégiques et tactiques.

Stratégie : ensemble des choix d'objectifs et de moyens qui orientent à moyen et à long termes les activités d'une organisation. Alors que la stratégie embrasse l'ensemble du plan d'action, la tactique regarde son application concrète dans les cas particuliers.

Tactique : les décisions tactiques consistent à jouer sur les moyens, à modifier les actions spécifiques pour s'adapter aux incidents de parcours et continuer à obtenir les avantages prévus sur le terrain. L'ensemble des tactiques s'inscrit dans la stratégie.

Tableau 17 : Facteurs de succès de projets de migration de [HOLL 1999-a]

Stratégiques	Tactique
Systèmes patrimoniaux Vision d'entreprise Stratégie ERP Soutien de la haute direction Plan et échéancier du projet	Consultation du Client Gestion des employés Acceptation du Client Gestion du changement des processus d'affaires et configuration logicielle Contrôle et rétroaction Communication Correction des problèmes

Dans le tableau 17, les facteurs de succès essentiels incluent les systèmes patrimoniaux, le soutien de la direction un plan et un échéancier du projet. Les systèmes patrimoniaux sont le point de départ de la migration. Par contre la correction de problèmes liés au déploiement, la relation entre le client et les autres intervenants sont plutôt considérées comme un enjeu tactique que stratégique.

[HOLL 1999-a] identifie le soutien de la direction et la stratégie ERP comme des facteurs stratégiques de succès. La stratégie ERP inclut les choix de déploiement du ERP. C'est-à-dire le choix du déploiement de tous les modules en même temps, ou graduellement avec un squelette de l'application permettant ensuite d'ajouter les fonctionnalités. La stratégie ERP établit la quantité de développement sur mesure.

4.2.2 Relation entre le fournisseur et l'acquéreur

Il peut exister plusieurs intervenants dans le projet de migration, notamment un fournisseur, un acquéreur, un intégrateur, et d'autres encore. Nous concentrons nos observations sur la relation entre le fournisseur et l'acquéreur, étant entendu que l'acquéreur correspond à l'organisation qui adopte le produit et le fournisseur étant toute organisation ou consultant qui aide l'acquéreur à déployer le produit. Nous examinons aussi brièvement l'enjeu des contestations entre divers fournisseurs.

4.2.2.1 Analyse du produit et du fournisseur

Pour [MEYE 2001] le fournisseur d'un produit peut être aussi important que le produit lui-même. Il faut donc être capable d'apprécier le degré de maturité du produit par rapport à la maturité du fournisseur. Les critères pour évaluer le produit doivent se concentrer sur les besoins essentiels que le produit doit satisfaire, et ne pas se buter sur des besoins secondaires. La maturité du fournisseur peut être mesurée par le modèle CMM. L'analyse du produit, du fournisseur et la délimitation du projet peuvent monopoliser beaucoup de temps. Dans l'article de [HOLL 1999-b], l'analyse du produit, du fournisseur et du projet par une grande entreprise de textile Européen « Threads », ayant des succursales à travers toute l'Europe, dura 4 ans : de 1993 à 1997. La négociation de l'étendue et de la portée du projet, le choix de l'intégrateur et les modules à déployer ne s'acheva qu'en novembre 1997.

L'analyse du produit et du fournisseur doit aborder les questions contractuelles. La disparition du fournisseur pour cause de faillite ou l'abandon d'une ligne de produit par le fournisseur, en cessant de le développer et l'entretenir, a un grand impact sur l'acquéreur.

L'acquéreur devrait avoir une clause pour obtenir le code source et tous les droits d'accès sur ce produit, pour pouvoir continuer à l'utiliser et l'entretenir lui-même, si nécessaire.

D'après [MEYE 2001], le vendeur d'un produit COTS, et l'acquéreur sont des partenaires. Leur partenariat ne prend pas fin avec la livraison du produit COTS.

Les produits COTS font souvent l'objet de tests rigoureux avant d'être mis en vente. Ils requièrent donc moins de tests lorsqu'ils sont achetés dans le court terme; mais plus de tests dans le long terme. En effet, en passant d'une version du produit à la suivante, et pour chaque mise à jour du produit, les tests de régression doivent être très nombreux pour s'assurer que des fonctionnalités n'ont pas disparu d'une version à l'autre. La conception du produit COTS n'étant pas sous le contrôle de l'acquéreur, ce dernier ne sait pas ce qui a été changé.

Les difficultés que peuvent poser le soutien technique d'un produit COTS :

- L'acquéreur perd le contrôle du produit, c'est le fournisseur qui décide librement quand et comment corriger les bugs.
- C'est le fournisseur qui choisit de mettre au point une nouvelle version. Si ce dernier ne soutient plus l'ancienne version il faut changer de version lorsque le fournisseur le décide même si on n'est pas prêt ou si on n'en a pas besoin.
- Les mises à jour continues des produits des différents fournisseurs peuvent donner lieu à des problèmes de coordination entre fournisseurs
- Les besoins en formation des employés peuvent varier considérablement à la hausse s'il s'agit d'un nouveau système ou à la baisse si le système est simple et d'apparence familière aux usagers.

Dans le long terme [MEYE 2001] identifie l'entretien d'un appareil comme un risque potentiel pour l'entreprise qui acquiert un produit. Si le fournisseur arrête d'entretenir le produit, l'acquéreur doit développer sa propre expertise dans le domaine. Si le fournisseur se met continuellement à rendre disponible de nouvelles versions de son produit avant qu'ils n'aient atteint leur age limite, l'acquéreur va subir ces coûts et évolutions plutôt que de les décider. Dans un tel environnement d'instabilité, il est difficile de fixer des plans de migration.

4.2.2.2 Relation de collaboration avec le fournisseur

Il semble évident qu'une relation de collaboration entre le fournisseur et l'acquéreur. Pourtant, dans certains projets des relations de confrontation ont été choisies dès le début. Dans le projet Gires [LYRE 2003] l'intégrateur, qui a été engagé, était directement imputable du respect des échéanciers, des coûts, et des niveaux de services par l'application de pénalités contractuelles. Une relation de confrontation n'entraîne que des litiges juridiques, comme en témoigne le projet Gires.

4.2.2.3 La vérification et la validation pendant la migration

Les tests doivent vérifier et valider le fonctionnement de la nouvelle application au cours de la migration, même si normalement les produits COTS sont correctement testés avant d'être mis en vente. En effet, pendant la migration, l'environnement patrimonial subit de nombreuses transformations. Il faut donc vérifier que le produit, ainsi que les outils d'administration fournis au soutien technique, fonctionnent correctement pendant ces transformations. Un test de grande envergure est un projet pilote.

4.2.2.4 Gestion des plaintes de l'utilisateur et de l'organisation

La gestion des problèmes du nouveau système informatique doit être assurée avec rigueur et d'une manière transparente pour les usagers et gestionnaires. Lors de la migration d'un système à un autre, la distinction entre les tâches de soutien technique, de maintenance et de développement peut devenir très difficile à faire. Les ERP sont souvent des applications infiniment configurables et pour lesquelles des mauvaises configurations sont possibles. Dans cette situation il peut être artificiel de distinguer une erreur dans une application d'une mauvaise configuration.

Le standard [IEEE/EIA 12207.0-1996] section 6.8 précise que lors du processus de déclaration et de résolution des problèmes, l'utilisateur n'a pas à connaître ou déterminer la nature du problème de fonctionnement du nouveau système pour soulever son problème et obtenir une résolution. Le projet doit donc fournir des outils et processus pour soulever, attribuer, examiner, corriger et fermer des dossiers d'erreurs. De nombreux processus de gestion de plaintes aussi appelés SPR (Software Problem Report) existent. Le standard

[IEEE 1044.1-1995] précise les catégories minimales à distinguer pour respecter le standard de gestion des plaintes logicielles lors de la vérification et la validation logicielle.

4.2.2.5 Gestion des plaintes par le fournisseur ou intégrateur du produit

Les standards de documentation et d'identification des erreurs logicielles sont utiles du côté de l'organisation qui adopte un produit, mais sont essentielles si le fournisseur ou intégrateur du produit veut assurer un service de gestion des plaintes à son client.

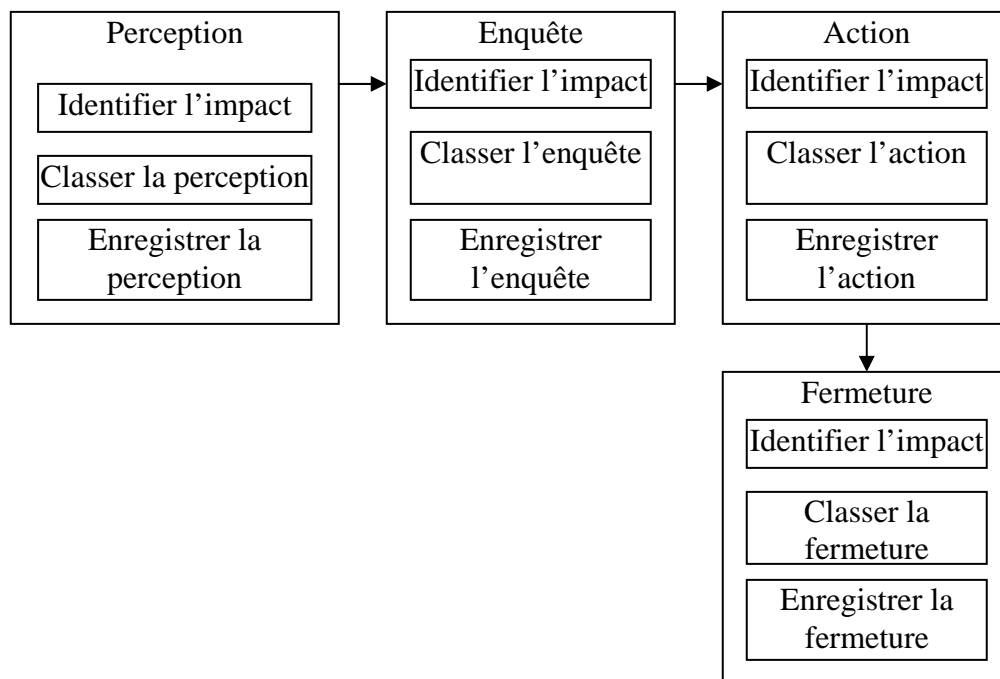


Figure 7 : Processus générique de gestion de plaintes du IEEE 1044.1-1995

La figure 7 présente le processus de gestion des plaintes du standard [IEEE 1044.1-1995]. Une priorité de correction devrait être associée à chaque erreur, avant même d'être corrigée. Lorsque des erreurs valides sont soulevées elles devraient être corrigées, identifiées et inventoriées par le fournisseur ou l'intégrateur d'une manière qui soit transparente pour l'utilisateur.

Malgré tout, [MEYE 2001] précise que c'est le vendeur qui détermine l'opportunité de corriger un bug soulevé par un client. Si une migration fait appel à plusieurs fournisseurs, ils peuvent se renvoyer les uns les autres la charge de corriger le problème. Le manque de

support externe peut allonger la durée d'un projet de migration. Ainsi, un projet de migration devrait établir clairement les responsabilités des divers participants.

4.2.3 Adoption et diffusion de l'innovation

L'adoption et la diffusion d'une innovation au sein d'une organisation repose sur le comportement de ses membres. Ce comportement peut être influencé par des attributs individuels et collectifs. Dans un premier temps, un attribut collectif : la culture de l'organisation, sera examinée. Ensuite les facteurs individuels qui influencent la décision d'adopter une application dans une organisation seront présentés. Enfin, la recherche de [GALL 2001] a démontré que l'adoption et la diffusion collective d'une innovation dans une organisation est influencée par la structure de l'organisation. Cette analyse et les limites qu'elle comporte seront résumées.

4.2.3.1 Comprendre la culture de l'organisation avec le système patrimonial

La culture d'une entreprise peut influencer l'utilisation d'un logiciel. Les employés sont parfois capables de fournir pendant des années des données correspondant à leurs besoins et qui ne respectent pas la conception des développeurs.

[ULRI 2002], dans son livre, donne l'exemple d'une application qui était utilisée depuis plusieurs années dans une organisation. Suite à un examen des données des analystes informatiques s'aperçurent qu'un champ "SEX_CODE" d'une table qui n'était censé contenir que des "H" ou des "F" (Homme / Femme), contenait à de nombreuses reprises la valeur "*". Les programmeurs, qui examinaient les données, ne comprenaient pas le contenu, fréquent, de ce champ. Interrogés, les employés expliquèrent qu'ils avaient décidé d'écrire "*" pour indiquer que l'employé était un stagiaire.

Pendant des années, l'informatique a été une discipline qui a subi beaucoup de changements et qui a beaucoup affecté le travail des employés. Trop de changements organisationnels peuvent démoraliser le personnel et diminuer leur intérêt pour des changements. Dans ces situations les membres de l'organisation vont ignorer les directives ou trouver des manières informelles de les contourner. Selon Dave Randall [RAND 1999], ces comportements sont

appelés des « gambits of compliance » pour éviter de déclencher des opérations non souhaitées.

4.2.3.2 L'adoption individuelle par les usagers

Certains auteurs distinguent des caractéristiques qui influencent l'adoption d'une innovation technologiques sous le nom de PCI : Caractéristiques Perçues d'une Innovation. Ces caractéristiques sont regroupées en catégories par [CHIA 2001] dans le tableau 18.

Tableau 18 : Facteurs influençant la perception des usagers, [CHIA 2001]

Caractéristiques	Description
Normes subjectives « Subjective Norms »	Ces normes correspondent aux perceptions de pression sociale que ressent l'utilisateur pour adopter une solution. Ces normes sont reconnues depuis les années 1990.
Compétence de l'utilisateur « User competence »	Les compétences informatiques de l'utilisateur avec de nombreuses applications différentes lui permettent de comprendre comment fonctionne un produit, et de l'adopter plus rapidement.
Facteurs organisationnels « Organizational factors »	Les facteurs organisationnels examinent l'environnement où la technologie et la tâche sont harmonisées. Ces facteurs examinent notamment comment les relations de pouvoir et la politique de l'organisation influencent l'adoption
Étape d'adoption « Stage of Adoption »	Les utilisateurs peuvent adopter une innovation à différents moments dans le processus d'implantation. Le moment de l'adoption peut caractériser leur relation avec l'innovation. Cette théorie date de 1998
Processus de mise en œuvre « Implementation processes »	Les processus de mise en œuvre ont pour but de préparer et de former les utilisateurs aux difficultés qui vont survenir pendant les premières étapes du déploiement. Ces processus incluent la participation de l'utilisateur, la formation, la gestion des attentes des utilisateurs, et développer un climat de confiance

La caractéristique de l'étape d'adoption est une théorie plus récente (1998). [CHIA 2001] distingue les utilisateurs selon l'étape à laquelle ils adoptent ou abandonnent un produit. Les utilisateurs qui adoptent rapidement une application ne l'abandonnent que pour adopter une autre technologie, alors que ceux qui adoptent une application plus lentement l'abandonnent et n'emploient aucune solution de rechange et sont insatisfaits. Selon [CHIA

2001] certaines études démontrèrent une corrélation entre l'adoption tardive d'une application informatique par l'utilisateur et l'abandon de celle-ci.

Les facteurs organisationnels contribuent à ce qu'une solution soit adoptée ou rejetée : notamment la pression sociale des autres employés et les relations de pouvoir entre les usagers et les services de soutien technique. La solution au problème n'est pas dans l'application, mais est formée chez l'utilisateur par la prise de conscience de la technologie dans un contexte social. Si les usagers voient qu'une application peut leur être utile, et accroître leur productivité (ce qu'on décrit comme un avantage relatif), ils vont vouloir être productifs et l'employer.

Les perceptions des usagers déterminantes pour l'adoption de l'application sont l'avantage relatif perçu de l'application et les caractéristiques perçues de l'innovation.

La vision des avantages perçus de l'innovation varie dans le temps. Quand un usager maîtrise correctement une technologie, les facteurs déterminants qui contribuent à ce que l'utilisateur continue à l'employer sont : la compatibilité et l'avantage relatif.

Dans l'article de [CHIA 2001], l'auteur examinait les facteurs individuels d'adoption d'une application de planification médicale par une employée dans un hôpital. Le déploiement de l'application se solda par un échec. Il découle de l'étude que la perception des avantages perçus de l'innovation est proportionnelle à la proximité entre l'application et la mission principale de l'organisation. Les avantages de l'application ne furent pas aussi bien perçus que s'ils avaient un lien direct avec l'application.

4.2.3.3 L'adoption collective par les usagers

La théorie de la diffusion d'une innovation, citée par [GALL 2001], est basée sur des décisions individuelles d'adoption volontaire, donc cette théorie comporte des limites lorsqu'on considère la diffusion de l'innovation au niveau du déploiement organisationnel après une décision imposée par la haute direction de l'organisation aux membres. La recherche en innovation a négligé d'examiner ce domaine, et demeure très exploratoire. L'adoption individuelle et organisationnelle d'une innovation sont différentes : l'une peut

connaître des succès alors que l'autre fait l'objet d'un échec. La recherche de [GALL 2001] s'efforce d'explorer ce domaine.

L'adoption collective d'une technologie dans une organisation passe par 2 décisions :

- La première décision correspond au processus d'adoption primaire. La direction décide d'objectifs de changement et trouve la technologie pour le faire. Un mandat est émis par la haute direction d'adopter la technologie. Ce mandat doit être ferme et soutenu par la haute direction et toute la hiérarchie.
- La deuxième décision correspond au processus d'adoption secondaire. Les employés décident d'adopter le produit.

Il est commun de rencontrer des situations où la direction adopte une innovation mais que les employés résistent à ce changement. [HAMM 2001] expliquait dans son livre que toute résistance est normale et qu'elle doit être gérée.

Des études citées dans [GALL 2001] ont examiné l'adoption d'innovations et ont déterminé que l'adoption est liée aux attributs de l'entreprise, notamment la centralisation, le niveau de compétences de l'utilisateur, et la perception de l'innovation par l'utilisateur. Pourtant [GALL 2001] critique cette approche, car certaines de ces études ont tiré des conclusions en n'interrogeant qu'un nombre limité d'utilisateurs finaux, selon la théorie des informateurs clés.

Dans sa recherche, [GALL 2001] s'attarde sur l'évaluation de l'impact de la centralisation ou la décentralisation dans l'organisation. Il conclut en déclarant que les attributs de l'innovation, de l'organisation (sa centralisation ou sa décentralisation) et les attributs de l'individu sont des conditions de facilitation, même s'il n'arrive pas à un consensus sur leur portée respective.

Pour l'auteur, l'intérêt de la recherche n'est pas tant l'adoption de l'innovation par l'utilisateur que la portée et le niveau de l'adoption par l'organisation pour la modification des processus, structures et culture de l'organisation. Il est facile de mesurer la portée de l'adoption en comptant le nombre de membres de l'organisation qui utilisent le produit. Il est plus difficile de mesurer d'une manière objective le niveau de l'adoption perçue des usagers.

Après avoir posé le cadre de sa recherche, [GALL 2001] présente les résultats recueillis par l'étude de l'adoption d'une technologie client/serveur qui remplace une technologie « main-frame » dans l'entreprise Insureco. De cette étude, [GALL 2001] détermine que certains facteurs constituent, lors des différentes étapes du projet, tantôt un avantage tantôt un inconvénient à l'adoption de l'innovation.

Facteurs de succès déterminants tout au long de l'assimilation

- Un message fort de la direction et bien diffusé aide toutes les étapes de la l'adoption
- Une allocation de ressources nécessaires pour faciliter l'adoption lors de toutes les étapes du projet : notamment avec des cours de formation, des livres, et examens pour mesurer les connaissances acquises sont toujours utiles.

Facteurs dont l'incidence peut varier pendant les étapes d'assimilation.

- Une structure hiérarchique, bureaucratique et organisationnelle aide à imposer et communiquer la volonté de la direction. Cette hiérarchie est très utile au début de l'adoption du produit mais constitue un inconvénient vers la fin du projet. En effet, des structures hiérarchiques bureaucratiques complexifient la rétroaction de la part des usagers vers les responsables.
- Une planification centralisée et une vue à long terme aident à démarrer le projet mais constituent des contraintes vers la fin, car les dernières étapes nécessitent des corrections de proximité.

Facteurs d'échec qui contraignent toutes les étapes de l'assimilation.

- Les normes culturelles de l'organisation qui attribuent à l'organisation la responsabilité complète pour le développement de carrières des employés peuvent être une contrainte vers la fin. Une vue étendue de la responsabilité qui voit l'apprentissage comme une initiative de l'employé peut faciliter l'assimilation.
- Les normes culturelles qui accentuent une vue étroite des rôles et responsabilités de l'employé peuvent être une contrainte vers la fin du projet. En effet, si l'employé interprète ses attributions d'une manière restrictive, il ne se sentira pas concerné par les processus d'affaires de l'entreprise. De même une vue étendue des rôles de l'employé peut faciliter les étapes subséquentes de l'assimilation.

Facteurs affectant l'assimilation, sans qu'on puisse déterminer qu'ils sont positifs ou négatifs.

- La sécurité d'emploi peut être autant un facilitateur qu'une contrainte pour motiver les employés à s'adapter au changement. Tout dépend des interactions entre les attributs des individus, mais l'incidence de ces interactions a surtout lieu vers la fin

de l'assimilation. Les projets d'adoption d'une innovation ne devraient pas avoir lieu dans un mauvais climat de travail, tel qu'une période de grève.

- Les attributs de l'individu ont peu d'influence sur les premières étapes de l'assimilation de la nouvelle application. Ces attributs s'exercent plutôt aux dernières étapes de l'assimilation car elles affectent la volonté de l'employé de s'adapter au changement des processus de travail.

Des attributs perçus comme un avantage, sont la tolérance de l'ambiguïté, la persévérance personnelle et l'esprit d'innovation. Des attributs perçus comme une contrainte sont un haut rang dans la hiérarchie de l'organisation et beaucoup d'ancienneté. [GALL 2001], comme [HAMM 2001] s'accordent pour reconnaître que les cadres intermédiaires représentent un risque pour le projet, lorsque ces derniers sentent que leur influence ou leurs responsabilités sont directement affectées.

4.2.4 La formation de la main d'œuvre

Dans sa recherche [TCHO 2002] étudie l'adoption du ERP de SAP par l'entreprise Pratt & Whitney. L'une des priorités poursuivies était d'avoir une formation disponible peu de temps avant la phase d'utilisation et qui réponde aux questions de base des employés telles que :

- Quelle est la nouvelle application ?
- Pourquoi l'installe t'on ?
- Pourquoi est elle installée maintenant ?
- Qui quand et comment va-t-on l'utiliser ?
- Qu'est ce que la nouvelle application va remplacer et quand ?
- À quoi peut on s'attendre en termes de formation ?

L'auteur déclare que des personnes, en moyenne, se souviennent d'abord de ce qu'elles font. Ensuite, par ordre de grandeur décroissant, des personnes en moyenne se souviennent de ce qu'elles disent, ce qu'elles voient, ce qu'elles entendent et en tout dernier lieu de ce qu'elles lisent.

Le plus souvent, il est difficile de retenir l'attention des employés lors de l'adoption d'une application. Les employés d'une organisation sont souvent déjà noyés dans l'information se rapportant à leurs tâches quotidiennes. Ils ne liront pas des documents d'apprentissage seuls. Ainsi, la formation de la main d'œuvre ne peut pas se limiter à la production de documents et à la budgétisation de temps nécessaire pour les lire. La priorité doit être le

contact direct. Une formation satisfaisante va permettre aux usagers de réaliser leurs opérations courantes eux-mêmes au moins une fois.

Il faut identifier et gérer la résistance aux changements chez les usagers. Lorsqu'on adopte un produit COTS et en particulier un ERP, il est impossible de faire participer les usagers à la conception ou à la configuration de l'outil. La seule participation des usagers est au niveau de l'adoption du nouvel outil.

Il ne faut pas compter uniquement sur les chefs d'équipe formels définis dans l'organisation. Il appartient aux responsables de la formation d'identifier une structure parallèle de changement en attribuant des rôles et responsabilités clairs à certains usagers. Ce n'est pas uniquement par l'autorité que le supérieur peut amener une innovation dans un groupe. Selon [TCHO 2002] il faut plutôt s'efforcer de créer une demande de formation de la part des usagers pour la gestion du changement. Évidemment on se demande immédiatement comment créer ou susciter cette demande, mais l'article de [TCHO 2002] n'y répond pas. Cependant quelques pistes de solutions sont mentionnées :

- Il faut essayer de se servir de la culture d'entreprise, plutôt que de l'affronter.
- Évaluer l'aptitude au changement et les risques
- Être spécifique, concret et factuel et dire toujours la vérité
- Impliquer et responsabiliser toutes les personnes touchées par le changement
- Utiliser la Communication comme principal outil de travail; appliquer les bonnes techniques en tenant compte des particularités spécifiques

4.2.4.1 Étude de cas : l'adoption de SAP R/3 par Pratt & Whitney

En 2002 Pratt & Whitney Canada adoptait la nouvelle version du ERP SAP R/3. Dans ce projet, examiné par [TCHO 2002], il y avait environ 2800 employés qui devaient utiliser la nouvelle application ERP. Pourtant 3000 employés suivaient des cours de formation, pour 10 processus principaux, regroupant 47 sous processus correspondant à 600 activités. Des cours de formation étaient donc dispensés à environ 10% d'employés qui n'allaient pas utiliser l'application.

Tableau 19 : Description des cours et manuels de formation [TCHO 2002]

Description des cours	Manuels	Participants	Contenu
Cours de navigation de base	2	2198	Ces cours devraient être très simples, généraux et répéter de plusieurs manières différentes les mêmes idées pour permettre aux usagers de comprendre les opérations de base
Cours qui présente sommairement les processus d'affaires	10	662	Ici il y a un manuel ou document pour chacun des 10 processus d'affaires principaux. Cette formation devrait être fournie aux gestionnaires.
Cours de formation à la tâche	145	2426	Ces cours devraient décrire des exemples très précis qui correspondent à des opérations que devraient normalement accomplir les usagers.

Les manuels des usagers du tableau 19 réunissaient un document commun de base qui serait enseignée à tous les employés, ainsi qu'un document spécifique qui se rapporte aux tâches de chaque catégorie d'employés. Ces manuels de cours ont fait l'objet d'une révision formelle.

Les cours de formation ont duré environ 3 jours par employé soit 28 heures de formation. La formation devrait constituer un renforcement des mesures de prévention ou de correction des difficultés pouvant survenir dans les premiers jours du démarrage.

Dans le projet de Pratt & Whitney, il y avait en moyenne 1 employé de soutien pour entre 3 et 6 employés au début de l'implantation. Soit des employés affectés au soutien technique parmi les usagers (un employé dans un groupe qui fait office de mentor en plus de ses tâches à l'intérieur du groupe) et central (employés à temps plein dans un centre de soutien technique pour tous les membres de l'organisation).

L'article de [TCHO 2002] insistait sur la notion de responsabilité partagée. C'est-à-dire qu'il faut convaincre les employés qu'ils doivent fournir des efforts d'apprentissage et être responsables de leur propre formation. Si les employés sont convaincus que c'est seulement l'employeur qui est responsable de leur formation, leurs efforts d'apprentissage seront moindres.

[YAKO 2001] distingue la formation technique pour les administrateurs et la formation technique, plus fonctionnelle, pour les usagers normaux. La formation des administrateurs doit avoir lieu lorsque la nouvelle application est installée sur un serveur, alors que la formation des usagers normaux doit avoir lieu peu de temps avant le déploiement officiel.

4.2.5 Négocier les changements aux processus d'affaires

Beaucoup d'entreprises se lancent dans la réingénierie, pour autant, d'après [OBER 2003], il n'existe pas de consensus au niveau de la notation et l'enregistrement des processus. Il n'en demeure pas moins que de nombreux standards sont proposés. L'article de [SCHE 2000] présente la modélisation ARIS des processus d'affaires utilisée par SAP. Par contre, [OBA 2001] emploie JointFlow et [KWAK 2002] emploie Workflow-XML pour modéliser les processus d'affaires, voir la section 2.2.2.

4.2.5.1 Technique en spirale de SAP

Dans son article [MAYA 2003] présente une technique développée par SAP pour encadrer la gestion des spécifications. Des études ont démontré que depuis son utilisation cette technique de gestion des spécifications a produit des bénéfices substantiels pour les projets qui l'ont employée. Pourtant l'article [MAYA 2003] avoue que certaines entreprises, qui ont employé la technique fort coûteuse de SAP pour accompagner le déploiement du ERP, n'en ont retiré aucun bénéfice. Ce qui démontre que certains facteurs de succès critiques n'ont pas pu être validés ou mêmes clairement identifiés.

La gestion des spécifications de SAP s'appelle le processus ASAP. C'est un processus en spirale qui contient 3 activités.

- l'élicitation des spécifications
- la modélisation de l'entreprise
- la négociation des spécifications

Le processus ASAP n'est censé donner lieu qu'à quatre spirales décrites au tableau 20.

Tableau 20 : Itérations du processus ASAP

Itération	Réalisation
1	Obtenir une image claire des structures de l'organisation.
2	Décider des objectifs et de la portée de la standardisation des processus.
3	L'architecture des processus d'affaires et des données de l'organisation se conforment à SAP
4	La spécification des conversions de données des interfaces est établie

4.2.5.2 Étude de cas

L'article de [MAYA 2003] a résumé des conclusions de 13 projets d'adoption de ERP entre 1997 et 2002 dans une entreprise de télécommunication sans fil Telus. Soit : 6 nouveaux déploiements, 3 améliorations, 2 mises à jour, et 2 alignements de processus à cause d'acquisitions d'entreprises.

Dans ces projets 67 équipes de 2 employés travaillaient sur la gestion des spécifications en liaison avec un seul architecte. La réingénierie des processus était analysée à l'aide d'un cadre de la maturité de la gestion des spécifications « Requirements engineering maturity framework ». De son examen [MAYA 2003] tire 15 recommandations. Pour résumer, [MAYA 2003] ne fait pas de distinction entre facteurs stratégiques et tactiques, donc ces recommandations ne sont pas priorisés. Les recommandations de [MAYA 2003] reprennent les facteurs de succès de projets de migration de [HOLL 1999-a], mais se concentrent sur des aspects tactiques de vérification au cours des projets de migration, alors que les réflexions de [HOLL 1999-a] présentent une vision d'ensemble.

4.3 Gestion des facteurs stratégiques

Les facteurs stratégiques selon [HOLL 1999-a] au tableau 17 regroupent plusieurs notions définies dans d'autres sections, notamment les systèmes patrimoniaux au chapitre 1 et la planification et l'échéancier du projet au chapitre 5. Par conséquent, cette section s'attardera plus spécifiquement sur les questions du soutien de la haute direction et de la stratégie ERP. Le soutien de la direction est un élément essentiel. Ce soutien a été vu comme déterminant dans de nombreux projets d'organisations prestigieuses, pourtant un soutien fort ne signifie pas un soutien aveugle. La stratégie ERP examine les questions liées à la personnalisation sur mesure de l'application acquise. Dans la littérature les auteurs

s'entendent pour dénoncer toute modification sur mesure de l'application mais négligent de différencier modification générique et modification spécifique.

4.3.1 Soutien de la direction

Pour [JARR 2000], les solutions ERP ne sont pas simplement des solutions logicielles, elles affectent la manière dont l'entreprise est gérée.

Les facteurs de succès considérés par des experts du domaine sont :

- l'engagement de la haute direction :
Le soutien de la direction est décisif. Il est néfaste pour la direction de ne voir un ERP que comme un problème technique. Seule la haute direction peut résoudre les conflits entre les impératifs de la technologie et les impératifs de l'organisation.

- la gestion du changement : Il faut changer la manière dont les employés travaillent.
- l'infrastructure informatique : Un ERP dépend du matériel et de l'infrastructure du réseau.
- la réingénierie des processus d'affaires :

Il faut le soutien de la direction pour modifier les processus de l'entreprise pour ceux des meilleurs standards de l'industrie. En adoptant l'ERP, l'organisation veut se conformer au modèle du ERP, pourtant même les meilleurs ERP ne peuvent souvent satisfaire que 70 % des besoins de l'organisation. Il ne faut pas s'éterniser sur les 30 % restants.

L'article de [JARR 2000] présente les méthodes suivies par plusieurs grandes entreprises qui ont adopté des ERP, notamment Microsoft, Hewlett Packard, et Monsanto.

En 1992 et 1994 Microsoft a échoué dans sa tentative de déployer un ERP et attribua l'échec à un manque de soutien de la direction. La troisième tentative réussit notamment à cause de l'engagement de la direction. La compagnie Hewlett Packard (HP) nota que le déploiement d'un ERP est plus lié aux problèmes humains qu'aux problèmes technologiques. L'organisation subit une transformation majeure et la direction doit avoir une vision d'affaires claire pour la soutenir. La compagnie Monsanto s'attacha à attribuer des responsabilités claires et centralisées pour les changements aux processus d'affaires.

Pour communiquer efficacement, il faut prévenir tous les employés à l'avance des objectifs, de la portée et des activités du projet et admettre que cela implique du changement.

Recommandations clés :

1. Obtenir le soutien de la direction
2. Modifier les processus d'affaires
3. Former des membres de l'organisation
4. Mesurer les performances
5. Choisir les meilleurs employés pour travailler sur ce projet

4.3.1.1 Soutenir les efforts de modernisation par la haute direction

Dans l'article de [BERG 1999], l'auteur fournit des recommandations pour la haute direction dans les organisations qui déploient un nouveau système. Ces recommandations sont valides qu'il s'agisse d'une application développée à l'interne ou achetée dans le marché. Ces recommandations ont été colligées à partir d'autres projets de modernisation.

- Avoir une stratégie globale, et fixer des objectifs mesurables pour chaque sous projet
- Surveiller et définir avec soin les rôles et responsabilités des sous-traitants
- Former et motiver les employés
- Conserver un contrôle sur le système patrimonial par la gestion des configurations
- Collecter et valider les besoins selon un processus clairement défini
- Privilégier l'architecture comme le cœur du projet
- Il doit exister un processus de réingénierie spécifique
- Créer un plan de réingénierie qui privilégie la dynamique de l'équipe, et le respecter
- Les gestionnaires doivent prendre des décisions pour le long terme
- Les gestionnaires ne doivent pas ignorer les contraintes techniques.

4.3.1.2 Limites au soutien de la direction

L'article de [JARR 2000] propose de mettre au point un ensemble de métriques qui pourraient assister la haute direction dans les choix qu'ils doivent effectuer. Cependant cet article n'offre aucune piste de solution sur la détermination des métriques à employer pour la haute direction. La direction ne peut pas se contenter de soutenir le projet en examinant des rapports de métriques. La direction doit s'impliquer dans le projet.

L'engagement des membres de la direction et des directions d'unités de l'organisation signifie se rendre disponible pour s'assurer que les questions soulevées par le projet soient promptement résolues. Les processus de communication et de décision, entre les responsables du projet et la direction, doivent être très rapides. L'engagement de la

direction signifie décharger les personnes, essentielles à la réussite du projet, de leurs responsabilités quotidiennes pour leur permettre de se concentrer sur le projet.

Il ne faut pas confondre un soutien fort de la direction pour le projet et un soutien inconditionnel. Dans le projet Gires, les coûts et délais du projet ont dépassé de plus de cinq fois le budget initial comme l'indique le rapport [LYRE 2003]. Seule une volonté ferme de la direction du projet d'accéder aux demandes de nouveaux crédits et de ressources à pu autoriser ces débordements. On pourrait aller jusqu'à dire que le projet était trop soutenu.

En outre, il découle de l'article de [JARR 2000] que si on veut modifier les processus d'affaires et donc gérer les problèmes humains, une difficulté supplémentaire est ajoutée lorsqu'une unité de l'organisation a fait l'objet d'une fusion forcée ou d'un rachat par une organisation principale.

4.3.2 Stratégie ERP

Dans son article, [PARR 2000] distingue trois catégories d'adoption :

L'adoption complète « Comprehensive »

Un ERP quasiment au complet est adopté à travers une organisation multi nationale, qui comporte plusieurs sites. Cette approche requiert beaucoup de réingénierie des processus d'affaires et peut durer jusqu'à 7 ans et coûter des dizaines de millions de dollars.

L'adoption moyenne « Middle road »,

Dans cette catégorie, l'organisation qui adopte le ERP peut y avoir un site principal ou plusieurs sites. Par contre, seul un nombre restreint de modules du ERP est adopté. Cette approche requiert elle aussi, beaucoup de réingénierie des processus d'affaires et peut durer de 3 à 5 ans et coûter environ 3 millions de dollars.

L'adoption vanille « Vanilla »,

Cette catégorie de déploiement est peu ambitieuse et peu risquée. On déploie le système dans un seul site et le nombre d'utilisateurs est petit (moins de 100 utilisateurs). On n'adopte qu'un nombre restreint de modules clés. C'est l'organisation qui s'adapte à l'ERP et non l'inverse. Cette approche peut durer de 6 à 12 mois et peut coûter de 1 à 2 millions de dollars.

On détermine la catégorie en fonction des caractéristiques de l'organisation, soit :

- La portée physique,
Le nombre de sites. Un site dans une région, ou plusieurs dans des pays différents
- La portée de la réingénierie des processus d'affaires,
- La portée technique,
Quels sont les changements techniques à apporter au ERP ?
- Le déploiement des modules
Par modules on entend les modules qui soutiennent les processus à haut niveau de l'organisation et l'ordre du déploiement des modules. Il est très rare de retenir tous les modules ERP dans tous les domaines de l'entreprise. Dans l'article de [PARR 2000] une seule entreprise étudiée sur 42 a adopté le ERP dans sa totalité.
- L'allocation des ressources.

Pour [PARR 2000], le déploiement des ERP est un domaine nouveau pour lequel il est difficile de valider des théories de gestion. La recherche dans le domaine se limite à enquêter auprès des experts du domaine et compiler les points communs d'études de cas.

4.3.2.1 Modification de l'acquisition

Toute modification sur mesure d'une application informatique commerciale achetée sur le marché est quasiment systématiquement critiquée par tous les auteurs en recherche, notamment [MEYE 2001], [OBER 2003], [PARR 2000], [BROW 1999] et [MAYA 2003], quoi que certains distinguent modification sur mesure et configuration.

L'application ne doit pas être modifiée uniquement pour satisfaire le client ou uniquement selon le bon vouloir du fournisseur. Dans son article [OBER 2003] parle du fournisseur et de l'acquéreur comme de deux centres de gravité que l'application doit accommoder. L'acquéreur doit modifier ses processus d'affaires pour s'adapter à l'application. Une modification spécifique de l'acquisition ne peut être le résultat que de spécifications essentielles que l'application choisie ne rencontre pas. Pour [OBER 2003] ces spécifications essentielles devraient être peu nombreuses. Les spécifications de l'application doivent être souples et pouvoir être négociées. Il faut mettre beaucoup d'attention au choix des spécifications essentielles pour ne pas inclure des spécifications simplement désirées ou des spécifications qui pourraient être négociées.

[OBER 2003] associe 6 principes au processus APCS

1. Il faut des spécifications flexibles
2. C'est l'acquéreur qui doit s'adapter au produit et modifier ses processus d'affaires
3. Une solution COTS doit évoluer avec le temps pendant sa durée de vie
4. Les produits COTS ne sont connus que d'une manière imparfaite, donc il faut prévoir et financer les tests directs sur le produit COTS
5. Les produits COTS sont créés indépendamment les uns des autres. Il faut donc étudier leur intégration lorsqu'ils devront travailler ensemble.
6. Tous les membres de l'organisation doivent participer à l'atteinte des objectifs continuellement : de la conception à l'installation et l'utilisation.

[BROW 1999] recommande de ne pas acheter le code source pour ne pas être tenté de le modifier. Dans l'article de [MAYA 2003] on identifie l'utilisation des modules de l'acquisition sans modifications générique ou spécifique comme début, milieu et fin du projet.

Il arrive effectivement des situations où un produit n'est pas conforme à certaines spécifications essentielles pour un grand nombre d'acquéreurs : on peut alors parler de modification générique. Lorsqu'une version d'un produit logiciel n'est utilisée que par un marché très restreint, elle devient dans les faits une version sur mesure. Il peut aussi exister des situations où des spécifications sont uniquement demandées par un seul acquéreur : on peut parler de changements sur-mesure ou modification spécifique. Ce type de changement à la version originale du logiciel peut représenter un risque élevé.

4.3.2.2 Modification générique

Il peut exister un bassin de clients qui partagent des intérêts communs pour la modification d'un produit logiciel. Cette modification peut concerner la localisation du produit ou l'adoption de processus communs. Par le terme « localisation » on entend les options régionales réalisées sur l'application pour qu'elle soit lisible ou compréhensible selon une région ou pays où elle est utilisée.

Exemple

Si une entreprise au Québec adopte un logiciel américain, et que ce système est peu utilisé au Canada, il y aura nécessairement au moins deux modifications génériques

à la version du fournisseur : tout d'abord pour que l'application soit disponible en français, ensuite pour que l'application puisse manipuler des dollars canadiens.

4.3.2.3 Modification spécifique sur mesure

Cette section recoupe les enjeux analysés dans la section 4.1.2. Les gouvernements ont souvent tendance à acquérir et à modifier des produits COTS pour qu'ils s'adaptent aux processus dans les divers ministères et organismes, sans qu'un consensus existe entre un grand bassin de clients.

Ce changement est valable s'il offre un avantage concurrentiel dans le long terme. Cette modification peut être due à un contexte exceptionnel pour le client ou à un manque de sérieux dans l'effort de réingénierie des processus d'affaires. Cette modification devrait être évitée autant que possible, ou se limiter à quelques sous routines. L'inconvénient principal de toute modification spécifique est qu'elle doit être développée, maintenue et testée aux frais de l'acquéreur, ainsi que pour toutes les versions ultérieures du produit logiciel. Les ressources de développement, de test et de maintenance sont souvent prohibitives dans le long terme.

4.4 Conclusion

Les applications patrimoniales qui sont remplacées, doivent donner lieu à un système ouvert, qui respecte les standards du domaine. Nous avons repris la distinction de [HOLL 1999-a] des aspects stratégiques et tactiques pour un projet de migration. Selon [HOLL 1999-a], il est essentiel d'avoir le soutien fort de la direction et des directives claires quant à la modification sur mesure de l'application. Avoir une formation de la main d'oeuvre adéquat, un processus de négociation des processus éclairé et une relation cordiale, entre fournisseur et acquéreur, n'offrent que des avantages tactiques.

Cette distinction fournit une vision claire à la direction du projet en identifiant les priorités essentielles du projet. Lors d'entrevues effectuées auprès de responsables de projets de migration à l'université Laval, ces derniers ont parfois contesté quels facteurs étaient stratégiques par opposition à tactiques mais n'ont pas contesté la distinction faite entre facteurs stratégiques et tactiques.

Chapitre 5 Plan de migration

La modernisation des procédures d'affaires repose sur une prise de contrôle des activités de l'entreprise et donc d'un effort de concentrer le pouvoir de décision et d'une centralisation des applications de gestion.

Le déploiement d'un ERP se décompose en déploiement de ses modules qui constituent des sous projets. On doit prioriser ces sous projets entre eux, car seuls les modules principaux devraient être déployés et on ne peut pas déployer tous les nouveaux modules en même temps et à brève échéance. Chaque sous projet doit se concentrer sur ce qui doit obligatoirement changer dans l'organisation, sur où l'organisation est la plus vulnérable au changement et où l'organisation prévoit de rencontrer le plus de résistance. Pour considérer ces enjeux ce mémoire s'attarde sur la question des applications concurrentes et de la migration multi site.

Les produits livrables d'un projet de migration incluent des documents et des composants logiciels. La documentation doit définir les interactions entre les intervenants et les informer correctement. Les composants logiciels doivent réaliser l'intégration de la nouvelle application et convertir les données nécessaires.

5.1 Imposer une centralisation

Dans les très grandes organisations, plusieurs applications différentes peuvent parfois être utilisées pour effectuer des opérations semblables dans les diverses unités. Un plan de migration qui vise à déployer un ERP et en tirer les bénéfices doit imposer une certaine centralisation de l'autorité et des applications autorisées. Cette centralisation peut être difficile lorsque le déploiement doit se faire dans plusieurs sites.

5.1.1 Concurrence entre applications

Des applications dans une organisation peuvent être déployées soit localement dans une unité de l'organisation ou centralement sous la direction et le contrôle de la haute direction. Généralement, les organisations sont divisées en unités spécialisées pour se répartir les tâches. Si une unité est de petite taille ou réunit des usagers qui effectuent un travail ou

opérations analogues le déploiement local d'une application est simple. Les applications déployées localement dans une unité d'une organisation sont moins complexes que des applications déployées centralement à travers des unités qui ont des besoins distincts. Des applications peuvent donc être déployées à moindres frais et plus rapidement dans une unité locale qu'à travers toute l'organisation.

Pour cette raison le déploiement d'une application de gestion à travers une grande organisation doit écarter les applications locales concurrentes pour pouvoir s'imposer comme outil de référence. La grande difficulté est de déployer des applications centralement, dans une organisation dont les unités constituantes peuvent être considérablement hétérogènes. Des avantages sont associés aux applications déployées centralement : la possibilité de réunir le maximum de membres de l'organisation dans le long terme d'une manière plus fiable, de fournir des économies d'échelle, et de fonctionner d'une manière optimale. Déployées centralement ces applications peuvent être les plus actives. La conquête du centre s'oppose à l'environnement patrimonial et les applications concurrentes. Cette conquête doit être rapide, et ciblée pour ne pas diminuer la productivité de l'organisation pendant la migration.

5.1.2 Approche Multi site

Pour [MARK 2000] les modules financiers sont les premiers qu'on doit choisir de déployer avec un ERP. Les finances et la comptabilité sont le premier moyen de contrôle des opérations de gestion d'une organisation. L'auteur note que, dans les grands projets de migration multi site, les succès sont l'exception et non la norme. La centralisation est un élément important de l'article de [MARK 2000]. Pour celui-ci, le terme multi site a plusieurs interprétations.

- Multi site correspond à des configurations multiples pour les modules du ERP dans une même organisation à cause d'une répartition des responsabilités de gestion des opérations et d'indépendance financière.

Un exemple d'organisation qui compte des opérations multiples et des finances multiples serait un constructeur mondial d'automobile. Ce type d'entreprise s'adapte au pays dans lequel il se trouve. Des versions distinctes d'un ERP, ou des

ERP différents peuvent être adoptés dans chaque pays, autant pour la production que pour les finances.

- Multi site du point de vue technologique, correspond à une combinaison de bases de données centrales ou de serveurs d'applications à plusieurs lieux géographiques distincts.

Soit on a une architecture avec un site primaire et des sites subordonnés soit des sites répartis qui ont des capacités équivalentes. Les sites répartis posent le problème de la synchronisation des données, du temps de réponse et les coûts de soutien, mais ont l'avantage d'être de taille moyenne et permettent de contrôler les politiques sur les données dans un site individuel d'une manière autonome.

- Le terme multi site s'applique aussi la distribution et la livraison des versions approuvées d'un ERP à travers l'organisation.

BICC Cables, une entreprise de télécommunication et de production de câbles électriques, adopta un ERP et décida que l'entreprise ne voulait pas avoir plus que trois versions du ERP installées à travers l'organisation en un temps donné. Soit une version ancienne qui doit être remplacée, une version la plus récente et une version, testée au quartier général de l'entreprise en tant que site pilote. Par contre, l'entreprise alimentaire Kraft, lors de son déploiement d'un ERP choisit une orientation différente. Kraft n'hésita pas à distribuer une nouvelle version du ERP dans certains sites alors que la version initiale standard n'était pas encore complètement installée.

5.2 Planifier plusieurs projets de migration

Un plan répartit les projets de migration des différents modules d'un ERP entre eux. Or, des sous projets peuvent souvent se greffer à un projet de migration ERP de grande envergure. Par grande envergure on exclut les projets vanille, tels que décrit à la section 4.3.2. Pour autant ces sous projets ne doivent pas distraire le gestionnaire du projet principal.

5.2.1 Éviter l'approche « Big Bang »

Une approche « Big Bang » effectue les changements technologiques et aux processus en un seul et même incrément. Ce type d'approche n'est valable que dans un contexte restreint, c'est-à-dire le déploiement de quelques modules d'un ERP dans une petite organisation. SAP propose une migration appelée « FastTrack 4 SAP ».

Cette approche fut retenue dans la migration multi site de Pratt & Whitney à SAP R/3 comme en fait état l'étude de cas de [TCHO 2002]. Cette migration s'étala sur un délai

court : un an et 8 mois, et fit l'objet d'un succès. SAP a développé la solution FastTrack pour les petites et moyennes entreprises où le prix de l'installation est fixe et garantie. Le succès relevé par [TCHO 2002] apparaît paradoxal puisque Pratt & Whitney n'est en aucun cas une petite ou moyenne entreprise et le déploiement concernait 6 modules, à travers 5 usines au Canada était un projet de grande envergure. Il faudrait nuancer en précisant que dans l'exemple, le logiciel SAP R/2 était déjà implanté dans certaines unités de l'entreprise, les autres unités employant d'autres systèmes patrimoniaux. La migration à la version R/3 n'était peut être pas trop difficile. De plus, le manque de transparence concernant le budget en argent et l'allongement de la durée du projet de 18 à 32 mois relevé par [TCHO 2002] réduit la pertinence des succès avec l'approche « Big Bang ».

Les changements technologiques incluent le déploiement de tous les modules d'un ERP, et l'intégration de l'ERP avec les applications de l'entreprise. Cette approche pose des inconvénients : si les changements technologiques sont réalisés en même temps, il devient difficile de déterminer l'origine d'une panne technique qui entraîne des interruptions du travail des employés, enquêtes du côté des administrateurs système.

L'espacement des changements techniques assure aux usagers un environnement de changement incrémental et pas radical. De plus les administrateurs et le soutien technique pourront identifier plus clairement les modules qui subissent des pannes. Dans une approche « Big Bang », tous les changements aux processus d'affaires des employés sont réalisés en même temps. Toute erreur avec les nouveaux processus sera difficile à identifier, et les employés seront, en attente d'éclaircissements.

Un projet de migration vers un ERP constitue souvent le plus important investissement informatique de l'organisation, et peut même représenter l'investissement le plus important de toute l'histoire de l'organisation. Choisir une approche « Big Bang » est très ambitieuse et ne permet que deux alternatives au projet : la réussite complète ou l'échec complet, pourtant elle reste utilisée dans la pratique. Le rapport [LYRE 2003] du projet Gires pour le gouvernement du Québec déclare que le projet Gires avait suivi une approche « Big Bang » pour effectuer en parallèle la migration des ressources humaines, financières du gouvernement du Québec. Cette approche fut suivie dans le but d'économiser des coûts.

L'intégrateur pensait qu'un seul projet coûterait moins que plusieurs petits projets. Le projet Gires fit l'objet d'un échec.

Dans sa recherche [MOTS 2002] présente des méthodes empiriques pour gérer l'adoption et l'utilisation de produits COTS. Le premier facteur de succès est de partager l'adoption en sous projets.

- Facteur 1 A-t-on la possibilité de partager le travail global en sous projets et de se concentrer sur chacun séparément ?
- Facteur 2 A-t-on la possibilité de former des partenariats avec des vendeurs ?
- Facteur 3 Existe-il des standards pour les composants du système et a-t-on la possibilité de les utiliser ?
- Facteur 4 Est-ce que tous les niveaux de l'organisation comprennent l'impact étendu que l'approche COTS va avoir sur l'organisation, y compris les nouvelles compétences requises ?
- Facteur 5 Le processus de déploiement de l'organisation a-t-il la flexibilité nécessaire pour accommoder la volatilité du marché ?

5.2.2 Déploiement par phases

Lorsque la portée du projet a été décidée, chaque déploiement de modules constitue un sous projet de migration individuel. Chaque sous projet comporte une étape à laquelle l'application devient une application officiellement reconnue et utilisée par l'organisation. Pour décrire ce concept [TCHO 2002] emploie le terme « Go Live » que nous traduisons par le terme « basculement ».

D'après [LANG 1998] la période de basculement devrait être choisie judicieusement. Soit en fin d'exercice ou pendant une période de faible activité de l'organisation. La fin d'exercice est une période de fermeture et de réouverture de l'organisation. La période de migration devrait être choisie plutôt que subie par l'organisation. Si on décide de basculer d'un système à un autre au début d'un nouvel exercice, il est nécessaire de bien comprendre les traitements de clôture et de réouverture d'exercice. Le choix du basculement en fin d'exercice est mieux adapté aux logiciels comptables, et la période de faible activité est mieux adaptée pour les logiciels réalisant les activités commerciales. Il vaut mieux, attendre les périodes de faible activité, pour basculer à un nouveau système avec les clients et fournisseurs. La conversion des données de commandes et d'inventaire interrompent les activités commerciales de l'organisation, jusqu'à ce que la conversion soit vérifiée et

que les données soient accessibles normalement. Ensuite a lieu une période d'ajustements, pendant laquelle l'application est instable. Pendant cette période, les administrateurs et employés de soutien corrigeant les pannes et s'habituant à l'application.

5.2.2.1 Établir un calendrier de déploiement

Le livre de [LANG 1998] propose un calendrier de déploiement en étapes qui fait subsister le système patrimonial et le système cible. Ce calendrier distingue 3 périodes

- Période 1 (instant donné) : on accepte le nouveau système. Le nouveau système est reconnu et les transactions qui y sont opérées, sont correctes et reconnues par l'organisation. Pendant cette période des modifications transitoires sont installées et servent d'interface pour s'assurer que le système cible et le système patrimonial sont cohérents.
- Période 2 (intervalle de temps) : Les 2 systèmes vont co-habiter. Selon des règles strictes les membres de l'organisation pourront effectuer des transactions dans un système ou dans l'autre.
- Période 3 (instant donné) : Le nouveau système est accepté définitivement et seuls sont reconnues les transactions effectuées avec le nouveau système, l'ancien système n'est plus reconnu.

Ce type de calendrier est utile dans les grandes organisations qui comportent beaucoup d'unités ou d'organisations décentralisées. En accordant un intervalle de temps il est possible de laisser une marge de manœuvre à la direction des unités locales de l'organisation pour décider du moment de la migration et de délimiter les tâches du soutien technique. Dans de petites organisations ou pour des migrations plus ciblées, ce type de calendrier apparaît mal adaptée.

Le livre de [LANG 1998] analyse les projets de migration des systèmes informatiques comptables à l'euro en Europe. La migration à l'euro, en France, a fait l'objet d'un calendrier de déploiement échelonné sur plusieurs années.

Période 1 : En 1999, l'euro est reconnue comme monnaie nationale par la bourse et marchés monétaires. La banque centrale européenne se substitue à la banque de France. Les taux de change entre les pays de la zone euro disparaissent.

Période 2 : Pendant une période de 3 ans, entre 1999 et 2002, pendant laquelle les entreprises s'adaptent à l'euro. Le franc et l'euro co-existent. Des ventes ou remboursements en francs et en euros sont valides et reconnus.

Période 3 : En 2002, La monnaie nationale française disparaît et on introduit des billets et pièces euro.

[LANG 1998] ajoute une quatrième période, pendant laquelle les données historiques comptables des entreprises sont converties en euros. Les convertisseurs dynamiques de monnaie sont donc supprimés.

Pour établir une description des tâches à budgétiser et réaliser lors d'une migration, le rapport technique de [STAL 2002] fournit un modèle de division et description des tâches pour l'adoption de produits commerciaux. Ce modèle est particulièrement détaillé et basé sur le modèle de Rational Unified Process, avec les étapes d'inception, d'élaboration de construction et de transition.

5.2.3 Durée de l'analyse et du déploiement

Plus les changements sont importants, plus l'étude de la portée du système est longue. Les délais de préparation du projet doivent être compensés par un déploiement rapide. En examinant le déploiement d'un ERP dans une entreprise de textile, [HOLL 1999-a] relève que cette entreprise a requis 6 ans pour préparer la migration (définir l'idée, l'étendue et la portée du projet) et 3 ans pour la période de migration. La complexité géographique et la rotation du personnel ont contribué à étendre la durée de vie du projet de 30 à 50 %. Enfin, le début d'un projet de migration peut être reportée à plusieurs reprises.

Au début d'un projet d'adoption d'un ERP, qui comporte la migration de plusieurs modules, l'intérêt n'est pas l'échange d'une application ancienne par une autre, l'échange ne doit avoir lieu que s'il apporte un avantage clair dans la stratégie de migration globale.

5.2.3.1 L'interruption des activités normales de l'organisation

Les projets de migration peuvent durer des années. Le déploiement, qui n'est que l'étape la plus visible de la migration, peut elle aussi durer des années pour des projets ERP dans des grandes organisations tel les projets « middle-road » définis dans la section 4.3.2.

Le déploiement de ERP peut avoir comme effet non souhaité de faire prendre du retard aux différentes unités de l'organisation. Le déploiement d'un projet ERP interrompt les nouvelles initiatives des unités de l'organisation, comme le confirme l'article de [YAKO 2001]. Cet inconvénient fut aussi noté dans le rapport du projet Gires [LYRE 2003]. Le projet avait monopolisé les ressources des ministères et organismes qui devaient l'adopter et avait pour effet de leur faire prendre du retard et de grever leurs budgets puisque le financement de l'adoption de Gires n'était pas exclusivement payé par le ministère des finances. Ce n'était qu'après la fin du déploiement de Gires que les ministères et organismes auraient pu considérer d'autres initiatives dans un environnement stable.

5.2.4 Étude de cas

L'article de [YAKO 2001] illustre la recherche récente qui décrit l'adoption d'outils ERP dans de grandes organisations ou de taille moyenne. Son article présente une petite université américaine qui adopte les modules de gestion des études de PeopleSoft. La migration au nouveau système bloqua tous les autres projets informatiques pendant la période du projet. Les meilleurs employés furent relevés de leurs tâches normales pour se concentrer sur le projet. Le ERP de PeopleSoft comporte quatre modules qui organisent les processus de l'université :

- Module admission,
- Module gestion des données étudiantes
- Module gestion des finances étudiantes
- Module gestion de l'aide financière.

Tableau 21 : Plan de migration suivi par l'université étudiée par [YAKO 2001]

Session	Description
1998 Automne	Formation technique, Sélection du nouveau matériel, Sélection du système d'exploitation et de la plate-forme
1999 Hiver	Installation du matériel, formation des administrateurs aux modules d'admission et de données étudiantes
1999 Été	Formation à l'admission. Installation des modules d'admission et données étudiantes sur le serveur. Conversion des données scolaires historiques des étudiants à partir du système patrimonial
1999 Automne	Conversion des données scolaires courantes. Conversion des demandes d'admission courantes et historiques
2000 Hiver	Activation du module admissions pour les étudiants admis et les étudiants potentiels. Activation des données étudiantes avec l'inscription par Internet. Installation des modules d'aide financière. Formation technique additionnelle.
2000 Été	Entrée double pour les modules étudiants potentiels et étudiant admis ainsi que pour les données étudiantes. Installation du module des finances étudiantes.
2000 Automne	Activation des finances étudiantes, activation de l'aide financière. Tous les modules sont actifs.

Ce plan mérite d'être commenté du point de vue de la conversion des données et du point de vue du calendrier du projet.

La conversion de données

Ce ne sont pas toutes les données du système patrimonial qui ont été ajoutées au nouveau système. On s'est contenté de migrer les données concernant les étudiants et les admissions. En effet, il faut pouvoir retrouver les dossiers des étudiants admis à cette université pour continuer à mettre à jour leur dossier. Il en est de même pour les données concernant les admissions. Si une personne a déjà fait une demande pour être admise à l'université, l'acceptation ou le refus doit être maintenue dans un dossier qui lui est associé. Les données les plus anciennes n'ont pas été converties ou transférées au nouvel environnement.

Le calendrier du projet

Les modules admission et données étudiantes furent activés en premier à l'hiver 2000, mais devaient fonctionner pleinement pour la période d'inscription pour l'automne 2000. L'activation de tous les autres modules eut lieu à l'automne 2000. Cependant en regardant l'emploi du temps on peut penser que la formation qui eut lieu à l'hiver 1999 fut donnée trop tôt, par rapport à l'utilisation à l'automne 2000. Malgré tout ce délai peut se justifier car beaucoup d'employés à l'université prennent des vacances pendant l'été.

Donc on peut dégager un calendrier générique de migration qui serait :

Tableau 22 : Calendrier de migration

Sélection du nouveau matériel, du système d'exploitation et de la plate-forme Formation technique des administrateurs Installation du matériel du système d'exploitation et de la plate-forme Pour chaque module <ul style="list-style-type: none"> Installation du module sur la plate-forme Intégration du module avec l'environnement Formation des usagers au nouveau module Migration des données dans le nouveau module (en période de faible activité) Activation du module (en début d'un exercice) Revue du module Examen post-mortem du projet complet

5.3 Produits livrables

Les produits livrables pendant un projet de migration incluent les documents livrables qui doivent définir comment échanger l'information pertinente entre différents intervenants, et les composants logiciels qui doivent être livrés ou acquis et configurés. Un des composants logiciels qui doivent être développés sont les convertisseurs de données qui feront l'objet d'un examen détaillé.

5.3.1 Documents livrables

[ALBE 2002] propose un processus évolutionnaire pour intégrer des produits commerciaux dans une organisation. Sa recherche est particulièrement détaillée et basée sur le modèle de Rational Unified Process, avec les étapes d'inception, d'élaboration de construction et de transition. Son rapport technique fournit un inventaire de documents livrables possibles et sont résumés ici.

5.3.1.1 Catégories de documents livrables

Le standard [IEEE/EIA 12207.0-1996] fournit deux gabarits de plans : un plan de migration et un plan de retraite logicielle, et spécifie quelles informations doivent impérativement être communiqués aux usagers.

Tableau 23 : Plan de migration

- | |
|---|
| <ul style="list-style-type: none"> a) Analyse de spécifications et définition de la migration b) Développement des outils de migration c) Conversion du logiciel et conversion des données d) Exécution de la migration e) Vérification de la migration f) Soutien pour l'environnement patrimonial dans l'avenir |
|---|

Tableau 24 : Plan de retraite logicielle

- | |
|--|
| <ul style="list-style-type: none"> a) Arrêt complet ou partiel du soutien après un certain laps de temps b) Archivage du logiciel et sa documentation associée c) Responsabilité pour tout enjeu résiduel dans l'avenir d) Transition au nouveau produit logiciel si applicable e) Accessibilité des copies de données en archive |
|--|

[ALBE 2002] distingue quatre catégories de documents livrables. Les trois premières catégories correspondent à des documents livrables standard de tout projet rigoureux de développement logiciel. La quatrième catégorie intitulée « Documents additionnels de planification et d'évaluation du risque » fournit des précisions utiles pour les plans de migration.

1. Les documents qui caractérisent les processus d'affaires et les besoins des intéressés. Ces documents sont essentiellement des cas d'utilisation patrimoniaux et cibles et des documents de spécifications.
2. Les documents qui décrivent la conception et l'architecture. Ces documents regroupent les documents de conception logicielle, plan de tests, documents de formation et d'installation, notes de mise à jour et documents de soutien de l'utilisateur.
3. Les documents de planification et d'évaluation du risque. Ces nombreux plans sont regroupés en 5 thèmes : processus de gestion, surveillance et contrôle, gestion du risque, plan des processus techniques, plan de soutien des processus.
4. Documents additionnels de planification et d'évaluation du risque.
 - Dossier commercial (Ce document définit les critères de succès, et les prévisions financières)
 - Plan de gestion du changement des processus d'affaires
 - Plan de déploiement
 - Accords de licence
 - Documents d'évaluation des itérations (un par itération)

5.3.1.2 Gabarits des documents additionnels de planification et d'évaluation du risque

Les documents identifiés par [ALBE 2002] par le nom de documents additionnels méritent un examen plus attentif. Un modèle de plan de gestion du changement des processus d'affaires et de document d'évaluation des itérations sera présenté. Nous présenterons aussi un modèle de plan de relation avec le fournisseur.

Le plan de gestion du changement des processus d'affaires proposé par [ALBE 2002] est très lourd et détaillé en 13 points que nous résumons ici aux points principaux. Se rapporter au document de référence pour plus de détails [ALBE 2002], chapitre 10.

Tableau 25 : Plan de gestion du changement des processus d'affaires

- | |
|--|
| <ol style="list-style-type: none"> 1. Introduction 2. Analyse d'appréciation 3. Planification préliminaire 4. Activités de diagnostique 5. Attribuer les rôles et responsabilités clés 6. Commanditaire responsable 8. Plan de communication 9. Plan pour renforcer le changement 10. Plan pour cibler la résistance 11. Plan contre la résistance culturelle 12. Plan de surveillance et de contrôle 13. Plan d'intégration |
|--|

[ALBE 2002] ne fournit pas de gabarit pour les documents d'évaluation des itérations. Nous proposons un gabarit pour ce document où les processus énumérés sont les processus à haut niveau de l'organisation (généralement moins d'une dizaine, même pour une grande entreprise tel que défini dans la section 3.2.1).

Tableau 26 : Document d'évaluation des itérations (après chaque itération)

<ul style="list-style-type: none"> 1. Introduction <ul style="list-style-type: none"> 1.1 Objectif 1.2 Portée 1.3 Définitions, acronymes, et abréviations 1.4 Références 2 Processus <ul style="list-style-type: none"> 2.1 Processus (pour chaque processus) <ul style="list-style-type: none"> 2.1.1 Consensus 2.1.2 Dissensions 2.1.3 Commentaires 3 Gestion du changement, adhésion de la communauté <ul style="list-style-type: none"> 3.1 Consultation 3.2 Adhésion au projet 3.3 Adhésion au produit 3.4 Culture et communications 4 Recommandations <ul style="list-style-type: none"> 4.1 Niveau du consensus 4.2 Maturité du produit [Fournir au minimum un exemple d'entreprise du même type qui a adopté le même produit] 4.3 Maturité du fournisseur [Standards publics utilisés par l'entreprise et les moyens de rétroaction des clients]

Le plan de relation avec le fournisseur de [ALBE 2002] résume le type de relations à établir avec les différents intervenants externes : les fournisseurs, vendeurs ou intégrateurs et a besoin d'être rédigé rigoureusement pour les organisations qui veulent rompre avec une culture de développement pour passer à une culture d'acquisition.

Tableau 27 : Plan de relations avec le fournisseur / vendeur

1. Introduction
1.1 Objectif
1.2 Portée
1.3 Définitions, acronymes, et abréviations
1.4 Références
2. Composants
[Inventaire des composants associés aux fournisseurs, vendeurs et entreprises fournissant du support technique.]
3. Nom du fournisseur ou vendeur
[Inclure une section pour chaque fournisseur ou vendeur.]
3.1 Composant critique
[Déterminer l'importance critique de chaque composant pour le système.]
3.2 Objectif de la relation
[Décrire les objectifs de la relation avec chaque vendeur ou fournisseur. Décrire à quel point le projet a besoin d'influencer le vendeur ou le fournisseur dans le moyen et long terme. Analyser la fréquence des nouvelles versions produites par le vendeur ou fournisseur]
3.3 Stratégie de la relation
[Décrire la stratégie du projet pour atteindre les objectifs décrits plus haut, y compris le financement et le degré d'implication avec le fournisseur / vendeur. Notamment la fréquence des rencontres avec le vendeur ou fournisseur, la participation aux organismes de normalisation ou aux groupes d'utilisateurs]

5.3.2 Composants logiciels

[LANG 1998] définit 3 types de composants logiciels qui sont utilisés pour le projet de migration au système cible. Le dernier type de composant logiciel retiendra plus particulièrement notre attention.

Tableau 28 : Types de livrables développés pour les projets de migration [LANG 1998]

Produits livrables	Description
Modifications définitives	Elles ne seront pas remises en cause lors des différentes étapes du projet.
Modifications transitoires	Elles sont réalisées pour une phase de projet. Dans une phase suivante elles devront être supprimées.
Utilitaires à utilisation ponctuelle	Programmes qui ne seront utilisés qu'une seule fois dans la vie du projet. Notamment les convertisseurs qui permettent de remplir le système cible avec des données patrimoniales

Modifications définitives

Les modifications définitives correspondent aux modules du ERP choisis, les interfaces entre le ERP et l'environnement. Tous ces composants ont déjà été examinés dans les chapitres précédents.

Modifications transitoires

Les modifications transitoires ne sont pas nécessairement des modifications rapides ou peu rigoureuses. Des modifications transitoires peuvent durer pendant de nombreuses années et nécessiter des efforts de développement très rigoureux.

Nous examinons dans la prochaine section les utilitaires à utilisation ponctuelle : les convertisseurs de données.

5.3.3 Convertisseur de données

Pour remplir le système cible avec les données du système patrimonial, sur demande, il est nécessaire de produire un convertisseur. Ce convertisseur ne servira qu'une seule fois et correspond à un utilitaire à utilisation ponctuelle tel que décrit dans la section 5.3.2. Le convertisseur peut se limiter à quelques simples fichiers de transactions séquentielles. Le développement du convertisseur monopolise moins de ressources que la définition des spécifications et la vérification et la validation de la conversion. Les données converties peuvent être testées par rapport aux données originales à l'aide de résumés ou de statistiques des données originales.

Une conversion de données peut produire des données qui ne sont pas rigoureusement identiques aux données originales. C'est-à-dire qu'une conversion inverse des nouvelles données ne reproduirait pas nécessairement les données originales. Pour illustrer ce thème, la conversion de données numériques dans le contexte du passage à l'euro en France, en 2002, sera examinée.

Pour [LANG 1998] la conversion requiert un document d'installation détaillé et sur mesure pour l'organisation qui décrit les opérations suivantes :

- Un archivage de sécurité
- Une conversion des données historiques
- Une sélection & traitement des données pour obtenir des résumés statistiques

- Un traitement des valeurs de référence, notamment des valeurs contractuelles et non contractuelles (Adaptation des valeurs de référence : donner aux articles des prix psychologiques que le consommateur reconnaît)
- Un traitement des documents : Traitement document par document (si des données sont seulement disponibles sur support papier).

L'analyse de [LANG 1998] s'applique à la conversion à l'euro dans les logiciels comptables et se préoccupe surtout de contrôler l'équilibre budgétaire après conversion et comment procéder à des ajustements en cas d'erreurs. Pour [LANG 1998], les données historiques archaïques, converties pour des motifs comptables, nécessitent moins de vérifications minutieuses et peuvent être réalisés indépendamment de la conversion principale.

L'analyse de [LANG 1998] apparaît moins rigoureuse que celle de [PANT 1999]. [PANT 1999] formule des recommandations quant aux capacités de traitement de la machine effectuant la conversion, de faire attention à l'espace mémoire et la performance des tables temporaires. La machine doit pouvoir supporter la manipulation de grands volumes de données et qui peut défragmenter les tables, réaliser des réindexations optimales, annuler des transactions erronées, et avoir un journal des opérations pour analyser les opérations.

[PANT 1999] propose un processus de conversion détaillé, dont chaque étape est itérative sauf l'étape 6.

Tableau 29 : Processus de conversion de données

Étape	Description
1	Écrire les spécifications pour les données cibles avec des messages d'erreur ou des avertissements.
2	Charger les données contenues dans le système patrimonial dans un outil temporaire
3	Lancer les vérifications
4	Convertir les données.
5	Chargement des données converties dans une BD de pré approbation. Étape de vérification et validation de la conversion
6	Chargement des données dans les tables définitives

Les spécifications du chargement des données provenant du système patrimonial peuvent inclure la vérification des types des données, la détection de la cohérence des données, des doublons, des valeurs vides, des valeurs incomplètes, de l'intervalle de valeurs autorisées. Les spécifications devraient définir comment choisir une seule donnée unique pour représenter la même information.

Les spécifications de la conversion des données doivent décrire les calculs de conversion ainsi que les valeurs à attribuer par défaut et les changements autorisés. En outre, la conversion peut inclure une réindexation des clés primaires des tables une fois que les vérifications des données fournies en entrée sont complétées. Le problème le plus fréquent qui peut résulter d'une conversion est que les données converties ne sont pas accessibles dans le nouvel environnement, et le problème le plus grave serait que les données soient réellement absentes.

5.3.3.1 Bases de données temporaires

Pour [PANT 1999], il est souhaitable de ne pas convertir directement du système patrimonial au système cible pour ne pas monopoliser les ressources ou interrompre le système patrimonial, surtout pour la manipulation de gros volumes de données. L'usage de bases de données temporaires contourne les contraintes d'intégrité sur les champs pour manipuler les données d'une manière opportuniste et simple. Dans une base de données temporaire on peut ajouter des champs de contrôle (des indicateurs), pour indiquer les liens entre les enregistrements et déterminer la séquence des conversions.

Une base de données de préapprobation, contenant une version du système cible, donne un outil aux administrateurs ou équipe de tests pour vérifier et valider le convertisseur. De plus la BD de préapprobation est un moyen d'organiser la formation aux employés. Dans cette BD les employés peuvent simuler les transactions sur le nouveau système avec les données de leur organisation. Pendant leur formation, les employés peuvent soulever des erreurs découvertes et vérifier et valider le convertisseur.

Il vaut mieux convertir par domaine d'affaires que par unité de l'organisation. La conversion de données selon les unités de l'organisation va fragmenter la conversion et

donc rendre la détection d'erreurs plus complexe, en effet la détection d'enregistrements en double ou les liens entre les enregistrements sont plus difficiles à détecter dans ce contexte.

5.3.3.2 Minimiser la conversion de données historiques

En examinant quelques projets de migration, un des objectifs recherchés pendant la conversion de données est d'effectuer une « conversion zéro » si possible. C'est-à-dire adopter au maximum les modules du logiciel sans données. Le principe étant qu'il faut bannir les informations inutiles et parmi les données qui sont absolument nécessaires, éviter de fournir des données inutiles ou incorrectes à l'entrée, sinon les résultats de la conversion le seront tout autant. Parmi les données qu'il faut absolument convertir, les problèmes de conversion les plus complexes sont les données concernant les transactions qui commencent dans une application et qui sont censées arriver à leur conclusion dans le nouvel environnement.

Une fois que les données essentielles sont ciblées, on doit cibler la période des données à convertir dans le temps (Exemple : les transactions comptables 5 dernières années). Des mesures statistiques pourraient déterminer la quantité de données historiques à convertir. Les données non converties pourraient soit être conservées dans leur environnement d'origine avec des accès spécifiques ou avec des copies sur support informatique pour des besoins ponctuels.

La conservation d'archives historiques répond à deux finalités : établir des comparaisons entre les années et la production d'informations dans le cadre d'un contrôle officiel. Par exemple un contrôle fiscal dans le cas d'applications comptables. Dans de nombreux pays il existe des dispositions réglementaires spécifiant que les archives comptables ne doivent pas être modifiées et doivent être gardées sur supports magnétiques pendant 3, 6 ou 10 ans.

5.3.3.3 Conversion de données numériques

L'erreur d'arrondi de conversion et de conversion inverse a été décrite dans le livre de Marc Langlois [LANG 1998] « L'euro et les systèmes d'information ». À cause de l'effet d'escalier ou « aliasing », les conversions et conversions inverses ne garantissent pas de retrouver les sommes initiales. Avec une erreur absolue de $\varepsilon_{x,y} \leq 5 \times 10^{-m}$, si une fonction

de conversion $f(x) = \frac{x}{n}$, respecte les critères suivants $n \in \mathfrak{R}, |n| > 1 + 10^{-m+1}$ alors la fonction est non injective. La précision insuffisante du résultat de la conversion implique qu'il $\exists y$ tel que $f(x) = y$ et $f(x \pm 10^{-m+1}) = y$. La conversion inverse $y \times n = x$ implique une marge d'erreur maximale de conversion de $\pm \frac{|n| \times 10^{-m+1}}{2}$.

En 2002, lors de l'adoption de l'euro en France, toutes les instances bancaires devaient employer l'euro comme monnaie de référence. Ceci impliquait une conversion des données financières en un seul format de données. Les libellés en francs et en euros n'étaient que l'affichage différent de la même valeur. Si un système est multi devise, le plus souvent, le noyau du système est en réalité mono devise et donc il faut changer ce noyau, lorsque l'application change de monnaie de référence. Pour plus de détails consulter l'annexe conversion du franc à l'euro.

5.3.3.4 Réorganisation et synchronisation

Lorsque la conversion est terminée, mais que les données ne sont toujours pas accessibles ou manipulables correctement, les transactions ne donnent pas les résultats escomptés dans l'environnement de production, il peut être nécessaire de procéder à une réorganisation des données.

En effet, il peut y avoir plusieurs manières de respecter des formats de données standard. Des données mal converties peuvent être inaccessibles ou difficilement manipulables dans la nouvelle application. Des innovations dans le domaine de l'indexation et des moteurs de recherche pourraient contribuer à des réorganisations pertinentes des données en fonction du forage de données. Le forage de données ("data mining") se fait à partir de plusieurs techniques d'apprentissage automatique. Toute réorganisation doit être analysée comme un coût associé à la conversion des données.

La synchronisation entre le système cible et le système patrimonial peut être un enjeu si le système cible doit fonctionner en même temps que sont réalisées la conversion et l'activation du système cible. Une synchronisation simple et opportuniste peut être faite en

convertissant toutes les données pendant une période de faible activité et réaliser une synchronisation des nouvelles données par la suite.

La synchronisation pose plus problème lorsqu'on doit réaliser la synchronisation de données entre plusieurs sites géographiquement distincts. En particulier, des spécifications formelles seraient nécessaires pour décrire comment synchroniser les données après qu'une première conversion ait eu lieu dans un site principal. Une application basée sur le modèle de la publication / abonnement aux changements des données pourrait être une piste de solution. Le modèle de la publication / abonnement permet à un site principal de garder un inventaire des sites secondaires qui requièrent une synchronisation et de publier un journal de toutes les transactions qui sont survenues. Les sites secondaires peuvent choisir de s'abonner ou d'annuler leur abonnement au site principal.

5.4 Conclusion

Le déploiement d'un progiciel de gestion est un effort de centralisation, qui doit écarter toute autre application concurrente, pourtant beaucoup de questions subsistent sur comment appréhender cette centralisation. La migration dans une organisation requiert des documents livrables standard pour la gestion des technologies de l'information, et pour le développement logiciel. Cependant d'autres types de documents livrables spécifiques sont nécessaires. La migration produit plusieurs composants logiciels parmi lesquels les convertisseurs de données retiennent d'avantage l'attention.

Conclusion générale

Les travaux de recherche, qui ont été menés, s'inscrivent dans le courant (tendance) de l'assurance qualité logicielle pour la migration de progiciels de gestion intégrés (ERP), des applications qui les soutiennent, de l'environnement cible et de l'organisation cible.

Ces recherches existent parce qu'une organisation qui compte plusieurs centaines de membres répartis en unités ne peuvent pas suivre la même démarche pour distribuer une application logicielle, qui est utilisée individuellement, que pour une ERP qui ne peut être utilisée que collectivement. À une époque où l'informatique est un outil de travail de masse, l'informatique est essentielle aux opérations quotidiennes. Une migration informatique, qui bouleverse le travail d'une majorité des membres de l'organisation, est donc souvent perçue comme une crise qui doit être gérée. Les échecs, parfois spectaculaires, de projets de migration, présentent l'image d'une industrie en manque de maturité. La conclusion s'impose : l'analyse informatique et la gestion de projets informatiques n'ont pas évolué suffisamment pour appréhender et résoudre les problèmes spécifiques posés par la migration des ERP commerciaux.

Ce mémoire démontre que la migration, l'intégration et l'adoption, par les usagers, d'une solution informatique dépendent de nombreux facteurs mais en premier lieu du point de départ : le système patrimonial. Le terme système patrimonial recouvre plusieurs réalités : notamment une application principale à remplacer, ainsi que les applications concurrentes, mais aussi les applications de l'entreprise qui vont demeurer actives après la migration. La recherche déclare que la portée des systèmes patrimoniaux est trop souvent négligée. Ce mémoire résume les modèles et processus à employer pour analyser les systèmes patrimoniaux.

Nous avons établi des catégories d'applications logicielles et tenté d'expliquer leur incidence sur les projets de modernisation. Les projets de modernisation ne sont pas en vase clos, ainsi nous avons examiné les différentes techniques de EAI pour intégrer les ERP. Ce mémoire s'efforce d'identifier les meilleures pratiques du domaine. Notamment, en énumérant les documents livrables les plus pertinents pour suivre et communiquer les

enjeux aux différents intervenants qui s'ajoutent aux documents livrables traditionnels de gestion de projet.

La modernisation d'un système patrimonial dans une organisation implique toujours, à des degrés différents, des développements internes et des acquisitions externes. La recherche atteste que toute solution retenue doit respecter les standards du domaine, pour profiter des avantages du marché. Malheureusement, pour de nombreuses organisations, le passage d'une culture de développement informatique à une culture d'acquisition informatique rencontre des résistances.

Dans ce mémoire, nous avons montré comment examiner le système patrimonial pour clairement appréhender l'application, qui doit disparaître, de l'environnement dans lequel le système cible doit s'intégrer, comment fixer des objectifs de migration réalistes, comment préparer les membres de l'organisation aux changements et comment profiter pleinement du cadre de l'organisation pour consolider l'adoption de l'application par les employés. Le domaine de l'intégration des applications de l'entreprise, réunit la modélisation de l'organisation et des contraintes techniques. Les objectifs d'un projet de migration ERP, dépendent de la modélisation des processus clés de l'organisation. Selon [HAMM 2001] il n'existe, même pour de grandes organisations moins d'une dizaine de processus clés.

À la suite de nos travaux nous pouvons établir que les enjeux principaux des projets de migration ne sont pas des enjeux techniques mais plutôt les enjeux de la réingénierie des processus d'affaires. En définitive, le succès peut être mesuré lorsque les processus d'affaires ont été changés conformément aux objectifs.

Dès le début de nos travaux nous avons été confronté à un problème, à savoir le manque de mesures, de résultats comparables, et de métriques. Tous les auteurs s'entendent pour énoncer que les tâches de modernisation ou de migration doivent être mesurées et que des métriques doivent être établies. Cependant, par leur nature même, les projets de migration vers des ERP sont peu fréquents et rarement identiques. L'examen des plans de migration et l'élaboration de recommandations, dans le domaine des ERP, sont des tâches immenses, tout d'abord parce qu'il est très ardu de comparer et connaître dans le détail beaucoup de projets de migration dans leur ensemble pour tirer des conclusions utiles. Par ailleurs,

seules les organisations publiques sont tenues de rendre disponibles les résultats de projets de migration. Les comptes rendus détaillés et complets de ce genre d'opérations sont souvent cachés au monde extérieur. Ce qui rend la validation de métriques difficile. Nous ne pouvons qu'espérer que les résultats positifs ou négatifs que connaissent les organisations, qui réalisent ces projets, soient diffusés.

La dichotomie entre facteurs stratégiques et tactiques développée par [HOLL 1999-a] a orienté notre approche. Nous avons suivi cette dichotomie pour sa simplicité et pour distinguer une priorité entre les risques au projet. Il demeure, dans la littérature, des nuances subtiles qui auraient intérêt à être diffusées auprès des informaticiens : notamment la concentration et la centralisation du pouvoir dans les organisations, et que ce sont les modules financiers d'un ERP qui doivent être déployés en priorité.

Ce mémoire n'aborde pas toutes les questions secondaires qui ne faisaient pas parti du cadre de l'étude mais qui ont un lien avec le sujet.

Une question d'intérêt est comment évaluer plus spécifiquement le lien entre un modèle de maturité de l'organisation et succès dans la migration. En effet, un outil de modélisation des systèmes patrimoniaux très embryonnaire était proposé par [LEWI 1998]. Pourtant, l'établissement d'un lien entre un modèle de maturité des systèmes patrimoniaux dans une organisation à de meilleures estimations des coûts du projet de migration serait utile.

Assurément, plus de recherches sont nécessaires pour organiser et gérer une migration multi site. L'article de [MARK 2000] affirme que les succès dans les migrations de ERP multi site sont l'exception et non la règle. La recherche ne fait pas état de beaucoup d'articles ou de consensus dans ce domaine.

Enfin une méthodologie plus rigoureuse pour mesurer les facteurs déterminant l'adoption collective d'une innovation par les membres d'une organisation est nécessaire. L'adoption par les usagers passe par la démonstration claire de la productivité de la nouvelle application pour l'organisation et pour l'individu. Pourtant, il n'existe pas de consensus sur la démonstration de la productivité de l'organisation et de l'individu avec un ERP. [ANDE 2003] et [POST 2000] parlent de paradoxe de la productivité.

Bibliographie

- [ALBE 2002], Albert Cecilia, Brownsword Lisa, Bentley David, Bono Thomas, Morris Edwin, and Pruitt Deborah , "Evolutionary Process for Integrating COTS-Based Systems (EPIC) Building, Fielding, and Supporting Commercial-off-the-Shelf (COTS) Based Solutions", (2002), CMU/SEI-2002-TR-005
- [ANDE 2003], Anderson Mark C., Banker Rajiv D. and Ravindran Sury, The New Productivity Paradox, ACM Press, *Communications of the ACM*, Vol. 46, No. 3 (2003) 91-94, ISSN : 0001-0782
- [BENN 2000], Bennett Keith , and Vaclav Rajlich, Software Maintenance and Evolution: A Roadmap, International Conference on Software Engineering, Copyright ACM Press, *Conference - Future of Software Engineering, Limerick Ireland* (2000) pp. 73-87, ISBN : 1-58113-253
- [BERG 1997], Bergey John, Linda M. Northrop and Dennis B. Smith, Enterprise Framework for the Disciplined Evolution of Legacy Systems, (1997), CMU/SEI-97-TR-007
- [BERG 1999], Bergey John, Dennis Smith, Scott Tilley, Nelson Weideman and Steven Woods, Why Reengineering Projects Fail, (1999), CMU/SEI-99-TR-010
- [BROW 1999], Brownsword Lisa, and Patrick Place, Lessons Learned Applying Commercial Off-the-Shelf Products, (1999), CMU/SEI-99-TN-015
- [CHIA 2001], Chiasson Mike W., and Chris Y. Lovato, Factors influencing the formation of a user's perceptions and use of a DSS Software Innovation, *ACM SIGMIS Database archive*, ACM Press; 'Special issue on adoption, diffusion, and infusion of IT' Vol. 32 , No. 3, (2001) 16-35, ISSN:0095-0033
- [CMMI V1.1, 2002] Capability Maturity Model Integration (CMMI), Version 1.1, (2002), CMU/SEI-2002-TR-029, ESC-TR-2002-029
- [ERPR 2004], ERP-PGI, en savoir plus, En ligne (2004), www.erprecut.com
- [GALL 2001], Gallivan Michael J., Organizational adoption and assimilation of complex technological innovations: development and application of a new framework, *ACM SIGMIS Database*, ACM Press; 'Special issue on adoption, diffusion, and infusion of IT' Vol. 32, No. 3, (2001) 51-85, ISSN: 0095-0033
- [HAMM 2001], Hammer Michael, and Champy James A., *Reengineering the corporation: a manifesto for business revolution*, New York: HarperBusiness, 2001, ISBN: 0066621127
- [HOLL 1999-a], Holland Christopher P. and Light Ben, A Critical Success Factors Model For ERP Implementation, *IEEE Software*, Vol. 16, No. 3, May-June (1999) 30-36, ISSN: 0740-7459
- [HOLL 1999-b], Holland Christopher P. and Light Ben, Global Enterprise Resource Planning Implementation, *Proceedings of the 32nd Hawaii International Conference on System Sciences* (1999) pp. 1-10
- [IEEE 1044.1-1995], IEEE 1044.1 - 1995, IEEE Guide to Classification for Software Anomalies, Institute of Electrical and Electronics Engineers, Inc. (1995), ISBN 1-55937-697-X
- [IEEE/EIA 12207.0-1996], ISO/IEC 12207, (ISO/IEC 12207) Standard for Information Technology—Software life cycle processes, Institute of Electrical and Electronics Engineers, Inc.(1996), ISBN 0-7381-0428-0, SS94581

- [ISO/IEC 9126-1:2001], Software engineering – Product quality – Part 1: Quality model, (2001)
- [JACO 1997], Jacobson Ivar, Martin Griss and Patrik Jonsson, *Software Reuse, Architecture Process and Organisation for Business Success*, Publisher: Addison-Wesley Pub Co; 1st edition, ISBN: 0201924765, 1997
- [JARR 2000], Jarrar Yasar F., Al-Mudirigh Abdullah and Zairi Mohamed, ERP Implementation critical success factors - the role and impact of business process management, *Software, IEEE*, Vol. 16 , No. 3, May-June (1999) pp. 30-36, ISSN: 0740-7459
- [JUNG 2003], Jung Ho-Won, and Goldenson Dennis R., CMM -Based Process Improvement and Schedule Deviation in Software Maintenance, (2003), CMU/SEI-2003-TN-015
- [KELL 1999], Kelly Sue, Nicola Gibson, Christopher P. Holland and Ben Light, Focus issue on legacy information systems and business process change: a business perspective of legacy information systems, *Communications of the AIS, Association for Information Systems*, Vol. 2 , No. 7 (1999)
- [KWAK 2002], Kwak Myungjae, Dongsoo Han and Jaeyong Shim, A Framework Supporting Dynamic Workflow Interoperation and Enterprise Application Integration, *Proceedings of the 35th Hawaii International Conference on System Sciences*, (2002) pp. 3798-3807, ISBN : 0-7695-1435-9/02
- [LANG 1998], Langlois Marc, et Renault Marie-Cécile, *L'euro et les systèmes d'information*, Dunod; Informatique et entreprise, ISBN: 2100037870, 1998
- [LEDR 2004], Ledroitpublic.com, L'état centralisé : centralisation et déconcentration, En ligne (2001), www.ledroitpublic.com
- [LEWI 1998], Lewis Ted, The Legacy Maturity Model, *Computer*, Vol. 31, No. 11 (1998) pp. 128-127, ISSN: 0018-9162
- [LOSA 2002], Losavio, Francisca, Ortega, Dinarle; and Perez, Maria, Modeling EAI [Enterprise Application Integration], *Proceedings. 12th International Conference of the Chilean Computer Science Society, 2002*, 6-8 November (2002) pp: 195-203, ISSN: 1522-4902
- [LYRE 2003], Lyrette Jacques, et Henry Jacques, Rapport Projet GIRES (Gestion intégrée des ressources), ADGA Group Consultants Inc, En ligne (2003), http://www.tresor.gouv.qc.ca/gires/rapport_gires.pdf
- [MARK 1994], Markosian Lawrence , Philip Newcomb, Russell Brand, Scott Burson and Ted Kitzmiller, Using an Enabling Technology to Reengineer Legacy Systems, *Communications of the ACM*, Periodical-Issue-Article, ACM Press New York, (1994) pp. 58-70, ISSN: 0001-0782
- [MARK 2000], Markus M. Lynne, Tanis Cornelis, and van Fenema Paul C., Enterprise resource planning: multisite ERP implementations, *Communications of the ACM*, ACM Press, Vol. 43, No. 4 (2000) 42 - 46, ISSN:0001-0782
- [MAYA 2003], Maya Daneva, "Lessons Learnt from Five Years of Experience in ERP Requirements Engineering", *Proceedings, 11th IEEE International Requirements Engineering Conference, 2003*, Sept (2003) pp. 45-54, ISSN: 1090-705X

- [MEHT 2001-a], Mehta Alok, and George T. Heineman, Evolving Legacy Systems Features using Regression Test Cases and Components, 2001, *International Conference on Software Engineering, Proceedings of the IWPSF 2001 Conference*, Publisher ACM Press; Vienna, Austria (2001) pp. 190-193, ISBN:1-58113-508-4
- [MEHT 2001-b], Mehta Alok, Evolving Legacy Systems Using Feature Engineering and CBSE, 2001, *International Conference on Software Engineering*, Publisher IEEE Computer Society (2001), ISBN ~ ISSN:0270-5257 , 0-7695-1050-7
- [MEHT 2002], Mehta Alok, and George T. Heineman, Evolving Legacy System Features into Fine-Grained Components, *International Conference on Software Engineering*, Publisher ACM Press; ICSE'02, May 19-25, 2002, Orlando, Florida, USA (2002) pp. 417-427, ISBN: 1-58113.-472-X
- [MEYE 1988], Meyer Bertrand, *Object-Oriented Software Construction*, Englewood Cliffs, N. J.: Prentice-Hall, ISBN: 0136290493, 1988
- [MEYE 2001], Meyers B. Craig, and Patricia Oberndorf, *Managing Software Acquisition: Open Systems and COTS Products*, Publisher: Addison Wesley Professional; 1st edition, ISBN: 0201704544, 2001
- [MOTS 2002], Motsko Michele, Patricia Oberndorf, Ellen-Jane Pairo and James Smith, Rules of Thumb for the Use of COTS Products, (2002), CMU/SEI-2002-TR-032
- [NICO 1995], Nicodemos Damianou, Naranker Dulay, Emil Lupu, and Morris Sloman, The Ponder Policy Specification Language, *Workshop on Policies for Distributed Systems and Networks, Bristol, UK, 29-31 Jan. 2001*, Springer-Verlag LNCS (1995) pp. 18-39
- [O'CA 1998], O'Callaghan Alan J., Adaptor : a pattern language for the reengineering of systems to object technology, *IEE Colloquium on Understanding Patterns and Their Application to Systems Engineering, Colloquium on IEEE*, Digest No. 1998/308 (1998) 1-6, INSPEC Accession Number: 5946094
- [O'CA 1999], O'Callaghan Alan J., Migrating large scale legacy systems to component-based and object technology: the evolution of a pattern language, *Communications of the AIS, Association for Information Systems*, Vol. 2, No. 3, Periodical-Issue-Article, (1999)
- [OBA 2001], Oba Michiko, and Norihisa Komoda, Multiple Type Workflow Model for Enterprise Application Integration, *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001*, (2001) pp. 2554 -2561, ISBN : 0-7695-0981-9/01
- [OBER 2003], Oberndorf Patricia A., David J. Carney, and Patrick R.H. Place, A Basis for an Assembly Process for COTS-Based Systems(APCS), (2003), CMU/SEI-2003-TR-010
- [PANT 1999], Pantano Paul J., Maximizing Accuracy in Large Data Conversions, *IT Professional*, Vol. 1, No. 5 (1999) 30 - 36, ISSN: 1520-9202
- [PARR 2000], Parr A. N. , and Shanks G. Dr., A Taxonomy of ERP Implementation Approaches, *Proceedings of the 33rd Hawaii International Conference on System Sciences - 2000*, Jan 4-7, (2000) pp. 2424 - 2433 , ISSN : 0-7695-0493-0

- [POST 2000], Poston Robin, and Grabski Severin, The impact of enterprise resource planning systems on firm performance, Association for Information Systems, *Proceedings of the twenty first international conference on Information systems*, Brisbane, Queensland, Australia (2000) pp. 479 - 493, ISBN : ICIS2000-X
- [RAE 1994], Rae Andrew K., *Software Evaluation for Certification : Principles Practice and Legal Liability*, Publisher: Books Britain, ISBN: 007709042X, 1994
- [RAND 1999], Randall Dave , John Hughes, Jon O'Brien, Tom Rodden, Mark Rouncefield, Ian Sommerville and Peter Tolmie, Focus issue on legacy information systems and business process engineering: banking on the old technology: understanding the organisational context of "Legacy" issues, *Communications of the AIS, Association for Information Systems*, Vol. 2 No. 8, (1999)
- [SCHE 2000], Scheer August-Wilhelm , and Habermann Frank, Making ERP a success, *Communications of the ACM*, Vol. 43, No. 4, April (2000) 57 - 61, ISSN:0001-0782
- [SEAC 2000], Seacord Robert, Santiago Comella-Dorda, Kurt Wallnau and John Robert, A Survey of Legacy System Modernization Approaches, (2000), CMU/SEI-2000-TN-003
- [SEAC 2001-b], Seacord Robert, Santiago Comella-Dorda, Grace A. Lewis, Pat Place and Dan Plakosh, Incremental modernization of Legacy Systems, (2001), CMU/SEI-2001-TN-006
- [SHET 1990], Sheth Amit P., and Larson James A., Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, *ACM Computing Surveys*, Vol. 22, No. 3, (1990) 183 - 236, ISSN : 0360-0300
- [STAL 2002], Staley Mary Jo, Oberndorf Patricia and Sledge Carol A., Using EVMS with COTS-Based Systems, (2002), CMU/SEI-2002-TR-022
- [STAN 1999], The Standish group international Inc., CHAOS : A recipe for success, Document de consultation, En ligne (1999), www.standishgroup.com/sample_research/PDFpages/chaos1999.pdf
- [TAIV 1995], Taivalsaari Antero, Roland Trauter and Eduardo Casais, Workshop on object-oriented legacy systems and software evolution, *Conference on Object Oriented Programming Systems Languages and Applications*, ACM Press, Series-Proceeding-Article (1995) pp. 180 - 185, ISBN: 0-89791-721-9
- [TCHO 2002], Tchokogué André, Benoît GRENIER et Claude R. DUGUAY, L'implantation d'un système ERP chez Pratt & Whitney Canada : de la réalité à la théorie, École des Hautes Études Commerciales., Cahier de recherche no 02-05 Avril (2002), ISSN : 1485-5496
- [THEM 2003], Themistocleous Marinos, and Zahir Irani, Towards a Novel Framework for the Assessment of Enterprise Application Integration Packages, *Proceedings of the 36th Hawaii International Conference on System Sciences, HICSS'03* (2003) pp. 234- 243, ISBN : 0-7695-1874-5/03
- [TRES 2004], Ministère du trésor, Le budget de dépenses 2003-2004 en bref, Gouvernement du Québec, Document de consultation, En ligne (2004), www.tresor.gouv.qc.ca/budget/pdf2003-04b/bref_franc.pdf
- [ULRI 2002], Ulrich William M., *Legacy Systems: Transformation Strategies*, Prentice Hall PTR; 1st edition, ISBN 013044927X, 2002
- [YAKO 2001], Yakovlev Ilya V., and Anderson Mark L., Lessons from an ERP implementation, *IT Professional*, Vol. 3, No. 4, July-Aug. (2001) 24 - 29, ISSN: 1520-9202

Bibliographie complémentaire

- [ELAZ 2003], El Azzouzi Ali, La structure de gouvernance comme facteur clé du succès d'une implantation ERP, Essai de maîtrise - Université Laval (2003), HF 91.5 UL 2003 E37
- [HILL 1999], Hill, Steve, Gestion de projet : première phase d'implantation d'un progiciel (PeopleSoft), Essai de maîtrise - Université Laval (1999), HF 91.5 UL 1999 H648
- [MORG 2003], Morgana, Muriel, Les points de contrôle pour limiter les risques d'échec d'un projet ERP? , Essai de maîtrise - Université Laval, (2003), HF 91.5 UL 2003 M849
- [SONI 2003], Sonia Mekadmi, ERP : réalité des implantations et conduite du changement : cas pratiques, Essai de maîtrise - Université Laval, (2003), HF 91.5 UL 2003 M516
- [TURG 2002], Turgeon, Mylène, La transformation de la fonction ressources humaines par les systèmes de gestion intégrés, Essai de maîtrise - Université Laval (2002), ISBN : 0612737268

Annexe : Conversion du franc à l'euro

Ce problème a été décrit dans le livre de Marc Langlois [LANG 1998] « L'euro et les systèmes d'information ». Lors de la transition à la monnaie unique européenne, en 2002, il fallait composer avec les différentes monnaies nationales des membres de l'union européenne. Lors de l'adoption de l'euro, il n'y avait pas de conversions massives entre les francs et les euros. Les valeurs en francs et euros n'étaient que l'expression de la même monnaie. Par ailleurs les conversions entre Francs et Deutschemarks (DM) ne se faisaient pas directement. Les conversions devaient passer par l'euro comme monnaie pivot : Francs → Euros → DM. Donc après cette date on faisait plus que des conversions multidevises. Pour mémoire rappelons le taux de conversion entre le franc et l'euro :

Euro * 6,40541 = Franc	$\frac{\text{Franc}}{6,40541} = \text{Euro}$
Entre 5 et 7 chiffres après la virgule possibles	Plus de 7 chiffres après la virgule

L'erreur absolue d'arrondi des devises est calculée d'après les trois chiffres après la virgule, et seuls deux chiffres sont affichés, donc le $\varepsilon \leq 5 \times 10^{-3}$. À cause de l'effet d'escalier ou « aliasing » les conversions, et conversions inverses ne garantissent pas de retrouver les sommes initiales, comme le démontre le tableau 30.

À cause de la valeur unitaire de l'euro qui est supérieure à celle du Franc :

- La conversion Euro → Franc → Euro ne génère pas d'erreur
- La conversion Franc → Euro → Franc génère des erreurs.

L'erreur de conversion de moins de 3 centimes de franc $\frac{0,01\text{euro} \times 6,40541}{2} \approx 0,03$, du à

l'effet d'escalier, peut se transmettre aux sommes de valeurs converties. Deux méthodes ont été employées pendant le passage à l'euro pour gérer les erreurs de conversion :

- Méthode A : L'erreur d'arrondi est ajoutée dans la ligne de l'addition au plus gros montant hormis la somme totale
- Méthode B : L'erreur d'arrondi est ajoutée à une nouvelle ligne pour régulariser le total

Tableau 30 : Effet d'escalier de l'arrondi de la conversion Franc – Euro – Franc

Franc	Conversion du Franc à l'euro	Conversion de l'euro au Franc
99,95	15,60	99,92
99,96	15,61	99,99
99,97	15,61	99,99
99,98	15,61	99,99
99,99	15,61	99,99
100,00	15,61	99,99
100,01	15,61	99,99
100,02	15,61	99,99
100,03	15,62	100,05

Index

- A**
- APCS 94
 - API..... 71
 - ARIS 88
 - ASAP..... 88, 89
 - Avantage relatif 18, 19, 82
- B**
- BICC Cables 98
 - Big Bang 98, 99
- C**
- CASE..... 32, 44
 - CMM 23, 57
 - CMMI 23
 - Cobol 42, 66
 - Conversion 104, 113
 - Convertisseur 110
 - CORBA 13
- D**
- Déploiement..... 100, 102
 - Document livrable 105
- E**
- EAI..... 24, 31, 39, 47, 116, 117
 - Concept 29
 - Dynamique..... 36
 - EAI ou ERP..... 31
 - Modèle Organisationnel..... 33
 - Modélisation technique 35
 - Nomenclature 36
 - Statique 36
 - Encapsulation..... 31
 - Euro 65, 125, 126
- F**
- FAR 73
 - Fasttrack..... 57
 - Fonction non injective 114
 - Formation..... 17, 65, 91, 104, 105
 - Fragmentation 26, 53, 55, 57
- G**
- Gires..... 64, 66, 67, 68, 78, 92, 99, 103
 - Go Live 100
- H**
- Hewlett Packard..... 90
- I**
- IEC..... 17, 20, 34
 - IEEE..... 17, 74, 78, 79
 - Insureco 84
- Intergiciel 13, 31, 39, 45
 - ISO 12, 17, 20, 34
- J**
- Java 13, 15, 42
 - Javadoc..... 15
- M**
- Microsoft..... 45, 90
 - Modifications transitoires 109, 110
 - Monsanto 90
- O**
- OMG 32, 35
 - Oracle..... 64
 - OSI..... 12
- P**
- PCI 81
 - PeopleSoft..... 64, 103
 - Pratt & Whitney 85, 86, 87
- R**
- Rational Unified Process..... 102, 105
 - Résistance au changement .. 19, 20, 56, 57, 83, 86, 96, 107, 117
 - RPC..... 32
- S**
- SAGIP 66
 - SEI 16, 23, 57
 - Sémantique..... 16, 34, 40
 - Sign-off 58
 - Spécification 72
 - Spirale 88
 - SPR 78
 - Standard 72
 - Stratégie 75, 89
 - SYGBEC..... 66
 - Système ouvert..... 35, 70, 74, 95
- T**
- Tactique 75
 - Threads..... 76
- U**
- UCITA 74
 - UML..... 38, 44
- V**
- Vanilla..... 92
- X**
- XML..... 35, 88