



UNIL | Université de Lausanne

Unicentre

CH-1015 Lausanne

<http://serval.unil.ch>

Year : 2015

EVOLIS : UN CADRE CONCEPTUEL POUR L'ÉTUDE DE L'ÉVOLUTION DES SYSTÈMES D'INFORMATION

Alexandre MÉTRAILLER

Alexandre MÉTRAILLER 2015 EVOLIS : UN CADRE CONCEPTUEL POUR L'ÉTUDE DE
L'ÉVOLUTION DES SYSTÈMES D'INFORMATION

Originally published at : Thesis, University of Lausanne

Posted at the University of Lausanne Open Archive <http://serval.unil.ch>
Document URN : urn:nbn:ch:serval-BIB_ODA98637D9362

Droits d'auteur

L'Université de Lausanne attire expressément l'attention des utilisateurs sur le fait que tous les documents publiés dans l'Archive SERVAL sont protégés par le droit d'auteur, conformément à la loi fédérale sur le droit d'auteur et les droits voisins (LDA). A ce titre, il est indispensable d'obtenir le consentement préalable de l'auteur et/ou de l'éditeur avant toute utilisation d'une oeuvre ou d'une partie d'une oeuvre ne relevant pas d'une utilisation à des fins personnelles au sens de la LDA (art. 19, al. 1 lettre a). A défaut, tout contrevenant s'expose aux sanctions prévues par cette loi. Nous déclinons toute responsabilité en la matière.

Copyright

The University of Lausanne expressly draws the attention of users to the fact that all documents published in the SERVAL Archive are protected by copyright in accordance with federal law on copyright and similar rights (LDA). Accordingly it is indispensable to obtain prior consent from the author and/or publisher before any use of a work or part of a work for purposes other than personal use within the meaning of LDA (art. 19, para. 1 letter a). Failure to do so will expose offenders to the sanctions laid down by this law. We accept no liability in this respect.



UNIL | Université de Lausanne

FACULTÉ DES HAUTES ÉTUDES COMMERCIALES
DÉPARTEMENT DE SYSTÈMES D'INFORMATION

**EVOLIS : UN CADRE CONCEPTUEL POUR L'ÉTUDE
DE L'ÉVOLUTION DES SYSTÈMES D'INFORMATION**

THÈSE DE DOCTORAT

présentée à la

Faculté des Hautes Etudes Commerciales
de l'Université de Lausanne

pour l'obtention du grade de
Docteur en Systèmes d'Information

par

Alexandre MÉTRAILLER

Directeur de thèse
Prof. Thibault Estier

Jury

Prof. Olivier Cadot, Président
Prof. Stephanie Missonier, experte interne
Prof. Yves Pigneur, expert interne
Prof. Jean-Henry Morin, expert externe
Prof. Michaël Petit, expert externe

LAUSANNE
2015



UNIL | Université de Lausanne

HEC Lausanne

Le Décanat

Bâtiment Internef

CH-1015 Lausanne

IMPRIMATUR

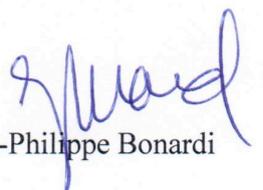
Sans se prononcer sur les opinions de l'auteur, la Faculté des Hautes Etudes Commerciales de l'Université de Lausanne autorise l'impression de la thèse de Monsieur Alexandre MÉTRAILLER, titulaire d'un master en Systèmes d'Information de l'Université de Lausanne, en vue de l'obtention du grade de docteur en Systèmes d'Information.

La thèse est intitulée :

EVOLIS : UN CADRE CONCEPTUEL POUR L'ÉTUDE DE L'ÉVOLUTION DES SYSTÈMES D'INFORMATION

Lausanne, le 8 octobre 2015

Le doyen



Jean-Philippe Bonardi

HEC Lausanne

Le Décanat

Tél. ++41 21 692 33 40 | Fax ++41 21 692 33 05

www.hec.unil.ch | hecdoyen@unil.ch



Jury

Docteur Thibault Estier

Maître d'Enseignement et de Recherche à la Faculté des Hautes Études Commerciales de l'Université de Lausanne
Directeur de thèse

Professeur Jean-Henry Morin

Professeur à l'Institut de Science de Service Informationnel de l'Université de Genève
Expert externe

Professeur Michaël Petit

Professeur à la Faculté d'Informatique de l'Université de Namur
Expert externe

Professeure Stéphanie Missonier

Professeure à la Faculté des Hautes Études Commerciales de l'Université de Lausanne
Experte interne

Professeur Yves Pigneur

Professeur à la Faculté des Hautes Études Commerciales de l'Université de Lausanne
Expert interne

University of Lausanne
Faculty of Business and Economics

Doctorate in Information Systems

I hereby certify that I have examined the doctoral thesis of

Alexandre METRAILLER

and have found it to meet the requirements for a doctoral thesis.

All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature : Th. Estier Date : 29 septembre 2015

Prof. Thibault ESTIER
Thesis supervisor

University of Lausanne
Faculty of Business and Economics

Doctorate in Information Systems

I hereby certify that I have examined the doctoral thesis of

Alexandre METRAILLER

and have found it to meet the requirements for a doctoral thesis.

All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature :  _____ Date : 1.10.2015

Prof. Jean-Henry MORIN
External member of the doctoral committee

University of Lausanne
Faculty of Business and Economics

Doctorate in Information Systems

I hereby certify that I have examined the doctoral thesis of

Alexandre METRAILLER

and have found it to meet the requirements for a doctoral thesis.

All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature :



Date :

October 5th 2015

Prof. Michaël PETIT
External member of the doctoral committee

University of Lausanne
Faculty of Business and Economics

Doctorate in Information Systems

I hereby certify that I have examined the doctoral thesis of

Alexandre METRAILLER

and have found it to meet the requirements for a doctoral thesis.

All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature :  Date : 5 octobre 2015

Prof. Stéphanie MISSONIER
Internal member of the doctoral committee

University of Lausanne
Faculty of Business and Economics

Doctorate in Information Systems

I hereby certify that I have examined the doctoral thesis of

Alexandre METRAILLER

and have found it to meet the requirements for a doctoral thesis.

All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature :  Date : 26 septembre 2015

Prof. Yves PIGNEUR
Internal member of the doctoral committee

Résumé

English version on the back

Dans cette thèse, nous étudions les évolutions des systèmes d'information. Nous nous intéressons plus particulièrement à l'étude des facteurs déclencheurs d'évolution, ce qu'ils représentent et comment ils permettent d'en apprendre d'avantage sur le cycle de vie des systèmes d'information.

Pour ce faire, nous avons développé un cadre conceptuel pour l'étude des évolutions qui tient compte non seulement des facteurs déclencheurs d'évolution, mais également de la nature des activités entreprises pour évoluer. Nous avons suivi une approche Design Science pour la conception de ce cadre conceptuel. Selon cette approche, nous avons développé itérativement le cadre conceptuel en l'instanciant puis en l'évaluant afin de raffiner sa conception. Ceci nous a permis de faire plusieurs contributions tant pratiques que théoriques.

La première contribution théorique de cette recherche est l'identification de 4 facteurs principaux déclenchant les évolutions. Ces facteurs sont des éléments issus de domaines généralement étudiés séparément. Le cadre conceptuel les rassemble dans un même outil pour l'étude des évolutions. Une autre contribution théorique est l'étude du cycle de vie des systèmes selon ces facteurs. En effet, l'utilisation répétée du cadre conceptuel pour la qualification des évolutions met en lumière les principales motivations des évolutions lors de chaque étape du cycle de vie. En comparant les évolutions de plusieurs systèmes, il devient possible de mettre en évidence des modèles spécifiques d'évolution des systèmes.

Concernant les contributions pratiques, la principale concerne le pilotage de l'évolution. Pour un gestionnaire de système d'information, l'application du cadre conceptuel permet de connaître précisément l'allocation réelle des ressources pour une évolution ainsi que la localisation du système dans son cycle de vie. Le cadre conceptuel peut donc aider les gestionnaires dans la planification et la stratégie d'évolution du système. Les modèles d'évolution, identifiés suite à l'application du cadre conceptuel, sont également une aide précieuse pour définir la stratégie de pilotage et les activités à entreprendre lors de la planification des évolutions.

Finalement, le cadre conceptuel a fourni les bases nécessaires à l'élaboration d'un tableau de bord pour le suivi du cycle de vie et le pilotage de l'évolution des systèmes d'information.

Abstract

Version française au recto.

In this thesis, we study the evolution of information systems. We particularly focus on the study of evolution triggers, what they represent and how they can be useful in the understanding of information systems lifecycles.

To achieve this, we developed a framework to study information systems evolutions that takes into account not only the factors triggering evolutions, but also the nature of activities undertaken to evolve. We followed a design science research approach to build our framework. According to this approach, we iteratively developed the framework by instantiating, evaluating and refining its design. This allowed us to make several contributions, both practical and theoretical.

The first theoretical contribution of this research is the identification of four main factors triggering evolutions. These factors are elements from domains that are usually addressed separately. The framework brings them together in a single tool for the study of evolution. Another theoretical contribution is the study of the information systems lifecycle according to these factors. Indeed, the repeated use of the framework for qualifying evolution highlights the main drivers of evolution during each stage of the lifecycle. By comparing evolutions of several systems, it becomes possible to identify specific evolution patterns.

Regarding practical contributions, the main one concerns the strategic management of evolution. For an information system manager, the use of the framework allows him to precisely know the allocation of the resources for every evolution and to determine the position of the system in its lifecycle. The framework can therefore help managers in planning and designing the strategy of information systems evolution. Specific patterns of evolution that are identified by the repeated use of the framework are also a great help to design the strategy of evolution and the activities to be undertaken when planning evolutions.

Finally, the framework provided the basis for the development of a dashboard to monitor information systems lifecycle and to manage the evolution of information systems.

Remerciements

Avant de commencer, j'aimerais écrire quelques lignes pour remercier chaleureusement les personnes qui m'ont soutenu dans l'accomplissement de cette thèse.

En premier lieu, je remercie vivement mon directeur de thèse, le Docteur Thibault Estier, pour avoir supervisé ce travail doctoral. Merci pour tous les excellents moments que nous avons partagés autant durant nos discussions concernant ce travail de recherche et les multiples activités d'enseignement ; que durant nos diverses discussions sur les turbines hydroélectriques, la navigation, les histoires de valaisans, etc. Un grand Merci à toi !

Un grand merci également aux membres du Jury qui ont pris le temps de relire attentivement ma thèse : la Professeure Stéphanie Missonier, le Professeur Yves Pigneur, le Professeur Jean-Henry Morin et le Professeur Michaël Petit. Leurs remarques constructives ont permis de grandement améliorer ce manuscrit.

Merci aux personnes avec qui j'ai pu collaborer lors de toutes mes magnifiques expériences d'enseignement, je pense évidemment à Thibault Estier et aussi tout particulièrement à Gabor Maksay.

Merci également à tous mes collègues et amis grâce à qui j'ai bénéficié d'un environnement de travail d'exception. Je pense tout particulièrement à l'équipe d'OA the fu** : Ulysse Rosselet, François Vessaz, Boris Fritscher et Samuel Bendahan ; sans oublier Thomas Boillat, Bertil Chapuis, Adrian Holzer, Fabio Daolio, Emmanuel Fernandes, Riccardo Bonazzi, Zhan Liu et toutes les autres personnes que j'ai eu la chance de rencontrer durant les pauses cafés, les repas à la Banane, etc.

Merci également aux membres de l'ISI que j'ai eu le plaisir de côtoyer et qui n'ont pas encore été cités : Prof Christine Legner, Prof Benoît Garbinato, Prof Marco Tomassini, Elisabeth Fournier et ceux que j'oublie.

Merci à mes parents qui m'ont toujours soutenu dans mes études que ce soit moralement et financièrement.

Je ne peux terminer ces remerciements sans une ligne pour ma compagne Rafaële qui depuis plus de 6 magnifiques années m'encourage sans relâche.

Table des matières

Chapitre 1	Introduction	1
1.1	Contexte de recherche	1
1.2	Questions de recherche	2
1.3	Cadre théorique	3
1.3.1	Evolution du logiciel	3
1.3.2	Processus d'évolution	3
1.3.3	Caractéristiques des évolutions	4
1.3.4	Facteurs déclencheurs d'évolution	4
1.4	Méthodologie de recherche	4
1.5	Contributions	5
1.6	Structure de la thèse	6
Chapitre 2	Motivations	7
2.1	Travaux précédents : déploiement incrémental d'ERP	7
2.2	Eléments déclencheurs d'évolution	8
2.3	Etude des évolutions	9
2.4	Cycle de vie des systèmes d'information	9
2.5	Un tableau de bord pour le pilotage de l'évolution	10
Chapitre 3	Recherche Design Science	11
3.1	Fondements design science	11
3.2	Critères d'une recherche design science	12
3.2.1	Les 3 boucles d'une recherche design science en systèmes d'information	12
3.2.2	Les critères clés d'une recherche design science en systèmes d'information	13
3.3	Différentes formes des contributions d'une recherche design science	14
3.4	Evaluation d'une recherche design science	14
3.5	Méthodologie de recherche design science	15
3.6	Notre processus de recherche design science	16
3.6.1	Identification du problème et motivation	17
3.6.2	Définition des objectifs de la solution	17
3.6.3	Design et conception	18
3.6.4	Démonstration	18
3.6.5	Evaluation	18
3.6.6	Communication	18
Chapitre 4	Environnement de recherche : Évolution Logicielle	19
4.1	Description du phénomène d'évolution	19

4.1.1	Interprétation dans le contexte logiciel	20
4.1.2	Définition du terme évolution dans le domaine logiciel	20
4.1.3	Plus loin que l'évolution : la transformation	21
4.1.4	L'évolution logicielle : activité et phénomène	22
4.2	Typologie de systèmes logiciels	23
4.2.1	Classification des programmes	23
4.3	Lois de l'évolution logicielle	25
4.4	Maintenance logicielle	27
4.5	Maintenance versus évolution	28
4.6	Le processus d'évolution	29
4.6.1	Macroprocessus	29
4.6.2	Microprocessus	30
4.6.3	Typologie des activités du processus d'évolution	32
4.7	Facteurs déclencheurs d'évolution	33
4.7.1	Le contexte d'utilisation	34
4.7.2	Les parties prenantes	34
4.7.3	Le contexte technique	35
4.7.4	L'architecture logicielle	35
4.8	Facteurs déclencheurs d'évolution dans EVOLIS	36
4.9	Notre apport à l'étude de l'évolution logicielle	37
Chapitre 5 Design du cadre conceptuel pour l'étude de l'évolution EVOLIS		41
5.1	Présentation des contextes	42
5.1.1	Présentation du contexte métier	42
5.1.2	Présentation du contexte académique	43
5.2	Besoins couverts par le cadre conceptuel EVOLIS	44
5.2.1	Chronologie des évolutions	44
5.2.2	Facteurs déclencheurs d'évolution (motivations)	44
5.2.3	Typologie des activités d'évolution (comment)	45
5.2.4	Importance de l'évolution	45
5.3	Facteurs d'évolution d'EVOLIS	46
5.3.1	Alignement métier et système d'information	47
5.3.2	Satisfaction des utilisateurs	51
5.3.3	Architecture	54
5.3.4	Technologie	59
5.4	Coût	62
Chapitre 6 Méthode d'étude de l'évolution du cadre conceptuel EVOLIS		63

6.1	Identification des différentes évolutions	65
6.2	Caractériser une évolution (taxonomie)	65
6.2.1	Propriétés temporelles (Quand).....	66
6.2.2	Objet du changement (Où).....	66
6.2.3	Propriétés du système modifié (Quoi).....	67
6.2.4	Support (Comment).....	67
6.3	Caractériser les motivations d'évolution.....	68
6.4	Caractériser les activités au sein des évolutions	68
6.4.1	Description des différents types.....	69
6.4.2	Typage des activités.....	70
6.4.3	Caractériser le but des activités.....	71
6.5	Exemple de scénario d'évolution	72
6.6	Modèle conceptuel d'EVOLIS	75
6.7	Issues de l'application	76
Chapitre 7 Mise en œuvre d'EVOLIS : Etude de cas 1.....		79
7.1	Etude de cas	79
7.1.1	Rôle de cette première étude de cas.....	80
7.1.2	Question de recherche	80
7.1.3	Unité d'analyse	81
7.1.4	Cas individuel ou multiple	81
7.1.5	Sélection des sites	82
7.1.6	Méthode de collecte des données	83
7.1.7	Description des données	83
7.2	Analyse des données.....	85
7.3	Résultats.....	87
7.4	Discussion.....	91
7.5	Apports de l'étude de cas vis-à-vis du processus de design.....	92
7.5.1	Instanciation d'EVOLIS.....	93
7.5.2	Démonstration d'EVOLIS	93
7.5.3	Evaluation et itérations sur le design et le développement d'EVOLIS.....	93
7.5.4	Communication	94
7.6	Conclusions.....	94
Chapitre 8 Mise en œuvre d'EVOLIS : Etude de cas 2.....		97
8.1	Etude de cas	98
8.1.1	Rôle de cette seconde étude de cas	98
8.1.2	Question de recherche	98

8.1.3	Unité d'analyse	99
8.1.4	Cas individuel ou multiple	100
8.1.5	Sélection des sites	100
8.1.6	Méthode de collecte des données	101
8.1.7	Description des données	102
8.2	Analyse des données	104
8.2.1	Objets de l'étude	105
8.2.2	Extraction des données	107
8.2.3	Modèles conceptuels de données	109
8.3	Résultats	112
8.3.1	Projet d'informatisation Dophin	113
8.3.2	Application Soarian	120
8.3.3	Application Axya	129
8.4	Discussion	140
8.5	Apports de l'étude de cas vis-à-vis du processus de design	141
8.5.1	Instanciation d'EVOLIS	141
8.5.2	Démonstration d' EVOLIS	142
8.5.3	Evaluation et itérations sur le design et le développement d'EVOLIS	142
8.6	Conclusions	143
Chapitre 9 Tableau de bord pour le pilotage de l'évolution		147
9.1	Fondations	148
9.1.1	Audience	148
9.1.2	But du tableau de bord pour l'audience	149
9.1.3	Quel type de tableau de bord	150
9.2	Prérequis	151
9.2.1	Application d'EVOLIS	151
9.2.2	Accès aux données	152
9.3	Structure	152
9.3.1	Forme	152
9.3.2	Structure	154
9.3.3	Principes de design	155
9.3.4	Fonctionnalités	155
9.4	Prototype	156
9.4.1	Page d'accueil pour le pilotage	156
9.4.2	Page de visualisation des cycles de vie selon EVOLIS	158
9.5	Apports du tableau de bord vis-à-vis du processus de recherche	161

9.6 Conclusions.....	161
Chapitre 10 Discussion et conclusions	163
10.1 Résumé de la recherche	163
10.2 Enseignements	164
10.3 Comparaisons avec les recherches du domaine.....	164
10.4 Importance théorique	165
10.4.1 Facteurs déclencheurs d'évolution.....	165
10.4.2 Etude des évolutions	165
10.4.3 Cadre conceptuel pour l'étude de l'évolution	165
10.5 Importance pratiques.....	166
10.5.1 Utilisation pour étudier des traces d'évolution	166
10.5.2 Un tableau de bord pour le pilotage de l'évolution	166
10.6 Limitations.....	166
10.7 Travaux futurs	167
10.8 Conclusions.....	167

Chapitre 1

Introduction

1.1 Contexte de recherche

La plupart des systèmes d'information utilisés dans les organisations humaines résultent d'une construction collective dans la durée de grands ensembles de connaissances et de données. Ils servent de support à de nombreux processus organisationnels et techniques sur de longues périodes de temps. Si les méthodes et processus de développement de ces systèmes ont fait l'objet de beaucoup de recherches et de contributions académiques, l'évolution de ces systèmes, c'est-à-dire l'ensemble de modifications successives qu'ils supportent pendant leur cycle de vie, ainsi que les facteurs qui conditionnent cette évolution ont été relativement peu étudiés et observés dans la littérature. Par « cycle de vie » nous entendons l'historique des évolutions rencontrées par le système.

Toutefois, l'évolution des systèmes d'information est un phénomène inévitable pour tous les systèmes utilisés afin de répondre à un problème ou une activité dans le monde réel. En effet, de par leur utilisation, ces systèmes deviennent une partie intégrante des domaines dans lesquels ils opèrent. Ces domaines étant toujours plus dynamiques, l'évolution des systèmes est incontournable pour que ces derniers ne perdent pas leur utilité et qu'ils restent satisfaisants. Afin que cette activité d'évolution des systèmes s'opère de la manière la plus efficiente et la plus efficace possible, le processus d'évolution doit être contrôlé, planifié et supporté en s'appuyant sur des méthodes, des procédures et des indicateurs. Pour encadrer ce processus d'évolution, il est important de le comprendre et de s'intéresser aux motivations qui poussent les organisations à faire évoluer leurs systèmes.

La littérature actuelle sur l'évolution des systèmes d'information ne propose pas de modèle pour étudier de manière longitudinale les motivations du phénomène d'évolution. En effet, l'étude des motivations de l'organisation à faire évoluer ses systèmes ou celle des facteurs capables de déclencher des évolutions semble ne pas avoir été une priorité pour le domaine. De plus, les différents enjeux du processus d'évolution sont généralement étudiés de manière isolée. Le but de notre recherche est de regrouper les connaissances actuelles afin de créer un outil capable

d'étudier les évolutions de systèmes d'information en les caractérisant de manière systématique tout en tenant compte des motivations d'évolution. Cet outil doit non seulement être utile aux chercheurs, mais également aux gestionnaires pour leur tâche de pilotage de l'évolution du système d'information.

1.2 Questions de recherche

Le but de cette thèse est de créer un outil permettant d'étudier et de mieux comprendre l'évolution des systèmes d'information. La première question de recherche à laquelle répond cette thèse concerne l'identification des connaissances nécessaires à cette compréhension ainsi que leur intégration dans cet outil. Elle se formule ainsi :

« Quelles notions doit englober un outil permettant de mieux comprendre le processus d'évolution des systèmes d'information? »

Cette première question est d'ordre général, les notions que doit englober cet outil dépendront du contexte dans lequel il en est fait usage. Le premier contexte qui découle de cette question est le contexte académique avec le domaine d'étude de l'évolution des systèmes d'information. Le second contexte qui peut découler de cette question est le contexte métier des personnes en charge de faire évoluer les systèmes d'information. Par processus d'évolution nous entendons les différentes étapes d'évolution par lesquelles le système est passé avant d'arriver dans son état actuel, son cycle de vie. Pour tenir compte des besoins de ce second contexte, la seconde question de recherche complétant la première se formule ainsi :

« Que doit inclure cet outil afin d'aider les gestionnaires à mieux comprendre et piloter l'évolution des systèmes d'information ? ».

Le processus de conception de cet outil nous a poussés à répondre à des sous-questions de recherche. En effet, les questions formulées ci dessus peuvent se décomposer en sous-questions concernant d'une part l'application de cet outil et d'autre part la manière dont l'outil peut répondre aux questions principales de recherche.

Ces sous-questions de recherche ont été traitées lors de deux études de cas.

Dans la première étude de cas, nous nous sommes concentrés sur l'application de l'outil ainsi que sur la manière selon laquelle l'outil permet d'apporter des éléments de réponse pour les questions de recherche principales.

Concernant l'application de l'outil, il est essentiel que le processus d'application et surtout que les résultats soient similaires lorsqu'une tierce personne met en pratique l'outil. De ce constat, il faut répondre à la question « comment ? » concernant l'application de cet outil. La première sous-question de recherche adressée par la première étude de cas se formule ainsi :

Première sous-question de recherche : « Comment cet outil peut-il aider à étudier l'évolution des systèmes d'information ? »

En répondant à cette question, nous produisons une nouvelle contribution que doit fournir l'outil : une méthodologie systématique d'étude de l'évolution.

Concernant la manière dont l'outil peut répondre aux questions principales, il faut que l'outil aide les gestionnaires et les chercheurs à la compréhension du phénomène d'évolution. Un premier élément permettant de mieux comprendre le processus d'évolution serait d'aider à révéler des modèles d'évolution. La seconde sous-question de recherche adressée par la première étude de cas se formule ainsi :

Deuxième sous-question de recherche : « Comment cet outil peut-il aider à révéler des modèles (ou des schémas) spécifiques d'évolution dans des traces d'évolution ? »

Pour répondre aux deux sous-questions précédentes nous avons conduit une étude de cas portant sur les évolutions de systèmes d'information de taille relativement contenue.

Dans la seconde étude de cas, nous avons naturellement réutilisé les connaissances acquises lors de la première étude de cas afin de les appliquer sur un plus grand volume de données d'évolution. Les réponses aux deux sous-questions de recherche précédentes ont été confirmées et nous avons répondu à deux autres sous-questions de recherche :

Troisième sous-question de recherche : « Comment un grand volume de données concernant l'évolution des systèmes d'information peut nous permettre d'améliorer le design de l'outil ? »

La réponse à cette troisième sous-question de recherche nécessite que l'outil soit doté de polyvalence. En effet, il doit pouvoir être applicable à toutes les tailles de systèmes d'information, que ces derniers soient de petite ou de grande ampleur.

Pour répondre à cette troisième sous-question de recherche, nous avons dû ajouter des éléments aux notions que doit englober l'outil afin de permettre l'étude d'une grande quantité de données d'évolution. Cette grande diversité de cas d'évolution a également permis d'améliorer les notions de caractérisation des évolutions.

La volonté de polyvalence de l'outil, conduit à une autre sous-question de recherche concernant l'aide procurée par l'outil aux gestionnaires et aux chercheurs quant à la compréhension du phénomène d'évolution. La quatrième sous-question de recherche adressée par la seconde étude de cas se formule ainsi :

Quatrième sous-question de recherche : « Comment cet outil peut-il aider à révéler des modèles (ou des schémas) spécifiques d'évolution dans un grand volume de données concernant l'évolution des systèmes d'information ? »

1.3 Cadre théorique

1.3.1 Evolution du logiciel

Cette étude puise ses fondements dans les domaines de recherches relatifs à l'évolution du logiciel. Pour situer cette thèse parmi les domaines de recherche connexes, nous pouvons citer les recherches sur l'évolution logicielle générale avec notamment les travaux du professeur Lehman à savoir : les Lois de l'évolution (M M Lehman & Ramil, 2001) et la typologie des logiciels (Cook, Harrison, Lehman, & Wernick, 2006). Les travaux de recherche du professeur Lehman ont apporté une meilleure compréhension de l'évolution logicielle en identifiant un ensemble de comportements que suivent ou que devraient suivre les systèmes logiciels évoluant. Ces recherches sont développées plus amplement dans le Chapitre 4 et plus particulièrement dans les Sections 4.2 et 4.3.

1.3.2 Processus d'évolution

Afin d'aborder l'évolution des systèmes d'information de manière longitudinale, cette recherche s'appuie également sur les travaux relatifs au processus d'évolution ainsi qu'au cycle de vie des logiciels (Bennett & Rajlich, 2000b). Toujours dans ce domaine de l'étude du processus d'évolution, nous intégrons également des notions concernant la dynamique de l'évolution avec les travaux de Sabherwal & Hirschheim (2001). La notion de cycle de vie des systèmes est incontournable lorsque l'on étudie la gestion des évolutions des systèmes d'information. La vie

du système passe par plusieurs étapes caractéristiques du cycle dans lequel se situe la vie du système. Le processus d'évolution et la notion de dynamique de l'évolution sont détaillés dans les Sections 4.6 et 4.1.

1.3.3 Caractéristiques des évolutions

Afin d'étudier les évolutions logicielles en tant que telles, nous utilisons la taxonomie des évolutions proposée par Buckley, Mens, Zenger, Rashid, & Kniesel (2005). Nous basons également notre étude sur les recherches adressant les modifications apportées aux logiciels avec les travaux relatifs aux activités de maintenance et d'évolution logicielle avec les travaux initiaux de Lientz, Swanson, & Tompkins (1978) ainsi que ceux de Chapin, Hale, Khan, Ramil, & Tan (2001). Ces éléments sont détaillés dans les Sections 4.6.3 et 6.4 pour les activités d'évolution et 6.2 pour la taxonomie des évolutions.

1.3.4 Facteurs déclencheurs d'évolution

L'étude des facteurs déclencheurs d'évolution se base sur les raisons évoquées par M M Lehman & Ramil (2001) dans les lois de l'évolution ainsi que dans la typologie des logiciels et les différentes extensions du modèle original (Cook et al., 2006). Cette étude des facteurs déclencheurs nous a conduits à nous intéresser aux domaines de l'alignement entre métier et système d'information, de l'architecture du système d'information, de la satisfaction des parties prenantes à l'utilisation du système ainsi qu'au domaine abordant les enjeux de la gestion technologique du système d'information. Ces éléments sont détaillés dans le Chapitre 5.

1.4 Méthodologie de recherche

L'évolution des systèmes d'information peut être étudiée selon deux approches différentes. Dans la première approche, l'évolution logicielle est étudiée en tant que phénomène alors que dans la seconde, l'évolution logicielle est étudiée comme étant une activité qui doit être exécutée, dirigée et contrôlée. Ces deux approches d'étude de l'évolution logicielle sont pertinentes, importantes et surtout complémentaires (M M Lehman & Ramil, 2003; Madhavji, Lehman, Perry, Ramil, & Fern, 2006). Avec cette recherche, nous voulons intégrer les deux approches de l'étude de l'évolution.

La nature du problème que nous essayons de résoudre dans cette recherche nous a conduits naturellement à suivre le paradigme « Design Science en systèmes d'information ». En effet, d'une part l'étude de l'évolution des systèmes d'information et d'autre part le pilotage de l'évolution des systèmes d'information nécessite la sélection d'un paradigme de recherche permettant de réduire l'écart entre les connaissances théoriques (étude de l'évolution) et les connaissances pratiques des professionnels (pilotage de l'évolution du système d'information). Cette recherche contribue donc à répondre aux deux missions des recherches design en systèmes d'information : produire des contributions théoriques rigoureuses et assister les praticiens dans leurs problèmes métiers.

Cette recherche s'inscrit donc dans le domaine de l'évolution logicielle en intégrant les deux principales interprétations du terme évolution en étudiant l'évolution comme un phénomène et comme une activité qui doit être pilotée.

Selon A. R. Hevner, March, Park, & Ram (2004), une recherche Design Science « crée et évalue des artefacts IT destinés à résoudre des problèmes organisationnels identifiés ». Une recherche Design Science en système d'information suit un processus itératif pour la création de l'artefact

de recherche. En effet, comme illustré dans la Figure 1, le processus de création peut être représenté comme une boucle entre de la conception et de l'évaluation de l'artefact. Nous sommes passés par plusieurs itérations afin de définir notre artefact final. Les publications ainsi que les études de cas nous ont permis de justifier les choix effectués, d'évaluer et d'améliorer le design de ce dernier.

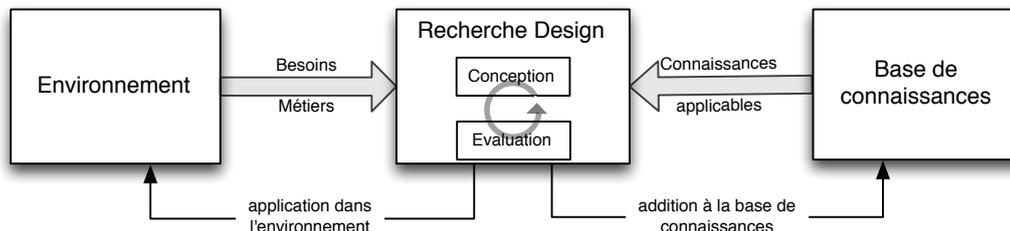


Figure 1. Cadre conceptuel des recherches design en systèmes d'information (adapté de Hevner et al. (2004)).

L'artefact que nous avons développé prend la forme d'un outil qui contient les ressources nécessaires (méthodologie systématique d'étude, caractérisation des évolutions selon différents facteurs, etc.) pour l'étude des évolutions de systèmes d'information.

Afin d'identifier les besoins de l'environnement, nous nous sommes basés sur nos précédents travaux concernant le déploiement de systèmes d'information et avons effectué une étude de la littérature du domaine de l'évolution logicielle. Ces besoins nous ont permis de définir les objectifs ainsi que les grandes lignes de notre artefact. Afin d'évaluer la pertinence de la recherche et de justifier les choix pour notre artefact, nous l'avons présenté à d'autres chercheurs du domaine.

A partir de là, le processus de design itératif a débuté avec de multiples itérations sur le contenu de l'artefact. Les deux grandes itérations prennent la forme de deux études de cas. Une première étude de cas sur un échantillon de données réduit afin d'évaluer l'applicabilité de l'artefact ainsi que la méthodologie d'étude de l'évolution qu'il contient (Chapitre 7). Une seconde étude de cas sur des données de grande taille pour démontrer et évaluer l'artefact (Chapitre 8). Lors de chacune de ces études de cas, nous avons corrigé les lacunes ainsi que les problèmes de conception de notre artefact. Ces multiples évaluations et affinements nous ont permis de formuler les contributions de l'artefact à la fois pour l'environnement et pour la base de connaissances du domaine.

1.5 Contributions

Cette recherche présente un outil pour analyser de manière systématique les évolutions des systèmes d'information en intégrant des notions qui sont généralement adressées séparément. Cet outil est un cadre conceptuel que nous avons nommé EVOLIS. EVOLIS est l'acronyme de EVOLution of Information Systems.

Pour un chercheur, l'application d'EVOLIS permet d'étudier les évolutions des systèmes d'information en se basant non seulement sur les motivations des évolutions mais également sur les autres éléments présents dans EVOLIS. Il peut aussi comparer les évolutions de plusieurs systèmes afin de mettre en évidence des tendances d'évolution selon le cycle de vie des systèmes.

Pour un gestionnaire de système d'information, l'application d'EVOLIS permet de connaître précisément l'allocation réelle des ressources pour une évolution. Une utilisation répétée d'EVOLIS sur les évolutions d'un système d'information ajoute la dimension temporelle et

permet de retracer le cycle de vie du système d'information. Cet élément peut aider les gestionnaires dans la planification et la stratégie d'évolution du système.

Avec le même objectif, aider les gestionnaires dans la planification et la stratégie d'évolution de leur système d'information, EVOLIS fournit les bases nécessaires à l'élaboration d'un tableau de bord pour le suivi du cycle de vie et le pilotage de l'évolution des systèmes d'information.

1.6 Structure de la thèse

Poursuivant avec le Chapitre 2, ce dernier présente les motivations qui nous ont poussés à entreprendre cette recherche ainsi que la justification de l'utilité que revêt cette recherche dans le domaine de l'évolution des systèmes d'information.

Le Chapitre 3 détaille la méthodologie utilisée pour atteindre nos objectifs de recherche. Après une introduction sur les différents points clés de la recherche design science en systèmes d'information, nous présentons notre processus de recherche.

Le Chapitre 4 présente le domaine académique dans lequel cette recherche se positionne. Cette revue de littérature permet de cerner le phénomène d'évolution des systèmes logiciels, les études faites jusqu'à présent pour typer les différents systèmes logiciels en fonction de leur utilité et naturellement le type de système étudié. Il contient la présentation des processus d'évolution ainsi que la revue des différents types d'activité de maintenance et d'évolution. Il se termine avec les éléments nouveaux sur l'étude des déclencheurs d'évolution, la démarche par laquelle nous sommes arrivés à ces éléments et le gap que nous désirons combler avec cette recherche.

Les Chapitre 5 et Chapitre 6 présentent le résultat de nos itérations de recherche. Le Chapitre 5 présente le point de départ de la recherche avec les populations susceptibles d'utiliser l'outil. Il détaille également ces nouveaux éléments concernant l'étude des déclencheurs d'évolution ainsi que ce que nous utilisons pour évaluer l'ampleur d'une évolution. Le Chapitre 6 présente la suite du contenu de l'outil, c'est-à-dire la méthodologie d'étude de l'évolution contenue dans de ce dernier.

Le Chapitre 7 est la première démonstration de l'application de l'outil, cette démonstration prend la forme d'une étude de cas qui nous a permis d'itérer à de multiples reprises sur le design de l'outil et d'évaluer l'adéquation des choix effectués en fonction des objectifs.

Le Chapitre 8 est la seconde étude de cas et application de l'outil dans un contexte réel. Il présente les éléments nouveaux à inclure à l'outil pour améliorer sa polyvalence ainsi que les enrichissements du design de l'outil. Cette seconde étude de cas a permis d'évaluer à nouveau le design de l'outil.

Le Chapitre 9 présente une démarche pour concevoir un tableau de bord de pilotage qui organise les informations générées par EVOLIS afin de les mettre à dispositions des praticiens pour leur tâche de pilotage de l'évolution.

Le Chapitre 10 discute de la recherche en présentant les contributions pour les domaines académique et pratique.

Chapitre 2

Motivations

Ce chapitre présente le cheminement par lequel nous nous sommes intéressés spécifiquement au sujet de cette recherche. Il présente aussi les éléments du domaine de recherche que nous souhaitons approfondir et dans quel but.

2.1 Travaux précédents : déploiement incrémental d'ERP

Nos précédents travaux concernaient le déploiement d'ERP (progiciel de gestion intégré) dans les petites et moyennes entreprises (PME). En se basant sur le fait que l'approche de déploiement classique d'ERP n'est que peu adaptée à la structure et aux besoins spécifiques des PME (Alleman, 2002; Equey, Kusters, Varone, & Montandon, 2008; Equey, 2006; Welsh & White, 1981), nous avons créé une méthodologie agile et incrémentale de déploiement d'ERP destinée aux PME (Métrailler & Estier, 2009).

Cette méthodologie ainsi que les outils logiciels de support de cette dernière ont été développés en partenariat avec des entreprises de services informatiques spécialisées dans le déploiement d'ERP en PME. La structure de cette méthodologie était basée sur les approches de développement Agile (Beck et al., 2001; Shore & Warden, 2007), les méthodologies traditionnelles de déploiement d'ERP (Ahituv, Neumann, & Zviran, 2002; Al-Mudimigh, Zairi, & Al-Mashari, 2001; Umble, Haft, & Umble, 2003) ainsi que sur l'expérience des praticiens des entreprises partenaires de ce projet. Elle a été présentée à la communauté des développeurs et des personnes en charge de l'implémentation d'un ERP open source. Elle a fait l'objet d'un accueil enthousiaste et a été utilisée pour plusieurs projets de déploiement d'ERP. Elle est devenue la méthodologie officielle de déploiement d'ERP pour les entreprises partenaires de ce projet.

Le processus de cette méthodologie agile de déploiement peut se résumer de la manière suivante : Un premier paquet de fonctionnalités de base de l'ERP étaient installées lors d'un

premier déploiement et par la suite, selon les besoins et les moyens de la PME la couverture fonctionnelle de l'ERP était complétée de manière incrémentale.

Cette approche de déploiement d'ERP, pousse les PME à utiliser un système d'information en évolution continue le long du processus incrémental de déploiement de l'ERP dans leur structure.

Afin de mieux comprendre ce processus incrémental d'évolution auquel étaient confrontés les systèmes d'information des entreprises clientes, nous nous sommes tournés vers le domaine de l'évolution logicielle et vers les théories, méthodes et construits permettant d'analyser ce processus. En effet une meilleure compréhension de ce phénomène aurait permis d'aiguiller d'avantage les professionnels du domaine sur la nature des éléments déclenchant des évolutions afin d'inclure ce paramètre dans le processus incrémental de déploiement des systèmes d'information.

2.2 Éléments déclencheurs d'évolution

Afin de comprendre la nature des raisons qui poussent les organisations à faire évoluer leur système d'information, nous nous sommes tournés vers la littérature académique du domaine de l'évolution pour trouver un début de réponse à la question suivante : De quelle nature est la raison qui déclenche une évolution ? Est-ce un enjeu métier, technologique, etc. qui occasionne une évolution ?

Le premier domaine de recherche que nous avons étudié pour déterminer les critères permettant de définir les raisons déclencheurs d'évolution est celui qui adresse le phénomène d'évolution logicielle dans sa globalité. Les travaux d'un groupe de chercheurs se distinguent dès le début des années 70 avec notamment la classification des programmes (SPE) (Cook et al., 2006; Cook, Ji, Harrison, Cook, & Ji, 2000; M.M. Lehman, 1980) et les lois de l'évolution logicielle (M M Lehman & Ramil, 2001) décrits dans le Chapitre 4.

Ce domaine contient les éléments qui expliquent pourquoi un système évolue et quelles sont les propriétés communes auxquelles sont soumis ces derniers lors de leurs évolutions. En observant les domaines connexes concernant la description des évolutions, ces domaines abordent les évolutions en tant que telles, mais pas les facteurs déclenchant ces dernières.

En effet, l'étude des motivations ou des facteurs déclencheurs d'évolution semble ne pas avoir été une priorité pour le domaine, tout comme leur influence dans le processus d'évolution tout au long de la vie des systèmes (Godfrey & German, 2008). Les nombreuses publications sur le domaine se focalisent sur l'aspect technique, les défis que représente l'évolution du logiciel plutôt que sur la dimension managériale de l'évolution du système d'information dans son ensemble. Selon Mens et al. (2005) les principaux défis du domaine portent sur l'amélioration de la qualité des logiciels au fil des évolutions, sur la coévolution des systèmes, sur les processus d'implémentation des systèmes, et ainsi de suite.

Des plus, ces différents enjeux du processus d'évolution sont généralement étudiés de manière individuelle. Le but de notre recherche est de regrouper les connaissances actuelles de ce domaine afin de créer un outil capable d'étudier les évolutions des systèmes d'information en les caractérisant de manière systématique tout en tenant compte des motivations d'évolution. Cet outil doit non seulement être utile aux chercheurs, mais également aux gestionnaires pour leur tâche de pilotage de l'évolution du système d'information.

Nos recherches concernant les éléments déclencheurs d'évolution se fondent sur les travaux effectués dans le domaine de l'évolution logicielle. L'analyse des recherches de ce domaine nous

a permis d'isoler la nature de différents facteurs déclenchant les évolutions que nous avons approfondi afin de les inclure dans la création d'un outil pour l'étude des éléments déclencheurs d'évolution.

2.3 Etude des évolutions

La littérature académique est riche pour ce qui concerne le processus de caractérisation d'une évolution et des différentes activités qui la composent. Néanmoins ces différents éléments de caractérisation des évolutions sont généralement adressés et développés séparément.

Parmi les études des évolutions que nous avons retenues, la taxonomie des évolutions proposée par Buckley et al. (2005) permet d'organiser, de classifier et de positionner les évolutions selon les dimensions essentielles qui sont : les propriétés temporelles, l'objet du changement, la manière dont le changement est effectué et les propriétés du système qui ont été modifiées. Nous avons également retenu les recherches adressant la typologie des activités de modification des logiciels avec les travaux relatifs à la maintenance logicielle de Lientz et al. (1978) et de Chapin et al. (2001).

La mise en commun de certains de ces éléments dans une même étude ou un même outil permettra de visualiser selon plusieurs approches les différents aspects des évolutions et de les étudier de manière plus complète.

2.4 Cycle de vie des systèmes d'information

Dès l'instant où un système logiciel est utilisé pour résoudre un problème dans un environnement dynamique, il n'est pas possible d'établir un ensemble complet et immuable de spécifications pour le système. L'évolution sera un résultat fondamental de l'utilisation de ce système dans son environnement, il faut donc la gérer ainsi et ne pas la considérer comme un effet de bord indésirable. Lorsque l'on mentionne la gestion des évolutions des systèmes d'information, la notion de cycle de vie des systèmes est incontournable. En effet, le système passe par plusieurs phases lors de sa vie. Dans la vision conventionnelle de la vie d'un logiciel, ce dernier est développé, puis il est déployé pour des utilisateurs, et ensuite il entre dans sa phase de maintenance. Bennett & Rajlich (2000) proposent une approche différente qui partitionne la phase conventionnelle de maintenance en plusieurs étapes qui détaillent le cycle de vie d'une manière plus utile et constructive (voir Section 4.6.1).

En effectuant une étude longitudinale basée sur les motivations des évolutions, il devient possible de déterminer quels sont les déclencheurs potentiels d'évolution lors des différentes étapes du cycle de vie du logiciel. De même, cette étude peut permettre d'aider les gestionnaires à déterminer et à anticiper dans quelle étape de son cycle de vie le système se situe. Bennett & Rajlich (2000) précisent que les transitions d'une étape à une autre peuvent être une décision délibérée ou se produire par contrainte. Ce second type de transition doit être évité par les gestionnaires du système car il engendre de lourdes conséquences métier. Cette étude longitudinale permet au responsable d'identifier les étapes du cycle de vie de chaque composant de son système d'information, et par la même occasion permet de planifier et de piloter les transitions d'une phase à une autre. C'est ce travail de contrôle et planification qui est appelé *pilotage de l'évolution* dans la suite de ce document.

2.5 Un tableau de bord pour le pilotage de l'évolution

Le rassemblement des connaissances sur les éléments déclencheurs d'évolution ainsi que des techniques et des approches de l'étude des évolutions dans un même outil aidera non seulement les chercheurs à mieux comprendre le processus d'évolution des systèmes d'information, mais également les gestionnaires pour le pilotage de l'évolution de leurs systèmes.

Ce rassemblement dans un même outil fournit les bases pour la conception d'un tableau de bord pour le pilotage de l'évolution des systèmes d'information. En effet, l'application longitudinale de ces éléments à l'ensemble des évolutions effectuées sur un système d'information fournit les éléments nécessaires pour retracer le cycle de vie des systèmes. Ce cycle de vie ainsi que les informations sur le déroulement des évolutions fournissent les indications nécessaires pour faciliter la planification et le pilotage des évolutions futures du système.

Chapitre 3

Recherche Design Science

La nature du problème que nous essayons de résoudre dans cette thèse nous a conduits naturellement à suivre le paradigme de recherche « Design Science en systèmes d'information ».

En effet, d'une part l'étude de l'évolution des systèmes d'information et d'autre part le pilotage de l'évolution des systèmes d'information nécessite la sélection d'un paradigme de recherche permettant de réduire l'écart entre les connaissances théoriques (étude de l'évolution) et les connaissances pratiques des professionnels (pilotage de l'évolution du système d'information). Cette recherche design contribue donc à répondre à la finalité des activités de recherche décrites par Hevner et al. (2004) à savoir : produire des contributions théoriques rigoureuses et assister les praticiens dans leurs problèmes métiers.

3.1 Fondements design science

La majorité des travaux dans le paradigme du design science en systèmes d'information trouvent leur fondement dans l'ouvrage *The Sciences of the Artificial* (Simon, 1996). Dans cet ouvrage, Simon établit une distinction claire entre les sciences naturelles (natural science) et les sciences de l'artificiel (science of the artificial) connues également sous le terme de design science. Simon décrit les sciences naturelles comme étant les connaissances concernant certaines classes de choses (objets ou phénomènes) qui décrivent et expliquent comment ces choses se comportent et interagissent mutuellement. A l'opposé, Simon décrit les sciences de l'artificiel comme étant les connaissances concernant des objets et des phénomènes artificiels (fait par l'homme) conçus afin d'atteindre un but spécifique.

La recherche en systèmes d'information s'intéresse à l'étude d'objets complexes et artificiels que sont les systèmes d'information. Ces objets artificiels, créés en vue d'une finalité précise génèrent les phénomènes étudiés par les chercheurs en systèmes d'information. La recherche en systèmes d'information s'intéresse donc à des phénomènes artificiels qui sont à la fois créés et étudiés par l'homme. De ce constat, March & Smith (1995) relèvent la dualité des recherches

en systèmes d'information et le fait qu'il ne faut pas opposer les deux approches (naturelles et artificielle) mais au contraire les traiter dans leur complémentarité. Sur ces bases, March et Smith proposent que les chercheurs en systèmes d'information contribuent à chacune de ces deux activités. Le domaine de recherche design en systèmes d'information se retrouve donc avec deux types légitimes d'intérêts scientifiques : 1. une approche science naturelle et sociale visant à comprendre la réalité et 2. une approche scientifique du design visant à créer des artefacts servant les objectifs des individus et des organisations.

L'objectif des chercheurs en design science consiste en effet à optimiser les performances des systèmes d'information en créant et en évaluant des artefacts qui s'appuient sur les connaissances des chercheurs afin d'améliorer la performance d'une situation ou de résoudre des problèmes organisationnels (Hevner et al., 2004).

3.2 Critères d'une recherche design science

Hevner et al. dans leur article fondateur « Design Science in Information Systems Research » (Hevner et al., 2004) décrivent la performance des recherches design science en systèmes d'information en proposant un modèle conceptuel permettant de définir les contours d'une « bonne » recherche en design science. Cet article met en évidence les sept critères clés afin de comprendre, exécuter et évaluer les recherches design science en systèmes d'information. De plus leur modèle conceptuel met en évidence trois boucles d'activités inhérentes à ce type de recherche à savoir la boucle de relevance (pertinence), la boucle de design et la boucle de rigueur (Figure 2) (Hevner, 2007).

3.2.1 Les 3 boucles d'une recherche design science en systèmes d'information

Avec comme objectif de répondre aux problèmes métiers des praticiens, la boucle de relevance trouve son enracinement dans l'environnement sociotechnique duquel est issu le phénomène d'intérêt. De cet environnement composé de personnes, d'organisations et de technologies sont issus les problèmes et les opportunités du terrain ainsi que les critères permettant de valider les résultats de la recherche adressant ce problème. Les recherches achevées sont donc évaluées en fonction de leur utilité vis-à-vis des problèmes ou des opportunités rencontrés. Si les résultats ne sont pas satisfaisants, une nouvelle itération de cette boucle est possible et ce, jusqu'à atteindre un résultat conforme aux critères de validité initialement émis.

Au cœur de l'approche design science, se trouve la boucle de design. Cette boucle met en œuvre les activités nécessaires au processus de création à savoir le développement et la conception de l'artefact ainsi que la justification et l'évaluation permettant d'établir si l'artefact conçu répond aux besoins identifiés dans l'environnement. Pour Hevner et al. (2004), ces artefacts doivent répondre à des problèmes non résolus ou fournir des solutions là où celles existantes ne sont pas pleinement satisfaisantes. L'artefact créé n'est pas l'unique point d'intérêt, les recherches design doivent s'intéresser à la fois au processus de design et à l'artefact créé car les deux se complètent. Le but d'une recherche design science est l'utilité, c'est à dire la capacité à répondre au problème initial.

La boucle de rigueur repose sur la capacité du chercheur à sélectionner à l'intérieur de « la base de connaissances » du domaine et à appliquer les fondements théoriques ainsi que les méthodologies appropriées pour la conception et l'évaluation de son artefact. La rigueur est de mise non seulement pour le processus de conception de l'artefact mais également pour l'évaluation des résultats de la recherche.

Tout comme la recherche de l'utilité est l'achèvement de la pertinence de l'artefact, l'extension de la base de connaissances à l'issue de la recherche est l'aboutissement de la boucle de rigueur.

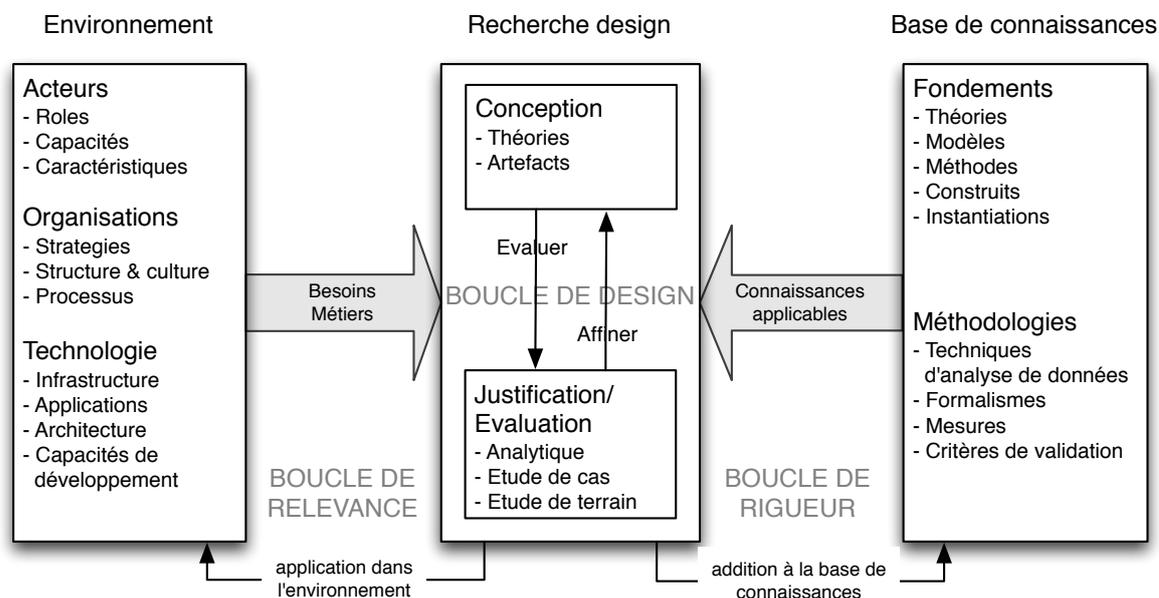


Figure 2. Boucles d'activités des recherches design (adaptés de Hevner et al. (2004)).

3.2.2 Les critères clés d'une recherche design science en systèmes d'information

Afin de supporter les chercheurs, réviseurs, éditeurs et lecteurs dans l'exécution, l'évaluation et la compréhension d'une recherche design science, Hevner et al. (2004) mettent en évidence sept critères clés d'une « bonne » recherche (« Guidelines for Design Science Research »).

Premièrement, par définition une recherche design science doit produire un artefact (Guideline 1) adressant un problème spécifique pertinent dans son environnement (Guideline 2). Cet artefact doit être utile, ses qualités ainsi que son efficacité doivent être rigoureusement évalués (Guideline 3). Une recherche design science doit produire une contribution claire et vérifiable soit dans l'artefact lui-même soit dans son processus de création et d'évaluation (Guideline 4). Les chercheurs doivent faire preuve de rigueur au cours du processus de conception et d'évaluation. Ceci nécessite de s'appuyer sur la base de connaissances disponibles afin de sélectionner les théories et méthodologies spécifiques à la création et à l'évaluation de l'artefact (Guideline 5). La mise en œuvre de ce processus de création / évaluation doit être itérative afin d'affiner l'artefact dans le but de produire une solution effective au problème (Guideline 6). Finalement, le résultat de la recherche doit être communiqué aussi bien aux audiences techniques que managériales. Ceci sous-entend de fournir suffisamment de détails quant à sa mise en œuvre dans son contexte ou de permettre à une organisation de déterminer si l'utilisation de cet artefact peut leur être bénéfique (Guideline 7).

Hevner et al. (2004) mentionnent également que ces guidelines ne sont pas juste une check-list à suivre étape par étape, mais elles doivent être utilisées consciencieusement selon les particularités de la recherche entreprise.

3.3 Différentes formes des contributions d'une recherche design science

Une partie de la communauté design science et pas des moindres avec notamment Hevner et al. (2004); livari (2007) et Hevner (2007) semble adopter une vision plutôt étroite de ce que doit être le résultat d'une recherche design science. En effet, ils semblent exclure acteurs et organisation en suggérant que le résultat doit être un artefact informatique technique (au sens IT artefact). Cependant selon March & Smith (1995), le résultat d'une recherche design science peut également être un construit, un modèle ou une méthode. Au final, il n'y a pas de réel consensus au sein de la communauté sur la nature des résultats attendus d'une recherche design science.

La vision que nous partageons de la contribution d'une recherche design science est celle présentée par March & Smith (1995) et également Carlsson (2007). Selon ce dernier, un système d'information peut être perçu comme un système sociotechnique. En effet, le lien est devenu tellement étroit entre les technologies de l'information et l'environnement métier que l'on ne peut plus traiter séparément technologie et métier, processus, relations intra et inter-organisationnelles. Cette vision est appuyée par Smith & McKeen (2006) qui relèvent que les problèmes systèmes d'information / technologies de l'information sont d'avantage managériaux que techniques. Actuellement pour certaines organisations, le déploiement d'une solution ERP (progiciel de gestion intégré) n'est plus perçu comme un projet technique, mais d'avantage comme un projet de réorganisation. Toujours selon Carlsson (2007), la recherche en systèmes d'information devrait développer des connaissances pour cet environnement sociotechnique et pas juste développer des artefacts technologique et des théories design autour de ces artefacts.

Plus précisément, Carlsson suggère que *« le but d'une recherche design science en systèmes d'information est de développer des connaissances pratique pour le design et la réalisation d'initiatives en systèmes d'information ou pour l'amélioration de la performance des systèmes d'information existants. Par initiative en systèmes d'information, nous entendons le design et l'implémentation d'interventions dans des systèmes sociotechnique où les systèmes d'information (incluant les artefacts informatiques) sont des moyens essentiels pour atteindre les résultats souhaités de l'intervention. De ce fait, la recherche design en systèmes d'information doit inclure organisation, personnes, systèmes d'information et artefacts informatiques. »* (Carlsson, 2007, p.79).

3.4 Evaluation d'une recherche design science

L'évaluation constitue un élément indispensable d'une recherche design science. Comme mentionné précédemment, March & Smith (1995) intègrent l'évaluation comme une des six étape du processus de recherche design science, au même titre que l'étape de design et conception de l'artefact. Hevner et al. (2004) mettent également l'accent sur l'évaluation comme étape cruciale de la recherche et indiquent de s'appuyer sur des méthodes d'évaluation rigoureuse (Guideline 3). Sur ce point Hevner et al. (2004) proposent cinq grands types de méthodes d'évaluation : observations, méthodes analytique, expérimentations, tests, méthodes descriptive. Ils précisent également que l'évaluation fait partie intégrante du processus itératif de design. Elle permet de recueillir du feedback afin d'améliorer la conception de l'artefact.

Pries-Heje, Baskerville, & Venable (2008) proposent une matrice stratégique pour l'évaluation des recherches design science. Cette matrice est formulée dans le but d'aider les chercheurs dans la sélection des stratégies d'évaluation afin de parvenir à d'avantage de rigueur dans leurs recherches design science.

Cette matrice (présentée dans le Tableau 1) fait la distinction entre évaluation *ex ante* et évaluation *ex post*. Une évaluation *ex ante* a lieu avant que l'artefact soit construit et une évaluation *ex post* à lieu après la construction de l'artefact. Le chercheur doit donc effectuer un choix quant au moment de l'évaluation. Comme en design science l'étape d'évaluation ne se résume pas à une seule activité exécutée uniquement à l'issue du cycle « design - construction - évaluation », la matrice propose deux moments opportuns pour effectuer des évaluation à savoir après la phase de design (de conception : *ex ante*) et après la phase de construction de l'artefact (*ex post*).

Cette matrice distingue également les modes d'évaluation. Deux modes sont mis en évidence : naturaliste et artificiel. Le mode d'évaluation naturaliste regroupe les types d'évaluation dans un contexte réel (par exemple les études de cas) et le mode artificiel qui regroupe les expérimentations en laboratoire (par exemple les simulations).

Le chercheur doit donc se poser trois questions :

- Quand l'évaluation doit être effectuée ? Ex ante ou ex post.
- Que faut-il évaluer ? Le processus de design ou l'artefact.
- Comment évaluer ? Choisir entre le mode naturaliste ou artificiel.

Tableau 1. Matrice stratégique d'évaluation (Strategic Evaluation Framework) adaptée de Pries-Heje et al. (2008).

	Ex Ante	Ex Post
Naturaliste	Processus de design / Produit (artefact)	Processus de design / Produit (artefact)
Artificiel	Processus de design / Produit (artefact)	Processus de design / Produit (artefact)

3.5 Méthodologie de recherche design science

Afin de conduire cette recherche, nous utilisons la méthodologie de recherche design science proposée dans l'article « Design Science Research Methodology (DSRM) » de Peffers, Tuunanen, Rothenberger & Chatterjee (2007). Le but de cette méthodologie de recherche est de fournir les principes, pratiques et procédures permettant aux chercheurs de conduire des recherches design science effectives.

Dans cet article, Peffers et al. (2007) détaillent en six étapes les deux processus permettant la mise en œuvre d'une recherche design science à savoir concevoir et évaluer (March & Smith, 1995). Ces six étapes sont : (1) Identification du problème et motivation, (2) définition des objectifs de la solution, (3) design et conception, (4) démonstration, (5) évaluation, (6) communication. La Figure 3 présente ces six étapes ainsi que les différents points d'entrée possibles de la recherche.

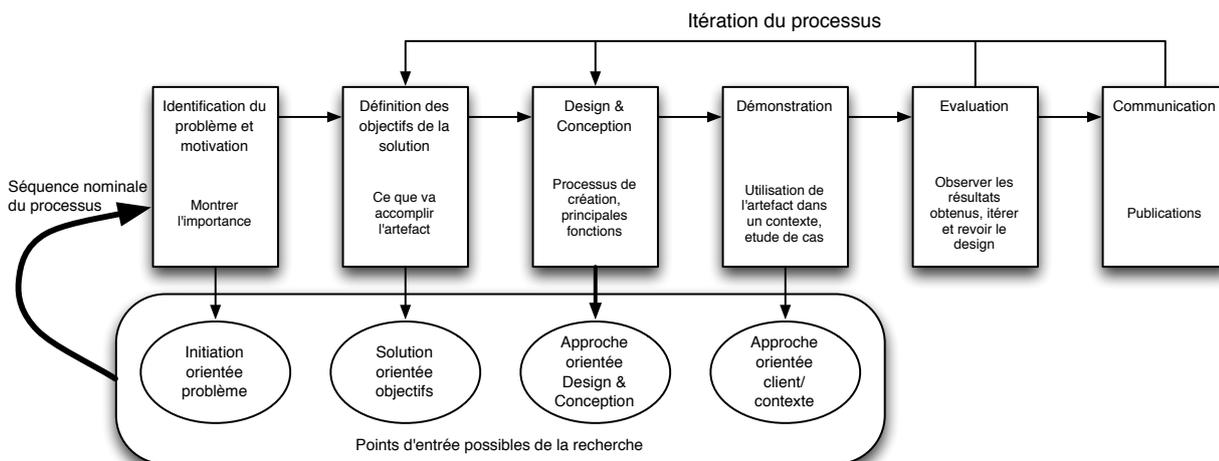


Figure 3. Processus de recherche Design Science selon Peffers et al. (2007).

Le séquençement nominal des étapes du processus de recherche est une suggestion de déroulement de la recherche. Les auteurs ne prévoient pas que les chercheurs procèdent toujours dans l'ordre séquentiel présenté. Une approche orientée problème suit le séquençement nominal et commence par l'activité « Identification du problème et motivation ». Les chercheurs pourraient choisir de procéder ainsi si l'initiation de la recherche est liée à l'observation d'un problème ou d'une recherche antérieure suggérant une opportunité de recherche. Une recherche orientée objectifs commence par la seconde activité : « Définition des objectifs de la solution ». Ce séquençement pourrait être déclenché par un besoin de l'industrie ou un besoin de la recherche elle-même qui nécessiterait l'élaboration d'un artefact et ainsi de suite pour les autres points d'entrée.

3.6 Notre processus de recherche design science

Selon la méthodologie de recherche de Peffers et al. (2007), nous avons effectué un processus de recherche initié par l'étape deux, c'est à dire le développement d'une « solution orientée objectifs ». En effet, nous avons comme objectif de développer un artefact qui soit pertinent pour l'étude du phénomène d'évolution et également pertinent pour accompagner les gestionnaires de systèmes d'information dans la compréhension de l'évolution de leur système. Notre processus de recherche est donc orienté sur les objectifs finaux que doit adresser notre artefact. La Figure 4 présente le processus de recherche selon Peffers et al. (2007) appliqué à notre recherche. Le point d'entrée de la recherche à savoir « Solution orientée objectifs » est mis en évidence et les différentes étapes sont détaillées dans les sections suivantes.

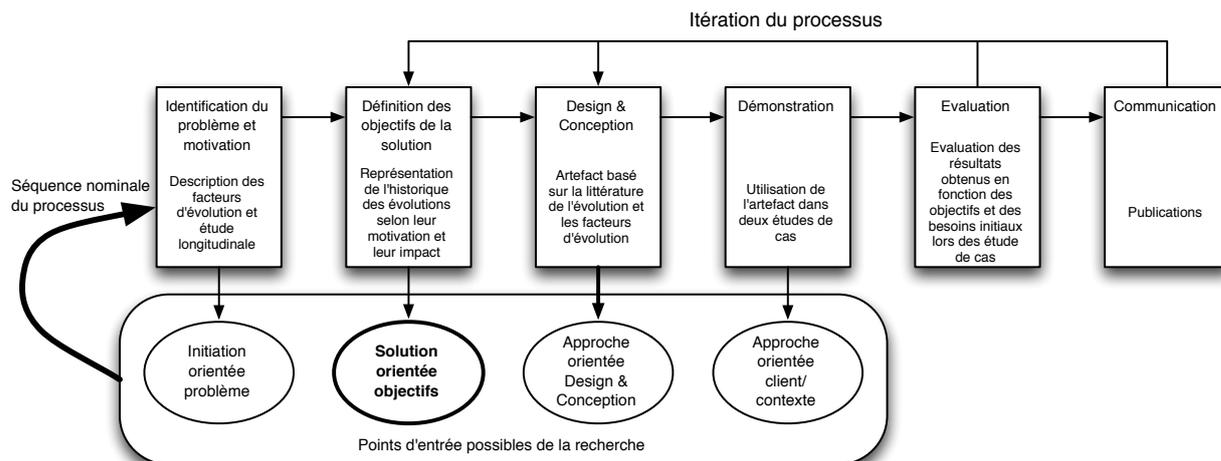


Figure 4. Processus de recherche selon (Peppers et al., 2007) adapté à notre recherche

Nous sommes passés par plusieurs itérations afin de définir notre artefact final. Les publications et études de cas nous ont poussés à redéfinir et à améliorer le design de ce dernier.

Nous avons entrepris deux études de cas, une première sur un échantillon de données réduit afin d'évaluer l'applicabilité de l'artefact ainsi que la méthodologie d'étude de l'évolution et une seconde sur un grand volume de données pour démontrer et évaluer l'artefact. Lors de chacune de ces études de cas, nous avons corrigé les lacunes ainsi que les problèmes de conception de notre artefact.

Les publications ainsi que les retours de la communauté, notamment des réviseurs nous ont également permis de réviser la conception et les éléments que devait couvrir l'artefact.

3.6.1 Identification du problème et motivation

Le Chapitre 1 : Introduction ainsi que le Chapitre 2 : Motivations présentent le contexte de recherche du problème que nous désirons traiter et les questions de recherche adressées dans cette thèse. Pour résumer, cette thèse vise à étudier les évolutions des systèmes d'information de manière globale. Un des éléments que nous voulons approfondir concerne l'étude des motivations ou des facteurs déclencheurs d'évolution. En effet ces derniers semblent ne pas avoir été une priorité pour le domaine de l'évolution des systèmes.

3.6.2 Définition des objectifs de la solution

L'artefact créé pour solutionner le problème doit avoir une utilité à la fois pour le contexte métier ainsi que pour le contexte académique. Il devra donc répondre aux spécificités de chacun des contextes. Les spécificités et les objectifs que doit couvrir l'artefact créé sont détaillés dans la Section 5.1.

Pour résumer, les objectifs de la solution sont de permettre l'étude globale de l'évolution des systèmes d'information en incluant l'étude des facteurs déclencheurs d'évolution. Afin de revêtir une utilité pour le domaine scientifique et professionnel, l'artefact doit se baser sur les connaissances reconnues du domaine et permettre de les étendre. Dans le même temps, ces connaissances doivent également être utilisables par les praticiens pour faciliter la compréhension de l'évolution de leurs systèmes ainsi que leur pilotage.

3.6.3 Design et conception

Les prémisses du design de l'artefact reposent sur l'étude de la littérature de l'environnement de recherche (Chapitre 4 sur l'évolution logicielle) ainsi que sur l'application et l'évaluation de l'artefact. Le processus de conception de l'artefact est principalement itératif, basé sur les connaissances préalables de la littérature du domaine et l'application de ce dernier sur des cas réels de différentes envergures.

Les choix de conception de l'artefact ainsi que la méthodologie suivie pour ces choix sont présentés dans le Chapitre 4.

Le détail de chaque élément retenu pour constituer l'artefact est présenté dans le Chapitre 5 et le Chapitre 6. Ce dernier contient une méthodologie de mise en œuvre de l'artefact ainsi que des éléments additionnels retenus pour l'étude des évolutions.

3.6.4 Démonstration

Pour démontrer l'application de l'artefact, deux études de cas concrets ont été conduites. Elles sont présentées dans le Chapitre 7 et le Chapitre 8. De plus, le Chapitre 6 contient une méthodologie de mise en œuvre qui démontre l'application de l'artefact ainsi que des éléments additionnels retenus pour l'étude des évolutions.

3.6.5 Evaluation

Nous avons effectué plusieurs types d'évaluations. Selon le cadre d'évaluation des recherches design en systèmes d'information développé par Pries-Heje et al. (2008), nous avons effectué les deux types d'évaluation naturalistes à savoir Ex Ante et Ex Post. Une première évaluation Ex Ante a été effectuée sous la forme d'un article qui nous a permis de valider et de compléter les facteurs déclencheurs d'évolution représentés dans l'artefact (Métrailler & Estier, 2012a).

Les principales évaluations du design de l'artefact ont été conduites Ex Post. Selon les catégories des évaluations Ex Post présentées par Yang & Padmanabhan (2005), nous avons utilisé une configuration dite abstraite et automatique en appliquant l'artefact sur des historiques de traces d'évolution.

Les résultats des évaluations sont présentés dans les sections correspondantes du Chapitre 7, Chapitre 8 et du Chapitre 9.

3.6.6 Communication

Afin de communiquer sur les recherches effectuées et les résultats obtenus lors de cette thèse, plusieurs articles ont été publiés dans des conférences internationales (Métrailler & Estier, 2014) et (Métrailler & Estier, 2012a) ainsi que dans une revue (Métrailler & Estier, 2012b).

La recherche et ses résultats ont également été présentés lors de séminaires de recherche interuniversitaires. Pour communiquer aux audiences autres qu'académiques, des posters ont été créés, notamment pour communiquer lors d'événements mélangeant académiciens et praticiens.

Sans oublier ce manuscrit qui participe également à la communication de la recherche et des résultats.

Chapitre 4

Environnement de recherche :

Évolution Logicielle

Ce chapitre contextualise le domaine de recherche dans lequel s'inscrit cette thèse en introduisant les différents types de logiciels, les recherches sur l'évolution des systèmes logiciels (software systems), ainsi que la maintenance des systèmes. Ce chapitre présente également les éléments de la littérature sur l'évolution des systèmes logiciels que nous avons utilisés pour mener à bien cette recherche ainsi que l'apport qui est visé par cette dernière.

4.1 Description du phénomène d'évolution

L'« évolution » décrit un phénomène présent dans de multiples domaines. Avec le temps, nos sociétés, nos villes, les concepts, les théories, les idées évoluent avec les contextes qui leurs sont propres. Le terme général « évolution » reflète un processus progressif de changement concernant les attributs de l'entité qui évolue voir également une modification des composants qui constituent cette entité.

L'évolution peut être vue comme l'antithèse de la dégradation. Par exemple, une entité ou un groupe d'entités est dit évoluant lorsque leur valeur ou leurs aptitudes augmentent avec le temps. En effet, individuellement ou collectivement les évoluant deviennent plus complets ou plus efficaces, en d'autres termes l'évolution valorise certains aspects de l'évoluant. D'un autre côté, le processus d'évolution peut également se manifester par la suppression de certaines propriétés qui ne seraient plus appropriées. Au final, les évoluant seront plus adaptés à leur environnement changeant (Madhavji et al., 2006).

Les changements évolutionnaires apparaissent généralement de manière incrémentale. Ils sont localisés et affectent une petite partie de l'évoluant proportionnellement à son entier. Il est cependant possible que des exceptions se produisent. Malgré le fait que les évolutions soient en général petites et localisées, leur impact peut être significatif pour l'évoluant.

4.1.1 Interprétation dans le contexte logiciel

Le phénomène de l'évolution dans le domaine logiciel a été identifié au début des années 1970 (Belady & Lehman, 1972). Cette identification est née suite aux besoins de maintenance ainsi que de développement continuels des logiciels utilisés dans le monde réel ou utilisés afin de répondre à des problèmes issus de domaines du monde réel.

De nos jours, l'utilisation courante des ordinateurs, que se soit de manière privée, dans l'industrie, l'éducation, le commerce, les gouvernements et ainsi de suite, nous conduit vers une dépendance de plus en plus accrue envers les logiciels. Cette situation fait naître un besoin pour les logiciels ; en effet, ces derniers doivent rester satisfaisants à mesure que le domaine opérationnel dans lequel ils sont ancrés change et évolue.

Comme l'utilisateur est dépendant du bon fonctionnement opérationnel du logiciel, il est important que ce dernier s'adapte au moment opportun ainsi que de manière fiable et rentable aux changements de son environnement.

Les avancées technologiques, les changements rapides des marchés, les nouveaux besoins, les nouvelles opportunités, tous ces éléments maintiennent une pression continue sur le changement et l'évolution des logiciels.

La rentabilité et au travers de cette dernière, l'amélioration des processus métiers ainsi que leur intégration conduit vers une dépendance organisationnelle envers les logiciels. Ceci a comme conséquence de rendre encore plus primordiale l'évolution commune entre le métier et le logiciel. Peu importe le logiciel, si ce dernier veut garder son utilité dans le monde réel, tôt ou tard il va devoir évoluer afin de s'adapter aux besoins changeant de son environnement.

Cette évolution entre métier et logiciel ne peut se faire d'elle même ; elle doit être encadrée et pilotée afin que les changements de l'environnement soient reportés dans le logiciel avec la plus grande fidélité possible. Pour mener à bien cet encadrement, une compréhension des enjeux de l'évolution logicielle est primordiale.

Il est donc important de connaître non seulement les logiciels, leur couplage, l'environnement, etc. mais également les caractéristiques et les facteurs qui déclenchent ce processus d'évolution. Pour arriver à ce niveau de pilotage, il faut comprendre les évolutions en essayant de les caractériser afin de déceler les relations spécifiques entre le logiciel, l'environnement et les évolutions.

4.1.2 Définition du terme évolution dans le domaine logiciel

Lorsque l'on cherche à définir le terme évolution, le sens commun ainsi que les dictionnaires font référence à la définition suivante :

« L'évolution est un processus de développement graduel d'une situation particulière ou d'une chose durant une période de temps.¹ »

Lorsque l'on s'intéresse aux systèmes logiciels, le terme évolution peut avoir plusieurs interprétations différentes selon le rôle de la personne concernée. Par exemple, un utilisateur ou un administrateur de base de données pourraient avoir une interprétation différente de ce que représente une évolution de la base de données. Un administrateur pourrait considérer un

¹ Traduction de : "Evolution is a process of gradual development in a particular situation or thing over a

changement comme étant de l'usage normal du système alors que pour l'utilisateur ce changement serait catégorisé comme évolution.

Afin de définir l'évolution de manière indépendante des points de vues subjectifs et qui capture les caractéristiques des systèmes logiciels, le professeur M.M. Lehman propose la définition suivante (Cook et al., 2006) :

'... a process of discrete, progressive, change over time in the characteristics, attributes, [or] properties of some material or abstract, natural or artificial, entity or system or of a sequence of these [changes]'

Cette définition englobe les caractéristiques d'évolution de diverses situations dont les systèmes logiciels. En limitant cette définition aux systèmes logiciels et plus précisément aux systèmes d'information d'une organisation, nous proposons la définition suivante (Métraiiller, 2011) :

'a process of discrete, progressive, change over time in architecture or workflows or features or functionalities of information systems.'

4.1.3 Plus loin que l'évolution : la transformation

La littérature du domaine de l'évolution logicielle converge vers une même direction. De manière générale, il existe un consensus définissant l'évolution logicielle comme étant un processus de changement incrémental. Lorsqu'un changement est de plus grande ampleur dans une perspective temporelle courte, on parle alors de transformation (Aier, Buckl, Gleichauf, & Matthes, 2011; Legner, Heck, & Möller, 2011) ou de révolution (Legner et al., 2011; Sabherwal & Hirschheim, 2001). La différence entre une évolution et une transformation réside dans le fait que la transformation change la structure profonde de l'environnement logiciel voir même le système logiciel dans son intégralité. L'évolution quant à elle, vise la stabilité de cette structure logicielle en opérant des ajustements. Selon Aier et al. (2011), l'évolution est en général initiée par une approche ascendante (bottom-up) : les utilisateurs s'aperçoivent de certains manques ou dysfonctionnements et initient le changement. A l'opposé, la transformation qui initie un changement fondamental est initiée par une approche descendante (top down) : le changement est initié par le haut de la hiérarchie.

Sabherwal & Hirschheim (2001) expliquent certains aspects de la dynamique entre évolution et transformation par le modèle de l'équilibre ponctué (Gould & Eldredge, 1977). Selon ce modèle, le système passe par de longues périodes de stabilité relative dans lesquelles s'opèrent des changements évolutionnaires. Ces périodes de stabilités sont interrompues par de courtes périodes où les changements sont fondamentaux, révolutionnaires. Les changements évolutionnaires sont ceux impliquant des modifications mineures de l'alignement entre le système et son environnement. Toujours selon Sabherwal & Hirschheim (2001), ces changements préviennent les problèmes en contrôlant les désalignements mais peuvent inhiber la nécessité de passer à un modèle différent et plus radical d'alignement. Par conséquent, les gestionnaires devraient rester vigilant quant aux alignements effectués car ils pourraient masquer les symptômes d'un futur échec. Des changements révolutionnaires dans la stratégie d'évolution du système d'information peuvent être nécessaires afin que l'organisation s'oriente vers une stratégie offrant un plus grand potentiel de performance plutôt que de continuer dans une stratégie d'alignement « de surface ».

Ces phases de révolution sont donc importantes dans l'étude de la vie du système d'information d'une organisation. En effet, les enjeux managériaux sont considérables, les gestionnaires doivent être capables d'identifier les signes précurseurs d'un besoin de changement fondamental, de révolution. Sabherwal & Hirschheim (2001) mettent en garde les gestionnaires conduisant des changements révolutionnaires de la structure profonde du système

d'information sur le fait qu'il faut continuer les changements évolutionnaires (mineurs) même après la révolution et surtout tout de suite après la révolution.

4.1.4 L'évolution logicielle : activité et phénomène

Dans le contexte de l'évolution logicielle, le terme évolution peut avoir deux principales interprétations. Dans la première approche, l'évolution logicielle est étudiée comme étant une activité qui doit être exécutée, dirigée et contrôlée ; dans la seconde approche l'évolution est étudiée en tant que phénomène. Ces deux approches d'étude de l'évolution logicielle sont pertinentes, importantes et surtout complémentaires. Chacune d'elles supporte l'importance de maintenir la compatibilité entre le système logiciel, les propriétés et les buts réels que le système supporte, ses utilisateurs, et les domaines dans lesquels le système opère (M M Lehman & Ramil, 2003; Madhavji et al., 2006).

Dans la littérature, la première approche est appelée l'approche verbale de l'évolution (M Lehman, 2003) ou amélioration du processus (process improvement) (Cook et al., 2006). Elle s'intéresse aux activités relatives à l'implémentation et à l'évolution du logiciel. Cette approche adresse les questions relatives aux développements et à l'amélioration des processus, des procédures, des méthodes et des outils visant à planifier, exécuter, supporter et contrôler l'évolution logicielle. Elle étudie le « comment » de l'évolution en développant ou en améliorant les activités, les processus, les méthodes, etc. afin d'augmenter l'efficacité vis-à-vis de facteurs tels que la réactivité, la fiabilité et les coûts.

La seconde approche est appelée l'approche nominale de l'évolution (M Lehman, 2003) ou explicative (explanatory) (Cook et al., 2006). Elle adresse l'évolution des systèmes comme un phénomène qui doit être systématiquement étudié. Elle étudie le « quoi » et le « pourquoi » de l'évolution. Elle s'intéresse aux causes, aux impacts, aux caractéristiques afin d'avoir une meilleure compréhension du phénomène d'évolution. Cette compréhension doit permettre de guider et d'inspirer le développement des méthodes, procédures et outils afin d'améliorer le processus global d'évolution. En d'autres termes, cette approche doit fournir une base ainsi qu'un cadre afin de permettre à l'approche verbale d'être la plus efficace possible. Les résultats d'études du phénomène d'évolution ne peuvent donc qu'améliorer les résultats attendus de la part de l'approche verbale, voilà donc pourquoi ces deux approches sont étroitement liées et complémentaires.

Cette recherche vise à créer un outil capable d'étudier le phénomène d'évolution en caractérisant de manière systématique les évolutions d'un système d'information. De ce point de vue, nous nous situons dans la seconde approche (approche nominale de l'évolution) en étudiant le « quoi » et le « pourquoi » des évolutions d'un système. Dans cette vision, un des objectifs principaux de cette recherche, en caractérisant précisément les évolutions, est de permettre aux chercheurs de mieux comprendre le processus d'évolution des systèmes d'information.

D'un autre côté, l'utilisation de cette recherche par les gestionnaires de systèmes d'information peut les aider à mieux comprendre l'état actuel de leur système en retraçant son évolution. De plus, cette recherche peut permettre aux gestionnaires d'évaluer l'impact d'un changement dans le système et peut donc les aider à la conception de la stratégie d'évolution du système d'information. De ce deuxième constat, cette recherche prend la forme d'un outil visant à l'amélioration du processus d'évolution (approche verbale de l'évolution). Cette recherche est donc parfaitement inscrite dans le domaine de l'évolution logicielle en intégrant les deux approches de l'évolution logicielle que sont l'activité d'évolution et le phénomène d'évolution.

4.2 Typologie de systèmes logiciels

Un des principaux chercheurs dans le domaine de l'évolution logiciel est le professeur Meir M. Lehman. A la fin des années 60 début des années 70, le professeur Lehman étudiait les processus de programmation des systèmes d'exploitation IBM OS360-370. Malgré que le rapport de ses résultats (MM Lehman, 1969) ne soit accessible que plus tard (MM Lehman & Belady, 1985), ce rapport examinait et décrivait les processus de changement continu des systèmes d'exploitation en se basant sur les propriétés des différentes versions. Des modèles d'évolution dérivés de ces recherches ont été proposés par la suite comme outils pour le planning, la gestion et le contrôle des versions (Belady & Lehman, 1972; M.M. Lehman, 1980).

Les travaux de Lehman et Belady (MM Lehman & Belady, 1985) sur la dynamique de croissance des logiciels ainsi que sur leur évolution concluent que cette dernière est intrinsèque aux programmes de grande taille (large programs). L'adjectif « de grande taille » est utilisé pour qualifier des programmes dont le nombre de lignes de code varie de 50'000 à 500'000 (Madhavji et al., 2006). Par la suite Lehman suggéra que cet intervalle arbitraire n'était pas pertinent dans l'étude de l'évolution. De plus, si la taille devait être un facteur pour déterminer des propriétés d'évolution, il est tout à fait possible que ces propriétés soient différentes pour des programmes de taille différente. De la même manière, deux logiciels ayant un nombre de ligne de code similaires pourraient avoir des propriétés d'évolution différentes selon leur environnement organisationnel, managérial, applicatif, etc.

Suite à ces éléments, Lehman suggéra que les phénomènes d'évolution observés étaient plus enclins à être liés avec les particularités du milieu dans lequel le logiciel a été développé, utilisé, ainsi qu'à son environnement applicatif. Il proposa alors qu'un logiciel soit considéré comme de « grande taille » si « ... il avait été développé ou maintenu par une structure managériale comprenant au moins deux groupes » (M M Lehman, 1979). Malgré tout cette dernière définition n'est toujours pas considérée comme suffisante pour expliquer le besoin intrinsèque d'évolution des logiciels.

4.2.1 Classification des programmes

Malgré ces précédentes tentatives de définition d'un « programme de grande taille », ce concept de taille n'était toujours pas satisfaisant comme base fondamentale pour l'étude de l'évolution logicielle. De ce constat, une classification non basée sur la taille des programmes a été proposée. Initialement cette classification regroupait les programmes selon trois catégories : les programmes de type S, P et E. Les programmes du type E sont ceux qui sont les plus à même d'être sujet à des études sur l'évolution. En effet, si les utilisateurs de ce type de programmes doivent rester satisfaits de leur usage, il a été démontré que leur l'évolution était inévitable. De plus, cette classification s'étend également aux applications, aux domaines d'applications, aux systèmes d'information, etc. (MM Lehman, 1991).

Applications et logiciels de type S

Un programme ou une application est caractérisée comme étant de type S si il peut être démontré qu'il satisfait la condition unique et nécessaire qu'il répond à une spécification de manière correcte d'un point de vue mathématique. Le S a été choisi pour montrer le rôle définitif que joue la spécification (S pour Spécification) dans la définition des propriétés du système. Ces programmes doivent donc rester statique en comparaison aux programmes de type E qui doivent être en continuellement changés et adaptés afin de rester satisfaisants.

Applications et logiciels de type E

Les programmes de type E étaient initialement définis comme étant : « les programmes qui mécanisent une activité humaine ou de société.² » (M.M. Lehman, 1980). Par la suite cette définition s'étend aux programmes qui opèrent ou adressent un problème ou une activité dans le monde réel. Afin de rester satisfaisant, ce type de programme doit, de par sa nature, être continuellement adapté et évoluant ; d'où la désignation E pour Evoluant.

Une propriété clé de ce type de programmes est qu'il devient une partie intégrante du domaine dans lequel il opère. L'évolution logicielle est donc une conséquence directe de la dynamique de l'environnement dans lequel le programme opère. Un programme de type E peut tout à fait inclure des éléments qui, de manière isolée, seraient de type S. Les systèmes d'exploitation, les bases de données, les systèmes de gestion sont tous des systèmes de type E qui contiennent des sous éléments de type S.

Applications et logiciels de type P

Un troisième type de programme, P, a initialement été défini d'une manière intermédiaire entre les programmes S et E. Ce troisième type devait initialement contenir les programmes répondant à un problème dont les spécifications étaient explicitées (P pour Problème). Ces spécifications résultent généralement de compromis entre les différents acteurs. De ce fait, ces spécifications ne pouvaient pas être définies de manière suffisamment stricte et la preuve de la conformité des résultats n'était pas démontrable de la manière formelle exigée par les logiciels de type S.

Rapidement, ce troisième type a été considéré comme redondant par Lehman de par le fait qu'un programme de type P pouvait être classifié dans l'une ou l'autre des deux autres catégories.

SPE+

Plus récemment, l'article : Evolution in software systems: foundations of the SPE classification scheme (Cook et al., 2006), dont Lehman figure parmi les auteurs, propose une version affinée de la classification S, P et E : la classification SPE+. Dans cette classification SPE+, les auteurs lient les concepts de l'évolution logicielle aux théories génériques de l'évolution dont notamment les concepts de réplicateur de Dawkins et la notion de paradigme de Kuhn. La notion de réplicateur ainsi que l'herméneutique phénoménologique vont permettre aux auteurs d'affiner la description et la compréhension du comportement des systèmes de type E.

Cette classification reste dans l'esprit de la précédente et n'entre aucunement en conflit avec les travaux préexistants du domaine. Dans les grandes lignes :

- Les définitions et les descriptions des types S, P et E sont dérivées non seulement du domaine de l'ingénierie logicielle mais également des théories philosophiques concernant l'évolution générale.
- SPE+ affirme explicitement que le type E représente la catégorie par défaut pour l'évolution des systèmes logiciels
- SPE+ définit les types P et S comme étant des cas particuliers qui découlent de certains types de besoins des parties prenantes

² « ... programs that mechanise a human or societal activity »

- SPE+ remplace la définition précédente des programmes de type P avec une définition dérivée du concept de paradigme. Le P signifie donc basé sur un Paradigme dans SPE+.

Apport de SPE+ pour les systèmes de type E

Dès la conception d'un système de type E, l'usage de l'herméneutique phénoménologique sera nécessaire afin d'analyser les besoins des futurs intervenants du système. En effet, développer le modèle conceptuel d'un futur système ayant un rôle dans le monde réel (système de type E) nécessite d'effectuer des choix issus de l'interprétation des besoins des différents intervenants. Les analystes devront interpréter les descriptions des besoins des intervenants du système afin d'atteindre une compréhension commune du domaine.

La notion de réplicateur est utilisée pour décrire le comportement d'un système évoluant. Pour résumer, chaque système évoluant, donc de type E, peut être vue comme une collection de réplicateurs. Chacun de ces réplicateurs dispose d'un certain potentiel de chance de survivre en étant copié ou non dans la prochaine version du système ou dans un autre système. Ces réplicateurs sont liés de manière plus ou moins forte avec leur entourage. Afin de décrire les réplicateurs ainsi que leur liens, les auteurs de SPE+ (Cook et al., 2006) prennent l'exemple d'un système de gestion de l'apprentissage dans lequel le *concept d'évaluation* peut dépendre d'un *concept de cours*. Ainsi, les chances de survie du réplicateur *concept d'évaluation* sont liées à celle du réplicateur *concept cours*. En d'autres termes si le *concept évaluation* est copié, il y a de fortes chances que le *concept cours* le soit aussi.

Nouvelle définition des systèmes de types P paradigme de SPE+

Le travail de (Cook et al., 2006) fait valoir qu'un système logiciel doit être défini comme étant de type P si ses parties prenantes ont pris la décision stratégique que le système doit se conformer à un paradigme unique dans sa représentation des connaissances du domaine. En d'autres termes, ce qui différencie un système de type S d'un système de type P réside dans le fait que ce dernier peut évoluer uniquement s'il reste cohérent tout au long de son cycle de vie dans un paradigme prédéfini par ses parties prenantes. Cette contrainte se concrétise de deux manières. Elle permettra d'éviter certains types de changements qui auraient autrement eu lieu. Elle peut également induire un changement, soit lorsque le paradigme est mis à jour, soit lorsque par exemple, grâce à l'évolution technologique, une occasion se présente d'améliorer la cohérence du système avec son paradigme.

4.3 Lois de l'évolution logicielle

Les lois de l'évolution logicielle (laws of software evolution) de Lehman sont une contribution majeure afin d'identifier les causes et les processus de ce phénomène complexe. Les lois de l'évolution sont au nombre de 8 (voir Tableau 1). Leur développement date du début des années 70 et elles ont été affinées et renforcées jusque dans les années 2000. Elles décrivent un ensemble de principes généraux relatifs à l'évolution des systèmes logiciels de type E.

Tableau 2 : Lois de l'évolution logicielle adaptées de M M Lehman & Ramil (2001).

N° et année	Nom	Brève description
I 1974	Modification perpétuelle (Continuing change)	Les systèmes de type E doivent être continuellement adaptés afin qu'ils ne deviennent pas moins satisfaisant dans leur environnement.
II 1974	Complexité croissante (Increasing complexity)	Au fur et à mesure qu'un système de type E évolue, sa complexité augmente. Il est nécessaire d'investir des ressources afin de maintenir le niveau de complexité ou de le réduire.
III 1974	Autorégulation (Self regulation)	Le processus d'évolution des systèmes de type E est autorégulé. Les propriétés du système comme sa taille, le temps entre les versions, ne varient quasiment pas entre deux versions. Ils suivent une dynamique dont la distribution est proche de la normale.
IV 1980	Conservation de la stabilité organisationnelle (Conservation of organizational stability)	Le taux d'activité moyen (interventions sur le logiciel) tend à rester constant sur la durée de vie d'un logiciel.
V 1980	Conservation de la familiarité (conservation of familiarity)	Durant la période d'activité d'un système de type E, les parties prenantes doivent conserver la maîtrise du système pour que l'évolution soit un succès. Il en résulte que l'incrément du changement de chaque nouvelle version est à peu près constant.
VI 1980	Croissance continue (Continuing growth)	Les fonctionnalités d'un système de type E doivent être enrichies afin de maintenir la satisfaction des utilisateurs durant la durée de vie du système.
VII 1996	Baisse de qualité (Declining quality)	Les acteurs vont percevoir une baisse de qualité du système logiciel si il n'est pas rigoureusement maintenu et adapté à son environnement opérationnel changeant.
VIII 1996	Système de feedback (Feedback system)	Le processus d'évolution d'un système de type E constitue un système de feedback multi-niveaux, multi-boucles et multi-agents. Il doit être traité en tant que tel afin d'obtenir une amélioration significative.

L'utilisation par Lehman du terme « loi » a subi des critiques. En effet, il ne faut pas considérer le terme loi de la même manière qu'en physique, les lois de Lehman ne représentent pas une relation mathématique invariante entre des éléments. Leur but est de représenter la connaissance sur les propriétés communes que partagent les systèmes logiciels en évolution. Le terme lois de Lehman s'apparente plus aux lois économiques avec par exemple « la loi de l'offre et de la demande » qui sera modifiée en fonction de la situation dans laquelle elle est appliquée. Les lois de l'évolution logicielle décrivent donc l'évolution de manière similaire à une situation sociale qui reste variable en pratique.

La loi du « Système de feedback » est importante afin de capter la complexité de l'évolution des systèmes de type E. En effet, l'évolution est sujette à une boucle de feedback positive (Cook et al., 2006). L'introduction d'un nouveau système ou d'améliorations à un système existant peut produire des effets secondaires inattendus. Peu importe si les changements effectués sur le système satisfont les attentes des acteurs, l'introduction de ces changements peut mettre en lumière d'autres problèmes qui devront être adressés par de futurs changements. Par exemple, un changement peut provoquer chez certains acteurs une modification de leur perception du problème qu'ils visaient à résoudre avec le changement initial, ou une modification de leurs attentes sur la manière dont le système produit ses résultats. Les compromis antérieurs entre les préoccupations des différents acteurs peuvent être remis en cause de manière imprévisible par l'introduction même des améliorations au système. Par conséquent, la dynamique du processus d'évolution d'un système de type E sera complexe, difficile à prédire et parfois même contre intuitive (Chatters & Lehman, 2000).

Dans le cas d'un système de type P, comme ce dernier est également utilisé dans le monde réel, il sera lui aussi sujet à certaines lois de l'évolution, notamment celle du système de feedback. Ainsi, pour un système de type P, le paradigme du domaine prédéfinit une structure complexe d'hypothèses et de théories dont les acteurs du système ont connaissance. Cela aura des effets sur les boucles de feedback. En effet, l'utilisation d'un système de type P peut changer son paradigme uniquement dans le cas où cette dernière contribue à de nouvelles connaissances dans son propre paradigme. Par exemple, un système simulant des lois de physique pourrait contribuer à son propre paradigme (la physique), ce qui engendrerait une nécessité de modification du paradigme afin d'y intégrer les nouvelles découvertes.

4.4 Maintenance logicielle

A la fin des années 70, les premiers travaux décrivant les activités de maintenance du logiciel sont ceux de Lientz et al. (1978). Actuellement la norme ISO 12207 (Software Life Cycle Processes) définit le processus de maintenance comme étant : « le processus mis en œuvre lorsque le logiciel subit des modifications relatives au code et à la documentation correspondante. Ces modifications peuvent être dues à un problème, ou encore à des besoins d'amélioration ou d'adaptation. L'objectif est de préserver l'intégrité du logiciel malgré cette modification. On considère en général que ce processus débute à la livraison de la première version d'un logiciel et prend fin avec son retrait. »

Les travaux initiaux de Lientz et al. (1978) identifiaient trois catégories d'activités de maintenance : corrective, adaptive et perfective. Parmi les normes et standards actuels évoquant la maintenance logicielle, la norme ISO 14764 (Software Maintenance) présente quatre catégories d'activités de maintenance :

- La maintenance corrective : modification d'un logiciel effectuée après la livraison afin de corriger les problèmes découverts.

- La maintenance adaptative : modification d'un logiciel effectuée après la livraison afin de maintenir un produit logiciel utilisable dans un environnement matériel ou logiciel modifié ou changeant.
- La maintenance perfective : modification d'un logiciel effectuée après la livraison afin d'en modifier les propriétés fonctionnelles suite à une modification des besoins d'utilisation.
- La maintenance préventive : modification d'un logiciel effectuée après la livraison afin de détecter et de corriger les défauts latents avant qu'ils se manifestent comme défaillances afin d'en améliorer l'efficacité ou la maintenabilité.

La maintenance logicielle comprend aussi bien des activités techniques que managériales. En effet, elle ne se résume pas uniquement aux actions relatives à la modification du code faisant suite à la découverte de défaillances de fonctionnement. Elle adresse également les enjeux managériaux tels que l'alignement sur les priorités des utilisateurs, la justification des coûts de maintenance, les plans et procédures de maintenance, etc.

Bien que la classification qui précède soit la plus couramment admise, il convient de noter que plusieurs auteurs et normes proposent des classifications différentes avec notamment l'article de Chapin, Hale, Khan, Ramil, & Tan (2001). Dans cet article, les auteurs proposent une typologie affinée des activités de maintenance et d'évolution. Cette dernière classe les activités de maintenance selon 12 types rassemblés dans 4 groupes. Nous détaillons ces 12 types ainsi que les 4 groupes dans la Section 6.4.

4.5 Maintenance versus évolution

Les deux termes maintenance et évolution sont souvent employés dans la littérature sans qu'il soit possible de positionner clairement l'un par rapport à l'autre. Historiquement, depuis les années 70, évolution et maintenance semblent être deux domaines distincts. En effet, la communauté « maintenance » s'intéresse essentiellement au code source du logiciel et à sa mise à jour après sa mise en service. Les normes et standards actuels fournissent une définition précise de ce qu'est la maintenance.

De l'autre côté, la communauté « évolution » s'intéresse non seulement aux modifications de code source mais également aux modifications d'architecture, d'environnement et plus généralement à tout ce qui touche et influe le logiciel. Selon Lehman, l'évolution logicielle est l'étude de la dynamique logicielle. Contrairement à la maintenance, il n'existe pas de standard ni de définition claire sur ce qu'englobe l'évolution. Dans la littérature, différentes définitions de l'évolution ont été proposées, cependant, elles ne sont malheureusement pas toutes similaires. Le terme évolution reste donc ambigu. Néanmoins les deux communautés se sont rapprochées au fil des années, la principale revue scientifique sur le domaine de la maintenance créée en 1989 : *Journal of Software Maintenance: Research and Practice*, a été renommée en y incluant le terme évolution en 2001 : *Journal of Software Maintenance and Evolution: Research and Practice*. Le terme évolution semble même redonner de l'attrait pour cette discipline.

A la fin des années 90 début 2000, le terme évolution est repris à plusieurs reprises pour qualifier un sous ensemble des activités de maintenance. Dans l'article *Software Maintenance and Evolution : a Roadmap* (Bennett & Rajlich, 2000a), les auteurs utilisent le terme maintenance pour faire référence à toutes les activités post-livraison du logiciel, tandis que le terme évolution est utilisé pour faire référence à une phase particulière de leur modèle de cycle de vie du logiciel. Durant cette phase dite d'évolution, les ingénieurs étendent les capacités et

fonctionnalités du système logiciel, tandis que dans les autres phases seules des activités mineures sont entreprises sur le logiciel. Parallèlement, (Chapin et al., 2001) décrivent l'évolution comme étant un sous ensemble des activités de maintenance. Ils décrivent 12 types d'activités de changement sur le logiciel dont certaines sont qualifiées de maintenance et d'autres d'évolution. Ce sont principalement les changements impactant les propriétés fonctionnelles du logiciel qui sont catégorisées comme activités d'évolution.

Cette dernière définition de l'évolution par Chapin et al. (2001) paraît réductrice à la vue des travaux réalisés sur l'évolution notamment par Lehman. La vision actuelle du terme évolution qui semble s'être imposée considère que l'évolution est un terme plus général décrivant la problématique de la vie du logiciel après sa mise en production. En effet, Parnas (1994) souligne que « maintenance » connote l'idée d'entretenir un système existant sans modifier son design. Cependant, la pratique de maintenance du logiciel requiert le changement du design de ce dernier : en corrigeant les bugs afin que le design soit correct, en adaptant le système pour un environnement changeant etc. (Godfrey & German, 2008). Le terme « évolution » offre une perspective qui diffère de la vue traditionnelle de la maintenance : il suggère l'idée de changement nécessaire dans un environnement en englobant naturellement les concepts des deux phénomènes planifiés et non planifiés ; tandis que la maintenance suggère plutôt ce qui est planifié dans une optique de préservation. On effectue de la maintenance *sur* un système. A l'opposé, l'évolution concerne ce qui arrive à un système au fil du temps, que ce soit planifié ou non (Godfrey & German, 2008).

Dans notre recherche, le terme « évolution » peut donc se substituer au terme « maintenance ». En effet, il étend ce dernier en incluant les aspects caractérisant les propriétés du logiciel, les processus ainsi que la dynamique des organisations et de l'environnement dans lequel le logiciel évolue. C'est cette définition plus systémique de l'évolution que nous avons choisi d'adopter dans cette recherche.

4.6 Le processus d'évolution

Il y a deux niveaux d'études du processus d'évolution d'un logiciel : le niveau macroprocessus, et le niveau microprocessus de l'évolution (Fleurquin & Tibermacine, 2007). Le niveau macroprocessus adresse des intervalles de temps vastes délimités par la mise en service du logiciel et son retrait. Il s'intéresse aux différentes phases par lesquelles passe la vie des logiciels. A l'inverse, le niveau microprocessus de l'évolution se préoccupe de la manière dont une seule évolution se déroule. Ce niveau adresse donc des intervalles de temps plus courts qui sont délimités par le moment lorsqu'une nouvelle version du logiciel est développée jusqu'à ce qu'elle remplace la version actuelle. Ci-après, le niveau macro du processus d'évolution est présenté dans lequel nous précisons la phase adressée par notre étude (Section 4.6.1); puis le niveau micro de ce processus d'évolution (Section 4.6.2).

4.6.1 Macroprocessus

Le modèle le plus populaire représentant ce niveau macroprocessus de l'évolution est le modèle de phases du cycle de vie du logiciel (Staged Model for Software Lifecycle) de Bennett & Rajlich (1999). Il affirme que durant sa vie un logiciel passe par 5 phases successives (Figure 5) : le développement initial, l'évolution, le service, la suppression progressive et l'arrêt de son exploitation. Les phases d'évolution, de service et de suppression progressive sont décrites comme étant les trois phases constituant la phase de maintenance. Pour synthétiser, lorsqu'un logiciel est mis en production, il entre dans sa phase d'évolution. Durant cette phase, le but est d'adapter le logiciel en fonction des changements continus de besoins des utilisateurs et de son

environnement opérationnel. Au fur et à mesure des évolutions, le logiciel va se complexifier rendant les évolutions de plus en plus coûteuses. Lorsque la décision est prise de ne plus faire évoluer ce logiciel, il entre dans la phase de service également appelée phase de maturité logicielle (MM Lehman, 1991). Durant cette phase seulement quelques corrections mineures apportées au logiciel. La suppression progressive et l'arrêt de l'exploitation servent à la gestion de la fin de vie du logiciel. Ce modèle se positionne comme étant dans la droite ligne des travaux de Lehman sur la dynamique des logiciels en référant aux lois de l'évolution.

Les phases du cycle de vie que nous adressons dans cette étude sont celle de l'évolution, du service et de la suppression progressive. Nous étudions donc tous les changements effectués sur le logiciel dès sa première mise en service et jusqu'à la décision du retrait de ce dernier.

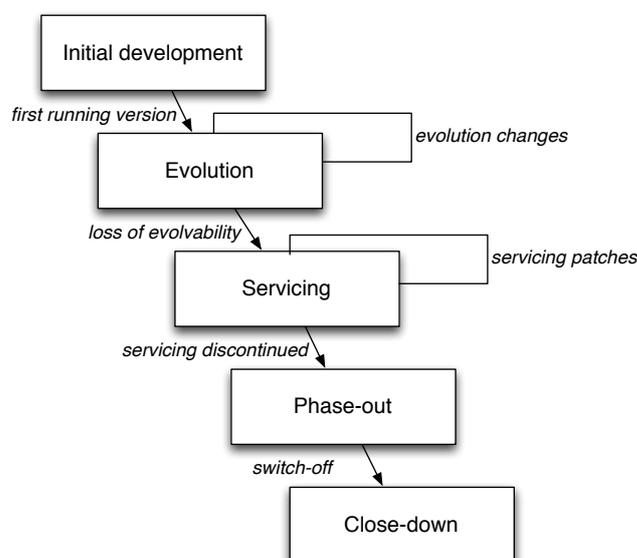


Figure 5. Macroprocessus d'évolution (Bennett & Rajlich, 2000a).

4.6.2 Microprocessus

Le microprocessus d'évolution concerne toutes les activités qui sont à entreprendre lors d'un acte de mise à jour du logiciel. La norme ISO 12207 donne une description détaillée de chacune de ces activités. La Figure 6 détaille le processus de modification d'un logiciel.

La réception d'une demande de changement est l'élément déclencheur de toute action d'évolution. Elle peut émaner de source externe (environnement, client) ou d'un acteur interne à l'organisation. La norme ISO 14764 (Software Maintenance) peut être appliquée ici afin de catégoriser globalement une évolution : Si la demande de modification fait suite à une anomalie de fonctionnement (on parlera alors de cycle de maintenance corrective), à la volonté de migrer l'application vers un nouvel environnement matériel ou logiciel (maintenance adaptative), au souhait de modifier les propriétés fonctionnelles ou non fonctionnelles de l'application (maintenance perfective) ou d'améliorer à titre préventif principalement sa maintenabilité et sa robustesse (maintenance préventive) (Fleurquin & Tibermacine, 2007).

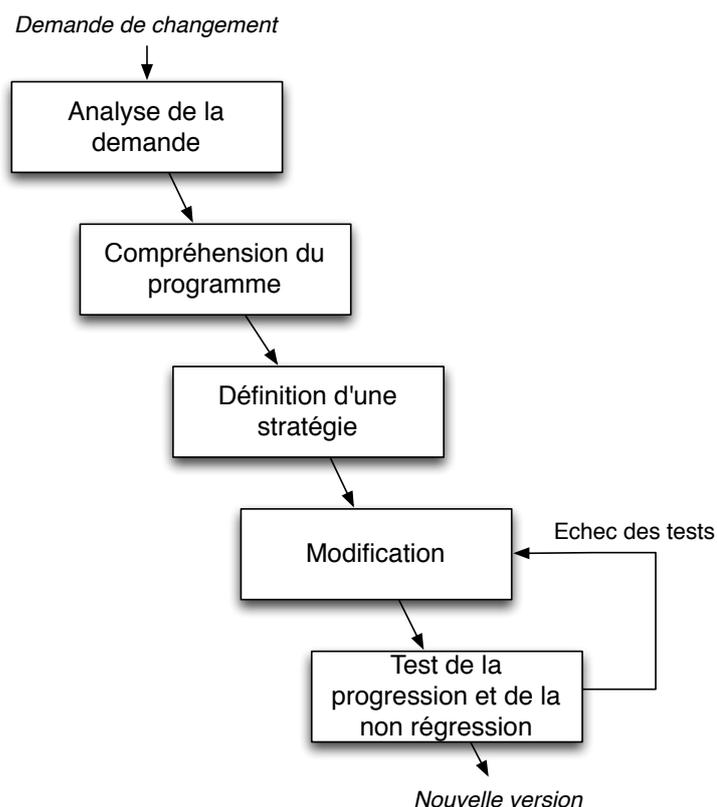


Figure 6. Microprocessus d'évolution adapté de (Fleurquin & Tibermacine, 2007).

La norme ISO 12207 précise un ensemble de processus décrivant le cycle de vie d'un logiciel. Cette norme détaille notamment un processus dit de maintenance. Dans ISO 12207, un processus est défini comme étant constitué d'un ensemble d'activités, elles-mêmes constituées d'un ensemble de tâches cohérentes, tel qu'illustré dans la Figure 7.

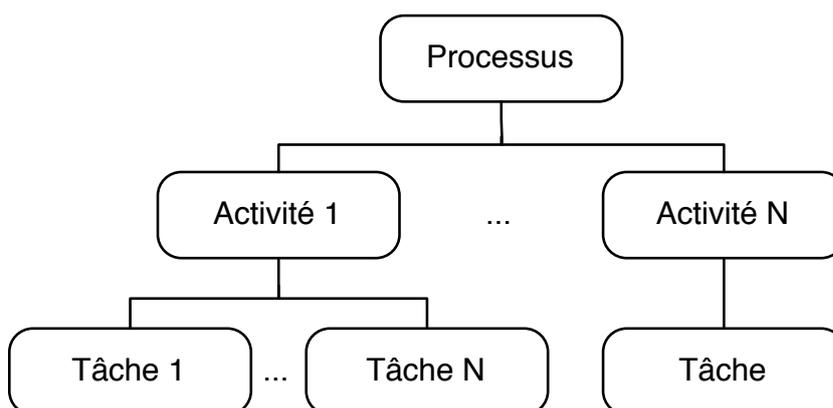


Figure 7. Structure d'un processus selon la norme ISO 12207.

Nous avons donc tout naturellement repris ce découpage du processus de maintenance en activités et tâches afin de l'appliquer aux évolutions étudiées par ce travail. A chaque fois que nous parlerons d'évolution, cette dernière pourra être décomposées en plusieurs activités.

Par analogie, une évolution représente le niveau macro du processus (Section 4.6.1) et une activité représente le niveau micro de ce processus d'évolution (Section 4.6.2).

4.6.3 Typologie des activités du processus d'évolution

Afin d'étudier les types d'activités qui composent une évolution, nous avons retenu la typologie de Chapin et al. (2001). En utilisant cette typologie, il devient possible de classifier les différentes activités dans chacun des processus d'évolution en fonction de leur type spécifique. Nous obtenons donc au final un processus d'évolution qui est composé de plusieurs activités précisément typées.

La typologie de Chapin et al. (2001) propose une classification plus étendue des différents types d'activités de maintenance et d'évolution logicielle. En effet, les activités de maintenance et d'évolution sont classifiées selon 12 différents types au lieu des 4 types initiés par Lientz et al. (1978) et complétés par la norme ISO 14764. La granularité des activités proposées par Chapin et al. (2001) est donc bien plus riche et permet une classification des activités en adéquation avec le travail effectué par les praticiens.

Nous avons choisi cette classification suite aux motivations qui ont poussé les auteurs à son élaboration. Premièrement, résumer les activités de maintenance à 4 différents types paraît réducteur vis-à-vis de la nature très diversifiée des activités effectuées par les praticiens. Deuxièmement, les gestionnaires impliqués dans l'évolution logicielle ont fréquemment des difficultés à définir, budgéter, justifier les ressources, la dotation en personnel, etc. pour des activités de maintenance et d'évolution qui ne sont pas définies de manière suffisamment précise. Troisièmement, les chercheurs ont besoin d'une terminologie descriptive précise pour les types d'activités qu'ils observent lorsqu'ils étudient l'évolution logicielle. Cette classification est détaillée dans la Section 6.4 ou son usage dans notre recherche est détaillé ainsi que dans la Figure 8.

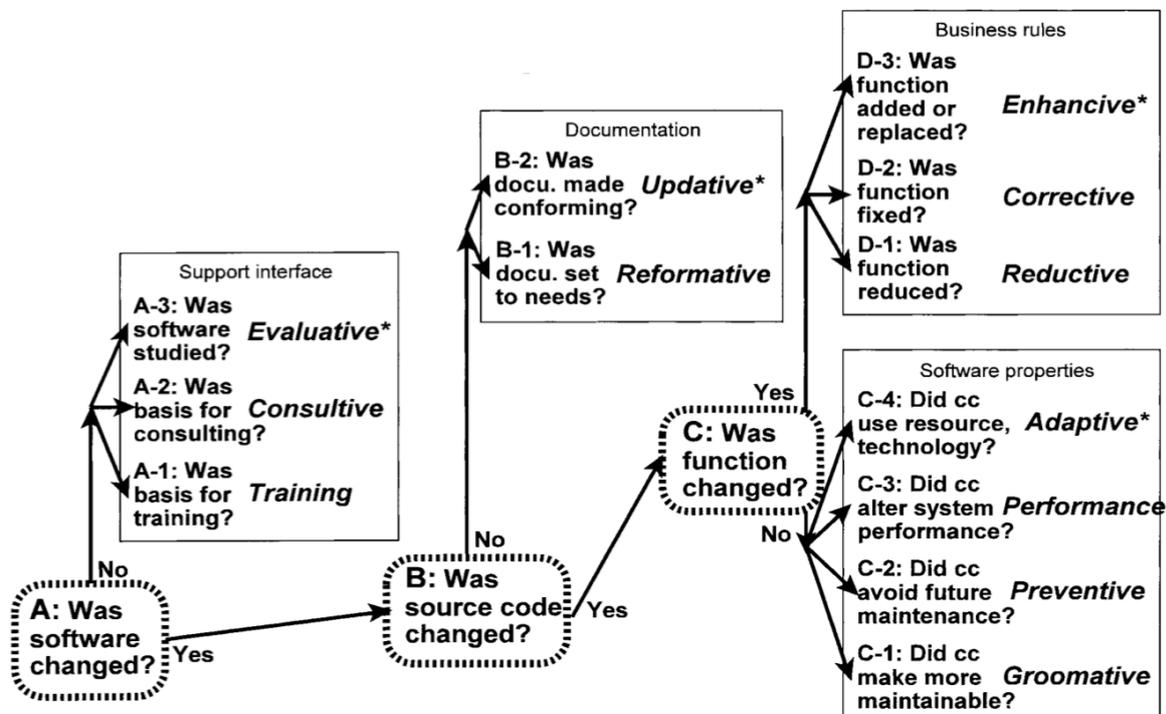


Figure 8. Arbre de décision pour déterminer le type des activités d'évolution (Chapin et al., 2001).

4.7 Facteurs déclencheurs d'évolution

Afin d'en connaître plus sur les évolutions des systèmes d'information, autant les praticiens que les chercheurs sont intéressés par connaître quelles sont la ou les raisons qui poussent une organisation à entreprendre une évolution. Est-ce un enjeu métier, technologique, etc. qui occasionne une évolution ?

Concernant cette problématique, la littérature du domaine ne contient que peu d'études se focalisant sur les éléments déclencheurs d'évolution. En effet, l'étude des motivations ou des facteurs déclencheurs d'évolution semble ne pas avoir été une priorité pour le domaine, tout comme leur influence dans le processus d'évolution tout au long de la vie des systèmes (Godfrey & German, 2008). Les nombreuses publications sur le domaine se focalisent sur l'aspect technique, les défis que représente l'évolution du logiciel plutôt que sur la dimension managériale de l'évolution du système d'information dans son ensemble.

Afin de sélectionner les critères les plus pertinents pour déterminer quelle est la motivation qui déclenche une évolution, nous avons effectué une revue de littérature concernant l'étude des évolutions, des changements ainsi que de la maintenance des systèmes logiciels.

Cette revue de littérature nous a permis de sélectionner et de combiner plusieurs sources afin de baser nos travaux sur des recherches reconnues. Nous avons donc combiné les éléments les plus pertinents et fiables afin de caractériser les évolutions logicielles.

Les retours lors de conférences ainsi que les commentaires des réviseurs et des personnes à qui nous avons présenté cette recherche nous ont également aiguillés pour la sélection des facteurs déclencheurs et la prise en compte de certains éléments dans le processus de d'identification des facteurs déclencheurs d'évolution.

Le premier domaine de recherche que nous avons étudié pour déterminer les critères permettant de définir les facteurs déclencheurs d'évolution est celui qui adresse le phénomène d'évolution logicielle dans sa globalité. Les travaux d'un groupe de chercheurs se distinguent dès le début des années 70 avec notamment la classification des programmes (SPE) (Cook et al., 2006, 2000; M.M. Lehman, 1980) et les lois de l'évolution logicielle (M M Lehman & Ramil, 2001) décrits dans les Sections 4.2 et 4.3. Ce domaine de recherche nous aide à répondre au besoin qui touche la qualification des déclencheurs et des raisons aboutissant aux évolutions d'un système logiciel. Ces recherches nous permettent d'établir les critères pour l'identification des déclencheurs d'évolution pour les systèmes d'information détaillés dans cette Section.

Le second domaine de recherche lié aux évolutions logicielles que nous avons étudié décrit ce qu'est une évolution ou un changement logiciel au travers d'une taxonomie des évolutions et d'une typologie des activités d'évolution. Premièrement afin d'organiser et de classifier les évolutions, nous avons choisi de nous référer à la taxonomie des évolutions logicielles de décrite par Buckley et al. (2005) (son usage dans notre recherche est détaillé en Section 6.2).

Puis, pour étudier les activités entreprises lors du déroulement d'une évolution, en d'autres termes pour qualifier ce qui est fait pour évoluer, nous nous appuyons sur les travaux de Lientz et al. (1978) sur les caractéristiques des activités de maintenance logicielle et plus précisément sur la typologie des activités de changements logiciels de Chapin et al. (2001).

Ces deux éléments, taxonomie et typologie, nous permettent de localiser et de caractériser une évolution ainsi que les activités de changement qui en découlent.

La suite de cette section décrit les principaux éléments influençant l'évolution des systèmes d'information. Les lois de l'évolution de Lehman présentent de manière précise les causes et les

processus du phénomène d'évolution. Ces éléments nous indiquent quels sont les domaines à approfondir afin de les intégrer pour déterminer les facteurs déclencheurs d'évolution. Les éléments issus du domaine de l'évolution logicielle que nous avons retenu pour la sélection des facteurs déclencheurs d'évolution sont les suivants : le contexte d'utilisation, les parties prenantes, l'architecture logicielle et le contexte technique.

4.7.1 Le contexte d'utilisation

Selon la première loi de l'évolution de Lehman, les systèmes logiciels utilisés dans le monde réel (systèmes de type E) doivent être continuellement adaptés pour qu'ils restent satisfaisant dans leur environnement (M M Lehman & Ramil, 2001).

Cet environnement dans le cas d'un système d'information représente le contexte métier ou organisationnel d'utilisation du système. De nos jours, les compagnies sont confrontées à un marché international qui est en changement continu. Pour répondre à ces marchés en perpétuelle évolution, les compagnies doivent sans cesse adapter leur processus de travail. Afin de faire face à ces changements et pour saisir les nouvelles opportunités de marché, le système d'information supportant ces processus doit être capable d'évoluer de manière adéquate.

De plus, dans leur article caractérisant les différentes activités d'évolution logicielle, Chapin et al. (2001) spécifient un regroupement de types d'activités en fonction de leur impact sur les règles métiers (business rules cluster). L'environnement du système, au travers des besoins et des règles métiers a donc une influence certaine sur l'évolution de ce dernier.

Le domaine de recherche en systèmes d'information qui couvre ce besoin d'évolution en fonction du contexte est le domaine de l'alignement entre le métier et le système d'information. Enormément de recherches ont été conduites concernant l'alignement entre le contexte d'utilisation et le système d'information. Une des premières théories et certainement la plus connue concernant l'alignement métier et les systèmes d'information est le Strategic Alignment Model (SAM) développé par Henderson & Venkatraman (1993). Ce n'est que plusieurs années plus tard que sont apparues les premières méthodes permettant d'évaluer l'alignement entre le système d'information et le métier. The Strategic Alignment Maturity Assessment Method (Luftman, 2000) est un des premiers modèles mesurant le niveau d'alignement et conseillant sur des voies possibles d'amélioration.

Ce domaine de recherche va dans le même sens que les constats M M Lehman & Ramil (2001) et de Chapin et al. (2001). L'environnement d'utilisation du système étant une des principales raisons poussant le système d'information à évoluer, il est donc primordial d'intégrer cette dimension comme facteur pouvant déclencher des évolutions.

4.7.2 Les parties prenantes

L'élément suivant que nous déduisons des lois de Lehman quant aux raisons poussant les systèmes logiciels à évoluer se trouve dans la Loi 6 : « Les fonctionnalités d'un système de type E doivent être enrichies afin de maintenir la satisfaction des utilisateurs durant la durée de vie du système » ainsi que dans la Loi 8 : « Le processus d'évolution d'un système de type E constitue un système de feedback multi-niveaux, multi-boucles et multi-agents ».

Lehman démontre que les acteurs du système sont également une partie importante du processus d'évolution de ce dernier (M M Lehman & Ramil, 2001; Madhavji et al., 2006). En effet, de par leur utilisation, leur perception, leurs décisions concernant le système, les acteurs sont au centre du processus de feedback. Ce sont eux qui perçoivent les incohérences entre la

manière dont un problème est résolu dans le système et dans le monde réel. Ce fait est également mis en avant par Lientz et al. (1978) qui précisent que les demandes des utilisateurs pour des améliorations et des extensions du système logiciel constituent le domaine le plus important des problématiques de gestion de l'évolution logicielle.

De plus, l'évolution d'un système de type E est influencée par les processus organisationnels ainsi que par les processus d'ingénierie technique du système. Ces processus sont eux-mêmes affectés par les décisions, interprétations et actions des parties prenantes. La conception, l'installation et de l'exploitation du système peut influencer et changer le domaine dans lequel le système doit être fonctionnel (Cook et al., 2006).

Il est donc primordial d'intégrer les acteurs du système comme facteur pouvant déclencher des évolutions.

4.7.3 Le contexte technique

Au milieu des années 90, il était fréquent d'observer des prédictions de durée de vie d'environ 10 ans voir plus pour un système d'information. Actuellement l'ingénierie logicielle évolue extrêmement vite, ce qui est aujourd'hui la dernière technologie deviendra rapidement obsolète. Dans leur article « Software Maintenance and Evolution : A Roadmap, Bennett & Rajlich, (1999) soulignent que le problème de « software legacy » n'est pas prêt de s'arrêter.

Lorsqu'on évoque l'évolution d'un système, les plateformes logicielles et matérielles jouent un rôle critique. Les coûts relatifs à l'utilisation d'une technologie inappropriée peuvent augmenter de manière significative lorsqu'un changement est nécessaire. Le contexte technique d'un système d'information doit tenir compte d'éléments autant technique que managériaux relatifs à la technologie. Nous trouvons dans ce contexte des éléments tels que le niveau d'innovation de la technologie utilisée, la flexibilité, la portabilité, la dimensionnabilité (scalability) ainsi que des éléments tels que le degré d'anticipation des changements.

L'aspect technologique du système d'information d'une organisation est également un domaine de recherche riche. Nous y relevons notamment les recherches portant sur la gestion de la technologie mettant en avant le fait que l'identification, l'acquisition, le développement et l'exploitation des technologies sont nécessaires pour la performance d'une organisation (Phaal, Farrukh, & Probert, 2004a). D'autres domaines de recherche connexes comme celui abordant la stratégie technologique d'une organisation ou celui relatif aux enjeux des transitions technologiques renforce ce besoin de la dimension technologique à la caractérisation des évolutions des systèmes logiciels. Nous avons donc intégré cette dimension aux facteurs déclencheurs d'évolution.

4.7.4 L'architecture logicielle

L'incertitude associée avec les systèmes logiciels de type E a des implications sur l'architecture de ces systèmes. Les systèmes de type E sont sensés évoluer lors de leur utilisation, dès lors il est important pour les parties prenantes que le système soit adaptable à l'environnement. Dans beaucoup de cas, l'architecture d'un système est un facteur important de maintien de la satisfaction des utilisateurs lors d'évolutions (Cook et al., 2006). Lorsqu'un système évolue, la stratégie de l'organisation vis-à-vis de l'architecture du système impacte directement la manière dont ce système évolue. L'architecture du système contribue donc fortement au processus d'évolution du système.

D'autre part, chaque évolution du système peut impacter directement l'architecture de ce dernier. L'ajout de fonctionnalités ou l'intégration de nouveaux composants vont modifier et

complexifier l'architecture actuelle du système. Il est donc important que les parties prenantes à l'évolution aient une vision globale des effets de l'évolution sur l'architecture.

Tout comme le contexte d'utilisation du système, l'architecture des systèmes logiciels est un domaine de recherche riche. La littérature fournit beaucoup de modèles pour comprendre et qualifier l'architecture ainsi que l'intégration des applications d'entreprise (Enterprise application integration EAI). Nous trouvons parmi les modèles les plus réputés : le Brown's Conceptual Model of Integration (Brown, Carney, Morris, Smith, & Zarrella, 1994) ainsi que le Zachman's Enterprise Architecture Framework (Zachman, 1999).

L'architecture du système est non seulement impactée par les évolutions du système, mais elle va également influencer la manière dont le système va évoluer. Il est donc naturel d'intégrer la dimension architecture comme facteur déclencheur d'évolution afin de suivre et de caractériser les évolutions architecturales du système.

4.8 Facteurs déclencheurs d'évolution dans EVOLIS

Basé sur les études relatives à l'évolution ainsi que sur les différents domaines de recherche abordant des problématiques influençant l'évolution des systèmes, nous avons retenu et intégré dans la conception de notre cadre conceptuel les quatre facteurs décrits précédemment. Ces facteurs influencent de manière directe le déclenchement de l'évolution des systèmes d'information. Les facteurs déclencheurs d'évolution représentés dans le cadre conceptuel EVOLIS sont illustrés dans la Figure 9 et sont les suivants :

- L'alignement entre les processus métiers et le système d'information qui représente le contexte d'utilisation comme déclencheur d'évolution (Section 4.7.1).
- La satisfaction des utilisateurs, sous entendu ici tous les acteurs du système, qui représente les parties prenantes comme élément déclencheur d'évolution (Section 4.7.2).
- L'architecture du système d'information incluant les interactions entre les différents composants du système, qui représente les motivations concernant l'architecture logicielle comme élément déclencheur d'évolution (Section 4.7.4).
- La technologie utilisée pour implémenter le système d'information représentant le contexte technique, ou l'innovation technologique comme élément déclencheur d'évolution avec le désir d'introduire de telles innovations pour des raisons de performance ou de capacité par exemple (Section 4.7.3).

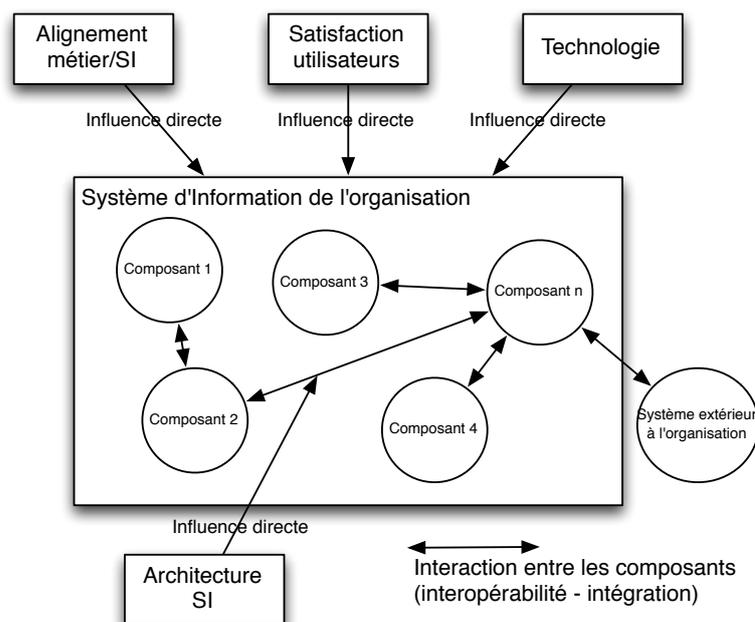


Figure 9. Facteurs représentés dans le cadre conceptuel EVOLIS influençant de manière directe l'évolution des systèmes d'information.

Dans la Figure 9, nous présentons schématiquement l'influence des facteurs d'évolution sur le système d'information de l'organisation. Les 4 facteurs déclencheurs d'évolution sont représentés autour du système d'information. Nous avons symbolisé le système d'information comme un système pouvant regrouper des composants. Pour le bon déroulement des activités métiers, les composants doivent interagir entre eux. Ces interactions sont représentées par des flèches bidirectionnelles entre les composants. Des composants sont dits interopérables lorsque leur couplage est faible et intégrés lorsque leur couplage est fort (d'avantages d'explications sont données dans la Section 5.3.3 Couplage des composants, intégration et interopérabilité). Les flèches entre les facteurs et le système d'information représentent la manière dont ces derniers agissent sur le système d'information. Par « influence directe », nous entendons que chacun des facteurs peut être la motivation qui va pousser l'organisation à faire évoluer le système. L'alignement métier, la satisfaction des utilisateurs, et la technologie auront une influence directe sur les évolutions et donc sur le cycle de vie du système d'information. L'architecture aura une influence directe non seulement sur le système mais surtout sur les interactions entre les composants ou lors d'un nouveau couplage de composants dans le système.

La sélection des facteurs s'est non seulement faite en se basant sur la littérature du domaine de l'évolution logicielle, mais également suite aux différents retours reçus par les réviseurs ainsi que lors de conférences auxquels ces travaux ont été présentés.

Le chapitre suivant (Chapitre 5) détaille le contenu de chaque facteur d'évolution représenté dans EVOLIS ainsi qu'un cinquième élément permettant de qualifier les évolutions à savoir l'investissement effectué pour une évolution. Nous avons nommé ce cinquième élément : le coût de l'évolution.

4.9 Notre apport à l'étude de l'évolution logicielle

Le domaine de l'évolution logicielle est décrit de manière riche dans la littérature, cependant un outil permettant l'étude et l'analyse de l'évolution globale d'un système d'information manque

aux contributions académiques du domaine. Les études du domaine se concentrent sur la qualification des évolutions, du processus d'évolution et des activités d'évolution. Ce sont là des éléments impératifs pour la compréhension et l'étude des évolutions. Néanmoins, les éléments déclencheurs, les causes des évolutions ne sont que rarement détaillées dans la littérature du domaine (Godfrey & German, 2008).

Une étude de cette littérature et une mise en commun des travaux de plusieurs chercheurs permet de se faire une première image des éléments de l'environnement du logiciel ou du logiciel lui-même qui motivent les évolutions logicielles.

Notre apport principal au domaine de l'évolution logicielle est la création d'un cadre conceptuel basé sur les connaissances du domaine pour étudier de manière globale les évolutions. Par manière globale, nous nous entendons l'étude des niveaux micro et macro du processus d'évolution, ainsi que l'étude des facteurs déclencheurs d'évolution. Ce cadre conceptuel regroupe et utilise les connaissances du domaine pour étudier les évolutions et apporte des éléments nouveaux sur les facteurs déclencheurs d'évolution.

Le Tableau 3 détaille visuellement les éléments de la littérature, à savoir la base de connaissances, sur lesquels nous nous sommes appuyés pour cette recherche ainsi que le gap que nous avons adressé avec nos contributions. Ces deux éléments (base de connaissance ainsi que gap et contributions) sont présentés selon deux axes : l'axe théorique ainsi que l'axe relatif aux méthodes et techniques du domaine de l'évolution.

Avec cette recherche nous comblons un principal gap théorique à savoir : les facteurs déclencheurs d'évolution. La contribution qui comble ce gap est regroupée les 4 principaux facteurs déclencheurs d'évolution pour les systèmes d'information. Ces derniers sont présentés dans la Section 4.7 ainsi que plus en détail dans le Chapitre 5. Concernant les méthodes et les techniques d'étude de l'évolution des systèmes d'information, nous avons combiné plusieurs éléments de la littérature avec les facteurs déclencheurs d'évolution dans le but de créer une méthode longitudinale d'étude de l'évolution (décrite dans le Chapitre 6). Aucune méthode similaire n'est présente dans la littérature, nous comblons donc ce gap afin de pouvoir étudier de manière longitudinale et systématique les évolutions des systèmes d'information. De plus, afin d'appliquer cette méthodologie sur un volume important de données relatives aux évolutions, nous présentons un modèle conceptuel pour organiser et traiter les données analysées pour l'étude de l'évolution (présenté dans la Section 6.6).

Ces nouvelles contributions sont rassemblées dans l'artefact produit par cette recherche à savoir : le cadre conceptuel d'étude de l'évolution des systèmes d'information nommé EVOLIS et décrit dans les chapitres suivants.

Tableau 3. Résumé des théories et méthodes Gap et contributions

	Base de connaissances	Gap et contributions
Théories	Lois de l'évolution Typologie des systèmes : SPE Cycle de vie logiciel Processus d'évolution	Facteurs déclencheurs d'évolution <ul style="list-style-type: none"> • Alignement métier et SI • Satisfaction Utilisateurs • Architecture • Technologie
Méthodes et techniques	Typologie des activités d'évolution Taxonomie des évolutions Description micro et macro du processus d'évolution	Méthode longitudinale d'étude de l'évolution logicielle <ul style="list-style-type: none"> • Analyse des évolutions selon les facteurs déclencheurs • Analyse des activités d'évolution Modèle conceptuel de données pour l'étude de l'évolution

Chapitre 5

Design du cadre conceptuel pour l'étude de l'évolution EVOLIS

Ce chapitre présente l'outil que nous avons conçu pour l'étude de l'évolution des systèmes d'information. Cet outil est un cadre conceptuel que nous avons nommé EVOLIS. EVOLIS est l'acronyme de EVOLution of Information Systems. EVOLIS sert à l'étude de l'évolution des systèmes d'information. Il permet d'analyser les évolutions en fonctions des motivations de ces dernières et des activités entreprises pour évoluer.

EVOLIS regroupe plusieurs éléments notamment : une description des facteurs déclencheurs d'évolution, une méthodologie systématique d'étude des évolutions ainsi qu'un modèle conceptuel permettant de soutenir sa mise en œuvre. Ces trois composants d'EVOLIS sont représentés graphiquement dans la Figure 10 et décrits dans les chapitres suivants :

Ce Chapitre 5 débute par la présentation des différentes populations susceptibles d'utiliser le cadre conceptuel, s'en suit la présentation des besoins pour l'étude de l'évolution et finalement le détail du contenu des facteurs déclencheurs d'évolution.

Le Chapitre 6 présente la méthodologie d'étude de l'évolution ainsi que le modèle conceptuel nécessaire à la mise en œuvre d'EVOLIS.

Ce Chapitre 5 est une adaptation de l'article « EVOLIS: A Framework for Evaluating Evolution of Information Systems » (Métrailler & Estier, 2012a) tenant compte des différents retours obtenus et également des applications d'EVOLIS.

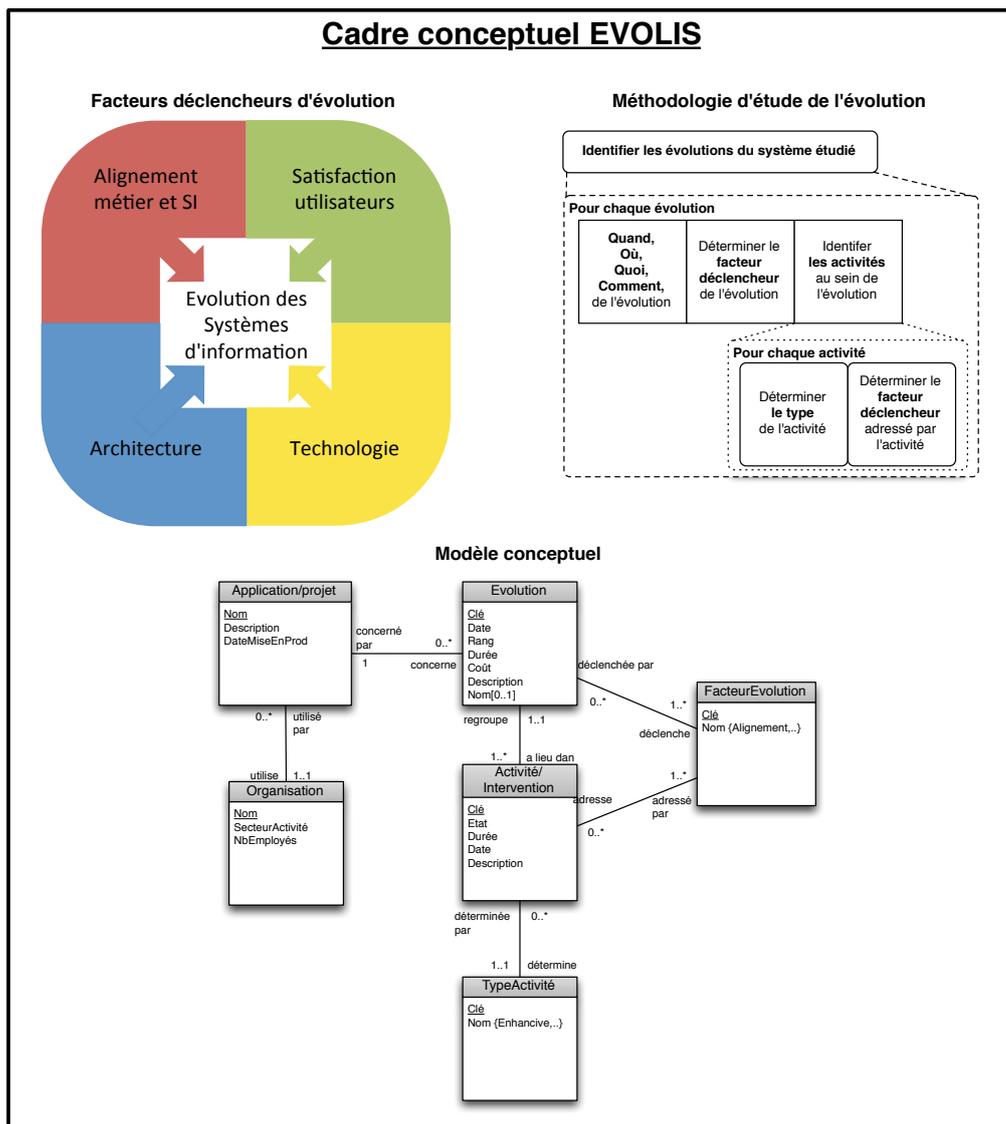


Figure 10. Composants du cadre conceptuel EVOLIS.

5.1 Présentation des contextes

Notre cadre conceptuel (appelé ici EVOLIS) s'adresse à deux populations : les chercheurs ainsi que les praticiens. Nous détaillons dans les sections suivantes les deux contextes d'utilisation d'EVOLIS.

5.1.1 Présentation du contexte métier

L'évolution d'un système d'information est incontournable et peut être issue de beaucoup de facteurs différents. Par exemple, des bugs qui doivent être corrigés, des utilisateurs qui demandent qu'un processus soit adapté, de nouvelles opportunités de marché qui nécessitent de nouvelles fonctionnalités, un nouveau niveau de performance que le système doit atteindre, des changements techniques dans l'environnement avec lequel le système interagit, l'obsolescence de certaines applications, et ainsi de suite.

Considérons l'exemple dans lequel, afin d'adresser un nouveau besoin métier, une organisation réalise qu'un composant spécifique de son système d'information doit interagir avec un système externe à l'organisation. La motivation principale de cette évolution sera l'alignement du système d'information avec le métier de l'organisation.

Une telle évolution englobe généralement des activités telles que : l'étude des besoins spécifiques des utilisateurs, l'étude du système logiciel externe à l'organisation, l'implémentation de l'interface de communication entre les deux systèmes, les tests du couplage des deux systèmes, l'évaluation de la nouvelle interface par les utilisateurs, la formation des utilisateurs, la mise à jour de la documentation, et ainsi de suite.

Le gestionnaire en charge de ce projet peut vouloir évaluer et quantifier selon plusieurs aspects l'impact de cette évolution sur le système. La première ébauche du cadre conceptuel EVOLIS est née suite à des discussions informelles avec des praticiens et des gestionnaires de système d'information.

Nous avons imaginé un cadre conceptuel pour étudier l'évolution qui :

- Doit être facilement compréhensible et rapidement mis en pratique pour les acteurs responsables de l'évolution du système d'information.
- Doit tenir compte des problématiques utilisateurs, comme par exemple la perception et l'efficacité des utilisateurs suite à l'utilisation du système d'information.
- Doit prendre en considération des facteurs objectifs comme le niveau de maturité technologique, l'alignement du système avec le métier qu'il supporte, son architecture ainsi que le niveau d'intégration des différents composants.
- Doit également permettre de situer globalement le coût des interventions et du système.

Le premier objectif visé par la création d'EVOLIS est donc d'épauler les praticiens en charge de l'évolution du système d'information. L'utilisation d'EVOLIS doit permettre de générer les connaissances nécessaires au pilotage de l'évolution du système d'information. Il doit aider les parties prenantes à créer la stratégie d'évolution du système. EVOLIS peut être interprété comme un tableau de bord de pilotage de l'évolution, il doit contenir les indicateurs pour comprendre dans quelle direction le système évolue, et pour mieux organiser les futures évolutions.

Dans ce dessein, nous avons conçu EVOLIS de manière à ce qu'il permette de comprendre et d'analyser les évolutions du système d'information selon des axes pluridisciplinaires. EVOLIS intègre donc des éléments qui jusqu'à présent étaient adressés individuellement.

Une analyse détaillée des activités entreprises pour une évolution permet de mieux détailler les impacts d'une évolution. Une meilleure compréhension du type d'impacts des évolutions permet de mieux contrôler les évolutions ainsi que leurs étendues sur les composants du système d'information et donc de minimiser les risques encourus.

5.1.2 Présentation du contexte académique

Le second objectif d'EVOLIS est l'étude a posteriori des évolutions des systèmes d'information. Pour cet objectif, EVOLIS permet au chercheur, comme au praticien, de retracer la vie des systèmes d'information et de localiser ces derniers dans leur cycle de vie.

Nous voudrions avec cette recherche être capable de retracer les activités d'évolution entreprises tout au long de la vie du système et de caractériser ces activités en fonction de critères particuliers. Les différents types d'activités d'évolution entreprises lors de l'exploitation d'un système ainsi que leur chronologie peuvent permettre de situer ce système dans son cycle

de vie et de mieux comprendre son processus d'évolution (Bennett & Rajlich, 2000b; Godfrey & German, 2008; Mens et al., 2005; Rajlich, 2006).

Cette classification chronologique et typologique des activités d'évolution aide à mieux comprendre les tenants et aboutissants de l'évolution des systèmes d'information. De plus, une étude longitudinale des activités d'évolution peut permettre de révéler des modèles spécifiques d'évolution mettant en exergue les différents types d'activités d'évolution pratiquées selon les différentes étapes du cycle de vie des systèmes.

5.2 Besoins couverts par le cadre conceptuel EVOLIS

Présentation des besoins des deux contextes : Les besoins relevés lors de nos discussions avec les praticiens et nos besoins pour l'étude de l'évolution ont des similitudes. Cependant la mesure pour qualifier ces besoins peut varier selon que l'on soit un praticien ou un académicien. En effet, si on prend par exemple le coût d'une évolution, le praticien aura tendance à préférer une mesure financière tandis que le chercheur sera davantage intéressé par une mesure permettant des comparaisons comme les jours/hommes.

Autant dans le contexte métier que dans le contexte scientifique, les principaux besoins pour l'étude et la compréhension des évolutions sont les suivants :

- Connaître le déroulement chronologique des évolutions du système d'information ;
- Connaître le type des raisons qui sont à l'origine de l'évolution ;
- Déterminer l'importance de l'évolution ;
- Connaître le type des activités entreprises pour accomplir cette évolution.

5.2.1 Chronologie des évolutions

Pour retracer et comprendre l'évolution d'un système, le fait de connaître le moment des évolutions est primordial. Pour les praticiens, il permet de représenter chronologiquement toutes les actions entreprises lors de la vie du système et d'avoir une meilleure compréhension de l'état dans lequel se trouve leur système. Cette mémoire chronologique des évolutions peut permettre également un meilleur pilotage et une planification plus précise des activités d'évolution.

Pour les académiciens, cette chronologie est primordiale afin de comprendre les étapes par lesquelles sont passés les systèmes d'information étudiés. De plus une étude longitudinale des évolutions de plusieurs systèmes d'information permet de mettre en exergue des modèles d'évolution similaires parmi les systèmes étudiés.

La classification chronologique des évolutions est une activité qui apparaît dans plusieurs études concernant les évolutions des systèmes d'information (Bennett & Rajlich, 2000b; Buckley et al., 2005; Sabherwal & Hirschheim, 2001). Dès qu'une étude est longitudinale, une telle classification est indispensable pour représenter le cycle de vie du système.

5.2.2 Facteurs déclencheurs d'évolution (motivations)

Afin d'en connaître plus sur les évolutions des systèmes d'information, autant les praticiens que les chercheurs sont intéressés par connaître quelles sont la ou les raisons qui poussent une organisation à entreprendre une évolution. Est-ce un enjeu métier, technologique, etc. qui occasionne une évolution ?

EVOLIS doit donc tenir compte des éléments déclencheurs d'évolution. Il doit être en mesure de répondre à la question : quel est la nature de l'élément déclencheur d'évolution ?

Pour chaque intervention sur le système, EVOLIS permet de qualifier la nature de la problématique qu'adresse une l'évolution.

Concernant les facteurs déclencheurs d'évolution, la littérature du domaine n'identifie pas de manière précise ces facteurs. Nous avons effectué une revue de la littérature afin de préciser les facteurs qui semblent les plus pertinents pour l'étude des facteurs déclencheurs d'évolution (Sections 4.7 et 4.8).

Chaque facteur d'évolution représenté dans EVOLIS est décrit dans la Section 5.3

5.2.3 Typologie des activités d'évolution (comment)

Lorsqu'une évolution est opérée sur le système d'information, cette dernière se compose généralement de plusieurs activités relatives à des domaines différents. En effet, une évolution peut contenir une grande diversité d'activités comme des activités techniques de développement, des activités de formation des utilisateurs, des activités de documentation et ainsi de suite.

Il est alors important autant pour le domaine métier que pour le domaine académique de connaître de quel type d'activités se compose une évolution. EVOLIS doit donc permettre de qualifier les activités entreprises lors de chacune des évolutions. Il doit être en mesure de répondre à la question : Quelle est la nature des activités entreprises pour effectuer une évolution ?

Pour répondre de manière précise à ce besoin, la littérature du domaine offre plusieurs classifications des activités d'évolution. Les Section 4.6.3 et 6.4 présentent la typologie que nous avons décidé d'intégrer dans le design d'EVOLIS.

5.2.4 Importance de l'évolution

Les recherches montrent qu'à la fois praticiens et chercheurs s'entendent sur le fait que les investissements dans le système d'information doivent être justifiés, mesurés et contrôlés (Milis & Mercken, 2004).

Afin de déterminer l'ampleur d'une évolution, nous avons choisi d'utiliser la notion de coût d'une évolution. Par le coût nous entendons les moyens mis en œuvre pour accomplir une évolution.

Dans le contexte métier, la qualification du coût d'une évolution est importante. Cette qualification peut se traduire en coût monétaire si l'organisation externalise l'administration de son système d'information ou en coût humain lorsque les évolutions sont exécutées en interne. Cette mesure de coût permet de qualifier une évolution vis-à-vis des ressources investies dans le système d'information. Le mot « coût » a été fréquemment utilisé dans cette partie, il ne faut pas oublier qu'un système d'information est un investissement, il a non seulement un coût mais également une valeur. Nous avons utilisé le terme « coût » car il semble plus explicite que le terme « valeur ». Cependant les deux termes peuvent se substituer l'un à l'autre.

D'un point de vue académique le plus important n'est pas le coût monétaire d'une évolution. En effet selon que l'évolution est externalisée ou non, ce coût peut varier de manière significative. Nous avons donc besoin d'une mesure unique permettant de comparer l'ampleur des différentes évolutions d'un système. Pour ce faire, nous avons choisi comme mesure le temps de travail accordé à l'évolution en jours/homme.

5.3 Facteurs d'évolution d'EVOLIS

La Section 4.8 présentait les recherches ainsi que les domaines sur lesquelles nous nous sommes basés afin de déterminer les facteurs pouvant déclencher des évolutions et qui devaient entrer dans la conception du cadre conceptuel EVOLIS pour l'étude de l'évolution. Nous détaillons ici les facteurs déclencheurs d'évolution représentés dans EVOLIS comme le résume la Figure 11. En plus des quatre facteurs influençant la vie d'un système d'information présentés dans la suite de ce chapitre, nous avons également relevé un cinquième élément important dans la vie de ce dernier : le coût. En effet, que ce soit pour les praticiens ou les chercheurs, une notion permettant de quantifier l'investissement nécessaire pour les évolutions est primordiale. Pour les praticiens ce peut être un frein à l'évolution de leur système et donc également un facteur influençant l'évolution. Pour les chercheurs et également les praticiens, il est nécessaire d'avoir un indicateur permettant de comparer plusieurs évolutions entre elles. Le coût ou l'investissement de la part de l'organisation dans une évolution permet d'estimer l'ampleur ou la taille d'une évolution.

Un point fort de ce cadre conceptuel est qu'il intègre et permet la comparaison d'éléments (les facteurs d'évolution) issus de domaines qui sont généralement étudiés séparément.

Alignement métier et système d'information

Evolution déclenchée par l'adéquation opérationnelle du système d'information :

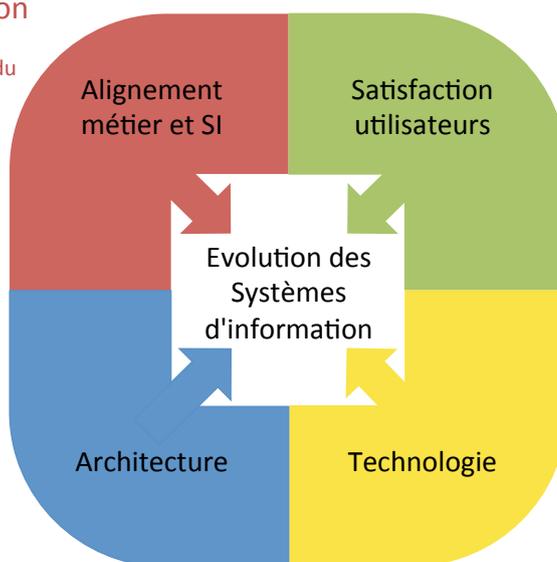
- Alignement interne
- Alignement externe
- Alignement orienté évolution

Architecture

Evolution déclenchée par un besoin d'intégration et d'interopérabilité :

- Entre composants du système
- Entre fonctionnalités
- Avec l'environnement

Adéquation stratégique du système d'information



Satisfaction utilisateurs

Evolution déclenchée par le feedback des parties prenantes (utilisateurs, acteurs) quant à l'adéquation du système :

- Facilité d'utilisation
 - Utilité perçue
- Implication des parties prenantes
- Aptitudes du système pour les parties prenantes

Technologie

Evolution déclenchée par de l'innovation technologique :

- Obsolescence technique
- Performance
- Dimensionnabilité
- Anticipation technologique

Figure 11. Facteur déclencheurs d'évolution représentés dans EVOLIS.

Le cadre conceptuel EVOLIS permet donc d'étudier les évolutions en fonction des motivations à l'origine de ces dernières mais pas seulement. La Figure 12 présente des évolutions de systèmes d'information déclenchées par chacun des facteurs d'évolution afin d'illustrer de manière pratique quels genres d'évolutions sont déclenchées par chaque facteur. En estimant l'ampleur de chacune de ces interventions sur les systèmes en fonction de leur coût ou d'une autre mesure comme l'investissement en ressources (jours/homme par exemple), il devient possible d'avoir une vue d'ensemble sur lequel de ces facteurs nécessite le plus « d'effort d'évolution ».

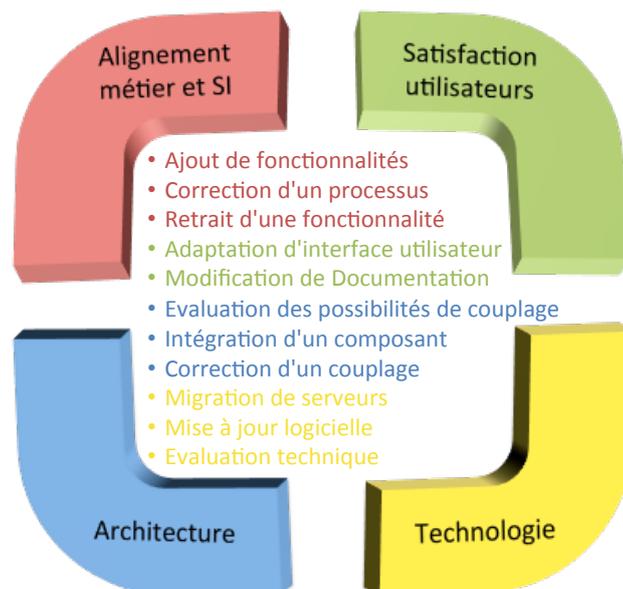


Figure 12. Exemple d'interventions d'évolution en fonction des facteurs déclencheurs.

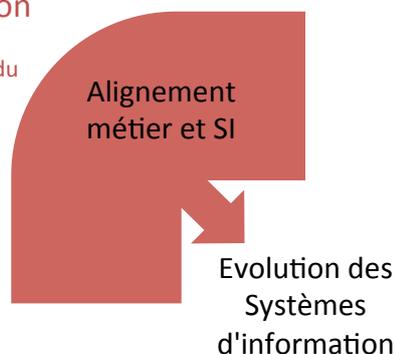
Les Sections suivantes présentent en détail chacun des facteurs d'évolution, les illustrent avec des exemples et font le corollaire avec les différents domaines de recherche en systèmes d'information d'où chacun d'eux puise ses origines.

5.3.1 Alignement métier et système d'information

Alignement métier et système d'information

Evolution déclenchée par l'adéquation opérationnelle du système d'information :

- Alignement interne
- Alignement externe
- Alignement orienté évolution



Ce facteur qualifie les évolutions qui sont déclenchées par des considérations d'alignement entre le système d'information et les processus organisationnels.

Une évolution est dite déclenchée par le facteur Alignement métier et système d'information si cette dernière est déclenchée par une motivation visant à améliorer l'adéquation entre les processus du système d'information et les processus organisationnels.

Nous distinguons plusieurs niveaux d'alignement, l'alignement interne entre les processus opérationnels et le système d'information, l'alignement externe qui vise à augmenter l'adéquation des processus avec l'environnement externe avec lequel le système cohabite et l'alignement orienté évolution qui a pour but d'identifier les évolutions déclenchées par une volonté d'anticipation des changements futurs.

L'idéal pour le pilotage de l'évolution du système est de détecter quel est le niveau d'alignement qui déclenche l'évolution. En effet, être en mesure de déterminer quel est le niveau d'alignement qui est adressé par une évolution permet de connaître le degré avec lequel l'alignement est effectué et si toutes les formes d'alignement sont adressées ou non.

Alignement opérationnel interne

Le but de la « perspective opérationnelle d'alignement » est de décrire l'évolution de l'adéquation entre les processus métier et les processus tels qu'ils sont implémentés dans le système d'information. Les évolutions sont classées comme « alignement opérationnel » lorsqu'elles impactent les processus concernant à la fois des fonctionnalités de support couvertes par le système et des fonctionnalités spécifiques au métier de l'organisation. En d'autres termes, la « perspective d'alignement opérationnelle » caractérise toutes modifications du système d'information de l'organisation afin qu'il corresponde au mieux aux processus organisationnels.

Exemples

Afin d'illustrer les évolutions déclenchées par ce facteur d'évolution, voici deux exemples :

- Un processus organisationnel nécessite davantage de données pour être effectué. Une évolution déclenchée par ce besoin typique d'alignement serait par exemple la modification d'un formulaire et d'une table de la base de données afin que l'utilisateur puisse saisir les informations requises et que ces dernières soient mémorisées.
- A plusieurs reprises un processus ne s'est pas terminé correctement et cette erreur s'est répercutée plus loin dans l'organisation. Une évolution déclenchée par un besoin d'alignement afin d'éviter cette erreur serait l'introduction d'une vérification et d'un message d'erreur ajoutés à fin du processus pour informer l'utilisateur de la bonne terminaison ou non de ce dernier.

Fondements

Camponovo & Pigneur (2004) suggèrent de considérer trois niveaux d'alignement. Le premier niveau correspond à ce que l'on considère généralement comme étant la notion d'alignement entre le système d'information et les buts, les activités et la stratégie de l'organisation. Le second niveau prend en compte l'alignement du système d'information avec l'environnement externe à l'organisation et assume que le système doit intégrer des fonctionnalités pour évaluer cet environnement. Le troisième niveau d'alignement englobe l'évolution dans le temps et souligne la nécessité de concevoir et d'adapter le système d'information afin qu'il soit en mesure d'évoluer en fonction des changements futurs de l'organisation et de son environnement. Malgré le fait que la notion classique d'alignement entre le système d'information et les stratégies métiers reste la première étape pour aborder l'alignement, les deux autres niveaux sont essentiels en ce qui concerne la complexité et l'incertitude présents dans les environnements métiers.

Les recherches ont montré que l'alignement du métier et du système d'information contribue à améliorer la performance organisationnelle (Chan, Huff, Barclay, & Copeland, 1997). Cette étude se base sur le « Strategic Alignment Model (SAM) » (Henderson & Venkatraman, 1993) qui identifie 4 domaines : la stratégie métier, la stratégie du système d'information, l'infrastructure organisationnelle et l'infrastructure technologique (illustré dans la Figure 13). Le SAM identifie deux types d'alignement entre le domaine métier et le système d'information. Le premier type, nommé intégration opérationnelle, adresse l'infrastructure organisationnelle et l'infrastructure du système d'information. Il concerne la cohérence entre les besoins et les attentes organisationnels et les capacités de l'infrastructure du système d'information. C'est ce type

d'intégration que couvre le facteur déclencheur d'évolution Alignement. Le second type, nommé intégration stratégique, représente le lien entre la stratégie métier et la stratégie du système d'information. Il concerne la capacité des fonctionnalités du système d'information à s'adapter et supporter la stratégie métier. Ce second type est couvert par le facteur déclencheur d'évolution Architecture présenté dans la Section 5.3.3.

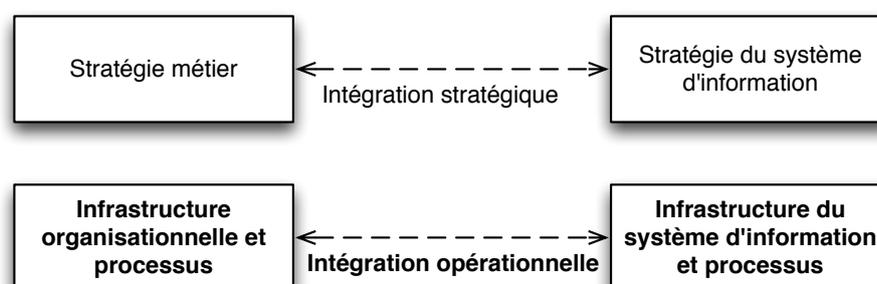


Figure 13. Composants majeurs du Strategic Alignment Model, intégration opérationnelle (Henderson & Venkatraman, 1993).

Alignement avec l'environnement externe

Ce niveau d'alignement caractérise les évolutions qui visent à améliorer l'adéquation du système avec l'environnement externe dans lequel ce dernier exerce une activité. En effet, la performance du système d'information progresse si ce dernier est conscient de l'environnement avec lequel il doit collaborer et évolue dans l'optique de développer cette collaboration. Il est donc non négligeable pour le système d'information d'intégrer cette perspective environnementale d'alignement. Une évolution du système d'information dans cette perspective est un indicateur de la volonté de rester attentif à l'environnement. EVOLIS doit donc caractériser ce type d'évolution orientée alignement avec l'environnement. Toute évolution visant à intégrer au système d'information des outils favorisant la veille de l'environnement sera donc caractérisée de ce type.

Exemples

Afin d'illustrer les évolutions déclenchées par ce facteur d'évolution, voici deux exemples :

- Modification d'un processus interne du système d'information afin de récupérer plus rapidement les informations concernant les prix pratiqués par l'organisation ceci dans le but de faciliter la veille concurrentielle des prix.
- Modification d'un processus afin de s'adapter à une nouvelle exigence légale.

Fondements

Faire évoluer le système d'information afin qu'il supporte au mieux les processus métiers de l'organisation est une étape essentielle dans la recherche de performance. Selon Camponovo & Pigneur (2004), le second niveau d'alignement du système d'information concerne l'environnement externe de l'organisation. Evaluer et surveiller l'environnement dans lequel une organisation exerce ses activités est une activité cruciale pour assurer à l'organisation une performance continue de ses activités.

Beaucoup d'études et de modèles sont à disposition des gestionnaires afin de les aider à évaluer leur environnement et améliorer leur alignement avec ce dernier. Pour n'en citer que quelques uns : la matrice SWOT qui permet de déterminer une stratégie en fonction des opportunités et des menaces actuelles et futures présentes dans l'environnement, ainsi que les forces et faiblesses du domaine d'activité stratégique. De façon similaire, le modèle des cinq forces de

Porter recommande de formuler la stratégie en considérant la structure concurrentielle de l'environnement organisationnel déterminée par les pressions des clients, des fournisseurs, des concurrents, les potentiels entrants sur le marché ainsi que les produits de substitution (Porter, 1979).

De plus, un grand nombre d'études démontrent l'importance d'évaluer et de surveiller l'environnement et suggèrent qu'il existe une relation positive avec les performances organisationnelles (Choo, 2001).

Alignement orienté évolution

Ce niveau d'alignement caractérise les évolutions déclenchées par une motivation d'anticipation des changements futurs auxquels le système d'information devra faire face. Ces évolutions ne sont pas forcément perceptibles par les utilisateurs mais sont effectuées pour simplifier ou favoriser une évolution future.

Toute évolution qui tient compte de scénarios futurs, ou qui est entreprise dans le but de faciliter un prochain changement doit être caractérisée par EVOLIS comme étant une évolution d'alignement de type orienté évolution.

Exemple

Afin d'illustrer les évolutions déclenchées par ce facteur d'évolution, voici un exemple :

- L'organisation s'apprête à acquérir un nouveau système de gestion de stocks. Afin d'anticiper le déploiement du système dans l'organisation, certains processus qui touchent à la gestion des stocks (vente, approvisionnement, etc.) sont modifiés progressivement afin de faciliter la mise en place du futur système.

Fondements

Lorsque l'alignement avec la stratégie organisationnelle et l'alignement avec l'environnement sont effectués, l'activité d'alignement n'est pas terminée. En effet, les multiples changements dans l'environnement ou dans la stratégie de l'organisation nécessitent un réalignement constant du système d'information. Lors de la conception du système d'information, il est important d'essayer d'incorporer des fonctionnalités permettant d'intégrer les potentielles évolutions futures. C'est particulièrement le cas dans des environnements incertains et très dynamiques (Camponovo & Pigneur, 2004).

Dans ces environnements, une technique d'anticipation possible est l'élaboration de scénarios d'évolution. L'approche de prévision basée sur les scénarios est adaptée pour aborder des environnements incertains et complexes (O'Brien, 2004). Ces scénarios fournissent un moyen de gérer et de concevoir à l'avance des systèmes d'information tenant compte de multiples possibilités d'évolutions. Des systèmes conçus de la sorte peuvent prétendre à une plus longue période d'alignement avec la stratégie et l'environnement de l'organisation.

5.3.2 Satisfaction des utilisateurs



Par utilisateurs, nous entendons toutes les parties prenantes dans la conception, l'exploitation et l'utilisation du système d'information.

Toute évolution ayant comme but principal d'améliorer la satisfaction des acteurs du système (autant les utilisateurs finaux que les administrateurs, développeurs, gestionnaires, etc.) doit être caractérisée par EVOLIS comme étant une évolution orientée satisfaction utilisateurs. D'autre part, afin d'étudier a posteriori les données d'évolution de plusieurs études de cas, nous classons chaque activité durant une évolution comme étant orientée satisfaction utilisateur si cette dernière concerne directement les acteurs du système ou lorsque cette dernière modifie uniquement la relation des acteurs avec le système sans modifier les fonctionnalités métiers.

Durant toute la phase d'analyse des besoins et de mise en place d'une évolution logicielle, nous retrouvons donc plusieurs types d'activités destinées à favoriser la satisfaction future des utilisateurs ainsi que leur propension à accepter ou à rejeter l'évolution.

Cette classification permet aux gestionnaires du système d'information de s'assurer que les futurs acteurs aient été suffisamment consultés et mis à contribution pour le développement de la future évolution du système.

Exemples

Afin d'illustrer les évolutions déclenchées par ce facteur d'évolution, voici deux exemples :

- Evolution des éléments graphiques de l'interface utilisateurs et ajout de raccourcis claviers spécifiques sans modifier les processus organisationnels. Cette évolution est sensée augmenter la performance et la satisfaction des utilisateurs.
- Nettoyage du code source d'un système spécifique à l'organisation et ajout de commentaires. Cette évolution, qui n'a aucune répercussion sur les processus organisationnels, est sensée favoriser la navigation dans le code source et sa compréhension.

Fondements

Les utilisateurs sont importants dans le processus d'évolution des systèmes logiciels. En effet, beaucoup de chercheurs mentionnent que les motivations principales des évolutions sont les feedbacks ainsi que les propositions de changement issus des utilisateurs (Tuan, Vo, & Elsner, 2007).

Le domaine de recherche relatif aux usages des systèmes logiciels par les utilisateurs finaux ainsi que la satisfaction procurée par ces usages foisonne de publications. The Technology Acceptance Model (TAM) (Davis, 1989) est certainement la théorie la plus célèbre à ce sujet, ceci est dû

autant à ses nombreuses mises en pratique qu'à ses critiques (Benbasat & Barki, 2007). Cette théorie modèle la façon dont les utilisateurs acceptent et utilisent les technologies. Le modèle suggère que lorsqu'un utilisateur est confronté à une nouvelle technologie certaines variables vont déterminer la propension de ce dernier à utiliser la technologie. Cette propension à accepter ou à rejeter une nouvelle technologie est mesurée en utilisant les facteurs : utilité perçue (*perceived usefulness*) et la facilité d'utilisation perçue (*perceived ease of use*). Les auteurs de TAM ont étendu le modèle original en un nouveau modèle nommé TAM2, afin d'expliquer l'utilité perçue et les intentions d'utilisation en s'appuyant sur les influences sociales (normes subjectives, volontariat, image) et les processus cognitifs (expérience, qualité du résultat, utilité de la tâche, etc.) (Venkatesh & Davis, 2000). Plus récemment, le modèle TAM3 a été proposé dans lequel un réseau nomologique compréhensif de l'adoption et de l'usage des technologies a été développé (Venkatesh & Bala, 2008).

Ces deux facteurs *utilité perçue* et *facilité d'utilisation* sont des indicateurs permettant aux gestionnaires de systèmes d'information d'évaluer l'acceptation ainsi que les bénéfices possibles d'une évolution du système d'information.

Beaucoup de recherches connexes aux recherches basées sur TAM présentent les facteurs influençant la satisfaction des utilisateurs finaux comme (Doll & Torkzadeh, 1988; Mahmood, Burn, Gemoets, & Jacquez, 2000). Certaines études sont d'avantages axées sur l'impact des propriétés du logiciel dans son rendu de l'information (contenu, précision, format, ponctualité de l'information et facilité d'utilisation) sur la satisfaction des utilisateurs (Doll & Torkzadeh, 1988), tandis que d'autres incluent des aspects de l'environnement comme facteurs de satisfaction. Dans cette seconde optique, un article de (Mahmood et al., 2000) analyse les résultats empiriques de 45 études publiées entre 1986 et 1998 en étudiant les relations entre la satisfaction des utilisateurs et 9 variables décrites dans la Figure 14. Cette analyse révèle une influence positive des 9 variables mais à des degrés divers. Les relations les plus significatives concernent la participation des utilisateurs dans le développement des systèmes, l'utilité perçue, l'expérience de l'utilisateur, le soutien organisationnel et l'attitude de l'utilisateur envers le système d'information.

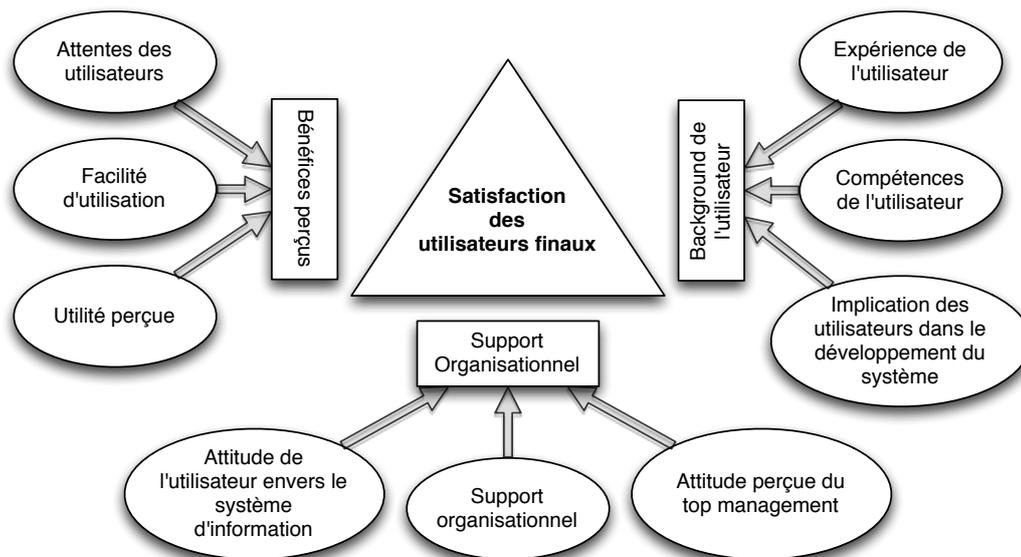


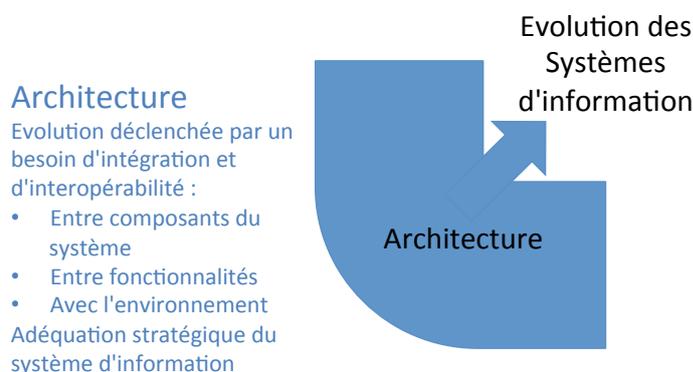
Figure 14. Facteurs de satisfaction des utilisateurs finaux adapté de Mahmood et al. (2000).

Ces études démontrent que non seulement les propriétés relatives au logiciel sont importantes dans la satisfaction des utilisateurs, mais que la préparation en amont de la mise en place du logiciel joue un également un rôle dans la satisfaction des utilisateurs.

Après la mise en service de l'évolution logicielle, il existe des outils afin d'évaluer la satisfaction des utilisateurs de manière plus rapide et pragmatique que ce que proposent les recherches sur TAM et ses dérivés. Notons par exemple l'article « A lightweight, industrially-validated instrument to measure user satisfaction and service quality experienced by the users of a UML modeling tool » (Islam, Koivulahti-Ojala, & Käkölä, 2010) proposant un instrument léger pour mesurer la satisfaction et la qualité de service perçue par les utilisateurs. Cet instrument fournit une bonne indication sur la mise en place d'un nouveau service et peut être facilement appliqué par les gestionnaires dans le but d'obtenir des feedbacks sur les évolutions du système d'information. Cet instrument prend la forme d'une dizaine de questions précises relatives à la satisfaction des utilisateurs et à la qualité du service fournis par le système (adapté de Islam et al. (2010)) :

- Q1. Dans quelle mesure êtes-vous satisfait de la vitesse du <Système à évaluer>?
- Q2. Dans quelle mesure êtes-vous satisfait de la disponibilité du <Système à évaluer>?
- Q3. Dans quelle mesure êtes-vous satisfait de la facilité d'utilisation du <Système à évaluer>?
- Q4. Dans quelle mesure êtes-vous satisfait avec les instructions et les guides de l'utilisateur disponibles pour le <Système à évaluer>?
- Q5. Si nécessaire, j'obtiens un support rapide et professionnel ?
- Q6. Dans quelle mesure êtes-vous satisfait de la formation disponible pour le <Système à évaluer>?
- Q7. Dans quelle mesure le <Système à évaluer> répondre à vos besoins ?
- Q8. Dans l'ensemble, êtes-vous satisfait avec le <Système à évaluer>?
- Q9. Combien de fois utilisez-vous le <Système à évaluer> (quotidiennement, hebdomadaire, mensuel, moins qu'une fois par mois)?
- Q10. Donnez s'il vous plait votre avis sur le <Système à évaluer> (améliorations, idées de développement)

5.3.3 Architecture



Toute évolution ayant comme principal déclencheur un aspect architectural, d'intégration ou de coordination de différents composants ou fonctionnalités du système d'information doit être caractérisée par EVOLIS comme étant une évolution orientée architecture. D'autre part, afin d'étudier a posteriori les données d'évolution de plusieurs études de cas, nous classons chaque activité durant une évolution comme étant orientée architecture si cette dernière concerne l'intégration ou l'interopérabilité de différents composants ou fonctionnalités du système d'information.

L'observation des différentes évolutions du point de vue architectural permet au gestionnaire du système d'information d'avoir rapidement une intuition sur le degré de couplage des composants impactés par l'évolution. En effet, le gestionnaire peut comparer parmi des évolutions de nature similaire les ressources investies pour des activités liées à l'architecture ainsi qu'à l'intégration des composants. Si de manière générale ce sont toujours les mêmes systèmes qui « consomment » le plus de ressources dans ce domaine, c'est que leur couplage est fort.

De plus, une information relative aux changements architecturaux du système permet au gestionnaire d'évaluer ainsi que de surveiller les variations de l'étendue de la couverture fonctionnelle du système d'information. Cette classification des différents changements architecturaux permet de maintenir une cartographie des interactions entre les composants du système qu'ils soient internes ou externes à l'organisation. Un avantage procuré par cette cartographie est d'avoir un aperçu des composants ou des fonctionnalités qui sont isolés du reste du système (couplage très faible) et inversement, de connaître les ramifications d'un système fortement intégré (couplage fort).

Classification des activités relatives au facteur Architecture

L'étude de différentes traces d'évolution l'analyse des données d'évolution concernant le facteur Architecture dans nos études de cas, nous a permis de détailler la nature des activités d'évolution liées à la communication entre les composants du système d'information.

En effet nous avons relevé le besoin de détailler les évolutions relatives à l'architecture. Par exemple, nous n'avons jusqu'ici aucun moyen de différencier les activités modifiant l'existant et celles élargissant le couplage avec de nouveaux composants.

Basé sur l'analyse des données d'évolutions des études de cas, nous avons relevé trois catégories d'évolutions liées à l'architecture du système et répondant à nos besoins :

1. lorsque l'architecture et les possibilités de couplage sont étudiées,
2. lorsque l'architecture est la motivation de l'évolution et
3. lorsque le changement architectural est un impact d'une évolution.

Nous avons identifié, pour ces catégories, des sous-catégories en se basant premièrement sur les motivations des interventions réalisées sur le couplage des composants du système d'information puis sur la nature des interventions et leur impact. Les voici en détail :

1. L'activité est motivée par l'étude de l'architecture et des possibilités offertes par les systèmes en vue de les faire communiquer.
2. L'interopérabilité ou l'intégration sont la motivation de l'évolution.
 - L'évolution porte sur un nouveau couplage de composants. Cette volonté de faire communiquer des composants peut se réaliser de deux manières :
 - Développement d'une interface de communication
 - Utilisation des possibilités offertes par les composants
 - Modification d'un couplage existant entre des composants. Cette modification peut avoir différents types :
 - Augmentation de la portée du couplage : ajout d'informations dans les communications entre composants.
 - Diminution du couplage : suppression d'informations dans les communications entre composants.
 - Correction de bugs relatifs au couplage des composants.
3. L'interopérabilité ou l'intégration ne sont pas la motivation de l'évolution. L'évolution qui touche la communication entre composants est ici un impact d'une évolution et non une motivation qui déclenche une évolution.
 - Lors de la modification d'un composant couplé à d'autres, il peut être nécessaire d'effectuer des adaptations sur les composants communicants ou sur les interfaces de communication de ces composants.
 - Lors du remplacement d'un composant couplé, il peut être nécessaire de redéfinir les interfaces de communication. Nous avons également remarqué que les possibilités de communications d'un composant entraînent dans les critères de choix pour un nouveau composant. En effet, lorsque un composant à remplacer est couplé avec d'autres composants, il est important de ne pas perdre des possibilités de couplages.

Le Tableau 4 présente un exemple de répartition entre les différentes catégories concernant toutes les interventions que nous avons analysées lors de la seconde étude de cas disponible au Chapitre 8. Nous relevons que lorsqu'une intervention concerne le couplage de composants, la motivation est en général le couplage des composants. Lors de rares cas, les modifications qui touchent au couplage de composants sont la conséquence d'une évolution et non la motivation. Nous observons que la solution de développement en interne d'une interface de couplage est utilisée plus fréquemment que la solution basée sur les possibilités de couplage offertes par les composants.

Tableau 4. Catégories de motivations architecturales concernant le couplage des composants pour toutes les interventions analysées.

Catégories de motivations sur le couplage des composants	Pourcentage
1. Etude des possibilités de couplage	14%
2. Le couplage motive l'intervention	83%
Nouveau couplage de composants	38%
Développement d'une interface de couplage	19%
Utilisation des possibilités offertes	5%
Non précisé	14%
Modification d'un couplage existant	45%
Augmentation de la portée d'un couplage	17%
Diminution de la portée d'un couplage	<1%
Correction d'un couplage	14%
Migration vers un autre type de couplage	3%
Non précisé	11%
3. Modification d'un composant couplé	4%
Adaptations sur les composants ou les interfaces	1%
Remplacement d'un composant couplé	3%
Redéfinition des interfaces de couplage	2%
Choix en fonction des possibilités de couplage	<1%

En se basant sur ces catégories et sous-catégories, il est intéressant pour le gestionnaire de faire ressortir les rôles ou les influences de chacun des composants dans l'architecture globale du système d'information.

Il pourra déterminer plus précisément :

- Quels sont les composants du système d'information qui sont d'avantage touchés par les décisions architecturales lors des évolutions : les composants qui subissent l'évolution des autres composants.
- Quels sont les composants qui vont déclencher les évolutions architecturales du système : les composants qui sont moteurs de l'évolution d'autres composants.

Nous avons appliqué la classification des activités relatives à l'Architecture dans l'analyse des traces d'évolution des trois objets d'étude.

Cette classification permet de cerner la nature des activités architecturales avec plus de précision que le typage des activités selon Chapin et al. (2001). Nous avons donc pu observer avec une granularité plus fine ce qui se passe au niveau architectural lors des différentes étapes des cycles de vie.

Couplage des composants, intégration et interopérabilité

L'information doit être partagée et accessible dans l'ensemble de l'organisation afin d'assurer et de garantir un meilleur planning, un meilleur pilotage des processus internes et externes à l'organisation. Cependant certains systèmes d'information supportant ces processus sont parfois considérés comme des « îlots » car ils ne peuvent pas communiquer aisément avec les autres

systèmes de l'organisation et également avec les systèmes externes comme les clients ou les fournisseurs. Afin de fournir un service efficient, le système d'information doit être intégré. Cette intégration signifie la connexion des systèmes indépendants dans un but de partage d'information afin d'être à même de supporter les processus dans leur intégralité.

Le but est d'avoir un système d'information qui couvre une grande partie des processus de l'organisation de manière transversale, c'est à dire en franchissant les frontières traditionnelles que sont les départements ou les fonctions d'une organisation.

Pour arriver à dépasser cette notion d'îlots parmi différents composants du système d'information, il est indispensable que les composants soient interopérables. L'interopérabilité est la capacité de deux systèmes à comprendre et à utiliser les fonctionnalités de l'autre. De manière générale, l'interopérabilité revêt une signification de coexistence ainsi que d'autonomie tandis que l'intégration se réfère d'avantage aux concepts d'uniformisation de coordination et de cohérence (Chen, Doumeingts, & Vernadat, 2008). D'un point de vue relatif au degré de couplage, l'intégration sous entend que les composants sont interdépendants et ne peuvent pas être séparés, on parle alors de couplage fort (tightly coupled). Tandis que l'interopérabilité sous entend que les composants échangent des informations tout en maintenant de manière autonome leur propre logique de fonctionnement, on parle alors de couplage faible (loosely coupled) entre les composants.

Exemples

Afin d'illustrer les évolutions déclenchées par ce facteur d'évolution, voici deux exemples :

- L'organisation a la volonté de déployer un système CRM (gestion de la relation client) au sein du système d'information en place. Cette évolution est déclenchée par une motivation d'étendre l'architecture en place afin d'y ajouter un nouveau composant et donc de nouvelles fonctionnalités métiers.
- La décision est prise d'implémenter une stratégie Web Services pour l'échange de données avec un fournisseur pour remplacer la stratégie actuelle. Cette évolution est déclenchée par une motivation architecturale. Non seulement les interactions actuelles avec ce fournisseur seront impactées par cette évolution mais les futures interactions seront également influencées par cette décision de stratégie architecturale. L'architecture est donc en même temps un moteur d'évolution et un cadre décrivant les futures évolutions.

Fondements

L'architecture logicielle se retrouve autant dans le domaine académique que dans l'industrie. L'importance que revêt la sélection des composants ainsi que leur connecteurs dans un système logiciel est un aspect reconnu et a conduit à un meilleur contrôle de la phase de conception, du développement et de l'évolution des systèmes logiciels les plus complexes et dynamiques (Bosch, 2004).

L'architecture logicielle est définie comme suit par le standard IEEE 1471 (IEEE, 2000) :

« L'organisation fondamentale d'un système représenté par ses composants, leurs relations les uns aux autres et à l'environnement ainsi que les principes influençant sa conception et son évolution. »³

³ Architecture: the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution.

La première phase des recherches sur l'architecture logicielle portait principalement sur l'étude des composants d'un système ainsi que la manière de les interconnecter. Beaucoup de publications lors de cette première phase se concentrent uniquement sur les problèmes relatifs aux changements affectant l'architecture logicielle. Cependant, l'architecture logicielle est d'avantage qu'un ensemble de composants et de connecteurs, elle est la composition d'un ensemble de décisions de conception architecturales (*architectural design decisions*) (Bosch, 2004). Ces décisions de conception architecturales vont influencer la structure de l'architecture logicielle. Selon ces décisions, des règles de conception vont être appliquées pour une partie ou l'ensemble des composants d'un système. Une règle spécifique généralement la manière dont une certaine action doit être effectuée. Par exemple lorsqu'une décision de conception est prise portant sur l'utilisation d'un certain type de base de donnée pour l'ensemble de l'organisation, tous les composants nécessitant de la persistance devront implémenter l'interface provenant du système de gestion de base de données afin d'avoir accès à ce dernier. Les décisions architecturales peuvent donc être transversales au système d'information et de ce fait affecter plusieurs composants et également d'autres choix architecturaux.

L'évolution d'un système logiciel est donc intimement liée aux choix de conception de l'architecture de ce système. Une évolution touchant l'étendue fonctionnelle du système sera influencée par les décisions architecturales. Dès lors, un monitoring des différentes actions entreprises pour des problématiques architecturales est un bon moyen d'obtenir de l'information sur la pertinence, l'efficacité voir l'obsolescence des décisions architecturales prises précédemment. De plus le monitoring de ces activités favorise la documentation des choix architecturaux et donc le respect des règles architecturales précédemment établies.

Comme mentionné dans la Section 5.3.1, les recherches ont montré que l'alignement stratégique du métier et du système d'information contribue à améliorer la performance organisationnelle (Chan, Huff, Barclay, & Copeland, 1997). Cette étude se base sur le « Strategic Alignment Model (SAM) » (Henderson & Venkatraman, 1993). Le SAM identifie deux types d'alignement entre le domaine métier et le système d'information. Le premier type, nommé intégration opérationnelle est pris en compte par le facteur déclencheur d'évolution Alignement (voir Section 5.3.1). Le second type, nommé intégration stratégique, représente le lien entre la stratégie métier et la stratégie du système d'information. Il concerne la capacité des fonctionnalités du système d'information à s'adapter et supporter la stratégie métier (illustré dans la Figure 15). Ce second type est couvert par le facteur déclencheur d'évolution Architecture.

Le but de la « perspective d'alignement stratégique » est de décrire lorsque les évolutions sont relatives à la couverture fonctionnelle du système d'information et lorsque les évolutions impactent stratégiquement les principales fonctionnalités métiers ou les fonctionnalités de support. Une évolution de couverture fonctionnelle du système d'information est déclenchée par un besoin d'expansion architecturale du système. En ajoutant de nouvelles fonctionnalités, l'architecture, et de ce fait le couplage entre composants, est une motivation directe d'évolution.

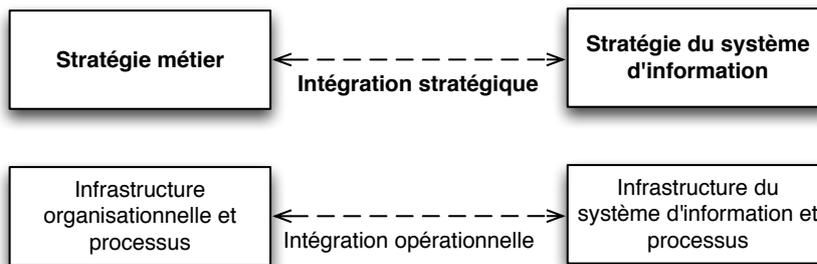


Figure 15. Composants majeurs du Strategic Alignment Model, intégration stratégique (Henderson & Venkatraman, 1993)

5.3.4 Technologie



La technologie revêt une importance stratégique dans la création de valeur ainsi que dans l'avantage concurrentiel des organisations. Cette problématique devient de plus en plus critique avec l'accroissement des coûts et de la complexité lorsque le nombre de changements technologiques augmente.

Lorsque l'on mentionne l'évolution, autant les environnements logiciels que matériels jouent un rôle déterminant. Les coûts inhérents à l'utilisation d'une technologie non optimale peuvent croître significativement quand un changement se révèle nécessaire. L'aspect technologique englobe des notions telles que le degré d'innovation, l'anticipation, la flexibilité, la dimensionnabilité (scalability), la portabilité, etc. des composants du système d'information.

Toute évolution ayant comme principal déclencheur un aspect technologique ou la technologie elle-même doit être caractérisée par EVOLIS comme étant une évolution orientée technologie. A titre d'exemple, lorsque des motifs tels que la performance, les mises à jour, la maintenance préventive et ainsi de suite motivent les évolutions logicielles ou matérielles, l'évolution est caractérisée comme orientée technologie. La caractérisation de ces évolutions orientées technologiques peut aider les gestionnaires à estimer la localisation de leur technologie sur la courbe en S de cette dernière (voir la Figure 16).

Exemples

Afin d'illustrer les évolutions déclenchées par ce facteur d'évolution, voici quelques exemples :

- Migration d'un serveur sous Linux pour palier à des problèmes de compatibilité.
- Ajout d'un système de load balancing pour un composant afin de répartir la charge et de pallier à des problèmes de lenteur du composant.
- Suite à un problème d'obsolescence, la décision est prise de migrer une application vers sa nouvelle version.

Fondements

D'autre part, afin d'étudier a posteriori les données d'évolution de plusieurs études de cas, nous classons chaque activité durant une évolution comme étant orientée technologie si cette dernière concerne directement les éléments cités ci-après dans les domaines de recherche principaux relatifs aux évolutions technologiques des organisations.

Gestion de la technologie

Il existe plusieurs définitions de ce que sont technologie et gestion de la technologie. Selon l'Association of Technology, Management, and Applied Engineering (www.atmae.org), la gestion de la technologie est le domaine concerné par la supervision du personnel technique dans une grande variété de systèmes technologiques complexes. La gestion de la technologie comprend généralement une instruction sur la gestion de la production et des opérations, gestion de projet, des applications informatiques, contrôle de la qualité, les questions de sécurité et de santé, les statistiques et les principes généraux de gestion.

Dans le cadre de cette recherche, nous adoptons la définition proposée par : *European Institute of Technology and Innovation Management (EITIM)* et (Phaal, Farrukh, & Probert, 2004b) :

La gestion de la technologie adresse l'identification, la sélection, l'acquisition, le développement, l'exploitation et la protection des technologies (produits, processus et infrastructures) nécessaires pour obtenir, maintenir [et développer] la performance de l'entreprise en conformité avec ses les objectifs.

Cette définition sous entend l'établissement et le maintien d'un lien important entre les ressources technologiques et les objectifs de l'organisation. Ce lien nécessite de la communication et de la compréhension entre les fonctions techniques et métiers de l'organisation. De plus, une gestion de la technologie effective requiert des processus de gestion de la technologie bien établis comme : l'identification, la sélection, l'acquisition, l'exploitation et la protection de la technologie (Phaal et al., 2004a).

Stratégie technologique

L'intégration des considérations technologiques dans l'établissement de la stratégie métier est un aspect important de la planification organisationnelle. Le principe clé de cette intégration réside dans le fait que la stratégie technologique doit être développée conjointement avec la stratégie métier en incluant les ressources technologiques comme partie intégrante de la planification organisationnelle.

Plusieurs approches visant à développer une stratégie technologique ont été publiées (Metz, 1996; Stacey & Ashton, 1990). Dans la même optique, plusieurs outils et techniques qui supportent ces approches ont été développés conjointement avec une vision de stratégie métier (Lankhorst, 2004). Par exemple, (Fritscher & Pigneur, 2011) font correspondre l'ontologie du business model (Osterwalder & Pigneur, 2010) et ArchiMate (Lankhorst, 2004) afin de mettre en évidence les interconnexions entre le métier et les services informatiques pour mieux les aligner.

Un aspect important de ce genre d'outils est qu'ils favorisent la discussion et comblent le fossé entre le métier et les opportunités technologiques.

Transitions technologiques

Les évolutions technologiques et ainsi que les transitions d'une ancienne vers une nouvelle technologie ont été étudiées principalement dans la littérature relative à la gestion de la technologie. Le principe central qui en découle est qu'initialement la performance d'une

nouvelle technologie est inférieure à celle d'une technologie existante, puis dépasse la performance de l'ancienne technologie pour finalement terminer à un niveau de performance plus élevé. Graphiquement, cela se traduit en traçant une courbe en forme de S visible dans la Figure 16 (Sood & Tellis, 2005). Un des enjeux les plus importants concernant cette théorie de la courbe en S est le processus de transition technologique. A un moment donné, les gestionnaires devront abandonner une technologie arrivée à maturité et en adopter une nouvelle afin de rester compétitifs. Cette transition technologique porte également le nom de transformation ou de révolution (Aier et al., 2011; Legner et al., 2011; Sabherwal & Hirschheim, 2001). Selon Sabherwal & Hirschheim (2001) cette dynamique est expliquée avec le modèle de l'équilibre ponctué (Gould & Eldredge, 1977), le système passe par de longues périodes de stabilité relative dans lesquelles s'opèrent des changements évolutifs. Ces périodes de stabilité sont interrompues par de courtes périodes où les changements sont fondamentaux, révolutionnaires.

Cette transition entre ancienne et nouvelle technologie est une tâche difficile car cela engendre une discontinuité technologique. D'un point de vue managérial, la connaissance de cette discontinuité technologique et de ces révolutions périodiques est un atout. Cependant, il est important pour un gestionnaire d'arriver à estimer la position de la technologie sur sa courbe en S afin de planifier au mieux possible les transitions.

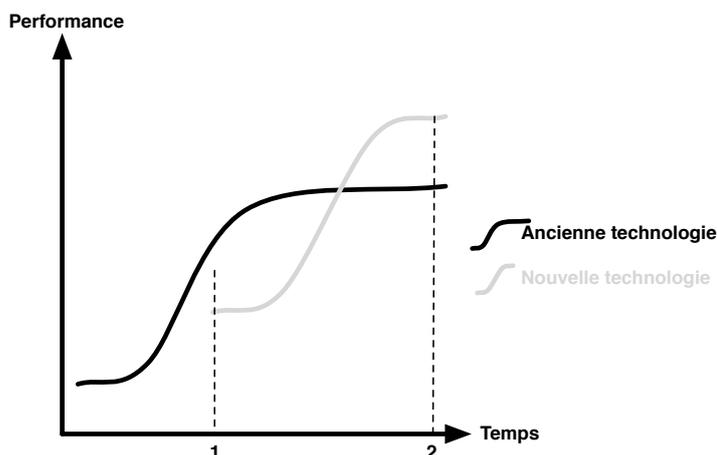


Figure 16. Courbe en "S" de la performance technologique (Sood & Tellis, 2005).

Technology roadmap

Afin de combler ce manque d'outils managériaux pour surmonter la phase de discontinuité lors de transitions technologiques, les « Technology roadmaps » représentent un outil adapté pour supporter la gestion et le planning technologique. Ils permettent d'explorer les liens entre les ressources technologiques, les objectifs organisationnels ainsi que les environnements changeants (Phaal et al., 2004b).

Le principal bénéfice de cet outil est qu'il repose sur un cadre conceptuel basé sur le temps afin de développer, représenter et communiquer les plans stratégiques en terme de coévolution des technologies, des produits et des marchés. Comme les représentations des technology roadmaps sont souvent graphiques, elles sont généralement reliées à d'autres approches de planification graphiques telles que les outils de planification Gantt (Phaal et al., 2004b).

5.4 Coût

Le coût ou le montant des ressources investies pour l'évolution du système d'information est un élément important autant du point de vue du gestionnaire du système que du chercheur. En effet, à la fois académiciens et les praticiens s'accordent sur le fait que les investissements devraient être soigneusement justifiés, mesurés et contrôlés. En pratique, cet élément a fait naître de nombreuses techniques traditionnelles développées dans le but d'évaluer le ratio «coûts-bénéfices» des investissements. Nous retrouvons des techniques d'évaluation comme par exemple le retour sur investissement (ROI), la période de récupération (Payback Period : PP), le taux de rendement interne (TRI) et ainsi de suite (Milis & Mercken, 2004).

Nous préconisons d'évaluer le coût de chaque évolution en parallèle avec la caractérisation des évolutions. Nous ne prônons pas de technique particulière pour évaluer le ratio coûts-bénéfices des évolutions. L'évocation de ces techniques d'évaluation est d'avantage pour attirer l'attention sur le fait que le niveau des ressources investies pour maintenir un niveau de service satisfaisant doit être contrôlé. De plus, le montant des ressources investies dans un système peut mettre en exergue le fait que ce système devient toujours plus coûteux à maintenir à un niveau suffisant de satisfaction. Ce fait est un bon indicateur concernant la position du système dans son cycle de vie et le besoin ou non de revoir sa stratégie d'évolution.

Dans notre contexte de recherche, les informations sur une structure de coûts détaillée sont difficiles à reconstituer ou même à obtenir, surtout dans l'étude des traces d'évolution a posteriori.

Par conséquent, afin d'avoir une base pour comparer les différentes évolutions, nous avons choisi de le remplacer la notion de coût par l'énergie humaine impliquée en jours ouvrables pour chaque activité d'une évolution. En effet, les informations sur l'effort en jours/hommes ou heures/homme sont parfois disponibles et traçables dans les journaux d'évolution, donnant un très bon indicateur indirect des coûts.

Nous pouvons donc quantifier l'importance que revêt chacune des activités entreprise pour l'évolution d'un système vis-à-vis des ressources investies et par analogie l'ampleur des évolutions les unes envers les autres.

Chapitre 6

Méthodologie du cadre conceptuel EVOLIS pour l'étude de l'évolution

Ce chapitre détaille la collecte ainsi que le traitement de traces d'évolution de systèmes d'information collectées a posteriori. Nous présentons les différentes étapes de l'étude des traces d'évolution contenues dans la méthodologie d'étude de l'évolution contenue dans EVOLIS. Nous exposons également un exemple concret d'application d'EVOLIS ainsi que le traitement des traces d'évolution. Nous présentons finalement le modèle conceptuel de données nécessaire au soutien de la mise en œuvre d'EVOLIS afin d'organiser les informations relatives aux évolutions. Ces éléments (méthodologie et modèle conceptuel) sont présentés visuellement en Figure 17.

Les différentes étapes de la méthodologie d'étude de l'évolution ont fait l'objet d'une publication lors de la conférence HICSS 2014 (Métrailler & Estier, 2014).

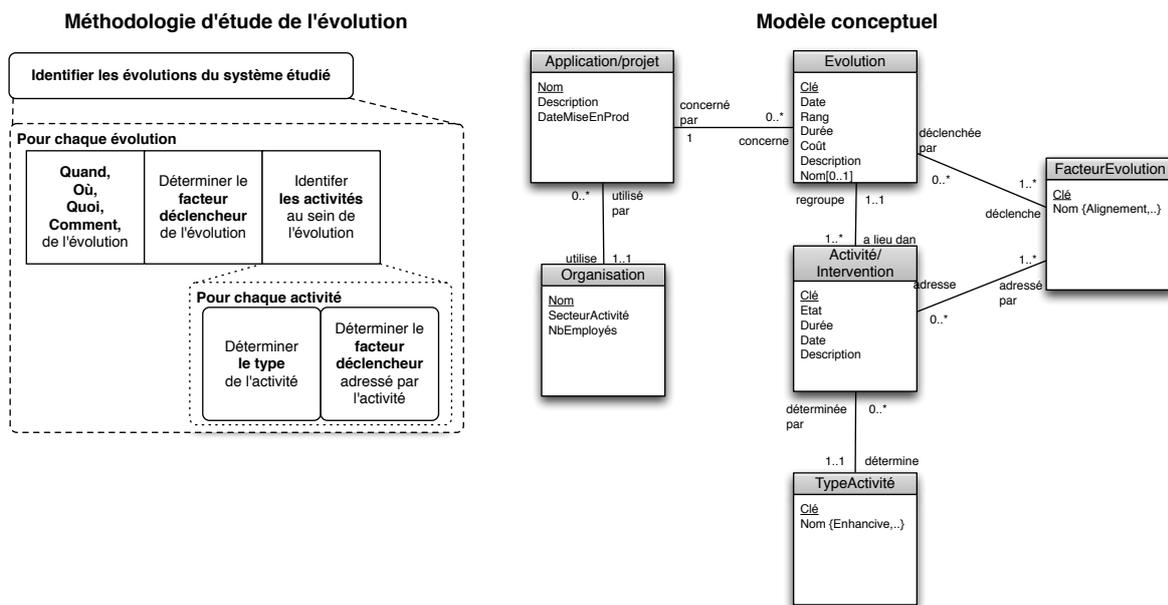


Figure 17. Méthodologie systématique d'étude de l'évolution des systèmes d'information.

Chaque organisation aura un comportement particulier concernant la conservation et l'archivage des multiples interventions effectuées dans le but de faire évoluer leur système d'information. Certaines archiveront uniquement les grandes évolutions de leur système et d'autres toutes les activités affectant de près ou de loin les propriétés de ce dernier. La collecte de traces d'évolution a posteriori ne permet donc pas de s'attendre à un modèle « standardisé » de traces à étudier. Il est donc nécessaire de chercher parmi plusieurs sources de documents, plateformes, archives, etc. pour dénicher un maximum d'informations sur la vie du système.

De la même manière, les personnes en charge de la maintenance et des évolutions des systèmes d'information vont chacune effectuer leur travail à leur manière. La description des activités concrètes entreprises sur le système dépendra non seulement de la politique de l'organisation relative à la documentation des interventions sur le système, mais également de la personne en charge de ces modifications. Parfois les différentes activités d'évolution seront détaillées, parfois uniquement les résultats, parfois rien.

Afin de gagner du temps et d'orienter au mieux les démarches auprès des personnes responsables de la gestion du système d'information, nous conseillons des actions préparatoires à la collecte des données relatives aux évolutions :

- Déterminer quelles sont et quelles ont été les pratiques de documentation liées aux interventions ultérieures peut aider à localiser les sources de traces d'évolution.
- Expliciter le type d'information ou de sources qui pourraient être exploitées.
- Définir le niveau de détail que les sources peuvent procurer.
- Obtenir accès aux sources de documentation qui peuvent être : base de données diverses, systèmes de gestion de tickets, factures faites par des prestataires externes, budgets liés aux systèmes, modification de documentation utilisateur, etc.

Ce chapitre présente donc la méthodologie systématique d'étude de l'évolution du cadre conceptuel EVOLIS en détaillant les différentes étapes de l'étude de l'évolution des systèmes d'information. Cette étude commence par l'identification des différentes évolutions (Section

6.1) puis vient la description de leurs caractéristiques (quand, où, quoi des évolutions ?) (Section 6.2) et enfin par l'identification de leur motivation selon les différents facteurs d'évolution d'EVOLIS (Section 6.3).

Après avoir identifié et décrit les évolutions, l'étude adresse le déroulement de chaque évolution. Au sein de chaque évolution, il s'agit de repérer les différentes activités accomplies (comment de l'évolution ?), leur attribuer un type (Section 6.4) et identifier le facteur d'évolution d'EVOLIS adressé par chaque activité.

En établissant quel est l'effort investi pour accomplir chaque évolution ainsi que la répartition de cet effort dans les activités d'évolution, il est possible de reconstituer de manière détaillée le cycle de vie du système d'information au travers de ses évolutions.

6.1 Identification des différentes évolutions

Lorsque les traces d'évolution sont collectées, il s'agit de repérer les évolutions. Comme une modification d'un système d'information nécessite généralement plusieurs activités différentes, nous avons identifié une évolution comme étant l'accomplissement d'une ou de plusieurs activités impactant le système et suivies par une mise en production. En d'autres termes, nous définissons une évolution comme étant une nouvelle version du système ou d'un composant de ce dernier. En se référant au processus d'évolution décrit en Section 4.6, une évolution telle que nous la décrivons est l'accomplissement d'un cycle total du microprocessus d'évolution (Section 4.6.2)

Néanmoins, il n'est pas nécessaire qu'une activité modifie fonctionnellement le système d'information pour la qualifier comme activité d'évolution. Des activités relatives à du consulting en vue d'une prochaine évolution, de la formation d'utilisateurs ou de la documentation font également partie intégrante des évolutions.

Nous avons observé à travers nos études de cas qu'il n'est pas toujours possible de regrouper des activités pour les considérer comme étant les composantes d'une même évolution. Ce cas de figure a plus de chance de se produire lorsque les compétences nécessaires à l'activité d'évolution se trouvent en interne de l'organisation. En effet, lorsque l'organisation dispose de personnes dédiées à l'entretien du système d'information, les évolutions de ce dernier ont tendance à s'opérer de manière continue. Tandis que lorsque ces compétences d'évolutions sont externes, les organisations semblent préférer regrouper les activités pour les faire effectuer en une fois auprès de leurs partenaires externes.

Dans le cas de figure où les compétences d'évolution sont internes, nous avons un système d'information en évolution continue, nous considérons donc une évolution comme étant l'accomplissement d'une intervention sur le système d'information. Une fois de plus, il n'est pas nécessaire que l'intervention modifie le fonctionnement du système pour la qualifier d'évolution.

Dans le cas de figure où les compétences d'évolution sont externes, les évolutions seront ponctuelles. Elles n'apparaissent donc plus dans le cycle de vie de manière continue mais de manière discrète.

6.2 Caractériser une évolution (taxonomie)

Après avoir identifié les différentes évolutions présentes dans les traces, nous utilisons la taxonomie proposée par Buckley et al. (2005) pour caractériser les évolutions. En effet, les éléments de taxonomie permettent d'organiser, de classier et de positionner les évolutions à

l'intérieur de leur domaine selon les dimensions essentielles qui sont : les propriétés temporelles (le quand), l'objet du changement (le où), la manière dont le changement est effectué (le comment) et les propriétés du système qui ont été modifiées (le quoi). (« how, when, what and where of software change »).

6.2.1 Propriétés temporelles (Quand)

Ces propriétés répondent au besoin de chronologie évoqué pour deux contextes pratique et académique (Section 5.1). Le premier facteur qu'adresse la question « quand ? » est le moment du changement. En effet, ce dernier est nécessaire pour retracer la vie du système et localiser le changement dans le cycle de vie. Sans une chronologie des changements effectués, il est impossible de comprendre par quelles évolutions est passé le système.

Un second facteur qui implique la notion de temps est la durée nécessaire pour effectuer le changement. Cette durée aura une relation directe sur les ressources allouées au changement. Le facteur Coût (Section 5.4) de l'évolution peut se substituer à la notion de durée si les ressources allouées sont évaluées en fonction de la durée plutôt qu'en fonction du coût réel de l'évolution.

Il est donc pertinent d'obtenir une date de début à laquelle le changement est initié et une durée afin de mieux réaliser l'importance et l'ampleur du changement.

Un autre facteur relatif au temps est la fréquence à laquelle un changement est effectué. Stratégiquement, les changements sur un système peuvent être entrepris de manière continue, périodiquement ou à intervalles irréguliers. Il est tout à fait envisageable que les bugs ne nécessitant pas de trop gros changements soient corrigés dès qu'ils sont détectés (de manière continue), et que les requêtes de changements émanant des utilisateurs soient regroupées et traitées périodiquement.

6.2.2 Objet du changement (Où)

Le second élément de la taxonomie adresse la question : Où est effectué le changement ? Le but de cette question est de décrire à quel(s) endroit(s) du système les changements sont effectués.

Les parties du système affectées par une évolution peuvent être multiple. Cela peut englober autant une simple modification de configuration qu'une remaniement de l'architecture du système.

Un autre thème adressé par l'objet du changement est la granularité du changement, autrement dit l'ampleur de ce dernier vis-à-vis de la taille du système global. La question à se poser ici est si le changement va modifier un élément isolé ou être plus étendu et toucher plusieurs fonctionnalités voir tout le système.

Lorsqu'un changement est réalisé, même si ce dernier semble affecter un composant précis, il ne faut surtout pas minimiser l'impact qu'il peut avoir. En effet, l'impact d'un changement peut avoir différentes portées. Il peut être uniquement « local », dans le cas d'un changement impactant qu'une petite partie du système, jusqu'à « global » dans le cas où tout le du système est impacté par le changement. Cette notion d'impact peut atteindre différents niveaux d'abstraction. Par exemple une modification d'un processus métier aura un impact non seulement au niveau logiciel, mais également au niveau métier, utilisateur final, documentation et ainsi de suite.

6.2.3 Propriétés du système modifié (Quoi)

Dans leur article, Buckley et al. (2005) présentent 4 dimensions à prendre en compte lors de l'analyse des changements effectués sur les systèmes logiciels.

La *disponibilité (Availability)* : la plupart des systèmes de type E évoluent de manière continue durant tout le long de leur vie. Cette dimension indique si le système doit être continuellement disponible ou non lors de l'évolution.

Le *dynamisme (Activeness)* : cette dimension peut être qualifiée de deux manières : réactive ou proactive. Le système logiciel est dit réactif lorsque les changements sont effectués de manière externe. Le système logiciel est dit proactif lorsque le système effectue de manière autonome ses propres changements.

L'*ouverture (Openness)* : les systèmes logiciels sont qualifiés d'*ouverts* si ils sont spécialement conçus pour permettre l'ajout d'extensions. A l'opposé les systèmes logiciels dits *fermés* ont leurs fonctionnalités fixées lors de leur conception. Ils ne sont cependant pas non évolutifs, mais ils ne sont pas conçus dans ce but. Leur évolution demandera donc d'avantage de ressources.

La *sécurité (Safety)* : cette dimension est qualifiée de deux manières : sécurité *statique* et sécurité *dynamique*. Le système dispose de sécurité statique si nous sommes capables d'assurer, au moment de la compilation, que les aspects spécifiques de sécurité sont préservés. Le système offre une sécurité dynamique si ils sont construits afin d'empêcher ou de restreindre un comportement indésirable lors de l'utilisation.

6.2.4 Support (Comment)

La dernière question qu'adresse la taxonomie de Buckley et al. (2005) concerne le mécanisme utilisé pour effectuer un changement. Les mécanismes de changement permettent d'analyser, de gérer, de contrôler, ou de mesurer les changements logiciels.

Trois dimensions sont proposées pour qualifier les mécanismes de support du changement :

- Le niveau d'automatisme : cette dimension va d'*automatique* à *manuel* en passant par *partiellement automatique*.
- Le degré de formalité : le mécanisme d'un changement peut être soit *ad hoc*, soit suivre une formalisme mathématique sous-jacent. Par exemple, la théorie des graphes est parfois utilisée dans des problématiques de propagation de changements ou de refactoring (Buckley et al., 2005).
- Le type de changement : dans leur article, Buckley et al. (2005) mentionnent deux types de changements : les changements structurels qui modifient le comportement du logiciel en ajoutant, altérant ou supprimant des fonctionnalités et les changements sémantiques comme le refactoring qui vise à modifier des propriétés telles que l'efficacité, l'utilisation de la mémoire etc.

Pour nos besoins, nous avons choisis d'utiliser la typologie des activités de maintenance et d'évolution proposée par Chapin et al. (2001). En effet cette dernière approfondi la typologie initiale découlant des travaux de Lientz et al. (1978) en proposant une typologie affinée des activités de maintenance et d'évolution classifiées selon 12 types rassemblés en 4 groupes. Sa mise en pratique est détaillée dans la Section 6.4.

6.3 Caractériser les motivations d'évolution

Après avoir caractérisé les évolutions selon les thèmes et les dimensions de la taxonomie présentée en Section 6.2, nous identifions les éléments déclencheurs des évolutions. Pour chacune des évolutions, nous nous intéressons donc à la nature des motivations ayant initié l'évolution en nous basant sur les différents facteurs d'évolution présents dans EVOLIS et décrits dans le Chapitre 5.

Nous déterminons a posteriori les motivations d'une évolution en nous basant sur les descriptions présentes dans les différentes sources d'information concernant les évolutions. Il est possible de déduire les motivations des évolutions en se basant par exemple sur les justificatifs d'une demande d'évolution. Les principales raisons qui ont engendré une évolution sont fréquemment présentes et explicitées dans les documents relatifs aux évolutions.

Une évolution est classifiée comme étant principalement orientée :

- Alignement métier et système d'information, lorsque la motivation principale de l'évolution est un alignement avec les besoins métiers ou l'introduction d'une nouvelle fonctionnalité métier.
- Satisfaction utilisateur, dans le cas où la relation entre l'utilisateur et le système d'information motive l'évolution.
- Technologie, lorsque l'évolution est motivée par des éléments logiciels ou matériels.
- Architecture, si l'évolution est motivée par l'organisation, les relations, l'intégration ou la consolidation entre les différents composants du système d'information.

Naturellement, une évolution peut avoir été déclenchée par une combinaison de motivations de plusieurs natures. A ce niveau, nous retenons la motivation dominante qui aura engendré l'évolution.

La propriété temporelle décrite par la taxonomie ne permet pas une évaluation suffisamment précise de l'ampleur de l'évolution. En effet, cette dernière nous donne une date à laquelle l'évolution est initiée et une date à laquelle elle est réalisée, malgré cela la période écoulée entre les deux dates n'est généralement pas suffisante pour déterminer l'ampleur de l'évolution ou l'investissement nécessaire pour cette dernière. Pour cette raison, nous établissons une estimation du coût de chacune des évolutions. Le coût ou l'investissement pour une évolution est calculé de manière à créer une base pour la comparaison de l'ampleur des évolutions. Comme mentionné en Section 5.4, nous utilisons notamment le nombre de jours/homme nécessaire pour l'accomplissement de l'évolution afin d'estimer les ressources investies. De plus, l'ampleur d'une évolution est un moyen aisé de déterminer son importance dans le cycle de vie du système d'information.

A ce point, nous sommes à même de retracer le cycle de vie du système d'information avec ses évolutions, leurs propriétés, leurs motivations et leur importance.

Afin d'étudier plus en détail les évolutions, nous analysons et qualifions les différentes activités nécessaires à l'accomplissement de chaque évolution. Cette analyse permet de répondre plus précisément à la question « comment » sont effectuées les évolutions de la taxonomie (Buckley et al., 2005).

6.4 Caractériser les activités au sein des évolutions

Un domaine de recherche sur les changements logiciels complémentaire à celui de la taxonomie de Buckley et al. (2005) est celui de la typologie des changements. Ce domaine est initié dans les

années 1970 avec les travaux décrivant les activités de maintenance du logiciel (Lientz et al., 1978). Ces travaux identifiaient trois catégories d'activités de maintenance : corrective, adaptative et perfective auxquelles s'est ajoutée la catégorie d'activités préventive. Plus récemment, Chapin et al. (2001) ont étendu cette typologie en proposant 12 types d'activité rassemblés dans 4 « groupes d'activités » selon leur orientation. Nous avons choisi cette typologie car elle propose des noms de types d'activité plus descriptifs et pertinents que les travaux précédents. Dans la continuité des travaux précédents, cette typologie est également indépendante des moyens techniques utilisés et des pratiques organisationnelles.

De plus, les praticiens accomplissent une grande variété d'activités en effectuant les travaux de maintenance et d'évolution. Cette variété d'activités est occultée par l'utilisation d'un terme trop général comme « maintenance perfective » (Chapin et al., 2001). De manière semblable, les gestionnaires impliqués dans l'évolution du système d'information doivent pouvoir décrire précisément les activités d'évolution afin de justifier, budgéter, mobiliser les ressources nécessaires, évaluer, superviser les évolutions du système. Une description plus précise permet donc d'éviter les malentendus à propos des activités à accomplir et facilite la communication entre les parties prenantes.

6.4.1 Description des différents types

Afin de classifier leurs 12 types d'activité d'évolution, Chapin et al. (2001) les ont regroupés en 4 groupes (*clusters*) selon deux dimensions. La première est l'impact des activités d'évolution sur la capacité de l'utilisateur à utiliser le système pour accomplir ses tâches, c'est à dire l'impact de l'évolution sur les processus métiers. Par exemple : l'ajout d'une nouvelle fonctionnalité logicielle a plus d'effet sur la capacité de l'utilisateur à arriver à ses fins qu'une modification de la documentation. La seconde dimension est l'impact de l'évolution sur le logiciel. Par exemple, une modification de la documentation a moins d'impact sur le logiciel qu'une correction de bug.

Le premier groupe d'activités est nommé « Support interface cluster ». Il regroupe les activités ayant un impact faible à modéré sur les processus métiers et faible sur le logiciel. Ce groupe rassemble les activités de type :

- Training : si les activités d'évolution sont relatives à de la formation quant à l'utilisation du logiciel.
- Consultive : si les activités d'évolution sont à des fins de consultation concernant le logiciel. Par exemple : collecte de données relatives à l'utilisation du logiciel, au coût d'un composant, à l'étude d'un possible changement, etc.
- Evaluative : si les activités d'évolution ont pour but d'évaluer le logiciel. Ce type est proposé comme étant le type dominant du groupe ainsi que le type par défaut lorsqu'aucun autre type du groupe ne correspond.

Le second groupe d'activités d'évolution est nommé « Documentation cluster ». Il regroupe des activités ayant un impact faible à modéré sur les processus métiers et modéré sur le logiciel. Ce groupe rassemble les activités de type :

- Reformative : si les activités d'évolution ont pour but de rendre la documentation plus conforme aux besoins des parties prenantes.
- Update : si les activités d'évolution ont pour but de mettre à jour la documentation en fonction du logiciel. Ce type est le type par défaut du groupe.

Le troisième groupe d'activités d'évolution est nommé « Software properties cluster ». Il regroupe des activités ayant un impact modéré sur les processus métiers et modéré sur le logiciel. Les actions de ce groupe modifient le code source du logiciel sans affecter les fonctionnalités. Ce groupe rassemble les activités de type :

- Groomative : si les activités d'évolution améliorent le code source de manière « cosmétique » dans un souci de meilleure lisibilité, compréhension, etc. Ces modifications impactent rarement les utilisateurs, leur but est principalement d'améliorer la maintenabilité du logiciel.
- Preventive : si les activités d'évolution permettent d'éviter ou de réduire les futures activités de maintenance du logiciel. Comme pour le type précédent, ces actions sont rarement sensibles pour les utilisateurs.
- Performance : si les activités d'évolution modifient les performances ou les propriétés du logiciel. On y regroupe des activités impactant la diminution des temps de latence, la diminution des périodes dans lesquelles le logiciel est hors service, améliorant la fiabilité et la robustesse du logiciel. Contrairement aux activités précédentes, celles-ci ont un effet perceptible par l'utilisateur.
- Adaptive : si les activités d'évolution modifient la technologie ou les ressources utilisées. Ce type regroupe des activités comme un changement de plateforme d'exécution ou un changement de système d'exploitation. Ce type d'activités n'altère pas les fonctionnalités du logiciel mais comme les autres types de ce groupe modifient les propriétés ou les caractéristiques du logiciel. Ce type est le type par défaut du groupe.

Le quatrième groupe d'activités d'évolution est nommé « Business rules cluster ». Il regroupe des activités ayant un impact important sur les processus métiers et modéré à important sur le logiciel. Ces activités sont parmi les plus fréquentes et les plus importantes dans le domaine de l'évolution logicielle. Afin que les activités de ce groupe soient menées à bien, elles reposent sur l'accomplissement des activités des trois autres groupes. Ce groupe rassemble les activités de type :

- Reductive : si les activités d'évolution restreignent ou réduisent les fonctionnalités utilisateurs du logiciel.
- Corrective : si les activités d'évolution modifient les fonctionnalités utilisateurs afin de les rendre d'avantage correctes.
- Enhancive : si les activités d'évolution remplacent, ajoutent ou étendent les fonctionnalités utilisateurs. En raison de la complexité croissante des opérations des utilisateurs, ce type est généralement le plus répandu et le plus important. Il est également le type par défaut de ce groupe concernant les règles métier.

6.4.2 Typage des activités

Afin de faciliter l'attribution des types aux différentes activités d'évolution, Chapin et al. (2001) proposent un arbre de décision représenté dans la Figure 18. Cet arbre se compose de deux niveaux. Le premier permet d'atteindre le groupe dans lequel se situe l'activité et le second de déterminer le type exact de l'activité dans le groupe. L'arbre se lit de gauche à droite en débutant par la classification de l'activité dans le bon groupe selon trois critères de décision (A-B-C). Puis, lorsque le groupe est déterminé, le second niveau de critères de décision (A-1-2-3, B-1-2, etc.) permet de déterminer le type de l'activité selon le but de cette dernière comme décrit précédemment. On note également que cet arbre de décision ne propose pas de type nommé « autre ». En effet, chaque groupe d'activités dispose d'un type par défaut si le typage d'une activité est ambigu.

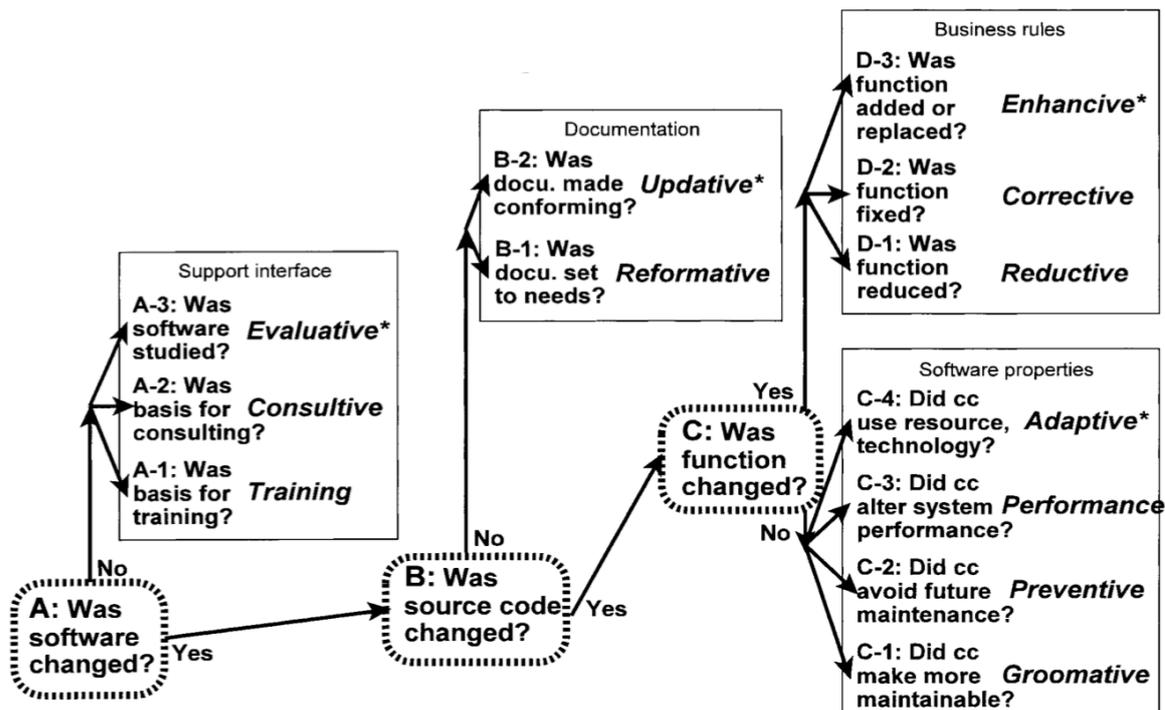


Figure 18. Arbre de décision pour déterminer le type des activités d'évolution (Chapin et al., 2001).

Comme mentionné précédemment, le processus d'évolution implique généralement plusieurs activités qui résultent en une ou plusieurs modifications du logiciel. Chacune de ces activités peut être classée selon l'arbre de décision présenté en Figure 18. Chapin et al. (2001) précisent que la typologie qu'ils proposent est exhaustive et composée de types mutuellement exclusifs. Chaque évolution logicielle est donc composée d'une combinaison d'activités de différents types voir de tous les types. Il se peut évidemment qu'un des types soit observé comme étant le type d'activités dominant de l'évolution.

6.4.3 Caractériser le but des activités

Comme une évolution peut entraîner des activités de différents types, une évolution motivée par un facteur d'évolution (voir Section 6.3) ne rassemblera pas forcément que des activités adressant ce même facteur d'évolution. C'est pourquoi il est intéressant de connaître le but des activités d'évolution en les classifiant selon les facteurs d'évolution représentés dans EVOLIS.

Le but de cette caractérisation est d'identifier quel facteur d'évolution d'EVOLIS est concerné par chacune des activités de l'évolution.

De manière similaire à la classification des motivations d'une évolution (Section 6.3), les activités sont classifiées comme étant principalement orientées :

- Alignement métier et système d'information, lorsque la finalité de l'activité est un alignement avec les besoins métiers ou l'introduction d'une nouvelle fonctionnalité métier.
- Satisfaction utilisateur, dans le cas où la finalité de l'activité concerne la relation entre l'utilisateur et le système d'information.
- Technologie, lorsque la finalité de l'activité concerne des problématiques purement logicielles ou matérielles.
- Architecture, si la finalité de l'activité adresse l'organisation, les relations, l'intégration ou la consolidations entre les différents composants du système d'information.

Par exemple, à l'intérieur d'une évolution, une activité typée « Consultive » adressera le facteur « Architecture » si son but est de déterminer la manière de réaliser le couplage entre différents composants du système.

La Section 6.3 mentionne qu'une estimation du coût de l'évolution est calculée afin d'obtenir une base pour la comparaison des évolutions. Il en est de même pour les activités. Comme une évolution est composée d'activités, il est possible d'allouer le coût global de l'évolution en fonction des ressources nécessaires pour accomplir chacune des activités. Cette répartition du coût permet donc de connaître l'allocation des ressources pour chaque activité d'une évolution et en même temps de connaître quel est le ratio d'importance de chaque facteur d'évolution dans l'accomplissement de l'évolution.

Par exemple, cette caractérisation pourrait faire ressortir que pour une évolution motivée par une volonté d'Alignement, 10% des ressources ont été investies pour des activités en rapport avec la satisfaction des utilisateurs, 60% pour des activités d'alignement, et le solde pour des activités en rapport avec l'architecture et le couplage de composants.

6.5 Exemple de scénario d'évolution

Jusqu'ici, nous avons présenté d'avantage les aspects de l'étude a posteriori des évolutions. Voici un exemple de situation représentant un cas d'utilisation d'EVOLIS en temps réel lors d'une évolution. La méthodologie d'étude de l'évolution d'EVOLIS reste invariante selon que l'évolution soit étudiée en temps réel ou sur des données d'évolution a posteriori.

Dans le but de répondre à un nouveau besoin métier, une organisation réalise qu'un composant spécifique de son système d'information (un « serveur d'application » par exemple) devra interagir avec un autre système hors de l'organisation. Ce scénario d'application ne repose donc pas sur l'étude a posteriori d'une évolution, mais une application en temps réel de la méthodologie d'étude d'évolution avec EVOLIS.

En appliquant la méthode systématique d'étude des évolutions, la première étape est de caractériser l'évolution dans sa globalité :

- Propriétés temporelles : actuellement, il faut donc mémoriser la date.
- Objet du changement : changement local sur un composant : « serveur d'application ».
- Propriétés du système modifiées : ouverture du système hors de l'organisation.
- Support : changement manuel détaillé avec les différentes activités.

Ainsi que selon la ou les motivations :

- Motivation principale : l'alignement du système d'information avec les besoins métiers de l'organisation.

Une telle évolution pourrait englober des activités telles que : l'étude des besoins spécifiques des utilisateurs, l'étude du système de logiciel externe à l'organisation, développement d'une interface de communication entre les deux systèmes, les tests de l'intégration des deux systèmes, le développement et l'évaluation d'une nouvelle interface utilisateurs, la formation des utilisateurs, la mise à jour de la documentation technique, et ainsi de suite.

Concrètement, dans ce scénario, les membres de l'équipe en charge de ce projet (techniciens, analystes d'affaires, etc.) répertorient et classent chaque activités mentionnées précédemment selon la typologie présentée en Section 6.4.

De plus, l'investissement nécessaire (le coût) pour chaque activité dans l'évolution est important pour déterminer l'allocation des ressources au sein de l'évolution. Afin d'établir le coût des

activités, les membres de l'équipe peuvent mesurer leur investissement en temps pour chaque activité.

Après avoir typé et pondéré en jours/homme les activités, la prochaine étape est de déterminer la finalité de chaque activité selon les facteurs d'évolution d'EVOLIS.

Dans ce scénario d'évolution, une grande partie des activités requises pour aligner le système d'information ont une finalité orientée « Architecture » puisqu'elles sont liées au couplage avec un système extérieur à l'organisation.

Des activités telles que l'étude du système logiciel externe, le développement de l'interface de communication entre les deux systèmes ainsi que les tests sont liées au facteur « Architecture » d'EVOLIS car l'architecture est à la fois impactée par l'extension du champ d'application du système et, en même temps, la stratégie architecturale de l'organisation détermine la technique utilisée pour le couplage des deux systèmes.

Le Tableau 5 présente les différentes activités entreprises pour l'évolution avec pour chacune leur type, leur durée (coût) et le facteur d'évolution d'EVOLIS qu'elle adresse.

Tableau 5. Détail des activités du scénario d'évolution.

Activités	Type	Investissement en jours/homme	Facteur d'EVOLIS
Etude des besoins des utilisateurs	Consultive	0.5	Alignement métier et système d'information
Etude du système logiciel externe à l'organisation	Evaluative	0.5	Architecture
Développement d'une interface entre les deux systèmes	Enhancive	1	Architecture
Tests du couplage des deux systèmes	Corrective	1	Architecture
Développement de l'interface utilisateurs	Enhancive	0.5	Alignement métier et système d'information
Evaluation de l'interface utilisateurs	Evaluative	0.3	Satisfaction utilisateurs
Formation des utilisateurs	Training	0.2	Satisfaction utilisateurs
Mise à jour de la documentation technique	Updative	0.2	Architecture
	Total	4.2	

Dès la fin de cette évolution ayant pour objectif principal l'alignement du système d'information avec les besoins opérationnels (l'interaction avec un système extérieur de l'organisation), le gestionnaire dispose d'une vue globale sur la répartition des ressources impliquées dans cette évolution en fonction des différents facteurs d'évolution d'EVOLIS. Il a donc accès à la répartition des ressources en fonction du but des activités à savoir : alignement avec le métier, architecture, utilisateurs, technologie (visible dans la Figure 19) ainsi qu'en fonction du type de chacune de ces activités (visible dans la Figure 20).

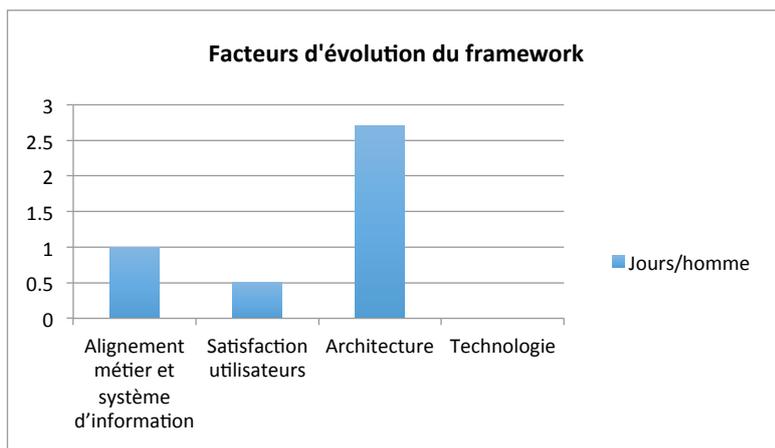


Figure 19. Répartition des ressources investies selon les facteurs d'évolution d'EVOLIS.

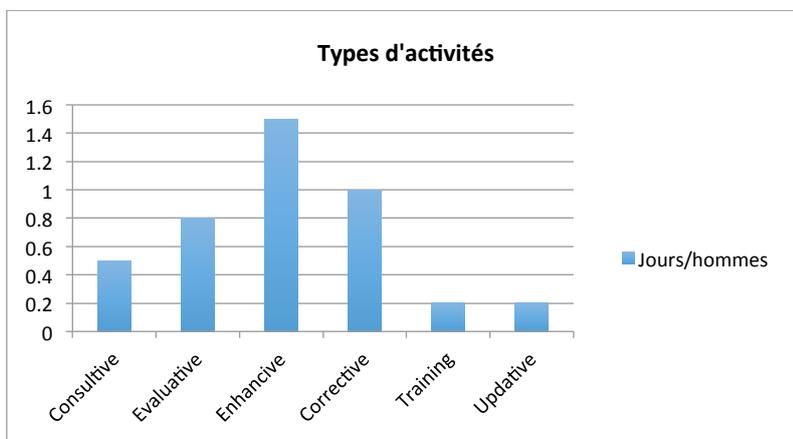


Figure 20. Répartition des ressources investies selon le type des activités de l'évolution.

L'utilisation répétée d'EVOLIS pour chaque évolution peut apporter des avantages de gestion multiples. En effet, une meilleure connaissance de l'allocation réelle des ressources sur les évolutions passées permet d'améliorer la préparation et la planification des évolutions futures.

L'application d'EVOLIS peut également fournir un indicateur sur les enjeux du système d'information. Par exemple si chacune des précédentes évolutions nécessite un fort investissement de ressources orientées architecture, on pourrait en déduire que la couverture du système s'agrandit ou que l'architecture actuelle se complexifie, etc.

Autre exemple : après avoir étudié la répartition des activités en fonction de leur type, le gestionnaire pourrait justifier l'allocation de davantage de ressources pour des activités classées comme «consultatives» ou «évaluatives» afin d'économiser sur d'autres activités ou sur le coût global d'une évolution.

6.6 Modèle conceptuel d'EVOLIS

Afin de mener à bien l'analyse et la classification des traces selon la méthodologie d'étude de l'évolution décrite précédemment, nous avons créé un modèle conceptuel de données basé sur les besoins nécessaires à l'application d'EVOLIS. En effet, les informations créées avec l'application d'EVOLIS doivent être organisées afin de pouvoir être exploitées.

Nous avons donc choisi de réaliser un modèle conceptuel générique ayant une orientation transactionnelle plutôt qu'analytique. Conceptuellement, l'approche transactionnelle permet de retrouver les concepts d'EVOLIS comme des entités associées entre elles plutôt que comme des éléments dissimulés dans une table de faits. Les systèmes de gestion de données actuels proposent en général des fonctionnalités analytiques et permettent de créer relativement facilement des structures pour l'analyse. L'enjeu pour passer d'un modèle transactionnel à un modèle analytique n'est donc pas crucial.

Ce modèle conceptuel (présenté dans la Figure 21) permet la création d'une structure de données générique pour effectuer les regroupements et la classification des différentes évolutions et activités d'évolution en fonction des systèmes d'information étudiés. Il peut également être utilisé par les praticiens pour enregistrer en temps réel les informations requises par EVOLIS pour l'étude et l'analyse future des évolutions.

Les attributs des différentes classes sont donnés à titre indicatif et peuvent être vus comme le minimum d'information nécessaire pour l'application d'EVOLIS afin de réaliser l'étude des différentes évolutions.

Voici la description des classes du modèle conceptuel de la Figure 22 :

- La classe Organisation représente les organisations qui exploitent les systèmes d'information étudiés (représentés dans la classe Application/projet).
- La classe Application/projet représente les systèmes d'information étudiés, application ou projet concerné par des évolutions.
- La classe Evolution représente les évolutions rencontrées par l'application/projet. Elle contient les éléments permettant de retracer l'ordre chronologique des évolutions.
- La classe Activité/Intervention permet de représenter les activités effectuées dans le cadre d'une évolution. Chaque objet de cette classe sera classifié avec un type d'activité (TypeActivite) et le facteur déclencheur d'évolution (FacteurEolution) adressé.
- La classe FacteurEolution représente les facteurs déclencheurs d'évolution présents dans EVOLIS. Les évolutions et les activités sont classifiées avec respectivement le facteur ayant déclenché l'évolution et le facteur adressé par l'activité.
- La classe TypeActivite représente les différents types d'activités entreprises lors d'évolutions. Chaque activité étant classifiée avec un type d'activité.

Ce modèle conceptuel est un modèle générique pouvant supporter les données d'évolution que nous avons rencontré lors des deux études de cas que nous avons effectuées (présentées dans le Chapitre 7 et le Chapitre 8).

Comme ce modèle est générique, il n'a pas pu être appliqué tel quel pour nos études de cas. Lors de la seconde étude de cas (Chapitre 8), nous l'avons adapté aux spécificités des données d'évolution collectées dans le but de conserver une partie de leur structure (voir la Section 8.2.3).

Ce modèle intègre les besoins des deux contextes adressés par EVOLIS. Pour les chercheurs, la classe organisation permet de regrouper les interventions pour une même organisation et autorise ainsi des comparaisons d'évolutions entre les différentes organisations étudiées.

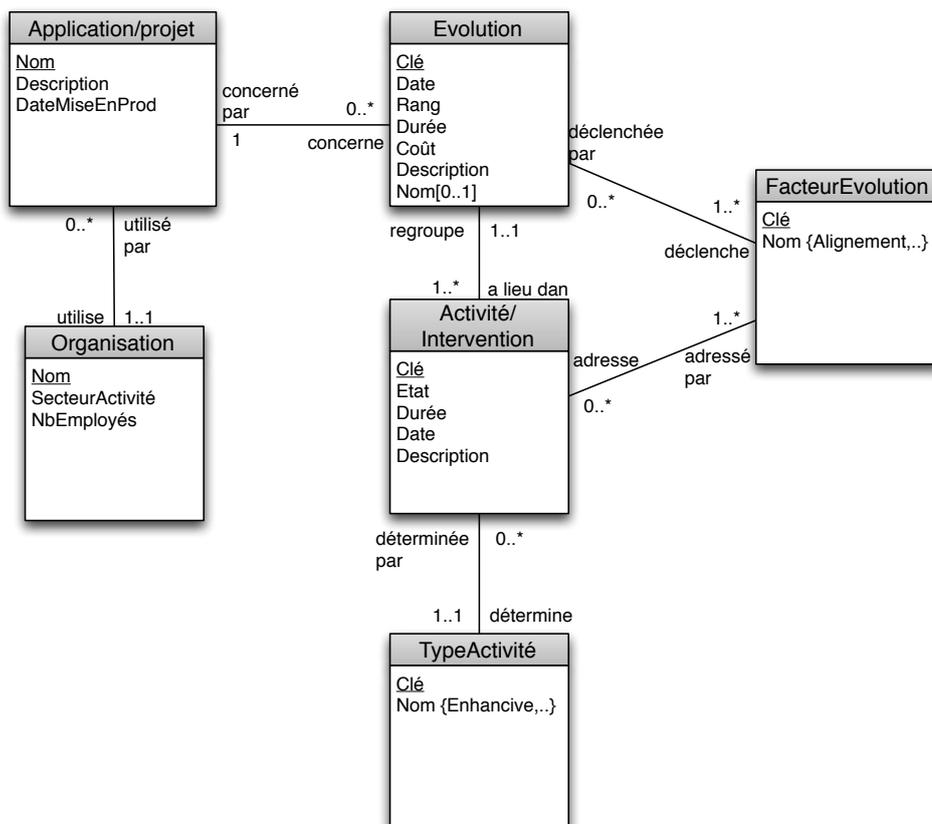


Figure 21. Modèle conceptuel supportant EVOLIS.

6.7 Issues de l'application

L'application d'EVOLIS produit une cartographie détaillée des évolutions d'un ou des systèmes d'information étudiés. Cette cartographie permet de retracer le cycle de vie du système et contient la chronologie des évolutions, quels ont été les objets de ces évolutions ainsi que les motivations ayant conduit le système dans son état actuel. De plus, pour chacune de ces évolutions, l'application d'EVOLIS renseigne sur la manière dont les évolutions sont accomplies grâce aux différentes activités d'évolution, à leur type et à leur finalité.

La cartographie des évolutions ainsi que cette classification des activités en fonction de leur type et de leur finalité revêt une utilité à la fois pour le gestionnaire et pour le chercheur.

En effet, le gestionnaire peut se baser sur les évolutions passées pour améliorer la planification et l'allocation des ressources pour les futures évolutions. Cette classification autorise également l'analyse des écarts entre planification et répartition effective des ressources. Basé sur ces écarts, le gestionnaire dispose d'une vision précise et chiffrée du déroulement de l'évolution et peut l'utiliser afin d'améliorer la planification des évolutions futures. Une connaissance détaillée du cycle de vie des systèmes aide également au pilotage de ces derniers.

En étudiant a posteriori des évolutions, le chercheur dispose d'un cadre conceptuel qui regroupe des critères de qualifications d'évolution qui sont généralement étudiés de manière isolée. Il peut dès lors étudier les impacts de chaque évolution en fonction des facteurs d'évolution représentés dans EVOLIS. De plus une application répétée d'EVOLIS pour l'étude de plusieurs

évolutions de systèmes d'information ajoute une dimension temporelle et permet donc de localiser les évolutions ainsi que leurs motivations dans le cycle de vie de chaque système.

Afin de démontrer et d'affiner l'application d'EVOLIS nous avons conduit une série d'études de cas exploratoires portant sur l'évolution de six systèmes d'information issus de petites et moyennes entreprises suisses et françaises. Ces études de cas sont présentées dans le Chapitre 7.

Chapitre 7

Mise en œuvre d'EVOLIS : Etude de cas 1

Ce chapitre détaille l'application d'EVOLIS pour l'analyse a posteriori de traces d'évolution. Cette première application réelle d'EVOLIS fait partie intégrante du processus itératif de design. En effet cette application nous a permis de souligner les aspects d'EVOLIS qui n'étaient pas suffisamment développés, d'affiner la méthodologie d'étude des traces d'évolution et également d'analyser les résultats obtenus.

Cette étude de cas a fait l'objet d'une publication lors de la conférence HICSS 2014 (Métraiiller & Estier, 2014).

7.1 Etude de cas

Nous avons choisi d'appliquer EVOLIS afin d'étudier des évolutions a posteriori. Le premier avantage d'une étude de cas a posteriori est que les données sont immédiatement disponibles. Nous économisons donc le temps d'attente nécessaire au processus d'évolution et comme la manipulation du sujet ou des événement n'est pas souhaitée, nous ne risquons donc pas d'influencer le processus étudié.

De plus, selon Benbasat, Goldstein, & Mead (1987), il y a trois raisons pour lesquelles les études de cas sont une stratégie de recherche en systèmes d'information. Premièrement une étude de cas permet d'étudier les systèmes d'information dans un contexte réel d'utilisation. Deuxièmement, elle permet au chercheur de répondre aux questions « comment » et « pourquoi » et de comprendre la nature du processus étudié. Troisièmement, cette approche est appropriée pour l'étude d'un domaine dans lequel peu de recherches précédentes ont été menées.

Selon les trois raisons évoquées précédemment, une étude de cas est appropriée pour appliquer et évaluer EVOLIS car elle permet de confronter EVOLIS avec un contexte réel d'utilisation. Les résultats peuvent donc être utiles non seulement pour les chercheurs mais aussi pour les experts du domaine et les personnes responsables des systèmes d'information pour les cas étudiés.

De plus, une étude de cas permet également de répondre aux questions que nous voulons aborder avec EVOLIS à savoir le « comment » et le « pourquoi » du processus d'évolution des systèmes d'information. Comme les recherches du domaine n'adressent que partiellement ces questions, une approche étude de cas permet d'apporter des compléments aux recherches actuelles en se focalisant sur les facteurs de déclenchement des évolutions.

Afin d'implémenter de manière judicieuse une étude de cas, nous avons suivi la marche à suivre proposée par Benbasat et al. (1987) afin de décrire et d'implémenter au mieux cette démarche de recherche. Il s'agit de présenter la question de recherche, d'établir quelle est l'unité d'analyse, de sélectionner l'étude d'un cas individuel ou multiple, de détailler le processus de sélection du site ainsi que la méthode de collecte des données.

7.1.1 Rôle de cette première étude de cas

Les rôles de cette étude de cas sont multiples concernant l'évaluation et les améliorations apportées aux composants d'EVOLIS :

- Premièrement, nous voulons tester la pertinence des facteurs déclencheurs d'évolution sélectionnés lors de l'étude de l'évolution de systèmes d'information. En d'autres termes, nous voulons évaluer quels sont les bénéfices que peut produire une telle étude pour la compréhension de l'évolution des systèmes d'information étudiés.
- Deuxièmement, afin de pouvoir étudier l'évolution de ces systèmes de manière systématique, il a fallu développer une méthodologie d'étude systématique des évolutions de systèmes d'information. Cette étude de cas nous permet de préciser et d'évaluer cette méthodologie en fonction des situations rencontrées.
- Troisièmement, nous voulons tester les aptitudes d'EVOLIS à faire émerger des modèles d'évolution (patterns) parmi les données étudiées.
- Dernièrement, nous voulons communiquer nos travaux et obtenir des retours de la communauté de recherche en systèmes d'information en publiant un article de recherche.

Outre ces objectifs principaux, nous avons également comme objectifs de tester différentes alternatives de typologies d'activités d'évolution. Nous avons finalement retenu la typologie de Chapin et al. (2001) qui catalogue les activités d'évolution selon 12 types décrits en Section 6.4.1.

De manière générale, le principal rôle de cette première application d'EVOLIS est l'amélioration et l'évaluation des composants d'EVOLIS à savoir les facteurs déclencheurs d'évolution ainsi que de la méthodologie d'étude de l'évolution.

7.1.2 Question de recherche

La définition d'une question de recherche précise représente une des étapes les plus importantes lors de n'importe quelle étude empirique, les études de cas ne dérogeant pas à la règle (Benbasat et al., 1987; Paré & Dubé, 2003). Comme mentionné précédemment, Yin (2009) ainsi que Benbasat et al (1987) expliquent que les recherches de type étude de cas sont plus appropriées à répondre aux questions « comment » et « pourquoi ». Afin de mieux comprendre

le processus d'évolution des systèmes d'information, cette étude de cas présente les deux questions de recherche suivantes (Métraiiller & Estier, 2014) :

« Comment le cadre conceptuel EVOLIS peut-il aider à étudier l'évolution des systèmes d'information ? »

La réponse à cette première question nécessite de classifier les évolutions ainsi que les activités d'évolution selon les facteurs déclencheurs d'évolution et d'effectuer les regroupements nécessaires afin de mettre en évidence la manière selon laquelle EVOLIS permet l'étude de l'évolution. Ce processus de classification des évolutions se doit d'être effectué systématiquement pour chacune des évolutions. Pour cette raison, nous avons appliqué la méthodologie systématique d'étude des évolutions présente dans EVOLIS. Cette question adresse donc le premier et le second objectif de cette étude de cas (voir Section 7.1.1 ci-dessus).

Et la seconde question de recherche qui précise la première :

« Comment le cadre conceptuel EVOLIS peut-il aider à révéler des modèles (ou des schémas) spécifiques d'évolution dans des traces d'évolution ? »

La réponse à cette seconde question nécessite d'analyser les résultats obtenus après la réponse à la première question (c'est-à-dire après l'application d'EVOLIS) afin d'identifier des modèles d'évolution. Cette question adresse donc le troisième objectif de cette étude de cas (voir Section 7.1.1 ci-dessus).

7.1.3 Unité d'analyse

La description de l'unité d'analyse est un composant essentiel du design d'une étude de cas, trop souvent spécifié de manière imprécise dans les études de cas en système d'information (Paré & Dubé, 2003).

Nous avons déterminé que l'unité d'analyse la plus appropriée pour cette itération est l'évolution globale d'un système, depuis sa mise en production. En effet, nous voulons appliquer et évaluer EVOLIS dès les premières évolutions du système d'information.

Nous avons trouvé judicieux comme première approche d'étudier les évolutions d'un système dans sa globalité et pas uniquement les évolutions d'une application ou d'un projet particulier. Les données d'évolution d'un système global sont de natures plus diverses et plus riches que les évolutions d'une sous partie de ce dernier ce qui est approprié pour l'application et l'évaluation des capacités d'EVOLIS.

Cette première itération ne vise pas à généraliser des résultats ou une théorie d'évolution particulière. Elle ne nécessite donc pas l'étude de systèmes d'information de très grande taille.

Le but de cette étude de cas étant d'itérer sur le design d'EVOLIS et d'évaluer son application, une unité d'analyse de taille contenue semble plus appropriée.

On peut voir cette première étude comme une étude pilote pour déterminer les bonnes unités d'analyse et pour nous familiariser avec le phénomène d'évolution dans son contexte.

7.1.4 Cas individuel ou multiple

Le nombre de cas inclus dans le projet étudié est un élément central de l'approche de recherche par étude de cas. La plupart des recherches de ce type requièrent d'inclure plusieurs cas. Les recherches portant sur un cas individuel sont surtout appropriées si c'est un cas révélateur

(inaccessible jusqu'à présent), un cas critique pour tester une théorie et si c'est un cas unique ou extrême (Yin, 2009).

Nous avons choisi d'étudier plusieurs cas d'évolution. Comme cette étude de cas porte sur des systèmes de taille contenue, l'application d'EVOLIS sur plusieurs cas permet de le confronter à d'avantage de situations. En effet, des données provenant de plusieurs systèmes d'information contiendront une plus grande diversité d'évolutions.

Même si ça n'est pas le but recherché ici, une étude sur de multiples cas produit des résultats de recherche plus généraux (Benbasat et al., 1987), ce qui est évidemment un atout pour l'évaluation de la polyvalence d'EVOLIS grâce à la diversité des situations étudiés.

7.1.5 Sélection des sites

Yin (2009) propose deux critères concernant la sélection des potentiels sites d'étude : les sites pour lesquels il est possible de prédire des résultats similaires et les sites pour lesquels des résultats contradictoires peuvent être prédits.

Avant d'avoir accès aux données d'études, il est difficile de prévoir le type de résultats escompté. Cette étude étant d'avantage exploratoire par rapport au domaine, il est plus judicieux d'étudier des organisations ayant des données comparables. Pour ce faire, Benbasat et al. (1987) relèvent que les recherches étudiant des phénomènes d'un niveau relatifs à l'organisation requièrent une sélection basée sur les caractéristiques des organisations étudiées. Nous avons donc sélectionné des organisations ayant une structure organisationnelle ainsi qu'un profil de système d'information similaires.

Caractéristiques des organisations sélectionnées :

- **Taille de l'organisation**

Nous avons choisi d'étudier les systèmes d'information de petites et moyennes entreprises (PME) car la taille « modeste » de leur système d'information semble plus appropriée pour une première étude de cas.

- **Niveau d'informatisation**

Malgré le fait que nous orientons cette étude de cas sur des petites structures, nous voulons une certaine maturité de leur système d'information. Nous n'avons retenu que les organisations ayant au minimum implanté un progiciel de gestion intégré (ERP) ou un système de gestion similaire.

- **Secteur d'activité**

Le secteur d'activité ainsi que l'aspect public ou privé de l'organisation n'ont pas joué de rôle dans la sélection. Cependant, afin de maximiser les chances de pouvoir retracer toutes les évolutions effectuées sur leur système, les PME que nous avons étudiées ne sont pas actives dans le domaine des services informatiques. Il est en effet plus compliqué de faire évoluer son système d'information « sans laisser de trace » lorsqu'aucune personne de l'organisation n'est experte dans le domaine.

- **Type de système**

Les types de systèmes utilisés ne sont pas primordiaux pour cette étude. Ils peuvent être similaires ou non, mais le niveau d'informatisation minimum doit être au moins un ERP ou un système de gestion intégré.

- **Type de données**

De préférence nous voulons retracer les cycles de vie de ces systèmes d'information depuis leur mise en production. Nous avons donc orienté la sélection des organisations sur celles qui disposent d'un historique des interventions effectuées sur le système depuis le déploiement de ce dernier dans l'organisation.

7.1.6 Méthode de collecte des données

Une description claire des données collectées ainsi que du processus de collecte de ces dernières est un aspect important pour la fiabilité et la validité des résultats d'une étude de cas (Benbasat et al., 1987; Paré & Dubé, 2003). Dans ce dessein, plusieurs méthodes de collecte de données peuvent être utilisées pour les recherches études de cas, voici les 6 méthodes principales présentées par Yin (2009) :

- La documentation : les documents écrits allant des coupures de journaux à des rapports officiels.
- Les documents d'archives : organigrammes, dossiers financiers, enquêtes, documents personnels.
- Les entrevues : conversations guidées ou ouvertes plutôt que des questionnaires.
- Les observations directes : observations passive sur site en relevant les subtilités sur le terrain, si le phénomène d'intérêt n'est pas historique.
- Les observations participatives : observation active en ayant un rôle dans les événements étudiés.
- Les artefacts physiques : objets technologiques, outils, productions, ou tout objet physique pertinent.

Les données collectées sont issues principalement de documentation ainsi que de documents d'archive relatifs aux modifications effectuées sur les différents systèmes d'information des organisations sélectionnées.

Afin d'avoir un accès plus direct à plusieurs organisations satisfaisant nos critères de sélection sans avoir à en démarcher un nombre conséquent, nous nous sommes tournés vers une société de services informatiques active dans le développement et l'installation d'un ERP Open Source ainsi que d'autres services associés (intégration avec des magasins en ligne, géolocalisation...). Leurs activités s'étendent entre la Suisse, la France et la Belgique.

Afin de sélectionner les organisations selon les critères préétablis, nous avons effectué une entrevue avec le responsable des projets de déploiement d'ERP. Cette entrevue nous a permis de préciser et de détailler les besoins de notre étude afin d'identifier les documents essentiels à cette dernière.

Les organisations retenues n'étant pas actives dans le domaine des systèmes d'information, toutes les interventions effectuées sur leur infrastructure ont été exécutées par cette société de service informatique.

Nous avons non seulement collecté divers documents, mais nous avons également effectué des entrevues avec le responsable des projets de déploiement d'ERP afin d'obtenir les éventuelles précisions nécessaires à notre étude.

7.1.7 Description des données

Les données collectées pour cette étude de cas proviennent de 5 PME suisses et une PME française. Toutes ces organisations sont suivies par la même société de service informatique pour ce qui concerne le déploiement ainsi que les évolutions de leur système de gestion.

Les 6 organisations ont toutes déployé la même base d'un système ERP avec, selon les cas et les secteurs d'activité, différentes extensions (intégration de shop en ligne, gestion de plusieurs sites, intégration de caisses enregistreuses, etc.). Le Tableau 6 présente les caractéristiques des organisations sélectionnées pour l'étude ainsi que la période étudiée et le nombre d'évolution durant cette période. Au total, parmi les 6 organisations, EVOLIS a été appliqué sur 18 évolutions.

Les documents collectés contiennent les offres faites par la société de services informatiques aux 6 organisations. Ces offres ont toutes été acceptées et effectuées. L'Annexe A.1 présente un extrait d'une offre étudiée.

Voici un détail des éléments contenus dans les documents analysés :

- Le but du document ainsi que de l'intervention effectuée,
- Le besoin couvert par cette intervention,
- La date d'établissement du document,
- La personne responsable de l'intervention,
- Un planning pour l'accomplissement des activités,
- Un détail du travail effectué pour chacune des activités ainsi que la durée du travail en jours/homme,
- L'impact du travail effectué sur le système (modules concernés),
- L'impact des activités sur les processus de l'organisation (si les activités modifient les processus internes).

Tableau 6 : Présentation des organisations.

Organisation	Localisation et marché	Secteur d'activité	Nombre de collaborateurs	Période étudiée	Nombre d'évolutions
Organisation A	Suisse romande et activité internationale	Industrie du câble réseau et télécom et plus précisément dans la métrologie du câble électrique.	Entre 20 et 49 collaborateurs	2009	3 évolutions
Organisation B	France	Vente d'outillage pour professionnels et particuliers (avec grand magasin en ligne)	Entre 20 et 49 collaborateurs	2011 à 2012	2 évolutions
Organisation C	Suisse allemande	Bureau d'ingénierie en géomatique, environnement et construction	Entre 15 et 20 collaborateurs	2009	4 évolutions
Organisation D	Suisse romande et activité internationale	Bureau d'ingénierie mécanique, fabrication, industrialisation	Entre 15 et 20 collaborateurs	2009 à 2012	4 évolutions

Organisation E	Suisse romande avec distributeurs dans toute la Suisse et activités internationales	Réalisation et mise en service d'installations photovoltaïques	Entre 20 et 49 collaborateurs	2009 à 2011	2 évolutions
Organisation F	Suisse romande, Suisse allemande et activité internationale	Promotion vente et distribution de vins, y compris oenothèques et bars à vins	Entre 10 et 15 collaborateurs	2010 à 2012	3 évolutions

7.2 Analyse des données

Selon la méthodologie d'étude de l'évolution d'EVOLIS décrite au Chapitre 6, la première étape est d'identifier les différentes évolutions.

Les documents collectés contiennent les offres faites par la société de services informatiques aux 6 organisations. Pour chacune de ces offres, l'état du système avant est différent de l'état du système après la réalisation de l'offre, cet élément nous permet d'identifier une évolution même si aucune nouvelle fonctionnalité n'est implémentée (cf. Section 6.1).

Pour cette étude, nous avons donc défini une évolution comme étant un ensemble d'activités suivi par une mise en production, en excluant la mise en place initiale du système. Comme mentionné précédemment, nous avons identifié 18 évolutions parmi les 6 organisations.

La seconde étape de la méthodologie d'étude de l'évolution d'EVOLIS est la caractérisation des évolutions selon la taxonomie décrite en Section 6.2.

Cette étude étant longitudinale, il s'agit de tenir compte des propriétés temporelles des évolutions (réponse à la question « Quand ? » de l'évolution) :

- 1) Déterminer la date de l'évolution.
- 2) Ordonner chronologiquement les évolutions afin de les positionner dans le cycle de vie du système.

Objet du changement (réponse à la question « Où ? » de l'évolution) :

- 3) Déterminer l'objet du changement. Quelle partie du système est impactée par les changements effectués.
- 4) Déterminer l'impact du changement (local / global) ainsi que les répercussions possibles.

Propriétés du système modifiées (réponse à la question « Quoi ? » de l'évolution) :

- 5) Quelles sont les propriétés du système qui ont été modifiées par l'évolution (différence entre avant l'évolution et après l'évolution).

Raisons de l'évolution (réponse à la question « Pourquoi ? » de l'évolution) :

- 6) De quelle nature est l'élément qui a déclenché l'évolution selon les facteurs d'évolution d'EVOLIS.

Réponse à la question « comment ? » de l'évolution :

- 7) Identifier quelles sont les activités entreprises durant l'évolution.
- 8) Classifier les activités selon leur type (Chapin et al., 2001) (cf. Section 6.4.2) ainsi que selon les facteurs d'évolution représentés dans le EVOLIS. (cf. Section 6.4.3).
- 9) Etablir l'effort investi pour chaque activité et pour l'évolution, en jours/homme.

L'évaluation des ressources impliquées dans une activité (au travers des jours/homme) nous permet d'allouer une pondération à chaque activité et donc à chaque facteur d'évolution adressé par les activités.

A la suite de cette pondération, il est donc possible d'établir l'allocation des ressources en fonction des différents facteurs d'évolution d'EVOLIS traités par une évolution.

Afin de garantir la reproductibilité de l'analyse des ressources, ces documents ainsi que la méthodologie d'étude de l'évolution et la description des facteurs d'évolution du cadre conceptuel EVOLIS ont été confiés à une étudiante pour analyse. Pour mener à bien ce travail, l'étudiante avait à disposition un article présentant les enjeux et que le positionnement d'EVOLIS (Métrailler & Estier, 2012a) ainsi que des documents détaillant la méthodologie d'étude de l'évolution (une première ébauche de l'article (Métrailler & Estier, 2014)). Cette seconde analyse a montré une similitude quasi parfaite avec les résultats d'analyse précédemment obtenus. Les différences mineures étaient dues à des connaissances globales concernant les évolutions qui ont été communiquées lors de la collecte de données et qui n'avaient pas été reportées dans les données analysées par l'étudiante. Nous sommes donc confiants sur le fait qu'une tierce personne puisse obtenir des résultats similaires avec une description détaillée d'EVOLIS.

La Figure 22 présente une grille d'analyse d'une évolution. Pour chacune des organisations il y a donc une grille d'analyse de ce type par évolution. Ces résultats sont ensuite regroupés par organisation puis mis en communs.

Sur la grille d'analyse, on retrouve la date à laquelle l'évolution a été effectuée, en l'occurrence la Figure 22 présente la seconde évolution du système de cette organisation.

Le tableau du haut contient la description des différentes activités entreprises pour l'évolution avec respectivement :

- le facteur d'EVOLIS adressé (colonne « 4 Facteurs d'EVOLIS »),
- la durée de l'activité en jours/homme (ressources investies),
- la nature des interventions,
- une justification de l'activité (description du but) et
- le type d'activité selon la typologie décrite en Section 6.4.2 (Chapin et al., 2001).

Les graphiques représentent la répartition ressources investies en fonction des facteurs d'EVOLIS, de la typologie ainsi que de la nature des interventions.

La colonne « Nature des interventions » présente sur la Figure 22 est une classification du type des activités que nous n'avons pas retenu pour EVOLIS. Cette classification provient d'une itération sur le design d'EVOLIS. Nous avons plutôt choisi de typer les activités d'évolution selon la typologie décrite en Section 6.4.2 (Chapin et al., 2001) plutôt que selon la nature des interventions. Cette classification selon la nature des interventions est toutefois intéressante car elle permet d'aborder différemment les activités d'évolution en se basant d'avantage sur les domaines de compétences requis pour effectuer une évolution plutôt que sur le type de l'activité en elle même. Nous avons finalement retenu la typologie de Chapin et al. (2001) car cette dernière est reconnue par la communauté et est largement utilisée dans les recherches abordant les sujets de maintenance et d'évolution. Elle confère également d'avantage de rigueur et quant à l'application des connaissances du domaine.

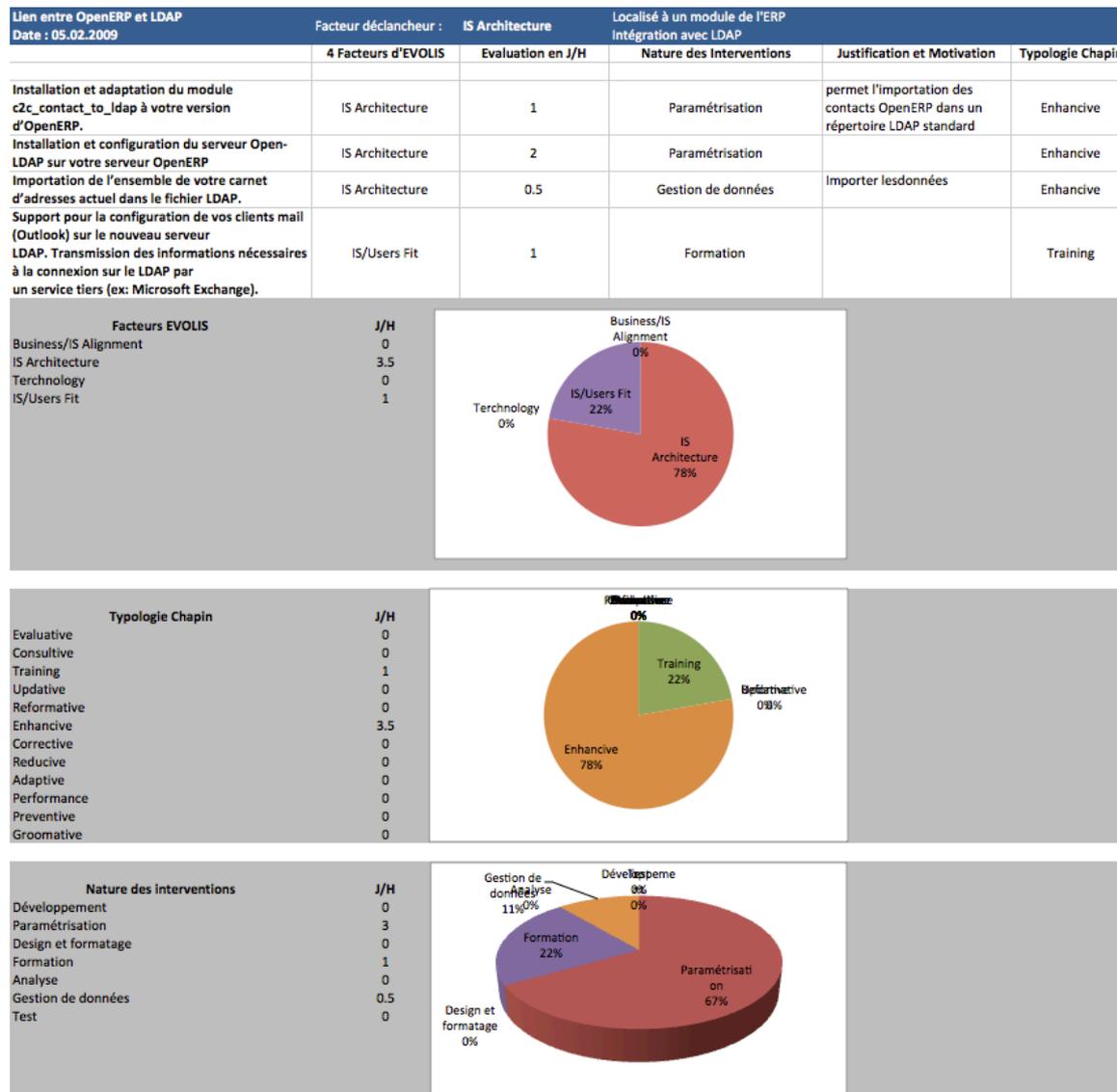


Figure 22. Exemple d'analyse d'une évolution.

7.3 Résultats

Après avoir analysé systématiquement toutes les évolutions identifiées, nous sommes en mesure de retracer l'évolution de chaque système et de mettre en évidence selon quel facteur d'évolution les ressources ont été investies (satisfaction utilisateurs, alignement métier et système d'information, etc.) ainsi que le type des activités effectuées. De plus, grâce à l'organisation chronologique des évolutions, nous pouvons déterminer à quel moment du cycle de vie de ces systèmes ces ressources ont été investies.

Dans cette première étude, nous nous sommes concentrés principalement sur la répartition de l'effort investi dans chaque évolution en fonction des facteurs d'évolution d'EVOLIS. En effet, pour cette première étude de cas, nous avons comme objectifs principaux d'améliorer le design d'EVOLIS et également d'évaluer la pertinence des facteurs d'évolution d'EVOLIS avec des cas concrets d'évolution.

Le Tableau 7 présente la répartition en jours/homme alloué à chaque activité, organisé par société et par chronologie d'évolution. Comme mentionné en Section 7.1.7, la charge de travail

est explicitée dans chaque document d'offre afin de justifier les montants facturés par la société de services informatiques.

Nous expliquons la faible quantité de ressources investies pour la majorité des évolutions (15 des 18 évolutions durent moins de 10 jours/hommes) car la complexité de leur système d'information reste modérée, malgré le fait que toutes les organisations utilisent un ERP, cette complexité est parfaitement maîtrisée par la société de services informatiques. En effet, les organisations sont suivies par les mêmes spécialistes depuis le déploiement de leur système, ces derniers ont donc une connaissance approfondie de leurs systèmes d'information. De plus, les organisations privilégient les technologies recommandées et maîtrisées par leurs spécialistes. Nous avons relevé ce fait suite à certains choix de technologies effectuées par les organisations lors des évolutions. Ce fait est également renforcé par le faible investissement en ressources pour des activités de type « Consultive » et « Evaluative ».

Toutes les organisations utilisent leur système depuis quatre années (2009-2012) ou moins. Sur cette base, nous pouvons faire l'hypothèse que la majorité des systèmes devrait être dans la première partie de leur cycle de vie au cours de laquelle les ressources sont investies dans l'alignement et l'extension du système, plutôt que dans la migration de systèmes obsolètes vers des systèmes à jours.

Tableau 7. Jours/hommes de chaque évolution alloués aux facteurs d'évolution d'EVOLIS.

		Satisfaction utilisateurs	Alignement métier et SI	Technologie	Architecture
Organisation A					
Evolution	1	0.5	4	0	0
	2	1	0	0	3.5
	3	1	1	0	3.5
Organisation B					
Evolution	1	0	3.5	0	1
	2	2	24.5	0	16
Organisation C					
Evolution	1	0.3	1.5	0.2	0
	2	1	1.5	0	2
	3	1.5	3	0	1
	4	1.5	0	0	3.5
Organisation D					

Evolution	1	0.2	10.5	0	0.3
	2	0.2	2	0	0.3
	3	0.2	1.5	0	0.3
	4	1.5	1	0	0.5
Organisation E					
Evolution	1	0	3	0	0.5
	2	1	3.5	0	2
Organisation F					
Evolution	1	0.5	2	0	0.5
	2	3	12.5	0	2
	3	0	2.9	0	1.1

Sur la base de la répartition des ressources entre les différents facteurs d'évolution d'EVOLIS, nous avons regroupé par ordre chronologique les évolutions. En d'autres termes, nous avons regroupé la répartition des ressources pour toutes les premières évolutions, pour toutes les secondes évolutions, et ainsi de suite indépendamment des organisations. Un tel regroupement permet de faire apparaître de façon globale si un ou plusieurs facteurs d'évolution d'EVOLIS ont été adressés de manière privilégiée selon la chronologie des évolutions.

Dans cette optique, le Tableau 8 et graphiquement la Figure 23 présentent le pourcentage d'énergie (de ressources en jours/homme) investi dans chaque facteur pour chaque évolution.

Tableau 8. Pourcentage de jours/hommes alloués à chaque facteur d'EVOLIS en fonction des évolutions.

		Satisfaction utilisateurs	Alignement métier et SI	Technologie	Architecture
Ensemble des évolutions	1	7.10%	78.89%	1.67%	12.35%
	2	14.95%	49.38%	0.00%	35.68%
	3	13.86%	55.06%	0.00%	31.08%
	4	40.00%	16.67%	0.00%	43.33%

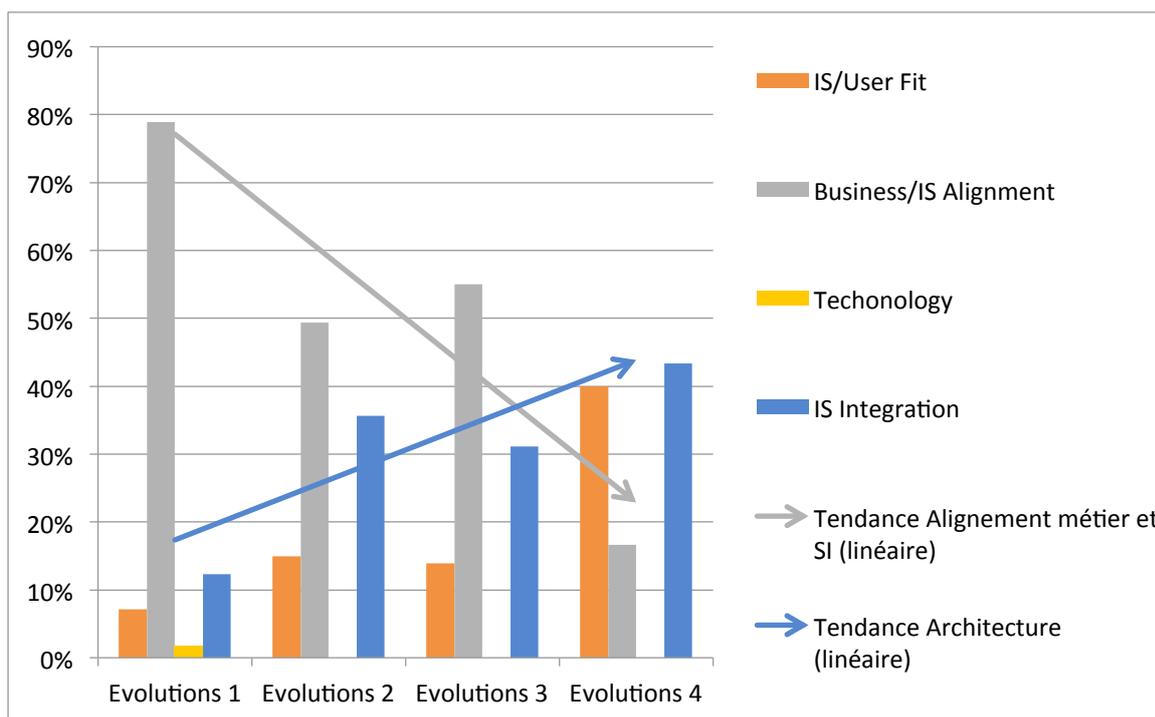


Figure 23. Pourcentage de jours/hommes de travail répartis selon les facteurs d'évolution d'EVOLIS par évolution (tiré de Métrailler & Estier (2014)).

Nous pouvons relever qu'au cours des premières évolutions les activités relatives au facteur « Alignement métier et système d'information » sont prédominantes dans l'ensemble des organisations. En analysant cet aspect d'alignement selon le détail du facteur présenté en Section 5.3.1, nous notons que les activités d'évolution sont relatives au premier niveau d'alignement. Le premier niveau correspond à ce que l'on considère généralement comme étant la notion d'alignement opérationnel entre le système d'information et les buts et les activités de l'organisation. Ce niveau interne d'alignement est une première étape essentielle et nécessaire pour atteindre l'alignement global. Ceci est confirmé par le fait que les activités de ce type sont très présentes lors des premières évolutions des systèmes d'information des organisations étudiées.

Une deuxième constatation intéressante est que, au fil du temps, une tendance à passer d'activités concernant « l'Alignement métier et le système d'information » vers des activités relatives à « l'Architecture » du système apparaît. Les ressources investies dans l'alignement diminuent dans le temps tandis que celles touchant l'architecture augmentent. Cette constatation se révèle en dessinant les courbes de tendances linéaires pour les deux facteurs d'évolution.

En d'autres termes, nous pouvons en déduire qu'après avoir aligné leur système avec leurs activités, ces organisations décident d'étendre la portée de leur système d'information grâce à l'intégration de nouvelles fonctionnalités. Toutefois, cette tendance n'apparaît clairement que dans l'organisation A et l'organisation C, par conséquent, d'autres investigations sont nécessaires pour confirmer cette tendance.

En se basant sur ces deux constats, l'alignement organisationnel semble être une priorité avant d'étendre les fonctionnalités du système d'information. Un troisième constat pouvant renforcer cette tendance est l'évolution des ressources investies dans des activités relatives aux parties prenantes (activités de type « Satisfaction utilisateurs »). En analysant le type de ces activités,

nous trouvons que le type prédominant est « Training ». Dans les évolutions analysées, les activités relatives à la « Satisfaction utilisateurs » viennent conjointement avec les activités relatives à « l'Architecture » du système. Cette constatation fait sous entendre que lorsqu'une organisation étend les fonctionnalités de leur système d'information, en complément, elle organise des activités de formation avec les différentes parties prenantes.

7.4 Discussion

Les résultats présentés ci-dessus découlent de six études de cas basées sur les évolutions de six systèmes d'information de PME. Ces systèmes ayant évolués entre deux et quatre fois, le nombre d'évolutions par système est clairement un trop faible afin de permettre de solides généralisations sur des tendances d'évolutions. Ce n'était d'ailleurs pas la finalité de cette étude. Nous voulions notamment :

- Démontrer l'application d'EVOLIS au travers de plusieurs cas afin de disposer d'une palette d'évolutions diversifiée.
- Evaluer et améliorer le design d'EVOLIS, notamment en précisant la méthodologie d'étude de l'évolution d'EVOLIS ainsi que le contenu des facteurs d'évolution.
- Evaluer la répétabilité de l'analyse. En effet, une tierce personne doit être en mesure d'effectuer une classification similaire sur ces données en partant d'une description d'EVOLIS ainsi que de la méthodologie.
- Soumettre ces résultats à la communauté afin d'obtenir des critiques externes pour améliorer encore le design d'EVOLIS, communiquer sur la recherche et valider les résultats.

Les résultats présentés dans la Section 7.3 mettent en avant les possibilités d'études encourageantes quant à l'application d'EVOLIS avec notamment des opportunités et des hypothèses de futures recherches. En se basant sur le Tableau 8 et la Figure 23 ainsi que sur les constats qui en découlent, nous avons relevé des exemples de futures questions de recherche potentielles :

1. Est-ce que la proportion d'activités concernant « l'Alignement métier et le système d'information » diminuent au fil du temps ?
2. Est-ce que la proportion d'activités relatives à « l'Architecture » croît au fil du temps ?

Pour la première question potentielle, l'interrogation serait de décrire le comportement dans le temps de la proportion d'activités concernant « l'Alignement métier et le système d'information ». Si l'hypothèse de recherche se vérifie et qu'il y a bien une plus forte proportion d'activités d'alignement au début du cycle de vie, il est serait intéressant de connaître quelle en est la cause? Serait-ce lié à la récente mise en production du système d'information, et à ses problèmes de jeunesse ou y a t'il d'autres raisons ?

Pour la seconde question potentielle, malgré le fait que cette tendance n'apparaisse que dans deux organisations, cette question reste intéressante. De plus, cette tendance semble être accompagnée par les activités concernant la « Satisfaction des utilisateurs ». Il serait intéressant d'analyser s'il existe un lien entre la proportion d'activités relatives à « l'Architecture » et la proportion d'activités concernant la « Satisfaction utilisateurs » avec notamment les activités de type « Training ». Ce questionnement pourrait permettre de découvrir des éléments descriptifs ainsi que des conséquences de cette tendance d'évolution.

Dans cette étude de cas, nous n'analysons pas le comportement du dernier composant d'EVOLIS : le « Coût » des évolutions (Section 5.4). En effet, comme vous pouvez le remarquer dans les tableaux et la figure, nous ne comparons pas le montant des ressources investies dans

chaque évolution principalement car le nombre de systèmes d'information ainsi que le nombre des évolutions sont trop faibles pour en extraire des résultats significatifs.

L'étude du « Coût » en parallèle avec celle du cycle de vie du système d'information peut apporter des résultats intéressants sur le lien entre les ressources nécessaires afin de maintenir un niveau de satisfaction suffisant et la position du système d'information dans son cycle de vie. Un second exemple d'étude sur les investissements pourrait être l'étude des variations du « Coût » des évolutions en parallèle avec la variation de couverture opérationnelle du système d'information. En d'autres termes, comment varie le « Coût » des évolutions en fonction de la complexité du système (seconde loi de l'évolution de Lehman, voir Section 4.3).

Selon les objectifs définis pour la conception d'EVOLIS, ce dernier doit revêtir une utilité pour deux populations : les chercheurs en systèmes d'information et les praticiens du domaine. Jusqu'ici, nous avons énuméré les intérêts potentiels de l'utilisation d'EVOLIS pour les chercheurs.

Concernant les intérêts potentiels que pourraient tirer les praticiens des résultats de cette étude de cas, ils sont les suivants :

La première utilité de l'application d'EVOLIS est de connaître précisément la répartition de l'investissement selon les différentes activités des évolutions. Cette connaissance permet une justification détaillée du coût des évolutions. De plus, cette répartition fournit un visuel sur la situation des évolutions ainsi que sur l'ensemble des investissements dans le système d'information.

La seconde utilité de l'application d'EVOLIS est orientée à des fins de pilotage du système. En effet, en s'appuyant sur les résultats de ce type de recherches ainsi que sur leur expérience, les praticiens peuvent interpréter les résultats de l'application d'EVOLIS et les utiliser pour le pilotage du système. Par exemple, une diminution des investissements dans les activités d'évolution de type « Corrective » ou relatives à l'alignement entre métier et système d'information pourrait indiquer que le système est mature et stable et que par exemple d'avantage de ressources pourraient être allouées à l'expansion des fonctionnalités du système. Ce type de résultat est un indicateur sur la position du système dans son cycle de vie et sur la « santé » de ce dernier.

La troisième utilité concerne la planification des évolutions ainsi que leur contenu. En effet, les résultats des recherches sur les évolutions peuvent montrer une relation entre certains types d'activités, par exemple les activités qui étendent des fonctionnalités du système et les activités de formation des différentes parties prenantes. Ces résultats rappellent de ne pas sous-estimer les investissements dans certaines activités « secondaires » à la finalité de l'évolution. Dans cette optique, EVOLIS peut être vu comme un outil d'aide à la planification des évolutions ainsi qu'à la répartition des ressources dans les activités des futures évolutions.

7.5 Apports de l'étude de cas vis-à-vis du processus de design

Cette étude de cas a produit de multiples apports pour le processus de recherche design science que nous avons sélectionné pour la conception de l'artefact EVOLIS. En effet, nous avons pu itérer de multiples fois sur le design d'EVOLIS selon la conformité des résultats obtenus par rapport aux objectifs fixés.

7.5.1 Instanciation d'EVOLIS

Selon Livari (2015), une des premières visions de la recherche design science semble supposer que premièrement on construit la recherche et qu'ensuite si possible on l'instancie. C'est le cas d'ailleurs dans les articles de March & Smith (1995) et A. R. Hevner et al. (2004) pour lesquels l'instanciation montre que les construits, les modèles ou les méthodes peuvent être implémentés. Cette première étude de cas permet justement d'instancier EVOLIS dans un contexte réel. En effet, cette étude de cas permet de passer de l'idée d'un artefact pour étudier l'évolution des systèmes d'information et de ses possibles résultats à une implémentation réelle de cet artefact avec les résultats escomptés.

7.5.2 Démonstration d'EVOLIS

Selon la méthodologie de recherche que nous avons adoptée (décrite en Section 3.5) et présentée dans l'article de Peffers et al. (2007), cette étude de cas s'inscrit dans l'activité dite de « Démonstration » dans laquelle l'usage de l'artefact est démontré. Cette démonstration nécessite une connaissance précise de l'utilisation de l'artefact. Cet élément nous a forcés à apporter des précisions sur la méthodologie d'étude de l'évolution d'EVOLIS. Afin de nous assurer un résultat reproductible par une tierce personne, cette méthodologie se doit d'être un « mode d'emploi » précis d'EVOLIS avec toutes les éléments permettant l'analyse des évolutions. Nous avons évalué de manière positive cette reproductibilité au travers d'un travail mené par une étudiante disposant d'un article présentant les enjeux et que le positionnement d'EVOLIS (Métrailler & Estier, 2012a) ainsi que des documents détaillant la méthodologie d'étude de l'évolution d'EVOLIS (décrite dans une première version de l'article Métrailler & Estier (2014)).

7.5.3 Evaluation et itérations sur le design et le développement d'EVOLIS

Cette démonstration de l'utilisation d'EVOLIS a permis de confronter sur des cas concrets le design d'EVOLIS avec les objectifs prédéfinis. Grâce à cette évaluation du design nous avons repéré les forces ainsi que les faiblesses d'EVOLIS pour l'étude des évolutions. Cette étude de cas peut donc également s'inscrire dans l'activité « Evaluation » de la méthodologie de recherche.

Voici les principaux objectifs évalués positivement lors de cette étude de cas :

- Grâce à l'utilisation d'EVOLIS dans un contexte réel, les résultats qui en découlent confirment la pertinence du choix des facteurs d'évolution représentés dans EVOLIS. En effet, l'utilisation des facteurs d'évolution a permis de représenter la répartition des ressources afin de mieux comprendre les enjeux des évolutions.
- Cette étude de cas confirme également la pertinence d'une étude longitudinale de l'évolution des systèmes d'information. En effet, les résultats permettent d'identifier des tendances d'évolution possibles dans le temps.
- La taxonomie (décrite en Section 6.2) utilisée pour caractériser les évolutions est adaptée aux besoins d'EVOLIS.

Voici quelques points qui ont été améliorés suite à l'évaluation de cette étude de cas :

- Besoin de typer les activités d'évolution pour l'interprétation des résultats. L'attribution d'un facteur d'évolution à chaque activité n'étant pas suffisamment riche.
- Essai de plusieurs typologies pour répondre au point précédent, notamment celles de Lientz et al. (1978), celle de Chapin et al. (2001) (voir Sections 4.6.3 et 6.4) et également une typologie « maison » basée sur la nature des interventions (voir Section 7.2).
- Préciser le contenu des facteurs d'évolution représentés dans EVOLIS.

- Etablir une méthodologie précise et systématique d'étude a posteriori de traces d'évolution afin de rendre l'analyse reproductible.

Nous avons donc effectué plusieurs itérations complètes sur l'activité de design et de développement d'EVOLIS afin de corriger les lacunes de ce dernier vis-à-vis des objectifs escomptés.

7.5.4 Communication

Cette étude de cas a également permis de communiquer sur nos recherches au travers d'une publication lors de la conférence HICSS 2014 (Métrailler & Estier, 2014). Les avis des relecteurs ainsi que les commentaires recueillis lors de la présentation ont permis d'évaluer positivement EVOLIS et d'en améliorer le design. Ces commentaires ont aussi été bénéfiques quant à la mise en forme de cette étude de cas et quant à la conduite de l'étude de cas suivante.

7.6 Conclusions

Cette étude de cas est donc la première confrontation d'EVOLIS avec un contexte réel. Au travers de cette dernière, nous voulions répondre aux deux questions de recherche suivantes : premièrement « Comment le cadre conceptuel EVOLIS peut-il aider à étudier l'évolution des systèmes d'information ? » et la seconde question qui précise la première : « Comment le cadre conceptuel EVOLIS peut-il aider à révéler des modèles (ou des schémas) spécifiques d'évolution dans des traces d'évolution ? ». En plus de ces deux questions de recherches, cette étude de cas fait partie intégrante du processus méthodologique de recherche décrit en Section 3.5. Afin d'effectuer une itération complète sur le processus de design d'EVOLIS, nous avons les objectifs suivants :

- Démontrer l'application d'EVOLIS :
 - Objectif atteint grâce à l'application d'EVOLIS pour l'étude des évolutions de plusieurs systèmes d'information.
- Evaluer et améliorer le design d'EVOLIS pour l'étude de traces d'évolution a posteriori :
 - Objectif atteint en raffinant notamment la méthodologie d'étude des traces ainsi que le contenu d'EVOLIS.
- Soumettre ces résultats à la communauté :
 - Objectif atteint avec la publication de la méthodologie ainsi que de l'étude de cas (Métrailler & Estier, 2014).

A l'intérieur de cette itération, nous avons également effectué plusieurs cycles « Design – Démonstration – Evaluation ». Au fil de ces cycles, le cadre conceptuel ainsi que cette étude de cas sont devenus suffisamment matures pour faire l'objet d'une publication. Les objectifs méthodologiques de construction d'EVOLIS sont donc atteints.

Pour ce qui concerne les deux questions de recherche, cette étude de cas permet d'y répondre de la façon suivante :

- A la question : « Comment le cadre conceptuel EVOLIS peut-il aider à étudier l'évolution des systèmes d'information ? », cette étude de cas permet de répondre qu'une classification systématique des évolutions ainsi que des activités d'évolution permettent effectivement de mieux comprendre quels sont les facteurs déclenchant les évolutions et quand ces facteurs déclenchent les évolutions. Pour les cas étudiés, l'application

d'EVOLIS nous fournit une cartographie claire des évolutions et permet de faire apparaître la répartition des ressources d'évolution en fonction des facteurs afin de retracer le début du cycle de vie de ces systèmes d'information. L'application d'EVOLIS aide donc à l'étude de l'évolution des systèmes d'information.

- A la question : « Comment le cadre conceptuel EVOLIS peut-il aider à révéler des modèles (ou des schémas) spécifiques d'évolution dans des traces d'évolution ? », cette étude de cas permet de répondre qu'en appliquant EVOLIS à l'évolution de plusieurs systèmes d'information, il est possible d'obtenir des modèles d'évolution pour les systèmes étudiés. Même si l'étude de cas actuelle ne permet aucune généralisation d'un modèle d'évolution, elle présente les résultats auxquels un chercheur peut s'attendre en appliquant EVOLIS. Cette étude de cas détaille donc comment l'application d'EVOLIS aux évolutions de plusieurs systèmes d'information permet d'aider à révéler des modèles d'évolution.

Chapitre 8

Mise en œuvre d'EVOLIS : Etude de cas 2

Le chapitre précédent détaillait l'application d'EVOLIS pour l'analyse a posteriori de traces d'évolution provenant des systèmes d'information de plusieurs PME. Cette première application d'EVOLIS sur des cas réels d'évolution nous a permis d'améliorer considérablement le contenu ainsi que la méthodologie d'application de ce dernier. De plus, les résultats obtenus nous donnent une vue d'ensemble sur les bénéfices potentiels quant à l'application d'EVOLIS.

Maintenant que la méthodologie a été testée dans une première étude de cas, nous avons choisi d'appliquer EVOLIS à des données d'évolution de plus grande ampleur. En effet, EVOLIS ne doit pas être destiné uniquement à l'étude de l'évolution de systèmes d'information de faible ampleur. Il se doit d'être polyvalent et adapté également aux systèmes de tailles importantes.

De plus, une seconde étude de cas sur des systèmes de « petite taille » aurait été bénéfique pour confirmer les résultats obtenus précédemment, mais n'aurait pas révélé suffisamment d'éléments supplémentaires pour l'évaluation du design d'EVOLIS. Nous comptons donc sur cette nouvelle confrontation d'EVOLIS avec un contexte réel d'une autre nature que le précédent pour mettre en lumière des éléments nouveaux concernant le design d'EVOLIS et son application.

Le cadre conceptuel EVOLIS ainsi que les résultats de la première étude de cas ont été présentés à un responsable d'un système d'information de plus grande ampleur. Son avis positif vis-à-vis d'EVOLIS et de nos recherches, nous a permis d'accéder à de nouvelles données pour l'étude d'un système d'information de grande taille.

Cette seconde étude de cas porte sur un système d'information de beaucoup plus grande ampleur que les précédents. En effet, nous avons étudié les évolutions d'un système d'information d'hôpital universitaire. Cette seconde application d'EVOLIS permet de confronter ce dernier à un système d'information sur lequel beaucoup d'interventions ont été entreprises.

8.1 Etude de cas

Pour des raisons similaires à celles présentées en Section 7.1 de la précédente étude de cas, nous avons choisi d'appliquer EVOLIS afin d'étudier des évolutions a posteriori. Les données collectées permettent donc d'étudier les évolutions d'un système d'information dans son contexte réel d'utilisation.

Nous avons choisi d'effectuer une étude de cas pour cette seconde application d'EVOLIS car l'objectif visé d'EVOLIS reste la réponse aux questions « comment » et « pourquoi » de l'évolution ainsi que la compréhension du processus étudié.

Comme pour l'étude de cas présentée au Chapitre 7, nous avons suivi la marche à suivre proposée par Benbasat et al. (1987) afin d'implémenter de manière rigoureuse la démarche de recherche pour cette étude de cas ; il s'agit de présenter la question de recherche, d'établir quelle est l'unité d'analyse, de sélectionner l'étude d'un cas individuel ou multiple, de détailler le processus de sélection du site ainsi que la méthode de collecte des données.

8.1.1 Rôle de cette seconde étude de cas

Alors que le rôle principal de la première étude de cas portait sur l'amélioration et l'évaluation des facteurs déclencheurs d'évolution ainsi que de la méthodologie d'étude de l'évolution, cette seconde étude de cas vise d'avantage à appliquer EVOLIS sur des données d'évolution non seulement réelles, mais également d'un volume suffisant pour en tirer des enseignements plus robustes.

Dans cette optique d'apport de nouvelles contributions théoriques, nous voulions avec l'application d'EVOLIS sur un grand volume de données d'évolution :

- Identifier des modèles spécifiques d'évolution (pattern).
- Affirmer ou infirmer les hypothèses de modèle d'évolution de l'étude de cas précédente.
- Représenter les cycles de vie sur le long terme des différents éléments étudiés.
- Identifier des possibilités de conduite de l'évolution et de représentation de données d'évolution avec un grand volume de données.

Ce dernier point est développé dans le Chapitre 9.

Nous avons également des objectifs d'amélioration et d'évaluation continue d'EVOLIS suite à son application sur un grand volume de données :

- Amélioration de la description des facteurs d'évolution d'EVOLIS suite à l'application sur un grand volume de données.
- Conception et évaluation d'un modèle conceptuel générique permettant de stocker et de traiter un grand volume de données d'évolution.

8.1.2 Question de recherche

La définition d'une question de recherche précise représente une des étapes les plus importantes lors de n'importe quelle étude empirique, les études de cas ne dérogeant pas à la règle (Benbasat et al., 1987; Paré & Dubé, 2003). Yin (2009) ainsi que Benbasat et al (1987) expliquent que les recherches de type étude de cas sont plus appropriées à répondre aux questions « comment » et « pourquoi ». L'étude de cas précédente nous a permis d'améliorer le design d'EVOLIS en se basant sur un volume de données relativement restreint. Dans cette seconde étude de cas, le but est toujours l'évaluation et l'amélioration des possibilités d'EVOLIS.

Afin de mieux comprendre le processus d'évolution des systèmes d'information de grande ampleur cette étude de cas présente les deux questions de recherche suivantes :

« Comment un grand volume de données concernant l'évolution des systèmes d'information peut-il nous permettre d'améliorer le design d'EVOLIS ? »

En appliquant EVOLIS sur un grand volume de données, nous désirons identifier ses faiblesses quant à l'étude de l'évolution. Les faiblesses que nous souhaitons corriger peuvent être en rapport avec la méthodologie d'application d'EVOLIS, avec la description des facteurs d'évolution, avec les éléments de littérature utilisés, et avec le traitement et l'analyse des données d'évolution.

« Comment EVOLIS peut-il permettre de retracer et de comprendre les différentes phases du cycle de vie des systèmes d'information ? »

En représentant de manière visuelle les résultats de l'application d'EVOLIS, nous voulons, en tant qu'observateur externe, pouvoir comprendre les différentes étapes par lesquels sont passés les éléments étudiés. En s'appuyant sur cette compréhension, nous voulons également identifier des possibilités de pilotage et de conduite de l'évolution relatives à ces cycles de vie.

« Comment EVOLIS peut-il aider à révéler des modèles (ou des schémas) spécifiques d'évolution dans un grand volume de données concernant l'évolution des systèmes d'information ? »

Grâce à un volume important de données d'évolution, nous espérons pouvoir identifier des suites d'activités d'évolution identiques, c'est-à-dire des modèles (patterns) spécifiques d'évolution. Ces modèles d'évolution ont plusieurs utilités, premièrement ils servent à mieux comprendre les enchaînements des différents types (selon les facteurs déclencheurs) d'évolution, et deuxièmement ils représentent une information importante dans le pilotage et la conduite stratégique de l'évolution des systèmes d'information.

8.1.3 Unité d'analyse

La description de l'unité d'analyse est un composant essentiel du design d'une étude de cas, trop souvent spécifié de manière imprécise dans les études de cas en systèmes d'information (Paré & Dubé, 2003).

Dans l'étude de cas précédente qui revêtait un aspect plutôt exploratoire et évaluatif des possibilités d'EVOLIS, nous avons choisi une unité d'analyse de taille contenue afin de ne pas perdre de temps dans le traitement des données. Pour cette seconde étude, l'objectif est de voir comment se comporte EVOLIS face à un grand volume de données. Il nous faut donc trouver une organisation disposant de source de données d'évolution suffisantes.

Selon le volume de données disponible, nous avons décidé d'orienter notre étude sur une unité d'analyse relative aux systèmes ou aux applications stratégiques de l'organisation. En effet, les données d'évolution sur ce genre de système ont, en plus d'un volume conséquent, bien plus d'intérêt que celles concernant les systèmes secondaires.

En plus des applications stratégiques, nous voulions également étudier les évolutions engendrées par des projets sur différents composants du système d'information. Cette approche projet peut être très intéressante pour le gestionnaire afin d'avoir une vision détaillée sur l'impact des projets en systèmes d'information ainsi que sur l'allocation des ressources du projet.

Pour l'étude de cas précédente, nous avons étudié l'évolution de systèmes dès leur première mise en production. Pour cette seconde étude de cas, cet aspect n'est plus primordial. Afin de

relever la polyvalence d'EVOLIS, nous voulons appliquer EVOLIS sur des systèmes peu importe si les données disponibles le sont depuis leur première mise en production ou non.

8.1.4 Cas individuel ou multiple

Le nombre de cas inclus dans l'étude est un élément central de l'approche de recherche par étude de cas. La plupart des recherches de ce type requièrent d'inclure plusieurs cas. Les recherches portant sur un cas individuel sont surtout appropriées si c'est un cas révélateur (inaccessible jusqu'à présent), un cas critique pour tester une théorie et si c'est un cas unique ou extrême (Yin, 2009).

Comme dans l'étude précédente, nous avons choisi d'étudier plusieurs cas d'évolution. Malgré le fait que l'application d'EVOLIS sur un cas disposant d'un volume de données important permettrait de le confronter à une multitude de situations, de multiples cas peuvent représenter avec plus de fidélité la position d'un gestionnaire de systèmes d'information.

Comme mentionné précédemment, une étude sur de multiples cas produit des résultats de recherche plus généraux (Benbasat et al., 1987), ce qui est évidemment un atout pour la polyvalence d'EVOLIS ainsi que son application en contexte réel.

8.1.5 Sélection des sites

Pour la sélection des sites, nous voulions trouver un ou plusieurs sites ayant la même caractéristique principale : un grand volume de données d'évolution.

Le site que nous avons sélectionné dispose de suffisamment de données pour effectuer plusieurs études de cas. Nous nous sommes donc concentrés sur ce seul site et n'avons pas eu recours à une sélection basée sur les caractéristiques de cette organisation. Ces caractéristiques sont détaillées ci-dessous.

Caractéristiques de l'organisation sélectionnée :

- **Taille de l'organisation**

Nous avons choisi d'étudier les systèmes d'information d'une organisation de grande taille car nous voulions absolument avoir un grand volume de données d'évolution. Exprimée en collaborateurs, il est communément admis qu'une petite organisation compte entre 10 et 250 collaborateurs, une organisation de taille intermédiaire en compte entre 251 et 5000, et une grande organisation plus de 5000. L'organisation que nous avons étudiée (un hôpital universitaire) compte plus de 9000 collaborateurs.

- **Niveau d'informatisation**

L'organisation compte plus de 9700 postes de travail⁴ et 1600 imprimantes. Cela est nécessaire afin d'exploiter plus de 250 applications métiers et environ 2000 logiciels spécifiques⁵.

L'organisation dispose de sa propre équipe d'experts informatique pour la gestion, la mise en place ainsi que l'évolution des différents composants du système d'information. La Figure 24 présente la structure du groupe systèmes d'information de l'organisation.

⁴ Chiffre de 2013

⁵ Source interne au 06.09.2010 et au 02.2013

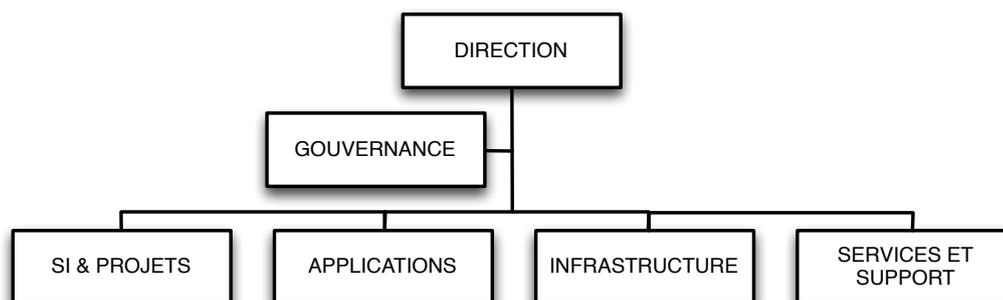


Figure 24. Structure de la Direction des Systèmes d'Information.

Nous avons étudié les activités de l'équipe « Applications ». Cette équipe est en charge de la gestion du parc d'applications. Toutes les modifications effectuées sur le comportement, les interconnexions, les mises à jour, etc. sont effectuées par cette équipe. Elle est donc au centre du processus d'évolution de la couverture fonctionnelle du système d'information. Elle est composée d'environ 50 collaborateurs qui s'occupent d'environ 250 applications.

- **Secteur d'activité**

En tant qu'hôpital universitaire, le secteur d'activité de l'organisation étudiée est public.

- **Types de systèmes**

Vu l'envergure et la diversité des applications du système d'information étudié, nous nous sommes concentrés sur les systèmes stratégiques de l'organisation (systèmes relatifs aux patients). En effet, l'équipe « Applications » s'occupe de systèmes dont les fonctionnalités vont de la gestion des parkings à la gestion des dossiers des patients.

- **Type de données**

Nous n'avons pas de préférence sur la nature des données d'évolution. Que les données soient disponibles depuis la mise en production du système ou non n'est pas de première importance. Nous voulons surtout une grande diversité ainsi qu'un volume important. La source des données n'est également pas déterminante, qu'elles proviennent d'un système de suivi de modifications ou d'autres méthodes de relevés.

8.1.6 Méthode de collecte des données

Une description claire des données collectées ainsi que du processus de collecte de ces dernières est un aspect important pour la fiabilité et la validité des résultats d'une étude de cas (Benbasat et al., 1987; Paré & Dubé, 2003). Dans ce dessein, plusieurs méthodes de collecte de données peuvent être utilisées pour les recherches études de cas. Les 6 méthodes principales méthodes sont les suivantes : la documentation, les documents d'archives, les entrevues, les observations directes, les observations participatives ainsi que les artefacts physiques. Ces méthodes sont décrites dans la Section 7.1.6 et dans Yin (2009).

Les données collectées sont issues principalement de documents d'archive relatifs aux interventions effectuées sur les différentes applications du système d'information de l'organisation.

L'accès aux différentes sources de données nous a été accordé par le responsable du groupe Application (cf. Figure 24). Le groupe Applications est chargé du support et de la maintenance des quelque 250 applications informatiques. Il a également la responsabilité de l'entrepôt de données institutionnel et de la plateforme d'intégration d'applications.

Plusieurs entrevues ont été nécessaires afin d'obtenir les accès requis pour la récolte des données. Ces entrevues ont été cruciales non seulement pour ce qui est de la description des différentes sources de données, mais également pour la compréhension du rôle des différentes applications ainsi que pour la compréhension des différentes sources.

8.1.7 Description des données

Données d'évolution

Les données collectées pour cette étude de cas proviennent de plusieurs sources. La source contenant le plus grand volume de données d'évolution est leur système JIRA. JIRA est un logiciel propriétaire de suivi de problèmes développé par Atlassian Software Systems⁶. Il peut être utilisé comme un système de suivi de bugs, un système de gestion des incidents ou un système de gestion de projets.

L'hôpital étudié utilise le système JIRA afin de regrouper toutes les demandes de modifications (sous forme de tickets) sur les différentes applications et systèmes utilisés. Nous y retrouvons toutes les traces de la maintenance effectuée sur l'ensemble du système d'information. En général, ces modifications sont des mises à jour d'interfaces, de processus, de couplage de composants, de documents, etc.

Les interventions sur les systèmes sont en règle générale regroupées par applications, il est donc aisé d'accéder à toutes les interventions effectuées sur une même application. Nous avons procédé de la sorte pour cibler l'extraction d'une partie des données d'évolution que nous avons étudiées.

Nous avons également remarqué dans le système JIRA des interventions regroupées par projet. En effet, certains projets sont transversaux à l'organisation et impactent donc plusieurs systèmes. Les regroupements d'interventions sont nommés « Projet » dans le système. Nous utiliserons ce terme pour qualifier les regroupements d'intervention indépendamment du fait que les interventions soient regroupées par application, projet ou autre.

Parmi ces projets, nous notons 3 projets (regroupements d'interventions) particuliers, les projets nommés :

- Demande d'évolution du système d'information
- Mise en production
- Intégration

Ces 3 projets regroupent des interventions particulières. Au lieu d'être classifiées en fonction de l'application concernée ou du projet concerné, elles sont classifiées selon leur nature ou leur impact sur le système d'information de l'organisation. Les interventions regroupées dans ces projets ne sont donc pas spécifiques à une application mais sont transversales à toutes les applications de l'organisation. Voici un détail de ce que ces projets contiennent :

- Projet : « Demande d'évolution du système d'information », ce projet regroupe des demandes d'interventions et non pas des interventions réalisées. Parmi les demandes contenues dans ce projet, environ 50% sont abandonnées.

⁶ <http://fr.wikipedia.org/wiki/Jira> - <https://www.atlassian.com/software/jira>

- **Projet** : « Mise en production », ce projet contient toutes les interventions sur les applications ayant nécessité un redémarrage de l’application ou du système qui l’héberge.
- **Projet** : « Intégration », ce projet regroupe les interventions qui impactent le couplage entre composants du système d’information.

Pour chaque intervention, les intervenants saisissent des informations au sujet du travail entrepris ou à entreprendre. Cependant, tous les projets n’ont pas les mêmes champs de saisie et chaque champ n’est pas systématiquement renseigné.

Lors du traitement des données, pour chaque projet, nous avons supprimé les champs qui n’ont jamais été renseignés pour aucune des interventions présentes dans le projet. Le détail du contenu des traces d’évolution est disponible en Annexe B.1 et B.2.

Données financières

Le second type de sources de données concerne les investissements externes effectués pour l’ensemble des applications. Ces sources contiennent tous les paiements sous forme de factures effectués aux divers prestataires du système d’information du début 1997 jusqu’au début 2012.

Nous avons ici deux sources distinctes. La première est une base de données Microsoft Access contenant les factures avec des relations sur les différents projets auxquels ces dernières sont attribuées ainsi que les prestataires de services ayant émis ces factures. Les factures entrées dans cette base concernent tous types de prestations allant de la facture pour du matériel en passant par les frais de licence des logiciels aux prestations effectuées par des prestataires externes. Cette source contient les factures qui s’échelonnent depuis le début 1997 jusqu’à la fin 2009. Elle contient plus de 4900 entrées. Le schéma de la base de données Microsoft Access est présenté dans la Figure 25.

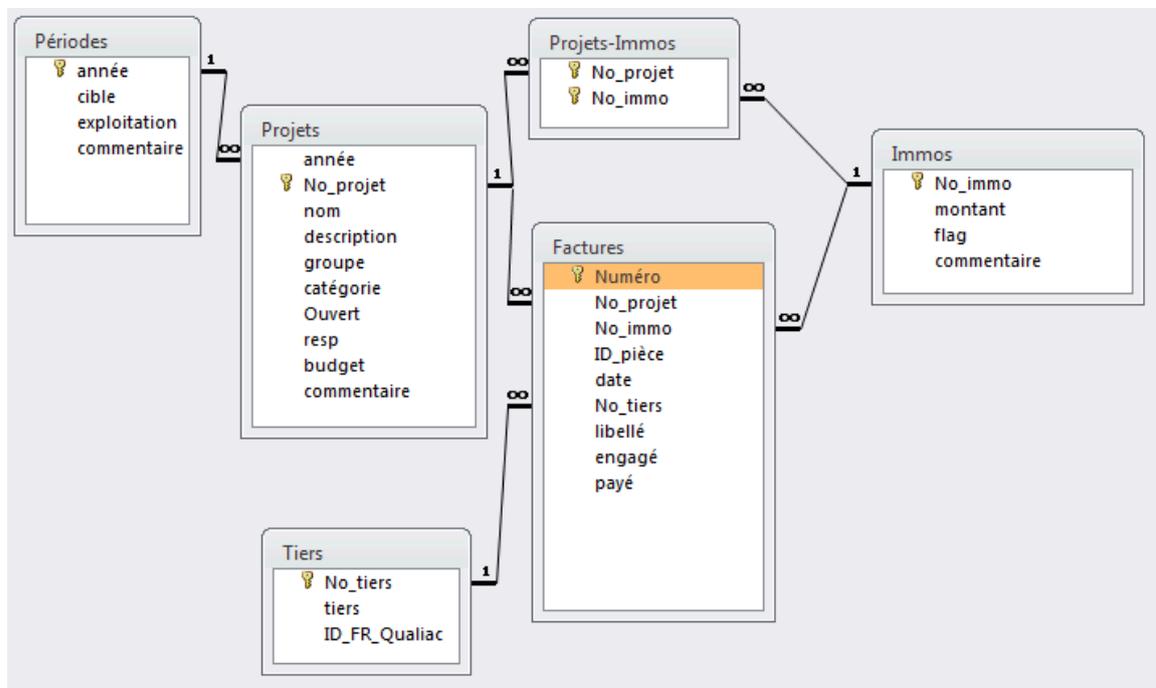


Figure 25. Source de données Microsoft Access contenant les factures informatiques de 1997 à 2009.

La seconde source de données financières contient des données d’un type similaire à la source précédente à savoir les écritures comptables concernant les budgets ainsi que les factures

payées aux divers prestataires du système d'information. Au lieu d'être en format base de données, elle est en format tableur (Microsoft Excel). Cette source contient moins d'informations sur les factures que la base de données précédente, néanmoins toutes les informations importantes pour notre étude s'y retrouvent. Pour imaginer le contenu cette source, ça semble être le résultat d'une jointure entre les tables Projets, Factures et Tiers avec une sélection de certaines colonnes et des noms de colonnes différents. Cette seconde source contient les factures pour la période allant du début 2009 au début 2012. Elle contient plus de 2800 entrées et permet donc de prolonger dans le temps le contenu de la base de données précédente.

Conclusion sur les sources de données

La mise en parallèle de plusieurs sources de données permet de retracer les évolutions des objets d'étude avec une plus grande fidélité. Il est en effet possible de compléter les zones d'ombres qu'il pourrait y avoir dans une partie des sources, voir de confronter les différentes sources sur des éléments communs.

Entre les données financière qui sont des dépenses externes et les données d'évolution qui proviennent d'activités réalisées en interne, la méthode utilisée pour estimer l'importance des ressources investies diverge. Pour les données financière, nous utilisons le coût afin de quantifier leur importance. Pour les activités réalisées en interne, le coût ainsi que la durée des activités ne sont pas disponibles, nous avons donc utilisé leur fréquence afin d'estimer le niveau d'activité. La comparaison précise entre le niveau de ressources investies en interne et en externe n'est donc pas réalisable.

Les données financières contiennent également des frais qui ne sont pas liés directement à l'évolution des objets d'étude comme par exemple le paiement des licences. Il est donc parfois difficile de relever des liens entre les deux sources de données. Cependant, l'observation en parallèle de l'activité financière d'une part et opérationnelle de l'autre peut permettre d'identifier certains événements de la vie de l'objet étudié et de les analyser plus en détail.

8.2 Analyse des données

Cette section détaille les objets analysés, leur histoire ainsi que leur rôle dans l'organisation. Nous présentons ensuite le processus d'extraction et de sélection des données relatives à chacun des objets analysés. Nous présentons également le modèle de données que nous avons implémenté afin de faciliter le regroupement et l'extraction des données pertinentes pour notre analyse.

Selon la méthodologie d'étude de l'évolution d'EVOLIS décrite au Chapitre 6, la première étape est d'identifier les différentes évolutions.

Comme mentionné dans la Section 6.1, nous avons relevé plusieurs façons d'identifier ce qui compose une évolution. En comparaison avec l'étude de cas précédente (Chapitre 7), l'identification des différentes évolutions ne peut se faire de la même manière. En effet, contrairement à l'étude de cas précédente, l'organisation dispose d'une équipe dédiée à l'entretien du système d'information. Environ 50 personnes travaillent à temps complet sur le maintien des applications du système. L'évolution de ce dernier a donc tendance à s'opérer de manière continue.

Nous considérons donc une évolution comme étant l'accomplissement d'une intervention sur le système d'information. Chaque intervention menée est donc considérée comme une évolution comportant une activité.

Comme pour l'étude de cas précédente, il n'est pas nécessaire que l'activité modifie le fonctionnement du système pour qu'elle soit qualifiée comme une évolution.

La seconde étape de la méthodologie d'étude de l'évolution d'EVOLIS est la caractérisation des évolutions selon la taxonomie décrite en Section 6.2.

Cette étude étant longitudinale, il s'agit de tenir compte des propriétés temporelles des évolutions (réponse à la question « Quand ? » de l'évolution) :

- 1) Déterminer la date de l'évolution.
- 2) Ordonner chronologiquement les évolutions afin de les positionner pour retracer le cycle de vie du système.

Objet du changement (réponse à la question « Où ? » de l'évolution) :

- 3) Déterminer l'objet du changement. Quelle partie du système est impactée par les changements effectués.

Concernant l'objet du changement, nous avons ciblé notre étude sur les évolutions de 2 applications ainsi que d'un projet d'informatisation détaillés dans la Section 8.2.1.

Propriétés du système modifiées (réponse à la question « Quoi ? » de l'évolution) :

- 4) Quelles sont les propriétés du système qui ont été modifiées par l'évolution (différence entre avant l'évolution et après l'évolution).

Raisons de l'évolution (réponse à la question « Pourquoi ? » de l'évolution) :

- 5) De quelle nature est l'élément qui a déclenché l'évolution selon les facteurs d'évolution d'EVOLIS.

Réponse à la question « comment ? » de l'évolution :

- 6) Identifier quelles sont les activités entreprises durant l'évolution. Concernant cette étude de cas, comme les activités d'évolution sont menées de manière continue, chaque intervention est considérée comme une évolution comportant une activité.
- 7) Classifier les activités selon leur type (Chapin et al., 2001) (cf. Section 6.4.2) ainsi que selon les facteurs d'évolution représentés dans le EVOLIS. (cf. Section 6.4.3).
- 8) Etablir l'effort investi pour chaque activité et pour l'évolution en jours/homme. Malheureusement il n'a pas été possible de retrouver de manière fiable cette information. Cependant, comme les interventions sont effectuées en continu, elles semblent être (selon leur description) pour la grande majorité d'une durée relativement courte et semblable.

8.2.1 Objets de l'étude

Nous avons concentré notre analyse sur les évolutions concernant deux applications stratégiques pour notre organisation :

- l'application de gestion administrative des patients nommée Axya ainsi que
- l'application supportant le dossier patient informatisé (Dophin) nommée Soarian.

En plus de ces deux applications, nous avons également étudié :

- le projet de mise en place du dossier patient informatisé nommé Dophin.

Ce projet Dophin concernant la mise en place du dossier patient informatisé est étroitement lié avec l'application Soarian. En effet, Dophin est le nom du projet pour l'informatisation du dossier patient et Soarian est la solution logicielle retenue pour le projet Dophin.

Projet Dolphin : Dossier patient informatisé

En milieu hospitalier, le « dossier patient » est l'assemblage des dossiers de tous les intervenants du système de soins de l'hôpital. A l'origine de chaque document se trouve un acte médical dans le sens large du terme. Il peut s'agir de l'admission du patient, de la réalisation d'un examen, d'une consultation, d'un transfert d'un service à l'autre, etc. Un dossier patient contient trois parties principales : le dossier administratif (données administratives, assurance, facturation), le dossier médical (résultats d'examens, prescriptions, etc.) et le dossier infirmier (feuilles de surveillances, notes de soin, etc.) (Zazani, 2003). Il permet donc aux intervenants hospitaliers de partager les informations de santé utiles à la coordination des soins du patient.

L'environnement hospitalier repose de plus en plus sur son système d'information dit hospitalier. Il est impensable que le dossier patient ne suive pas cette évolution vers un dossier électronique du patient. Vers la fin des années 1990 début 2000, le dossier patient informatisé a vu le jour dans l'organisation. Ne pouvant plus assurer le stockage du nombre croissant de documents papiers, il a été décidé de numériser ces derniers. Cette numérisation a donné naissance au premier projet de dossier patient informatisé nommé projet ARCHIMEDE (Archives Médicales Informatisées). Suite au nombre croissant de patients pris en charge conjointement par l'organisation ainsi que par les autres établissements hospitaliers de la région il est devenu essentiel que tous les établissements disposent d'un même système d'information clinique afin de partager les données, d'assurer l'interopérabilité des systèmes d'information et d'améliorer la communication entre les partenaires sur la prise en charge du patient. Ces objectifs se sont concrétisés grâce au projet nommé Dolphin qui a débuté en 2008 avec la collaboration de l'organisation étudiée et les autres établissements hospitaliers de la région. C'est la société Siemens qui remporte l'appel d'offre avec le produit Soarian étant le plus adapté à l'informatisation du dossier patient. Dès 2009, la préparation et le déploiement de cette solution se fait dans l'organisation pour être par la suite progressivement étendue dans les autres établissements. La mise en service de la solution s'est faite en 2011. L'investissement global pour cette solution se monte à 21 millions de francs suisses dont 12 millions pour l'organisation étudiée et 9 millions pour les autres établissements⁷.

Dolphin est donc le nom du projet du dossier patient informatisé commun entre l'organisation étudiée et les autres établissements hospitaliers de la région et Soarian en est la solution logicielle.

Soarian : Application supportant le dossier patient informatisé

Soarian Clinicals est un logiciel développé par la société Siemens Health Services. C'est l'outil hospitalier dédié à la mise en place du dossier patient informatisé en vue d'assurer le partage des données médicales permettant d'améliorer la collaboration entre les différentes institutions participant au projet Dolphin. Soarian pourra ainsi constituer un élément central de la stratégie «e-Health» suisse⁸.

Soarian sert donc de plateforme d'information homogène pour l'intégration et le renfort de la collaboration entre les différentes institutions participant au projet de dossier patient informatisé Dolphin. La plateforme Soarian a été mise en service au milieu de l'année 2011.

⁷ <http://www.bicweb.vd.ch/communiquel.aspx?pObjectID=441575>

⁸ www.e-health-suisse.ch

Axya : Application de gestion administrative des patients

Avant l'arrivée de l'application Axya, la gestion des patients était assurée par l'application BHIS (Burroughs Hospital Information System). Mise en service en janvier 2002, cette application marque la transition entre un système d'information hospitalier centralisé vers un système distribué. Cette application de gestion administrative des patients assure le suivi administratif du patient de son arrivée à l'hôpital jusqu'à la facturation (Zazani, 2003). L'application Axya a été développée par la société Axya Medical, Inc. fondée en 1996 au Massachusetts. Depuis 2007, c'est la société Cerner France⁹ qui assure le support du système Axya pour l'organisation étudiée.

Afin d'effectuer sa tâche de suivi du patient, l'application Axya est connectée à plusieurs autres applications que l'on peut qualifier de « départementales ». En effet, la majorité des départements de l'hôpital (laboratoires, radiologie, imagerie, endoscopie, etc.) utilisent des applications spécifiques à leur métier. Malgré cette diversité applicative, il est essentiel que la communication entre toutes ces applications spécifiques et Axya soit assurée. Ainsi les différentes applications ont accès dans Axya aux informations administratives nécessaires quant à leurs patients et peuvent, à la sortie du patient, communiquer à Axya les prestations effectuées en vue de la facturation. De manière similaire, si un patient change de service lors de son séjour, il est crucial que les résultats, analyses, etc. suivent le patient et soient accessibles par le nouveau service (Zazani, 2003).

8.2.2 Extraction des données

Comme mentionné en Section 8.1.7, la principale source de données d'évolution étudiée est le système JIRA. Nous avons donc extrait depuis le système JIRA les données relatives aux trois objets d'étude à savoir les projets :

- Dophin (projet d'informatisation)
- Soarian (application)
- Axya (application)

L'Annexe B.1 présente le détail des informations contenues dans ces projets JIRA.

De plus, afin de regrouper toutes les activités relatives à ces trois objets d'étude, nous avons également extrait les interventions concernant ces trois objets que nous avons retrouvés dans les projets JIRA suivants :

- Demande d'évolution du système d'information
- Mise en production
- Intégration

Les informations étudiées proviennent donc de six projets JIRA, les trois objets d'étude ainsi que les trois projets transversaux contenant des interventions sur toutes les applications de l'organisation y compris les applications Soarian et Axya ainsi que le projet d'informatisation Dophin.

L'Annexe B.2 présente le détail des informations contenues dans les projets JIRA transversaux.

Projets : Dophin, Soarian, Axya

Ces projets regroupent les interventions effectuées concernant chaque objet d'étude.

⁹ www.cerner.fr

- Le projet concernant Le projet Dolphin contient 51 interventions effectuées en 2009.
- Le projet concernant l'application Soarian contient 191 interventions effectuées entre 2011 et 2013.
- Le projet concernant l'application Axya contient 346 interventions effectuées entre 2006 et 2013.

Projet : Demande d'évolution du système d'information

Ce projet regroupe des demandes d'interventions concernant toutes les applications en fonction dans l'organisation. Comme ce sont des demandes d'intervention et pas des interventions réalisées, il contient également l'indication si la demande a été réalisée ou non. Parmi les demandes contenues dans ce projet, environ 50% sont abandonnées. Ce projet regroupe 629 entrées au 27.02.2013.

Projet : Mise en production

Ce projet regroupe toutes les interventions nécessitant redémarrage d'une application ou d'une machine parmi la totalité des composants du système d'information. Ces interventions ont parfois des références à d'autres interventions qui leurs sont liées. Ce projet contient 816 interventions au 27.02.2013 dont environ 580 sont réalisées ou en cours de réalisation.

Projet : Intégration

Ce projet contient toutes interventions qui concernent plusieurs applications et qui nécessitent la modification ou la création d'un moyen de communication entre les applications. Il contient 45 entrées au 27.02.2013. Ces interventions ont parfois des références à d'autres interventions qui leurs sont liées. Lors de l'analyse, nous avons remarqué que les autres projets contenaient également des interventions relatives à la communications entre applications, cependant le projet Intégration regroupe les interventions pour lesquelles une interface de communication a dû être spécialement mise en place afin de permettre la communication.

Processus d'extraction et de sélection des traces d'évolution

Les données concernant les six projets JIRA précités ont été extraites depuis le système JIRA en utilisant le système de requêtes de JIRA (JIRA Query Language (JQL)), très semblable à du SQL. Cette extraction nous a permis de rassembler des fichiers .csv contenant toutes les interventions effectuées concernant les objets d'étude.

Les données ainsi récoltées ont été nettoyées à l'aide d'un tableur pour supprimer les colonnes d'informations qui n'ont jamais été renseignées.

Afin de sélectionner les interventions relatives aux trois objets d'étude parmi les entrées présentes dans les projets transversaux aux applications de l'organisation, nous avons comparé deux démarches d'extraction. Dans la première démarche, les interventions relatives à chaque objet d'étude ont été sélectionnées dans les trois projets transversaux grâce à une requête JIRA QL dans le système JIRA. Dans la seconde démarche, nous avons créé une base de données dans laquelle nous avons importé la totalité des données des trois projets transversaux avant d'effectuer des requêtes pour extraire individuellement les interventions relatives aux objets d'étude. Les deux démarches ont fourni des résultats quasi similaires (les différences étant certainement dues aux subtilités du langage JIRA QL). Comme la seconde démarche nous permettait de faire des requêtes en s'affranchissant de JIRA QL et sans avoir à se connecter au réseau de l'organisation puis au système JIRA, nous avons continué avec la seconde démarche pour les autres objets d'étude.

Nous avons effectué une extraction large des interventions relatives à chaque objet d'étude parmi les trois projets transversaux. Le tri final afin de déterminer si les interventions sont bel et bien liées aux objets d'étude a été fait manuellement lors de l'analyse des traces d'évolution.

Pour chaque objet d'analyse, voici les sources des interventions que nous avons étudiées :

- Projet Dolphin, 87 interventions au total dont :
 - Dolphin, 51 interventions
 - Demande d'évolution du système d'information, 28 interventions
 - Mise en Production, 7 interventions
 - Intégration, 1 intervention
- Application Soarian, 254 interventions au total dont :
 - Soarian, 191 interventions
 - Demande d'évolution du système d'information, 20 interventions
 - Mise en Production, 32 interventions
 - Intégration, 11 interventions
- Application Axya, 525 interventions au total dont :
 - Axya, 346 interventions
 - Demande d'évolution du système d'information, 57 interventions
 - Mise en Production, 102 interventions
 - Intégration, 20 interventions

8.2.3 Modèles conceptuels de données

Modèle générique

Afin de mener à bien l'analyse et la classification des traces selon la méthodologie d'étude de l'évolution (Chapitre 6), nous sommes parti du modèle conceptuel générique présenté dans la Section 6.6 et repris dans la Figure 26 afin d'en déduire un modèle qui reprenne la structure des données de cette étude de cas.

En effet, plutôt que de conserver la structure de données du modèle de base, nous avons préféré l'adapter aux spécificités des données d'évolution collectées dans le but de conserver une partie de leur structure en vue des analyses et des regroupements de cette étude de cas.

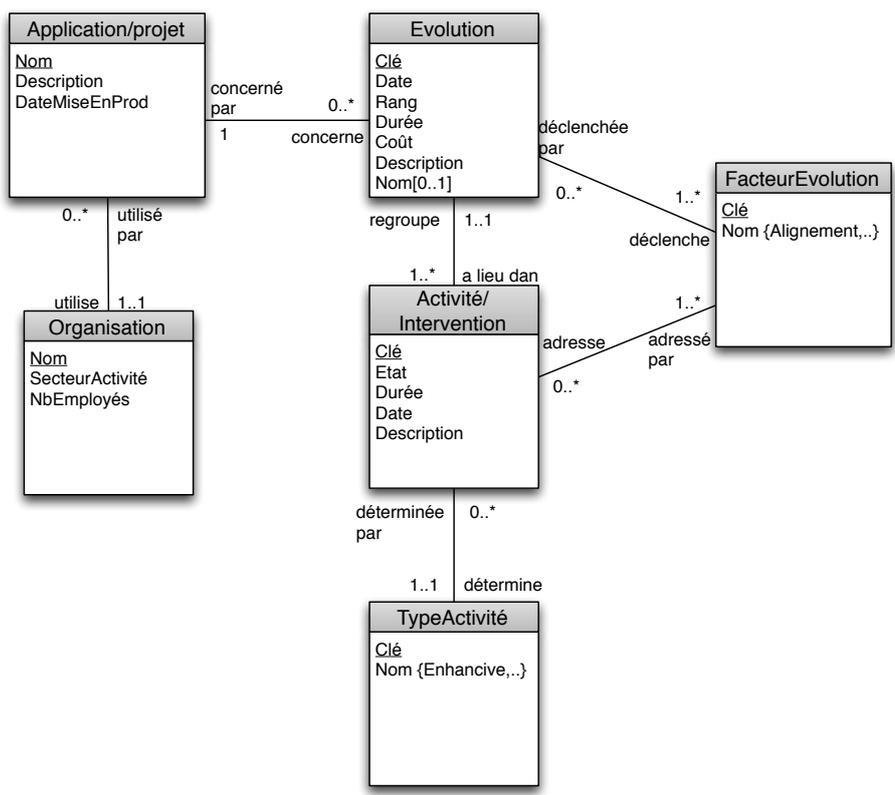


Figure 26. Modèle conceptuel supportant EVOLIS.

Modèle adapté aux données

En se basant sur le modèle conceptuel générique précédent (Figure 26), nous avons créé un nouveau modèle conceptuel pour représenter les données d'évolution que nous avons extraites de la plateforme JIRA. Ce modèle est présenté dans la Figure 27.

Parmi les données extraites, comme il n'est plus possible de faire de regroupements d'activités qui composent une évolution (cf. étude de cas précédente), toutes les activités sont considérées comme des évolutions. Elles sont représentées par la classe Intervention. Ces interventions concernent une application ou un projet étudié. Cette information est représentée par la classe ObjetEtude qui représente les objets que nous avons étudiés tels que décrits dans la Section 8.2.1.

Les traces d'évolution sont organisées dans la plateforme JIRA de la manière suivante : des traces concernant les objets d'étude regroupés dans des projets JIRA spécifiques et des traces d'évolution regroupées en fonction de leur nature dans 3 projets JIRA transversaux (Demande d'Evolution du SI, Intégration et Mise en Production). Toutes ces traces d'évolution étant des interventions effectuées sur les objets d'étude, elles sont regroupées dans la classe Intervention. Les informations spécifiques aux 3 projets JIRA transversaux sont quant à elles regroupées dans les sous-classes (Demande d'Evolution du SI, Intégration et Mise en Production) respectives. De cette manière nous ne perdons aucune information relative à la nature et à la provenance de chaque trace d'évolution.

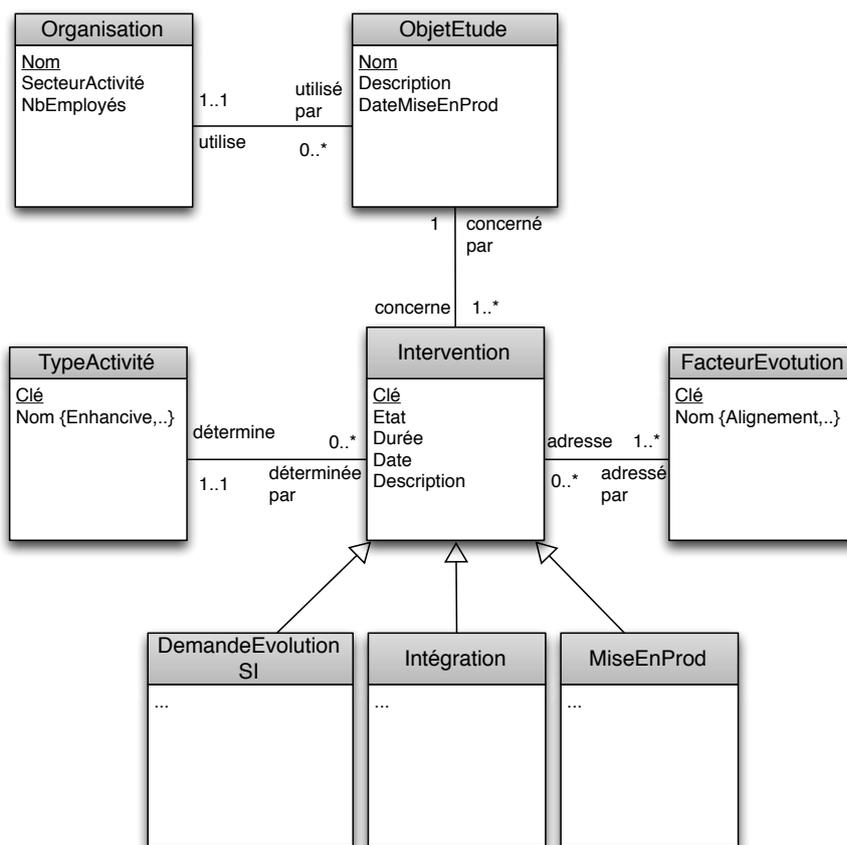


Figure 27. Modèle conceptuel de données adapté aux traces d'évolution étudiées.

Sans ce modèle et les outils de bases de données, l'identification des interventions relatives aux objets d'étude parmi les projets transversaux aurait été extrêmement difficile. Avec ce modèle et la base de données, il a été possible de créer des requêtes effectuant un premier tri parmi les toutes les interventions des projets transversaux. Le tri final a été effectué manuellement lors de l'application d'EVOLIS.

La classification de chacune des interventions selon la méthodologie d'EVOLIS a été effectuée en utilisant un tableur. En effet, l'analyse manuelle des interventions ainsi que leur classification s'effectuent avec une plus grande rapidité d'exécution. Les données ainsi classifiées ont été importées dans la base de données par la suite.

Le modèle conceptuel de données que nous avons implémenté n'est pas totalement similaire au modèle présenté dans la Figure 27. Le schéma de la base de données utilisée est présenté en Annexe B.3. Afin de faciliter l'étude, nous avons effectué nos analyses sur une « vue » de cette base. En effet, les colonnes des classes TypeActivite, FacteurEVOLIS, ObjetEtude ont été jointes avec la table Intervention afin de simplifier l'analyse, le traitement et l'import des données après classification.

Type de modèle OLTP ou OLAP

Au moment de créer le modèle de données sur lequel cette étude allait s'appuyer, la question s'est posée sur le choix du type de modélisation. Allons-nous plutôt nous baser sur un modèle de type transactionnel (OLTP) ou sur un modèle de type analytique (OLAP) ?

Si on se base sur l'utilisation que nous allons faire de ce modèle de données qui est naturellement orientée analytique plutôt que transactionnelle, un modèle analytique semble

plus approprié. D'autant plus qu'aucune transaction (INSERT, UPDATE, DELETE) ne sera effectuée après l'import des données.

En effet, les traces d'évolution étudiées à savoir les interventions pourraient être représentées comme les faits du modèle et dimensionnées avec, par exemple, une dimension temps pour l'aspect longitudinal de l'étude, une dimension pour les facteurs EVOLIS, une dimension pour le typage des activités et ainsi de suite.

D'un autre côté, nous avons plusieurs avantages à ne pas choisir un modèle purement OLAP. Premièrement, si on se base sur les objectifs de cette recherche, la création d'un modèle qui supporte EVOLIS plutôt qu'un modèle qui supporte les données d'évolution à disposition revêt plus de sens dans le processus de recherche suivi pour l'élaboration d'EVOLIS. Deuxièmement, un modèle conceptuel supportant EVOLIS peut être réutilisé pour d'autres études. Troisièmement, un modèle conceptuel est une contribution supplémentaire de notre recherche concernant l'étude des évolutions.

Nous avons donc choisi de réaliser un modèle conceptuel ayant une orientation transactionnelle plutôt qu'analytique. Conceptuellement, l'approche transactionnelle permet de retrouver les concepts d'EVOLIS comme des entités associées entre elles plutôt que comme des éléments dissimulés dans une table de faits.

Dans cette étude de cas, le modèle conceptuel adapté de la Figure 27 propose une structure en quasi étoile suggérant un modèle OLAP d'analyse de données classique. Toutefois, les objets de la classe centrale Intervention contiennent beaucoup plus d'attributs qui seraient représentés sous la forme de dimensions dans un schéma OLAP normal. Néanmoins cette modélisation n'apporterait ici aucun élément d'étude supplémentaire sur l'ensemble des interventions considérées.

A l'usage, lors de nos analyses nous n'avons pas rencontré le besoin d'un modèle de type analytique (OLAP). L'outil tableaux croisés dynamiques de Microsoft Excel possède suffisamment de fonctionnalités pour effectuer les regroupements nécessaires à notre étude. De plus, la vue sur le modèle de donnée que nous utilisons effectue un regroupement préliminaire des informations étudiées.

Malgré tout, si une modélisation analytique devrait être nécessaire dans le futur, elle semble être réalisable sans trop de difficultés. En effet, les interventions sont les objets de l'analyse et les éléments d'EVOLIS et d'autres attributs des interventions peuvent être transformés en dimensions.

8.3 Résultats

Après avoir analysé systématiquement toutes les interventions identifiées, nous sommes en mesure de retracer l'évolution de chaque objet d'étude et de mettre en évidence dans quel facteur d'évolution les ressources ont été investies (satisfaction utilisateurs, alignement métier et système d'information, etc.) ainsi que le type des interventions effectuées. De plus, grâce à l'organisation chronologique des évolutions, nous pouvons déterminer à quel moment des cycles de vie ces ressources ont été investies.

Le coût des interventions, que ce soit en jours/homme ou en valeur monétaire n'est pas présent dans nos sources de données. Nous ne pouvons donc pas déterminer avec exactitude le montant des ressources investies pour chaque intervention. Toutefois, afin de déterminer le niveau d'activité pour chacun des objets étudiés, nous nous sommes basés sur la fréquence des

interventions par rapport au temps. Cette technique nous permet de faire ressortir les pics d'activité ainsi que les périodes plus calmes du cycle de vie de chaque objet d'étude.

Contrairement à la première étude de cas qui se concentrait principalement sur les facteurs d'évolution d'EVOLIS, cette seconde étude inclue également la typologie des activités d'évolution (voir Section 6.4) dans l'étude. Cette typologie permet d'en savoir davantage sur l'orientation des changements effectués lors des interventions.

En plus de l'étude des évolutions de ces systèmes, nous avons pu observer en parallèle l'activité « financière » et « opérationnelle » au long du cycle de vie des différents objets d'étude. Comme les données financières ne concernent que les investissements externes sur les objets d'étude, il n'est pas possible d'associer ces investissements aux interventions effectuées par l'équipe « Application » qui elle est interne à la structure de l'organisation. Toutefois, les investissements permettent de connaître les périodes des cycles de vie de chaque application qui nécessitent le plus de ressources financières ainsi que la répartition de ces ressources en fonction des différents types d'investissements.

Cette étude de cas nous a permis non seulement d'appliquer EVOLIS sur les traces d'évolution d'un système d'information de grande ampleur, mais également d'améliorer le contenu d'EVOLIS notamment en affinant le contenu du facteur déclencheur d'évolution Architecture et en proposant un modèle conceptuel supportant EVOLIS.

Les sections suivantes présentent les résultats de l'application d'EVOLIS aux objets d'étude à savoir le projet Dophin dans la Section 8.3.1, l'application Soarian dans la Section 8.3.2, et l'application Axya dans la Section 8.3.3.

8.3.1 Projet d'informatisation Dophin

Après analyse manuelle du contenu des interventions qui touchent le projet d'informatisation du dossier patient Dophin, au total 87 interventions ont été sélectionnées. Parmi ces 87 interventions 25 ont été annulées ou abandonnées et n'entrent donc pas dans l'étude des traces d'évolution. Les 87 interventions ont été extraites de l'application JIRA selon la répartition suivante pour les projets :

- Dophin, 51 interventions entre mars 2009 et décembre 2009 dont 43 réalisées
- Demande d'évolution du système d'information, 28 interventions dont 11 réalisées
- Mise en Production, 7 interventions dont 7 réalisées
- Intégration, 1 intervention réalisée

Au travers de ces interventions et des dates des différents événements clés, nous avons retracé la vie du projet. Le cycle de vie de ce projet peut s'observer selon différents points de vue : le point de vue financier qui permet de confirmer les événements de la vie du projet et le point de vue évolution qui permet de mieux comprendre les étapes par lesquelles ce projet est passé.

Nous avons choisi d'afficher les cycles de vie en se basant sur les trimestres. Cette unité est le meilleur compromis entre l'échelle des mois et des années afin de représenter le cycle de vie en conservant suffisamment de précision sans entraver la lisibilité des différentes figures.

Dates importantes

Dans les traces d'évolution, quelques dates importantes permettent de retracer la vie du projet d'informatisation Dophin (Figure 28). Pour retracer la vie du projet, nous avons inclus toutes les interventions peu importe qu'elles soient réalisées ou non. La raison est la suivante : les interventions non réalisées montrent qu'il y a eu de l'activité autour de ce projet. Nous avons pris en compte ces interventions afin de déterminer les dates qui illustrent la vie du projet. Ces

interventions ne sont pas prises en compte pour l'étude des traces d'évolution qui elles ne contiennent que les interventions réalisées ou en cours de réalisation lors de la collecte des données.

Les premières traces concernant le projet Dophin dans le système JIRA débutent en janvier 2007. Jusqu'au mois de mars 2009 toutes les interventions qui mentionnent le projet Dophin sont regroupées dans le projet transversal « Demande d'Evolution du Système d'Information ». La première intervention saisie dans le projet JIRA intitulé Dophin est effectuée le 20 mars 2009, et la dernière intervention saisie dans ce projet est effectuée le 22 décembre 2009.

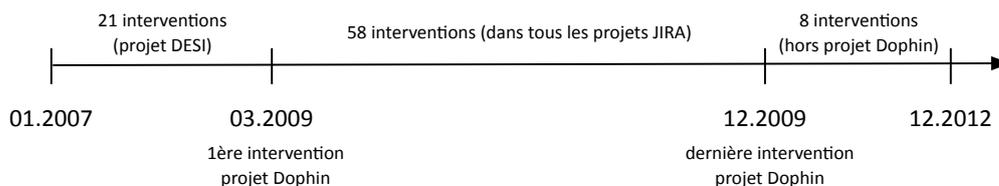


Figure 28. Dates importantes concernant le projet d'informatisation Dophin.

Il est intéressant de relever que les traces confirment les multiples sources externes (presse et autres cf. Section 8.2.1) mentionnant les activités relatives au projet Dophin. En effet, le projet Dophin a débuté en 2008 (les quelques interventions antérieures sont principalement préparatoires pour le début du projet). Dès 2009, la préparation et le déploiement de la solution retenue pour le projet Dophin se fait dans l'organisation. La mise en service de la solution (l'application Soarian) s'est faite en 2011, le projet d'informatisation Dophin n'est plus concerné par de nouvelles interventions. Les interventions postérieures à la mise en service concernent des éléments relatifs au projet Dophin, notamment un module informatique nommé DOPHIN qui continue son activité.

Données financières

Financièrement, il n'y a aucune trace du projet Dophin avant son lancement en 2009. Les données financière représentées dans la Figure 29 concernent des prestations tierces ainsi que des frais de licences et d'infrastructure.

Ces relevés confirment également les événements de la vie du projet. Les principaux paiements de prestations tierces et d'infrastructure ont lieu durant l'année 2009 et le premier trimestre 2010. Le pic d'investissement du second trimestre de 2010 concerne pour plus de 80% les frais de licences. Les trimestres 2 et 3 de l'année 2011 contiennent également des frais de licences, à faible hauteur pour le second trimestre (environ 10 %) et environ 50% pour le troisième trimestre 2011.

Nous n'avons pas pu appliquer EVOLIS aux données financières car ces dernières ne contiennent pas suffisamment de détails concernant les motivations des investissements.

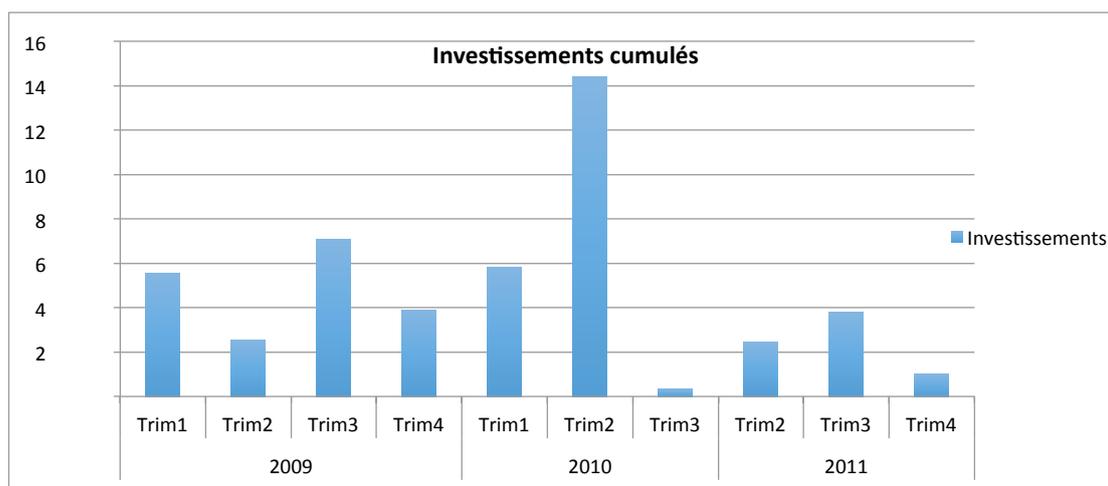


Figure 29. Données financières par trimestre concernant le projet d’informatisation Dophin.

Facteurs déclencheurs d’évolution

Le Tableau 9 présente la répartition des interventions selon les facteurs déclencheurs d’évolution d’EVOLIS. Le pourcentage est établi en fonction du nombre d’interventions concernant chaque facteur dans l’ensemble des interventions relatives au projet Dophin. Pour certaines interventions, l’importance de la motivation les ayant engendrées était difficilement attribuable à un seul des deux facteurs Alignement ou Architecture. Nous avons décidé de les classer comme étant déclenchées par une combinaison de ces deux facteurs.

Tableau 9. Pourcentage d’interventions déclenchées par les facteurs d’évolution d’EVOLIS.

Facteurs déclencheurs	Pourcentage d’interventions
Alignement métier et SI	27%
Alignement et Architecture	10%
Architecture	58%
Technologie	5%

Les interventions effectuées sont principalement déclenchées par des besoins architecturaux (plus de 58% en tenant compte des interventions « Alignement et Architecture »). Ce projet Dophin qui, rappelons-le, est le projet d’informatisation du dossier patient comporte donc une majorité d’enjeux liés à l’architecture du système d’information de l’organisation. En effet, le dossier patient informatisé se doit d’intégrer les informations produites par une multitude d’autres applications hospitalières spécifiques. De plus, ce projet d’informatisation se veut au centre du partage de données entre les différents établissements hospitalier qu’un patient pourrait être amené à côtoyer.

La Figure 30 présente la répartition des intervention durant la vie du projet Dophin. Nous voyons clairement que le pic d’activité en relation avec Dophin à lieu durant l’année 2009.

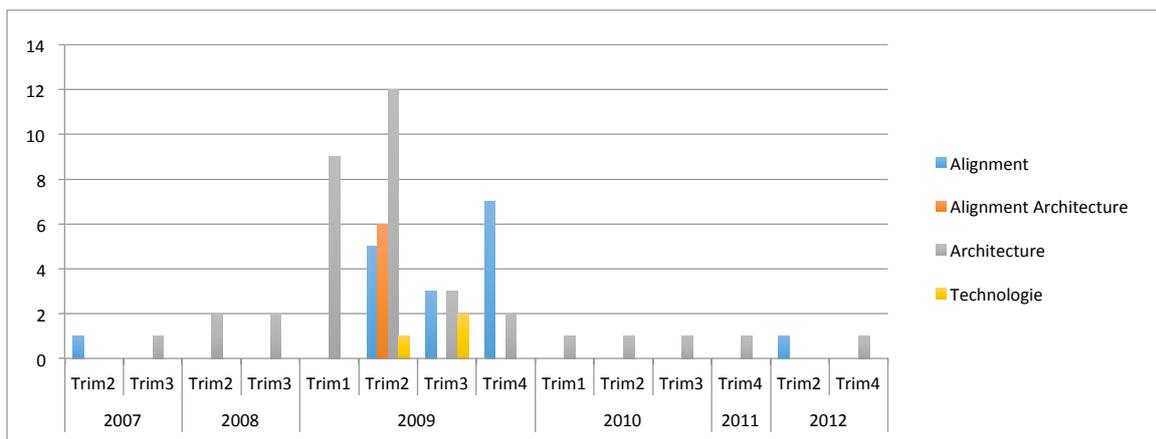


Figure 30. Répartition par trimestre de 2007 à 2012 des interventions sur le projet Dophin en fonction des facteurs d'évolution d'EVOLIS.

Entre 2007 et 2008 : sur les 21 interventions, 6 n'ont pas été annulées ou abandonnées. Parmi ces 21 interventions, 15 (71%) sont issues d'un besoin architectural dont 5 sont réalisées et 6 (29%) d'un besoin d'alignement dont une seule est réalisée. Il n'y a pas de grosse modification de ratio vis-à-vis des interventions réalisées : 83% Architecture et 17% Alignement.

Durant l'année 2009, année principale d'activité du projet, 50 interventions ont été réalisées, parmi ces interventions, la répartition est la suivante : 52% Architecture, 30% Alignement, 6% Technologie et 12% avec des motivations mixtes Alignement et Architecture. De nouveau l'aspect Architecture motive plus de la moitié des interventions.

Les années suivantes les interventions sont d'avantage déclenchées par des motivations d'architecture. Ces interventions portent principalement sur le maintien et l'amélioration d'éléments mis en place durant le projet Dophin (notamment le module informatique nommé DOPHIN).

Jusqu'ici, l'application d'EVOLIS a permis de retracer le cycle de vie du projet Dophin. La période la plus active de ce projet fut l'année 2009 et globalement ce projet provoque des évolutions majoritairement architecturales dans le système d'information de l'organisation.

Des détails concernant la classification des interventions relatives à l'Architecture tel que présentée dans la Section 5.3.3 sont donnés dans la conclusion de cette section. Ils renforcent les résultats obtenus après application de la typologie des activités d'évolution.

Type d'activité des interventions

Le Tableau 10 présente la répartition des interventions selon leur type d'activité. Le pourcentage est établi en fonction du nombre d'intervention concernant chaque type d'activité dans l'ensemble des interventions relatives au projet Dophin. Parmi les groupes d'activités (détails disponibles en Section 6.4.1), 41% des interventions sont du groupe « Support ». Les activités de ce groupe ont pour objectif l'étude du logiciel avec la consultation d'information sur le logiciel et l'évaluation de ce dernier. L'autre groupe d'activités fortement représenté est le groupe des « Règles métier », il concerne 39% des interventions. Ce groupe rassemble les activités ayant un impact important sur les processus métiers et modéré à important sur le logiciel. Parmi ces interventions, 21% remplacent, ajoutent ou étendent les fonctionnalités utilisateurs et 20% corrigent les fonctionnalités. Parmi les interventions restantes, la moitié (10%) concernent les mises à jour et la correction de la documentation et 10% concernent des modification logicielles afin d'adapter les logiciels ou de modifier leurs performances.

Tableau 10. Pourcentage d'interventions pour chaque type d'activité d'EVOLIS pour le projet Dolphin.

Groupe d'activités	Type d'activité	Pourcentage d'interventions
Support	Consultive	31%
	Evaluative	10%
Documentation	Reformative	5%
	Updative	5%
Règles métier	Enhancive	20%
	Corrective	19%
Propriétés logicielles	Performance	2%
	Adaptive	8%

En observant la Figure 31 qui présente la répartition des types d'activités dans le temps, il en ressort que les interventions de type « Enhancive » s'effectuent à un niveau relativement stable tout au long de la vie du projet. A l'inverse, les interventions de type « Consultive » et « Corrective » s'effectuent de manière plus regroupée durant l'année 2009 et particulièrement durant le second trimestre pour les interventions de type « Consultive ». Si on se concentre sur cette année 2009, 48% des interventions sont du groupe « Support » et concernent donc l'étude et l'évaluation logicielle. Ces relevés correspondent donc à l'objectif du projet Dolphin et confirment l'aspect préparatoire de l'informatisation du dossier patient.

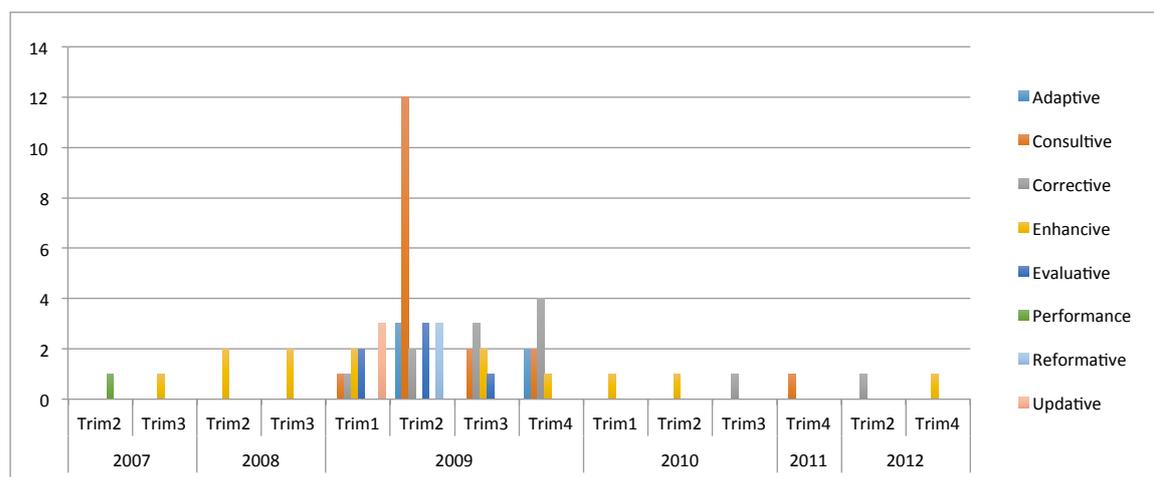


Figure 31. Répartition par trimestre de 2007 à 2012 des interventions par type d'activité sur la vie du projet Dolphin.

Combinaison entre facteurs EVOLIS et types d'activités

Afin d'en apprendre plus sur les interventions, la combinaison des facteurs déclencheurs d'évolution et des types d'activités renseigne de manière précise sur la nature de ce qui a

réellement été entrepris durant les interventions. De cette combinaison, il devient possible de retracer le cycle de vie en le complétant avec des renseignements comme par exemple :

- Si les interventions de la première partie du projet sont issues du groupe « Support » et motivées par l'Alignement et l'Architecture, nous pouvons en déduire que cette période a été consacrée à l'étude des processus et à l'évaluation des possibilités d'intégration.
- Nous pouvons déduire que suite à la mise en service de la solution, une période de correction de cette dernière et d'alignement avec le métier débute, dans le cas où les interventions de cette période sont motivées par l'Alignement et l'Architecture mais qu'elles sont, cette fois-ci, issues du groupe « Règles métiers ».
- Et ainsi de suite afin de retracer la vie du système.

L'année la plus active concernant le projet Dolphin est l'année 2009. Pour mieux comprendre ce qui s'est passé dans la vie du projet pendant cette période, le Tableau 11 et graphiquement la Figure 32 présentent la combinaison entre chaque facteur EVOLIS et les types d'activités qui y sont associés. Cette combinaison fait ressortir le fait que ce sont les facteurs Architecture et Alignement qui déclenchent les 48% d'interventions du groupe « Support ». Les interventions concernent donc l'étude et l'évaluation logicielle pour ces deux facteurs.

Ces éléments conduisent au résultat suivant : les interventions entreprises lors du projet Dolphin sont principalement orientées à la préparation de l'informatisation du dossier patient. En effet, l'étude et l'évaluation des possibilités architecturales ainsi que des aspects d'alignement confortent ce résultat. De plus, certaines corrections d'alignement sont effectuées en rapport avec la future mise en production de l'application Soarian choisie pour supporter l'informatisation du dossier patient.

Tableau 11. Répartition des interventions par facteur d'évolution et par type d'activité pour l'année 2009.

Type d'activité par facteur	Nombre
Alignement	15
Adaptive	1
Consultive	5
Corrective	7
Enhancive	1
Reformative	1
Alignement Architecture	6
Consultive	6
Architecture	26
Adaptive	3
Consultive	5
Corrective	3
Enhancive	4
Evaluative	6
Reformative	2
Updative	3
Technologie	3
Adaptive	1
Consultive	2
Total	50

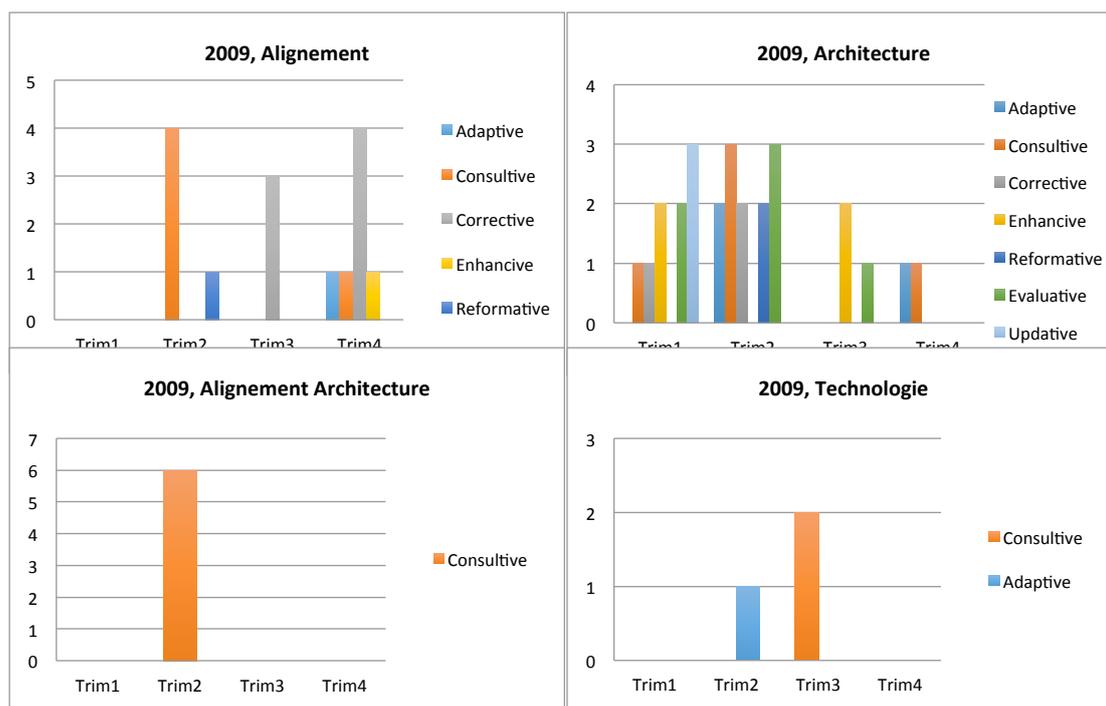


Figure 32. Répartition des interventions du projet Dolphin par facteur d'évolution et par type d'activité par trimestre pour l'année 2009.

Conclusion de l'étude des traces d'évolution du projet Dolphin

L'étude des traces d'évolution selon EVOLIS permet de retracer le cycle de vie du projet d'informatisation du dossier patient. Ces traces nous montrent que la grande partie des interventions en lien avec ce projet a été effectuée durant l'année 2009, avant la mise en production de la solution logicielle en 2011.

L'étude de ces traces d'évolution nous montre que le projet d'informatisation Dolphin génère des activités bien avant l'ouverture du projet JIRA destiné à Dolphin en 2009 et qu'il engendre des interventions après la dernière saisie dans le projet JIRA dédié (fin 2009). Les interventions post 2009 sont éparpillées dans le temps et ont, pour la plupart, une motivation liée à l'architecture du système d'information.

La classification de chaque intervention selon sa motivation et selon la typologie des activités d'évolution a permis de déterminer que l'enjeu principal de ce projet concerne non seulement l'architecture du système d'information mais également que cet enjeu concerne des activités de type étude et évaluation au niveau architecture et alignement. En effet, l'analyse selon EVOLIS des interventions effectuées lors de la période de forte activité (2009) conduit au résultat suivant : les interventions entreprises lors du projet Dolphin sont principalement orientées à la préparation de l'informatisation du dossier patient et donc à la préparation de la mise en service de l'application Soarian au sein du système de l'organisation. Le dossier patient étant un élément central du système hospitalier, son informatisation engendre de multiples interconnexions entre les différents composants du système d'information qu'il faut étudier et évaluer.

Ce résultat est également confirmé par la classification des interventions relatives à l'Architecture tel que présenté dans la Section 5.3.3. Cette classification est effectuée en catégorisant les motivations des interventions relatives au facteur Architecture. Elle met lumière les éléments suivant : sur les 42 interventions motivées par le facteur Architecture, 32 ont été effectuées en 2009. Sur ces 32 interventions, 63% sont motivées par l'étude des différentes

possibilités de couplage des composants du système. Le Tableau 12 présente la répartition des interventions motivées par l'Architecture selon les catégories couplage des composants du système.

Tableau 12. Catégories de motivations architecturales concernant le couplage des composants pour l'année 2009.

Catégories de motivations sur le couplage des composants	Pourcentage
1. Etude des possibilités de couplage	63%
2. Le couplage motive l'intervention	28%
Nouveau couplage de composants	3%
Modification d'un couplage existant	
Augmentation de la portée d'un couplage	3%
Correction d'un couplage	16%
Modification indéterminée	6%
3. Modification d'un composant couplé	3%
Indéterminé	6%

Cette classification renforce le résultat trouvé selon la typologie des activités et enrichit cette dernière en précisant les différentes interventions relatives au couplage des composants.

Les données financière confirment également que l'année 2009 était la période la plus active pour ce projet. Cependant, si l'année 2009 était la plus active au niveau des interventions, une activité financière plus soutenue que l'activité d'évolution persiste en 2010 et 2011 confirmant que le projet continue sa vie malgré le faible nombre d'interventions. Cette activité financière met en lumière l'importance des frais de licence incombant à ce projet ainsi que les travaux effectués par les prestataires externes.

8.3.2 Application Soarian

L'application Soarian est l'outil hospitalier dédié à la mise en place du dossier patient informatisé en vue d'assurer le partage des données médicales permettant d'améliorer la collaboration entre les différentes institutions participant au projet Dophin. La section précédente nous a permis d'établir que le projet d'informatisation Dophin avait eu comme principal objectif la préparation de l'informatisation du dossier patient et donc la préparation de la mise en service de l'application Soarian au sein du système de l'organisation. L'étude des traces d'évolution concernant cette application peut être considérée comme la suite de l'étude des traces du projet Dophin.

L'analyse des traces d'évolution concernant cette application comporte au total 254 interventions dont 247 n'ont pas été abandonnées ou annulées. Les interventions ont été extraites de l'application JIRA selon la répartition suivante pour les projets :

- Soarian, 191 interventions entre juin 2011 et février 2013 dont 190 réalisées
- Demande d'évolution du système d'information, 20 interventions dont 16 réalisées
- Mise en Production, 32 interventions dont 31 réalisées
- Intégration, 11 interventions dont 10 réalisées

Au travers de ces interventions et des dates des différents événements clés, nous avons retracé la vie de l'application. Le cycle de vie de l'application a été représenté du point de vue évolution en caractérisant les interventions selon EVOLIS. Cette caractérisation apporte une meilleure compréhension des étapes par lesquelles cette application est passée.

Comme pour le projet Dolphin, nous avons choisi d'afficher les cycles de vie en se basant sur les trimestres. Cette unité est le meilleur compromis entre l'échelle des mois et des années afin de représenter le cycle de vie en conservant suffisamment de précision sans en entraver la lisibilité.

Dates importantes

Dans les traces d'évolution, quelques dates permettent de retracer la vie de l'application Soarian (Figure 33). Pour retracer la vie de cette application, nous avons, comme pour le projet Dolphin, inclus toutes les interventions peu importe qu'elles soient réalisées ou non. La raison est la suivante : les interventions non réalisées nous montrent qu'il y a eu de l'activité autour de l'application. Nous avons donc tenu compte de ces interventions afin de déterminer les dates qui illustrent la vie de l'application. Toutefois, elles n'ont pas été prises en compte pour l'étude de l'évolution qui ne contient que les interventions réalisées ou en cours de réalisation lors de la collecte des données.

Les premières traces concernant l'application Soarian dans le système JIRA débutent en janvier 2010. Jusqu'au mois de juin 2011 toutes les interventions qui mentionnent Soarian sont regroupées dans les projets transversaux « Demande d'évolution du système d'information », « Mise en production » et « Intégration ». La première intervention saisie dans le projet JIRA intitulé Soarian est effectuée le 29 juin 2011, le système JIRA continuait d'enregistrer des interventions lors de la collecte des données en février 2013.

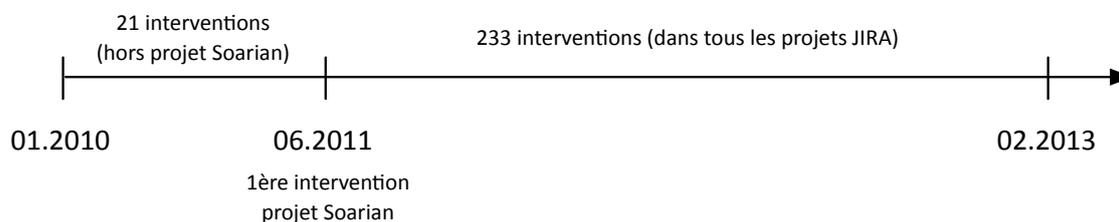


Figure 33. Dates importantes concernant l'application Soarian.

Les dates présentées dans la Figure 33 coïncident avec les dates mentionnées dans les sources externes (presse et autres cf. Section 8.2.1). L'application Soarian a été mise en service au milieu de l'année 2011, date à laquelle la première intervention a été saisie dans la plateforme JIRA sous le projet dédié à Soarian. Avant la mise en service de l'application, les interventions effectuées concernent pour la plupart le déploiement de Soarian dans l'organisation ainsi que le couplage de cette application avec les autres composants du système d'information. Ces éléments sont détaillés ci-dessous dans l'étude des facteurs d'évolution.

Après la mise en service de l'application, les traces d'évolution étudiées contiennent des interventions effectuées à la fois dans le projet JIRA dédié à Soarian et dans les trois autres projets JIRA transversaux.

Données financières

Concernant l'application Soarian, les seules données financières spécifiques à l'application concernent uniquement des factures émises pour des prestations externes (visibles dans la Figure 34). Durant le second trimestre 2011 (avant la mise en service), le fournisseur d'un autre

composant du système d'information (système de gestion de laboratoire) a réalisé une interface de communication permettant le couplage entre ce composant et Soarian dans le but d'établir des prescriptions connectées. Fin 2011, début 2012, des factures ont été payées à un autre prestataire externe qui a développé des formulaires et des documents cliniques dans le cadre du déploiement de Soarian.

Toutefois, nous supposons qu'une grande partie des frais externes liés à l'utilisation de Soarian dans l'organisation sont comptabilisés dans le projet Dophin, notamment les frais de licences. En effet, l'application Soarian (développée par Siemens) est l'application permettant d'informatiser le dossier patient entre les différentes institutions prenant part au projet Dophin. Hors, les données financières liées à Dophin contiennent les paiements relatifs à des licences auprès de la société Siemens.

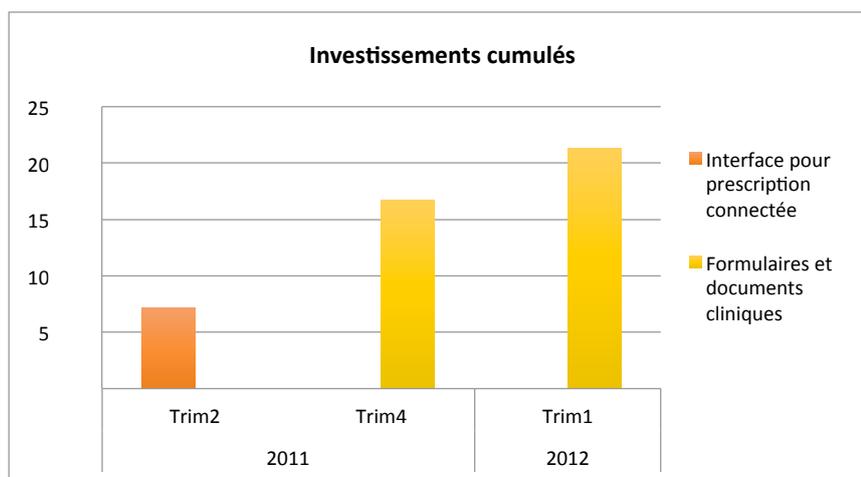


Figure 34. Données financières par trimestre concernant l'application Soarian.

Facteurs déclencheurs d'évolution

Le Tableau 13 présente la répartition des interventions selon les facteurs déclencheurs d'évolution d'EVOLIS. Le pourcentage est établi en fonction du nombre d'interventions réalisées concernant chaque facteur dans l'ensemble des interventions relatives à Soarian.

Tableau 13. Pourcentage d'interventions déclenchées par les facteurs d'évolution d'EVOLIS.

Facteurs déclencheurs	Pourcentage d'interventions
Alignement	77%
Architecture	16%
Technologie	4%
Satisfaction utilisateurs	3%

En comparant ces pourcentages avec ceux du projet Dophin, le principal facteur faisant évoluer l'application Soarian est l'Alignement entre métier et système d'information. Durant le projet Dophin qui créait les conditions favorables pour le déploiement de Soarian dans le système d'information, les motivations des interventions étaient majoritairement orientées Architecture. Maintenant que l'application est en service, une phase du cycle de vie contenant des évolutions motivées par le facteur d'Alignement débute.

La Figure 35 présente la répartition dans le temps des interventions selon leurs motivations. Avant la mise en production de l'application à la fin du second trimestre 2011, les interventions relatives à Soarian sur le système d'information sont motivées principalement par l'architecture de ce dernier. Cette répartition des interventions est similaire avec celle du projet Dophin lors de la même période. Dès la mise en production de Soarian, les interventions motivées par l'alignement du système font leur apparition. Le trimestre qui suit la mise en service rencontre un pic d'interventions motivées par l'alignement puis, les trimestres suivants, le nombre d'interventions relatives à l'Alignement baisse pour atteindre un niveau qui semble se stabiliser, du moins jusqu'à la fin de la collecte de données.

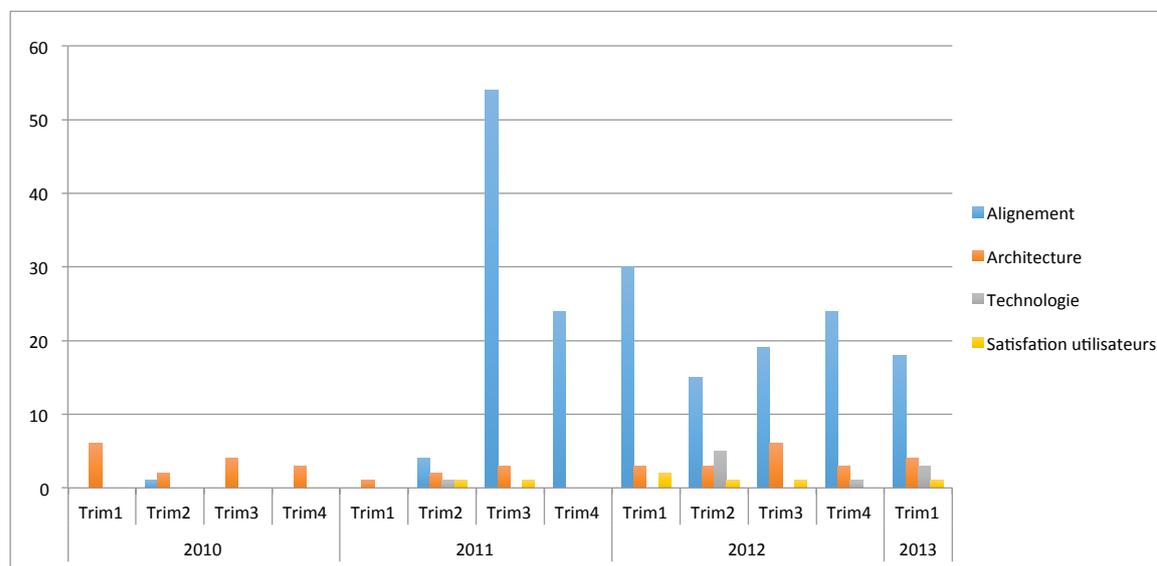


Figure 35. Répartition par trimestre de 2010 à 2013 des interventions sur l'application Soarian en fonction des facteurs d'évolution d'EVOLIS

Les évolutions effectuées sur l'application Soarian sont donc principalement motivées par l'alignement de cette application sur les besoins métiers. Afin de comprendre en quoi consistent ses interventions d'alignement, il est nécessaire d'observer les types d'activités qui les composent.

Type d'activité des interventions

L'étude des interventions selon leur type d'activité renseigne sur la nature des activités entreprises. Le Tableau 14 présente la répartition des interventions selon leur type d'activité. Contrairement au projet Dophin dans lequel les types d'activités dominants étaient ceux du groupe support (étude et évaluation logicielle), les activités modifiant les règles métier (65%) sont dominantes pour l'application Soarian. Un peu moins de la moitié du total des interventions (47%) étendent les fonctionnalités utilisateurs, tandis que 16% corrigent des fonctionnalités en service. Les interventions qui adaptent les propriétés du logiciel sont également un type d'activité fréquent (30%). Ce type d'activité n'ajoute ni ne retire de fonctionnalités, il rassemble les activités qui ajustent et améliorent ce qui se fait déjà.

Tableau 14. Pourcentage d'interventions pour chaque type d'activité d'EVOLIS pour l'application Soarian.

Groupe d'activités	Type d'activité	Pourcentage d'interventions
Règles métier	Enhancive	47%
	Corrective	16%
	Reductive	2%
Propriétés logicielles	Adaptive	30%
Support	Consultive	3%
	Evaluative	<1%
Documentation	Reformative / Updative	<1%
Mixte	Adaptive Enhancive	<1%

La Figure 36 présente la répartition des types d'activités dans le temps. En observant la période qui suit immédiatement la mise en service, le type d'activité dominant de quasi chaque trimestre (excepté lors du 1^{er} trimestre 2012) est le type Enhancive qui étend les fonctionnalités de l'application. Nous relevons également que les activités de type Adaptive et Corrective augmentent également après la mise en service de Soarian.

L'augmentation des interventions de type Enhancive peut s'expliquer de deux manières : premièrement l'utilisation de Soarian est étendue et adaptée progressivement aux différents services de l'organisation (pour la plupart des interventions) ; deuxièmement les utilisateurs finaux remarquent des améliorations à apporter aux processus couverts par Soarian.

L'augmentation des interventions de type Adaptive et Corrective peut également s'expliquer par le fait qu'à l'usage des modifications de l'application Soarian se sont révélées nécessaires à son bon fonctionnement. Ces modifications ne pouvaient pas être anticipées sans l'expertise des utilisateurs finaux.

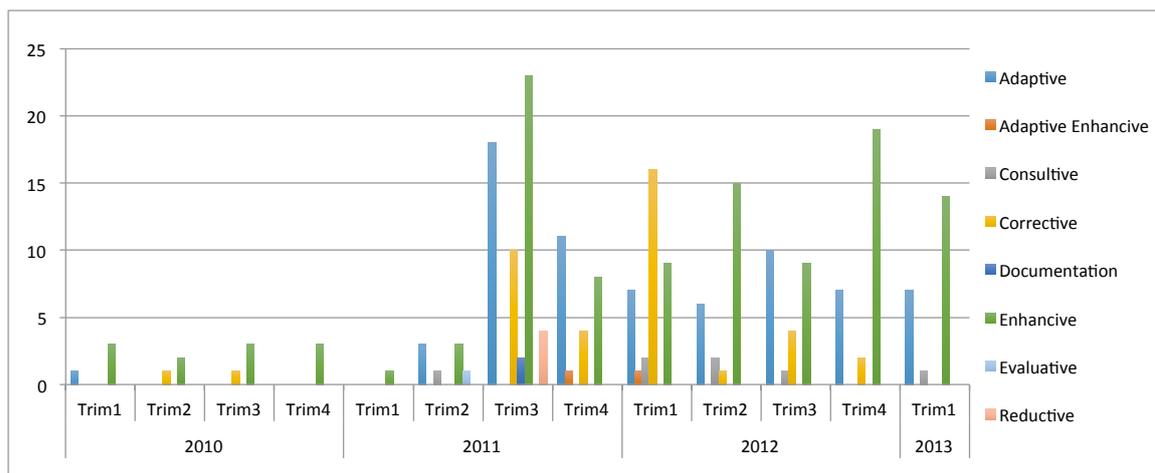


Figure 36. Répartition par trimestre des interventions par type d'activité sur l'application Soarian.

Combinaison entre facteurs EVOLIS et types d'activités

La combinaison des facteurs déclencheurs d'évolution et des types d'activités renseigne de manière précise sur la nature de ce qui a réellement été entrepris durant les interventions.

L'année 2010 ne contient que des interventions motivées par le facteur EVOLIS Architecture à l'exception d'une correction d'alignement. Les figures représentant la combinaison pour les facteurs Technologie et Satisfaction utilisateurs (Figure 37) présentent donc les années 2011 à 2013.

Nous remarquons dans la Figure 37 que peu d'interventions sont déclenchées par les facteurs Technologie et Satisfaction utilisateurs. En observant le facteur Technologie, nous trouvons des interventions de type Enhancive. Ce sont des interventions qui améliorent les capacités techniques de Soarian, elles contiennent des activités telles que des mises à niveau de certains composants. Les interventions de type Adaptive contiennent des migrations sur d'autres serveurs. Concernant le facteur Satisfaction des utilisateurs, ces derniers ont été questionnés sur leurs utilisations de Soarian et leurs attentes notamment sur leurs besoins d'accessibilité à l'application. Une intervention porte également sur une modification de la documentation relative à Soarian.

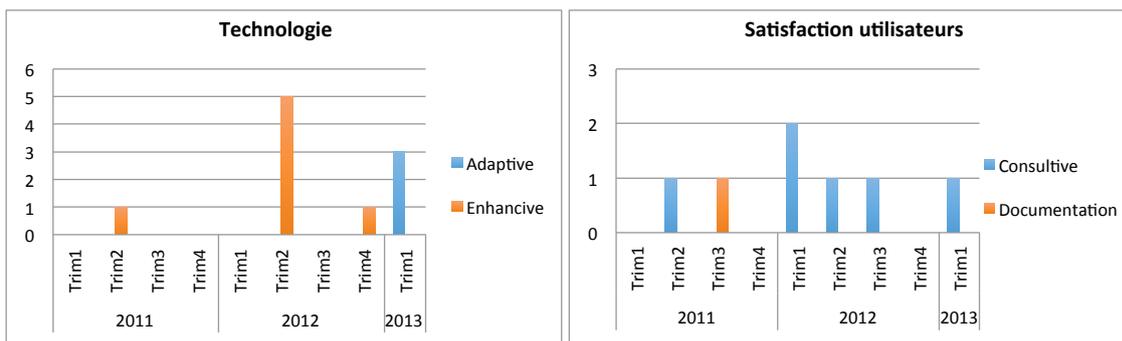


Figure 37. Répartition des interventions par type d'activité pour les facteurs Technologie et Satisfaction des utilisateurs pour l'application Soarian.

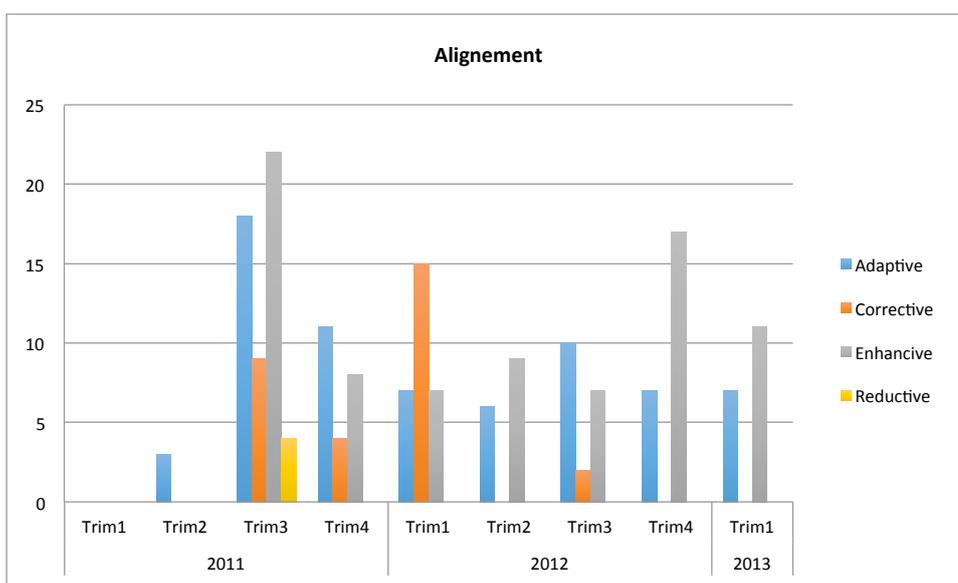


Figure 38. Répartition des interventions par type d'activité pour le facteur Alignement métier et système d'information pour l'application Soarian.

La Figure 38 présente la répartition des interventions par type d'activité pour celle qui sont motivées par l'alignement entre le système d'information et le métier. Pour des raisons de lisibilité, nous n'avons pas affiché les interventions représentant moins d'1% des types d'activités pour le facteur Alignement ainsi que l'année 2010 qui ne contenait qu'une intervention de correction d'alignement.

Dès le 3^{ème} trimestre 2011, tout de suite après la mise en service de Soarian, des interventions motivées par l'Alignement apparaissent. Elles sont de trois principaux types : Enhancive (43%), Adaptive (37%), Corrective (16%). On retrouve également tout de suite après la mise en service des interventions qui réduisent certaines fonctionnalités de l'application. Ces fonctionnalités inappropriées ou inutiles ont été détectées uniquement après la confrontation avec les utilisateurs finaux.

En observant en détail le contenu des interventions d'alignement Enhancive, nous observons qu'une grande partie d'entre elles portent sur la mise en place de Soarian dans des nouveaux services de l'organisation. Pour résumer la situation : Soarian a été mis en production (probablement dans quelques services initiaux) puis de nouveaux services ont été intégrés au dossier patient informatisé dans Soarian. Entre le 3^{ème} trimestre 2011 et le début 2013, 21

services de l'organisation ont intégré le dossier patient informatisé avec Soarian. La Figure 39 présente pour chaque trimestre le nombre de nouveaux services ayant intégré le dossier patient informatisé avec Soarian. Les interventions du type Enhancive fluctuent en effet de manière coordonnée avec l'expansion du dossier patient informatisé dans de nouveaux services de l'organisation. Ces fluctuations ne sont pas parfaitement alignées car la création de certains dossiers patients dans Soarian ont parfois nécessité plus d'interventions sur l'application que d'autres.

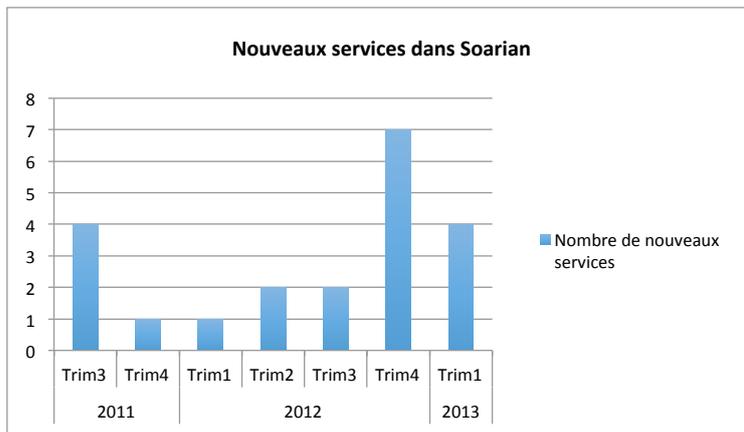


Figure 39. Nombre de nouveaux services par trimestre ayant intégré le dossier patient avec Soarian.

Le second type d'intervention le plus répandu en ce qui concerne les interventions motivées par l'alignement est le type d'activité Adaptive. Après observation de la répartition de ces interventions dans le cycle de vie de Soarian, nous relevons que le niveau des interventions de ce type reste relativement constant lors de chaque trimestre. Toutefois, nous notons une valeur plus élevée lors du trimestre qui a suivi la mise en service de Soarian. Nous expliquons cette valeur élevée par la « jeunesse » de l'application et le besoin d'adaptation de cette dernière avec les attentes des utilisateurs. En effet ce type d'activité rassemble des interventions telles que les adaptations de formulaires, les changements de mises en formes, etc. qui n'étendent pas la couverture ou les fonctionnalités de l'application Soarian.

Le troisième type d'intervention le plus répandu en ce qui concerne les interventions motivées par l'alignement est le type d'activité Corrective. En observant la répartition de ces interventions, nous relevons un pic de 15 interventions lors du 1^{er} trimestre 2012. En se penchant sur ces dernières, sur les 15 interventions, 10 sont effectuées pour corriger le dossier patient d'un nouveau service dont la mise en place a débuté lors du trimestre précédent. Ce pic met donc en lumière une intégration d'un service qui a posé d'avantage de difficultés. De manière générale, les autres interventions de ce type corrigent des défauts, bugs dans l'ensemble des fonctionnalités et prestations offertes par Soarian.

La Figure 40 présente la répartition des interventions par type d'activité pour celle qui sont motivées par l'architecture du système d'information. Contrairement aux figures précédentes, toutes les années sont représentées ici car le facteur Architecture a déclenché des interventions avant et après la mise en service de Soarian. Nous relevons que le niveau des interventions liées à l'architecture reste relativement constant lors de chaque trimestre. Toutefois, il est intéressant de relever qu'après la mise en service de Soarian, d'avantage d'interventions de type Corrective sont effectuées. Ce type d'intervention est présent lors de quasiment tous les trimestres alors qu'avant la mise en service une seule intervention de ce type était recensée.

Le type d'intervention le plus représenté pour le facteur Architecture est le type Enhancive. Ce type regroupe des interventions qui vont de la « simple » extension de couplages existants (en

ajoutant des informations) à l'établissement de nouveaux couplages entre composants du système d'information. Afin de déterminer plus précisément la nature de ces interventions de type Enhancive, nous avons appliqué la classification des interventions relatives à l'Architecture tel que présenté dans la Section 5.3.3. Le Tableau 15 présente la répartition des interventions motivées par l'Architecture selon les catégories de couplage des composants du système. Il est intéressant d'observer les variations de répartition entre la période précédant et suivant la mise en service. Avant la mise en service de Soarian, la répartition entre un nouveau couplage de composants et une modification d'un couplage existant est quasi similaire, respectivement 44% et 39% des interventions. Après la mise en service, cette répartition se modifie et passe de 44% à 22% pour les nouveaux couplages de composants et de 39% à 78% pour la modification de couplages existants. Nous relevons donc une tendance qui passe d'une activité répartie quasiment à part égale entre nouveau couplage et modification de l'existant à une activité qui, dans presque $\frac{3}{4}$ des cas, modifie un couplage existant soit en l'étendant soit en le corrigeant.

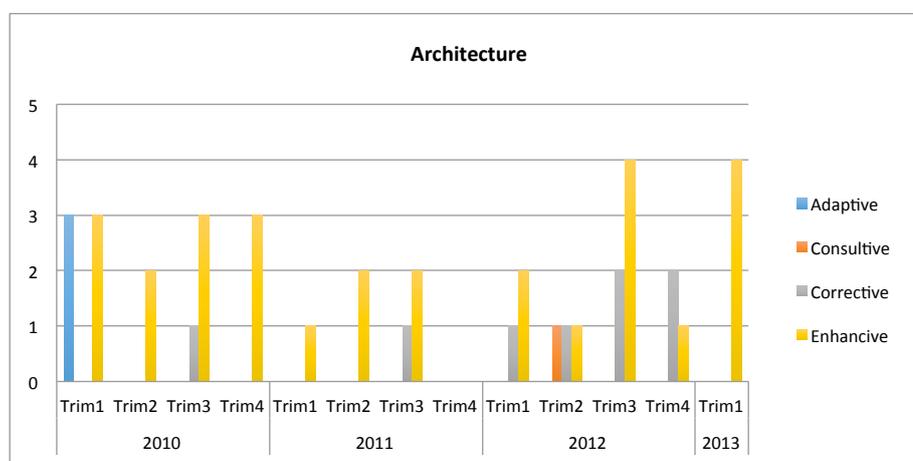


Figure 40. Répartition des interventions par type d'activité pour le facteur Architecture pour l'application Soarian.

Tableau 15. Catégories de motivations architecturales concernant le couplage des composants, répartition avant et après la mise en service de Soarian.

Catégories de motivations sur le couplage des composants	Pourcentage avant la mise en service	Pourcentage après la mise en service
1. Etude des possibilités de couplage		
2. Le couplage motive l'intervention	83%	100%
Nouveau couplage de composants	44%	22%
Développement d'une interface de couplage	22%	8%
Utilisation des possibilités offertes	11%	
Non précisé	11%	14%
Modification d'un couplage existant	39%	78%
Augmentation de la portée d'un couplage	33%	32%
Correction d'un couplage	6%	32%
Non précisé		14%
3. Modification d'un composant couplé	6%	
Indéterminé	11%	

Conclusion de l'étude des traces d'évolution de l'application Soarian

Avec l'étude des traces relatives au projet Dophin (Section 8.3.1), nous avons pu représenter le cycle de vie et comprendre les enjeux de la phase d'étude et d'analyse des besoins pour le déploiement de l'application Soarian. Dans l'étude concernant traces d'évolution pour l'application Soarian, nous avons pu, en quelques sortes, prolonger la vie de ce projet Dophin avec l'étude des interventions qui ont suivi la mise en place de l'application Soarian sélectionnée pour ce projet.

Cette étude selon les facteurs déclencheurs d'évolution d'EVOLIS fait ressortir qu'après la mise en service de l'application Soarian, l'enjeu le plus important était l'alignement de Soarian avec les besoins de l'organisation. En appliquant la typologie des activités sur les interventions, nous avons déterminé que cet alignement s'est fait non seulement en adaptant et en corrigeant l'application mise en place, mais surtout en étendant son utilisation dans les différents services de l'organisation.

Nous avons également remarqué un changement de tendance concernant les interventions liées au facteur Architecture. En effet, les natures de ces interventions ne sont pas similaires entre la période qui précède et celle qui suit la mise en service de l'application Soarian. Dans la période précédant la mise en service, la répartition entre les interventions concernant de nouveaux couplages de composants et celles concernant une modification d'un couplage existant est quasi similaire avec respectivement 44% et 39% des interventions. Dans la période qui suit la mise en production, cette répartition se modifie et passe de 44% à 22% pour les nouveaux couplages de composants et de 39% à 78% pour les interventions concernant la modification de couplages existants. Nous observons donc une transition des efforts vers la modification des couplages entre Soarian et les autres composants du système d'information.

8.3.3 Application Axya

L'application Axya est l'application de gestion administrative des patients. Elle a été mise en service au début de l'année 2002.

Deux moyens ont été utilisés pour retracer la vie de l'application selon les données à notre disposition : les données financières et les interventions effectuées pour faire évoluer Axya.

De 2000 à 2007, nous ne disposons que de données financières concernant des prestations tierces ainsi que des frais de licences. De 2007 à 2012, nous disposons des données financières et des données relatives aux évolutions effectuées sur l'application Axya.

Afin de retracer la vie de l'application Axya entre 2000 et 2007, nous avons appliqué EVOLIS aux données financières. Malheureusement les données financières contiennent des paiements qui regroupent plusieurs prestations et qui ne contiennent que peu de détails sur la nature des activités effectuées sur l'application Axya.

Concernant les traces d'évolution extraites de la plateforme JIRA, la première intervention remonte à l'année 2000. Cependant, entre 2000 et 2007, seulement 8 interventions sont répertoriées dans JIRA. Compte tenu de leur éloignement et de leur faible importance quant à l'ensemble des traces d'évolution, nous avons choisi de ne pas les représenter dans les figures présentant la répartition des interventions dans la vie de l'application.

L'analyse des traces d'évolution concernant cette application comporte au total 525 interventions dont 461 n'ont pas été abandonnées ou annulées. Les interventions ont été extraites de l'application JIRA selon la répartition suivante pour les projets :

- Axya, 346 interventions dont 313 réalisées
- Demande d'évolution du système d'information, 57 interventions dont 37 réalisées

- Mise en Production, 102 interventions dont 92 réalisées
- Intégration, 20 interventions dont 19 réalisées

Comme pour les objets d'étude précédents, nous avons choisi d'afficher les cycles de vie en se basant sur les trimestres. Cette unité est le meilleur compromis entre l'échelle des mois et des années afin de représenter le cycle de vie en conservant suffisamment de précision sans en entraver la lisibilité.

Dates importantes

Nous avons retracé la vie de l'application Axya depuis son acquisition jusqu'en 2012 au travers des données financières. Ce sont donc les grands événements pour lesquels l'organisation a eu recours à des prestataires externes qui sont représentés. Le détail et la chronologie de ces événements sont décrits dans le Tableau 16 et la Figure 41.

Sur la base de ces événements, nous relevons comme pour l'application Soarian, que directement après la mise en service de l'application, une évolution déclenchée par un besoin d'alignement a été nécessaire (Axya – 02). La mise en service a donc contribué à faire ressortir les défauts de « jeunesse » et les besoins d'amélioration de l'application.

Une fois qu'Axya a passé cette première étape d'alignement, la couverture de l'application a été étendue à un autre service de l'organisation : la Polyclinique Médicale Universitaire.

Les évolutions suivantes de la vie d'Axya ont porté sur l'amélioration des performances de l'application avec Axya – 03 et Axya – 04, puis une consolidation a eu lieu.

Une fois de plus, après ces évolutions relatives à des améliorations, la couverture de l'application a été étendue à un autre service de l'organisation : la Polyclinique Médicale Universitaire Dentaire.

Comme en 2003 et 2004, les évolutions suivantes de la vie d'Axya ont porté sur des améliorations. Ces améliorations concernent une augmentation des performances, des évolutions techniques, une amélioration de l'ergonomie et également l'augmentation de la couverture fonctionnelle de l'application.

Les années 2008 et 2009 ont permis de faire à nouveau évoluer la couverture fonctionnelle d'Axya en y intégrant un nouveau composant : AxyaNet.

L'année 2009 a permis d'internaliser dans l'organisation le processus de facturation global.

Les années 2010 à 2012 (fin des données) ont été consacrées à l'introduction de TARMED. TARMED est la structure tarifaire unique pour toute la Suisse pour la médecine ambulatoire. Chaque acte médical est associé à un certain nombre de points qui permettent le calcul du prix de la prestation en fonction de l'emplacement des institutions.

Tableau 16. Description des événements de la vie de l'application Axya.

Figure	Dates	Evénement
1	2000 à 2002	Achat et installation d'Axya
2	Janvier 2002	Mise en service d'Axya
3	mi 2002 – début 2003	Alignements suite à la mise en service : Axya – 02
4	fin 2002 – début 2003	Mise en service d'Axya à la Polyclinique Médicale Universitaire
5	mi 2003 – fin 2003	Amélioration des performances : Axya – 03
6	2004	Amélioration des performances : Axya – 04
7	fin 2004 – début 2005	Finalisation de la consolidation de la chaîne de facturation
8	fin 2005 – 2006	Mise en service d'Axya à la Polyclinique Médicale Universitaire

		Dentaire
9	fin 2006 – 2007	Evolution technique, migration, augmentation des performances, de l’ergonomie et nouveaux services
10	2007 – 2008	Renouvellement de l’outil agenda institutionnel (composant)
11	2007 – 2008	Evolution technique, migration, augmentation des performances, de l’ergonomie
12	2008 – 2009	Mise en place d’AxyaNet
13	2009	Internalisation de la facturation
14	2010 – 2012	Introduction TARMED (structure tarifaire uniformisée Suisse)

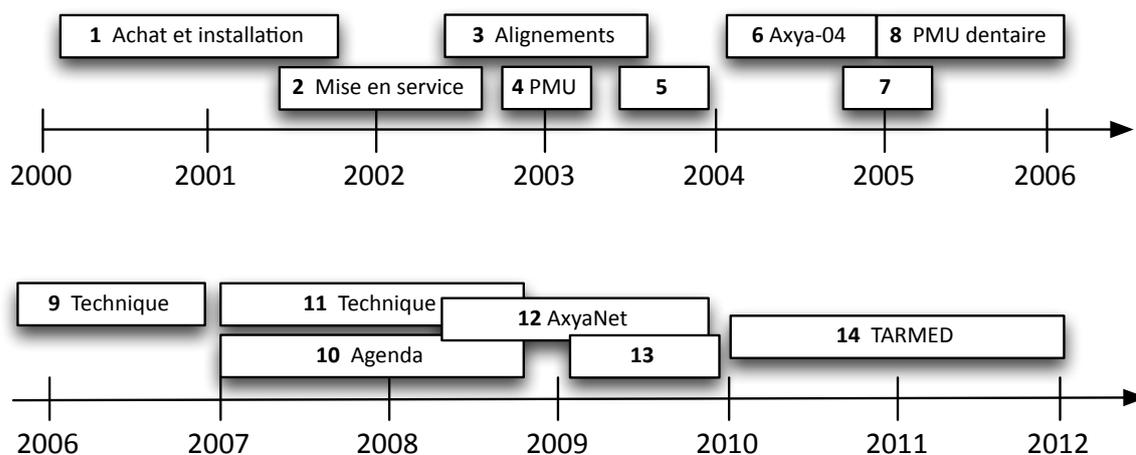


Figure 41. Dates importantes concernant la vie de l'application Axya (voir la description des événements dans le Tableau 16).

Données financières et facteurs déclencheurs d'évolution

Afin de retracer le cycle de vie de l'application Axya pour la période précédant la saisie des traces d'évolution dans la plateforme JIRA, nous avons appliqué EVOLIS aux factures relatives à l'application Axya. La Section 5.4 présentait l'importance des ressources investies (le coût) dans le système. Nous proposons d'évaluer les ressources investies soit en fonction du coût financier lorsqu'il est disponible, soit en fonction de l'énergie humaine investie dans les évolutions lorsqu'elle est disponible.

En appliquant EVOLIS aux données financières de l'application, nous pouvons retracer le cycle de vie selon les facteurs déclencheurs d'évolution (2002 à 2012) et également comparer les résultats avec les résultats de l'analyse de traces d'évolution de la plateforme JIRA pour la période correspondante (2007 à 2012). Comme mentionné précédemment, les données financières contiennent des paiements qui regroupent plusieurs prestations et qui ne contiennent que peu de détails sur la nature des activités effectuées sur l'application Axya. Les résultats obtenus ont donc une granularité relativement grossière par rapport à l'étude des traces d'évolution issues de JIRA. Il reste cependant possible de distinguer les multiples événements de la vie d'Axya.

Notre étude selon EVOLIS ne tient pas compte des frais occasionnés par l'acquisition d'Axya. Nous avons conservé uniquement les frais liés aux évolutions. A titre indicatif et pour mettre en proportion les investissements effectués pour les évolutions, l'acquisition de cette application

avait été budgétée à 3 millions. Les paiements effectués pour l'acquisition ont été échelonnés entre le dernier trimestre 2000 et le 3^{ème} trimestre 2002.

La Figure 42 présente la répartition des ressources financières investies dans Axya de 2002 à 2012 selon les facteurs d'évolution d'EVOLIS.

Lors des second et troisième trimestres 2002, nous retrouvons la période d'évolutions déclenchées par un besoin d'alignement qui a été nécessaire afin de corriger les défauts de « jeunesse » directement après la mise en service de l'application (Axya – 02).

Entre fin 2002 et début 2003, les ressources sont principalement investies pour des évolutions relatives au facteur Architecture. Ce résultat s'explique avec l'étendue de la couverture de l'application à un autre service de l'organisation : la Policlinique Médicale Universitaire. Puis la suite de l'année 2003 et l'année 2004 sont marquées par des évolutions d'alignement de l'application (Axya – 03, Axya – 04 et la consolidation). Lors du premier trimestre 2004, l'essentiel des ressources investies concerne des évolutions déclenchées par l'architecture. En effet, durant cette période Axya a également été déployée dans deux autres services de l'organisation.

Fin 2005 début 2006, l'application est déployée dans la Policlinique Médicale Universitaire Dentaire, ce qui explique les investissements déclenchés par le facteur Architecture.

Fin 2007 début 2008, les investissements sont déclenchés par le facteur Architecture afin d'intégrer le composant Agenda à Axya. Durant cette même période, nous retrouvons les investissements technologiques liés à Axya. Ces investissements sont également motivés par de l'alignement suite à des améliorations d'ergonomie et de fonctionnalités.

L'année 2009 est marquée par des investissements dans l'alignement du processus de facturation et également par l'introduction d'AxyaNet. Les dépenses faites pour AxyaNet ont toutes été classées comme étant motivées par l'Alignement car les informations à disposition dans les données financières ne permettaient pas de faire de distinctions plus précises.

Depuis 2010 les investissements ont été consacrés à l'introduction de TARMED. Ils sont classifiés comme étant motivés par le facteur Alignement car, en effet, TARMED impose d'adapter le processus de facturation en modifiant la façon de calculer la facturation des prestations destinées aux patients.

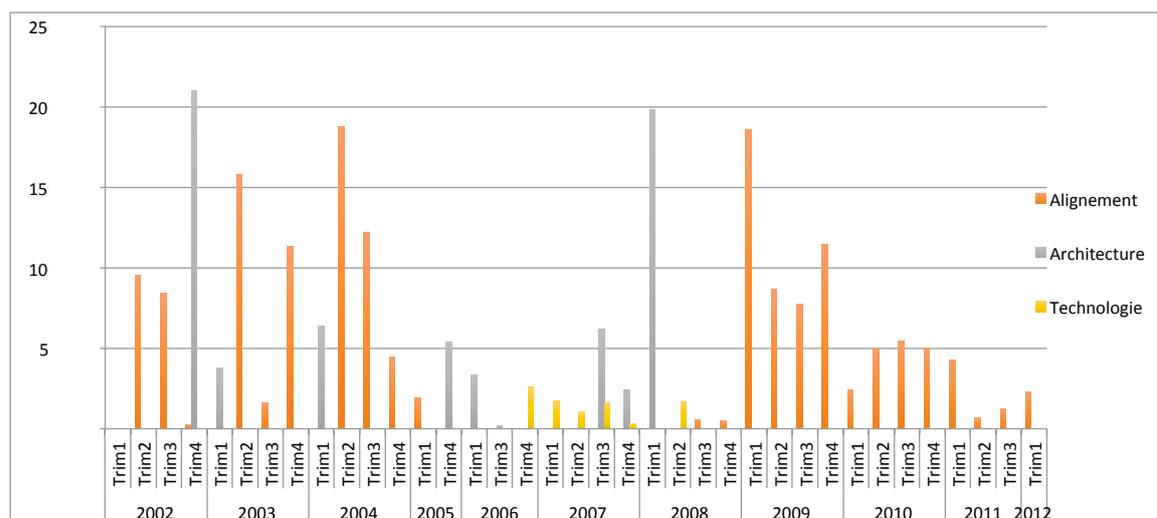


Figure 42. Investissements financiers pour l’application Axya répartis selon les facteurs d’évolution d’EVOLIS.

Facteurs déclencheurs d’évolution

Cette section présente les résultats de l’application d’EVOLIS aux interventions effectuées par l’équipe interne qui s’occupe des applications. Ces interventions ont été extraites de la plateforme JIRA. Comme mentionnée précédemment, la première intervention saisie dans JIRA concernant Axya remonte à l’année 2000. Cependant, entre 2000 et 2007, seulement 8 interventions sont répertoriées dans JIRA. Compte tenu de leur éloignement et de leur faible importance quant à l’ensemble des traces d’évolution, nous avons choisi de ne pas les représenter dans les figures présentant la répartition des interventions dans la vie de l’application.

Le Tableau 17 présente la répartition des interventions selon les facteurs déclencheurs d’évolution d’EVOLIS. Le pourcentage est établi en fonction du nombre d’interventions réalisées concernant chaque facteur dans l’ensemble des interventions relatives à Axya.

Tableau 17. Pourcentage d’interventions déclenchées par les facteurs d’évolution d’EVOLIS.

Facteurs déclencheurs	Pourcentage d’interventions
Alignement	67%
Architecture	27%
Technologie	5%
Satisfaction utilisateurs	1%

La répartition des interventions est presque de 2/3 d’interventions déclenchées par le facteur Alignement et 1/3 par le facteur Architecture. En comparant cette répartition avec celle de l’application Soarian, nous pouvons déduire que Axya est d’avantage couplée avec les autres composants du système d’information. Ce résultat se confirme en analysant le contenu des interventions saisies dans le projet JIRA transversal Intégration. En effet, environ la moitié des

interventions saisies dans ce projet concernent un couplage avec l'application Axya. Ces résultats confirment la description du rôle de l'application Axya faite dans la Section 8.2.1. Afin d'établir la facturation pour des prestations faites aux patients, Axya est couplée avec les autres composants du système d'information qui sont spécifiques aux différents services de l'organisation.

Axya est donc une application importante en terme de couplage avec les autres composants du système d'information. Les évolutions de cette application peuvent donc provoquer des évolutions architecturales concernant le couplage des composants.

La Figure 43 présente la répartition dans le temps des interventions selon leurs motivations. Contrairement à l'application Soarian, les évolutions de l'application Axya sont effectuées en interne comme en externe. Suite au manque de précision des données financières concernant les évolutions effectuées, il est difficile de retrouver une correspondance entre les facteurs d'évolution des deux sources.

Nous retrouvons toutefois des « sujets » d'évolutions communs entre ces deux sources de données. Notamment durant les années 2009 et 2010 concernant l'événement « Internalisation de la facturation ». Ces deux années regroupent à elles seules plus du 60% des interventions effectuées en interne sur l'application Axya. La répartition des facteurs d'évolution d'EVOLIS est quasi similaire à celle de l'ensemble des années étudiées avec environ 70% d'alignement et 20% d'architecture. Nous notons tout de même deux pics d'interventions relatives à l'alignement aux 2^{ème} et 4^{ème} trimestres 2009. Comme mentionné au début de ce paragraphe, le 80% des interventions internes qui affectent le composant facturation ont lieu durant ces deux années, ce qui correspond bien à l'événement « internalisation de la facturation » du cycle de vie. Concernant les interventions de ces deux années relatives au facteur architecture, elles concernent toutes le couplage d'Axya avec les autres composants du système d'information.

Au cours des années 2011 et 2012, la répartition des facteurs d'évolution d'EVOLIS change par rapport à celle de l'ensemble des années étudiées. Lors de ces deux années, la répartition est d'environ 55% pour les interventions motivées par l'alignement et de 45% pour celles motivées par l'architecture. L'année 2012 contient même d'avantage d'interventions liées à l'architecture qu'à l'alignement sachant que le nombre annuel d'interventions liées à l'architecture n'évolue que peu.

Pour résumer, Axya a nécessité beaucoup de ressources pour l'alignement en 2009, 2010 et début 2011, puis le niveau des ressources d'alignement a baissé. On peut en conclure que l'application est depuis mi 2011 relativement bien alignée avec les besoins métiers de l'organisation. Pour ce qui concerne les ressources d'architecture, elles sont relativement stables depuis 2009 avec quelques pics par périodes. Néanmoins leur niveau reste relativement élevé en comparaison à l'application Soarian. Nous noterons également que les interventions motivées par la technologie utilisée ou par la satisfaction des utilisateurs sont regroupées lors des deux années les plus actives à savoir 2009 et 2010.

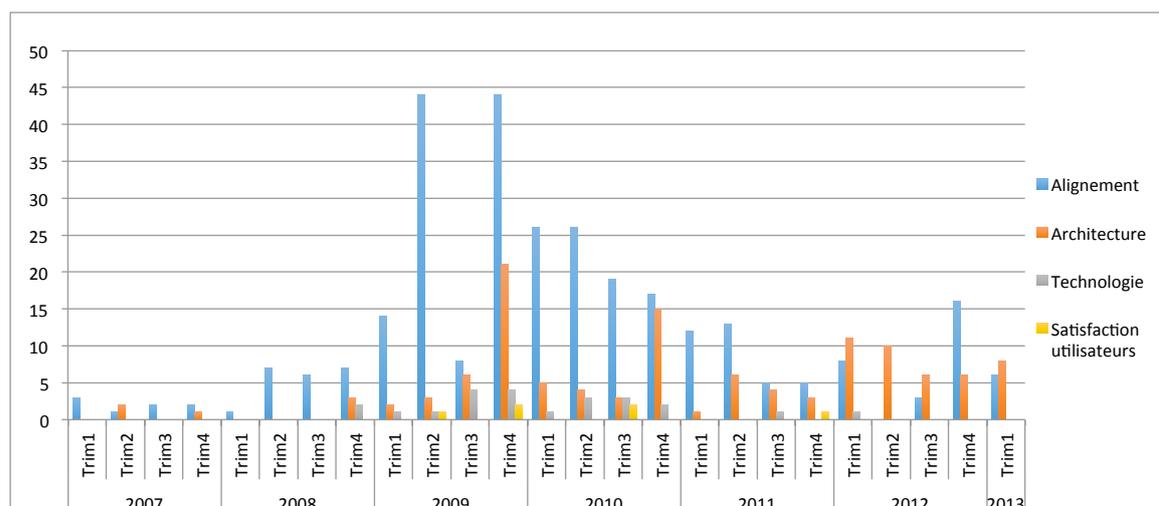


Figure 43. Répartition par trimestre de 2007 à 2013 des interventions sur l’application Axya en fonction des facteurs d’évolution d’EVOLIS.

Type d’activité des interventions

L’étude des interventions selon leur type d’activité renseigne sur la nature des activités entreprises. Le Tableau 18 présente la répartition des interventions selon leur type d’activité.

Plus de 80% des interventions sur l’application Axya modifient les règles métier soit en étendant les fonctionnalités (65%) soit en les corrigeant (15%) ou encore en les réduisant (2%). Viennent ensuite les interventions qui adaptent les propriétés du logiciel (11%). Le solde des interventions ne modifient pas le logiciel et concernent l’étude et l’évaluation du logiciel ainsi que de la mise à jour et des refontes de documentations.

Tableau 18. Pourcentage d’interventions pour chaque type d’activité d’EVOLIS pour l’application Axya.

Groupe d’activités	Type d’activité	Pourcentage d’interventions
Règles métier	Enhancive	65%
	Corrective	14%
	Reductive	2%
Propriétés logicielles	Adaptive	11%
	Performance et Groomative	<1%
Support	Consultive	2%
	Evaluative	1%
Documentation	Updative	4%
	Reformative	1%

La Figure 44 présente la répartition des types d'activités dans le temps. Pour des raisons de lisibilité nous n'avons pas représenté les types d'activités rassemblant moins de 2% des interventions.

Nous retrouvons donc le haut niveau d'activité des années 2009 et 2010 pour lesquelles nous remarquons que le principal type d'activité est Enhancive à raison de 64%. Les interventions d'alignement faites à cette période visent donc en priorité à étendre les fonctionnalités de l'application puis à les corriger (14%) et à les adapter (12%).

Nous remarquons que l'année 2009 regroupe quasiment toutes les activités de type Update, il s'agit de mises à jour de lettres, de dossiers de renseignements etc. qui n'impactent pas le code source ni les fonctionnalités de l'application Axya.

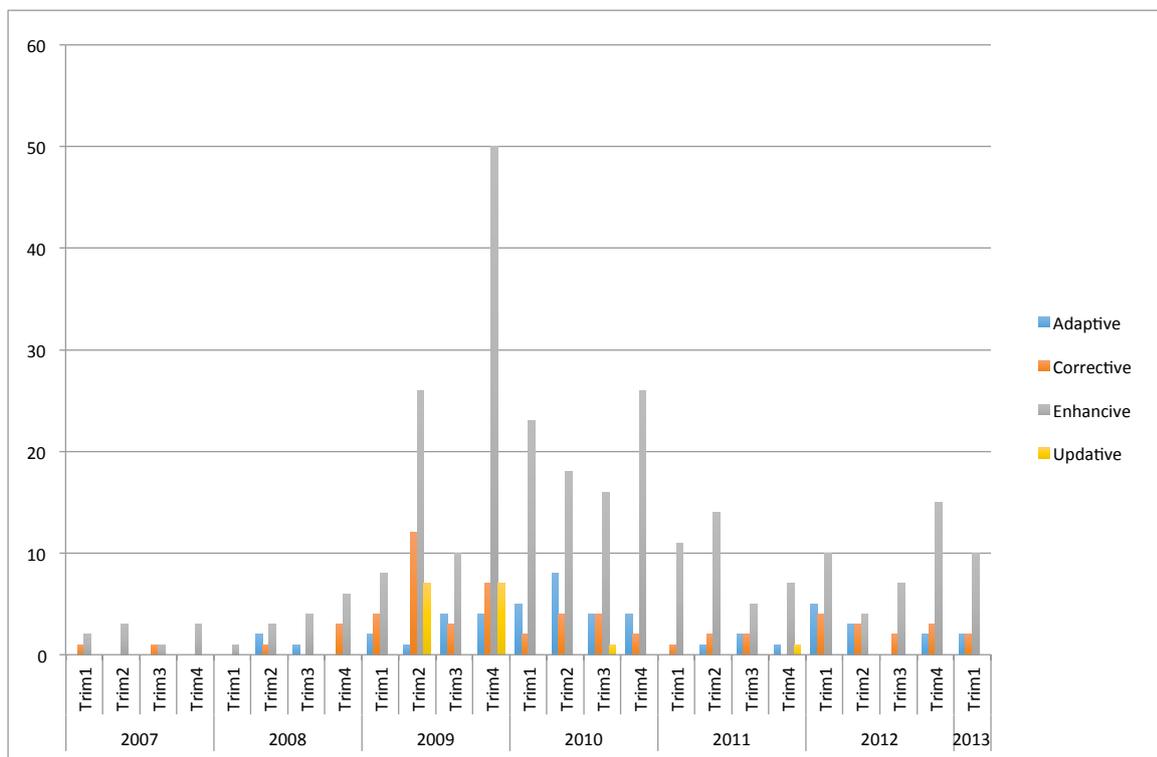


Figure 44. Répartition par trimestre des interventions par type d'activité sur l'application Axya.

Combinaison entre facteurs EVOLIS et types d'activités

La combinaison des facteurs déclencheurs d'évolution et des types d'activités renseigne de manière précise sur la nature de ce qui a réellement été entrepris durant les interventions.

Aucune intervention motivée par les facteurs Technologie et Satisfaction des utilisateurs n'a lieu en 2007 et durant 2008 pour la Satisfaction des utilisateurs. Ces années ne sont donc pas présentées dans les figures correspondantes (Figure 45)

En observant le facteur Technologie, nous trouvons des interventions principalement de type Adaptive. Ces interventions contiennent des conversions de langages de programmation et des migrations. Concernant le facteur Satisfaction des utilisateurs, les types d'activités que l'on retrouve dans les interventions sont variés. Deux activités modifient une documentation expliquant l'accès à Axya via le wifi et d'autres mettent à jour des lettres sur la demande d'utilisateurs.

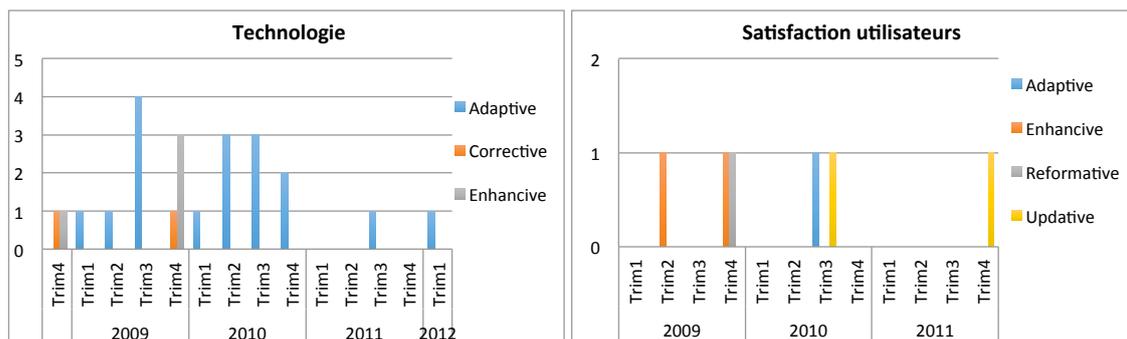


Figure 45. Répartition des interventions par type d'activité pour les facteurs Technologie et Satisfaction des utilisateurs pour l'application Axya.

La Figure 46 présente la répartition des interventions par type d'activité pour celles qui sont motivées par l'alignement entre le système d'information et le métier. Pour des raisons de lisibilité, nous n'avons pas affiché les interventions représentant moins de 2% de types d'activités pour le facteur Alignement.

Les figures de cette section nous renseignent sur le fait que les interventions de type Corrective portent principalement sur des problèmes d'alignement pour l'année 2009. Les corrections d'alignement sont faibles les années suivantes ce qui indique que l'application est bien alignée avec les besoins métiers. Les fonctionnalités nécessaires au métier sont étendues (Enhancive) chaque année avec beaucoup d'apports les années 2009, 2010 et 2011. La fréquence des interventions de type Enhancive diminue au fil des années ce qui indique une bonne couverture de besoins par l'application Axya. L'analyse du pic d'interventions de type Enhancive du 4^{ème} trimestre 2012 ne permet pas de déterminer un quelconque événement les ayant provoquées.

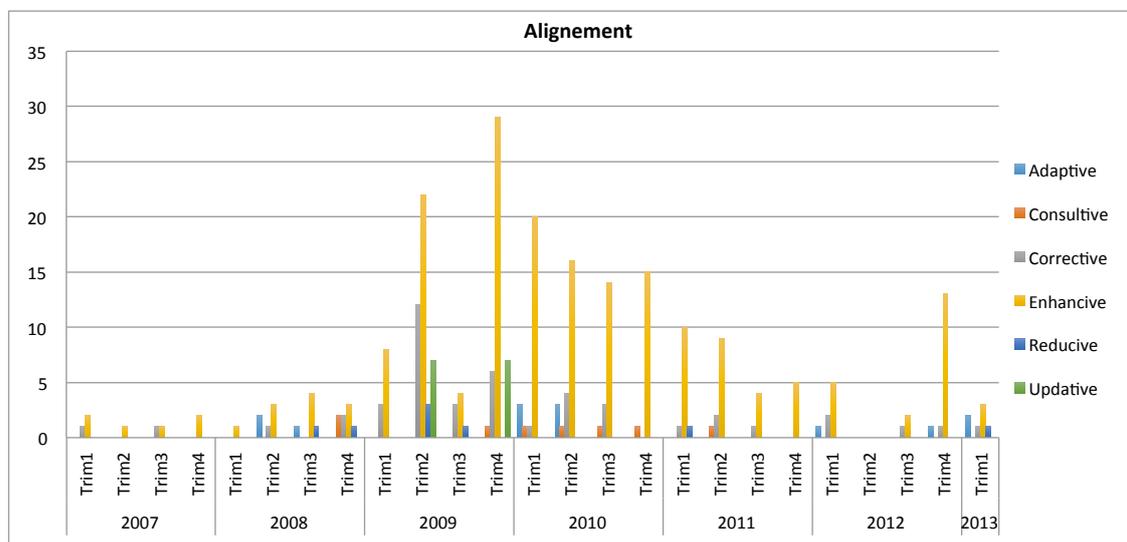


Figure 46. Répartition des interventions par type d'activité pour le facteur Alignement métier et système d'information pour l'application Axya.

La Figure 47 présente la répartition des interventions par type d'activité pour celle qui sont motivées par l'architecture du système d'information. Parmi ces interventions liées à l'architecture, 68% sont de type Enhancive qui, soit étendent les informations dans les couplages existants, soit mettent en place de nouveaux couplages entre Axya et les autres composants du système. Nous avons donc plus du 2/3 des interventions qui augmentent le couplage tandis que

17% sont de type Adaptive et 13% Corrective. Il est donc possible d'en déduire qu'Axya à un fort besoin de couplage et que ce couplage ne nécessite que peu de corrections suite à des erreurs.

Pour déterminer la répartition des interventions entre les différentes catégories de couplage des composants du système, nous avons appliqué la classification décrite en Section 5.3.3. Le Tableau 19 présente cette répartition. Le 95% des interventions sont motivées par le couplage entre Axya et les autres composants du système d'information. 40% concernent le couplage de nouveaux composants à l'application Axya et 54% la modification des couplages existants. Lors du pic d'interventions du 4^{ème} trimestre 2009, 60% des interventions concernent le développement de nouvelles interfaces de communication afin d'augmenter le couplage d'Axya avec son environnement et 15% utilisent les possibilités offertes par les applications. Nous avons donc un gros travail de couplage d'Axya avec son environnement qui a lieu fin 2009 puisque 75% des activités du pic y sont dédiées. Le second pic d'interventions du 4^{ème} trimestre 2010 ne privilégie aucune catégorie de couplage.

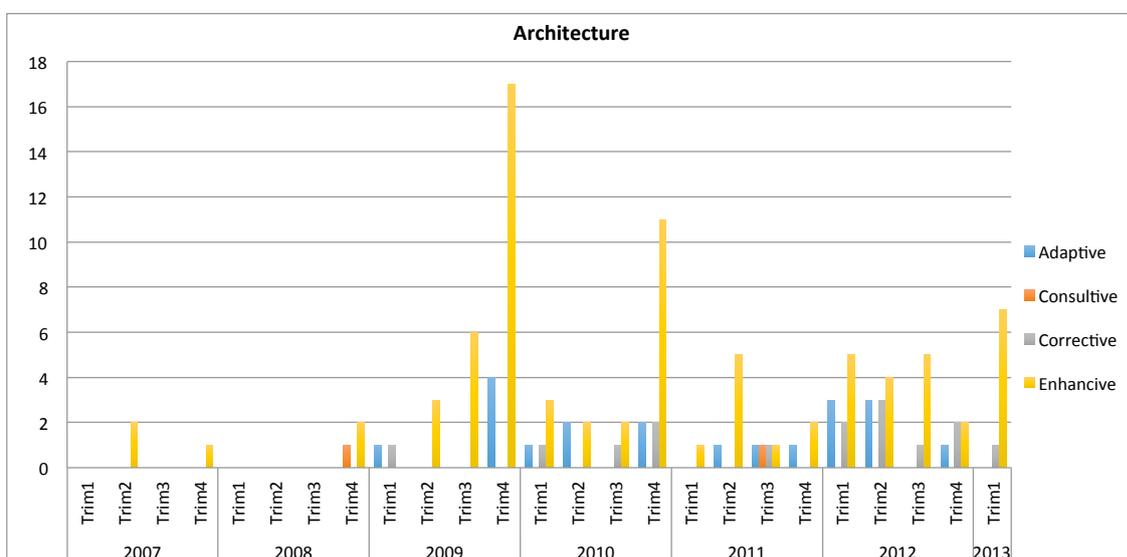


Figure 47. Répartition des interventions par type d'activité pour le facteur Architecture pour l'application Axya.

Tableau 19. Catégories de motivations architecturales concernant le couplage des composants pour Axya.

Catégories de motivations sur le couplage des composants	Pourcentage
1. Etude des possibilités de couplage	1%
2. Le couplage motive l'intervention	94%
Nouveau couplage de composants	40%
Développement d'une interface de couplage	26%
Utilisation des possibilités offertes	8%
Non précisé	7%
Modification d'un couplage existant	54%
Augmentation de la portée d'un couplage	21%
Diminution de la portée d'un couplage	1%
Correction d'un couplage	13%
Migration vers un autre type de couplage	4%
Non précisé	15%

3. Modification d'un composant couplé	5%
Adaptations sur les composants ou les interfaces	2%
Remplacement d'un composant couplé	
Redéfinition des interfaces de couplage	3%

Conclusion de l'étude des traces d'évolution de l'application Axya

L'étude des différentes sources de données selon EVOLIS permet de retracer le cycle de vie de l'application Axya. Grace aux données financières, le cycle de vie a pu être retracé depuis la mise en service de l'application en 2002. Pour les années 2007 à 2012 nous disposons, en plus des données financières, de données relatives aux évolutions effectuées sur Axya par l'équipe interne.

Pour retracer la vie d'Axya, l'application d'EVOLIS sur les données financières a permis de faire ressortir la nature des éléments ayant motivé les grands investissements de la vie d'Axya.

Comme pour l'application Soarian, l'enjeu le plus important qui a suivi la mise en service d'Axya était son alignement avec les besoins de l'organisation. Une fois cet alignement réalisé, l'étape suivante dans le cycle de vie d'Axya fut d'étendre la couverture de l'application à un autre service de l'organisation. Après cette augmentation de la couverture, une nouvelle série d'évolutions motivées par l'alignement et par des améliorations de performance a eu lieu. Ces dernières ont à nouveau été suivies par un nouvel accroissement de la couverture à un autre service de l'organisation.

Il en ressort une boucle alternant des évolutions motivées par de l'alignement et suivies par des évolutions motivées par l'architecture visant à étendre soit le couplage soit la couverture de l'application. Cette boucle s'est répétée plusieurs fois tout au long du cycle de vie d'Axya.

L'étude des évolutions faites en interne selon les facteurs déclencheurs d'évolution d'EVOLIS nous a montrés que d'avantage de ressources avaient été investies dans des évolutions déclenchées par le facteur Architecture que pour l'application Soarian. Ce résultat se confirme en analysant le contenu des interventions saisies dans le projet JIRA transversal nommé Intégration. En effet, environ la moitié des interventions saisies dans ce projet concernent un coupage avec l'application Axya. Nous en concluons que l'application Axya occupe une place importante en terme de couplage avec les autres composants du système d'information.

Une évolution de la politique de couplage de l'application Axya entraînera des conséquences non négligeables en terme de couplage avec d'autres composants. Axya peut être vu comme un moteur d'évolutions architecturales dans le système d'information.

Nous avons également observé qu'au cours des années 2011 et 2012, la répartition des facteurs d'évolution d'EVOLIS change par rapport à celle de l'ensemble des années étudiées. Les années précédentes, la répartition était d'environ 65% alignement et 25% architecture, lors de ces deux années, la répartition passe à environ 55% pour les interventions motivées par l'alignement et 45% par l'architecture. L'année 2012 contient même d'avantage de motivations liées à l'architecture qu'à l'alignement sachant que le nombre annuel d'interventions liées à l'architecture n'évolue que peu sur les années étudiées. Cette diminution des ressources investies dans des évolutions motivées par l'alignement nous conduit à conclure que depuis mi 2011 Axya est relativement bien alignée avec les besoins métiers de l'organisation.

8.4 Discussion

Pour cette seconde étude de cas, un des objectifs était de confronter EVOLIS à un grand volume de données, ceci afin de l'évaluer et d'en améliorer le contenu pour l'analyse de l'évolution de systèmes d'information de grande ampleur.

Comme nous ne pouvons pas appliquer EVOLIS à l'ensemble des applications du système d'information de l'organisation sélectionnée, nous avons choisi les composants stratégiques pour l'organisation. Au travers de cette seconde étude de cas, nous voulions notamment :

- Démontrer l'application d'EVOLIS sur un système d'information de large ampleur avec un volume de données d'évolution important.
- Evaluer et améliorer le contenu d'EVOLIS, en précisant les facteurs d'évolution et en fournissant des outils pour l'analyse d'un volume de données d'évolution important.
- Retracer les cycles de vie de chacun des composants étudiés en analysant les facteurs déclencheurs des évolutions ainsi que les types d'activités entreprises.
- Montrer comment EVOLIS permet de révéler des modèles spécifiques d'évolution.

Les résultats présentés dans la Section 8.3 démontrent l'application d'EVOLIS sur un grand volume de données. Pour chacun des objets étudiés, nous avons retracé le cycle de vie en fonction des éléments présents dans EVOLIS. L'analyse des traces d'évolution avec EVOLIS permet non seulement de confirmer les intuitions initiales quant à la nature des évolutions effectuées en fonction du rôle des objets étudiés dans l'organisation, mais également de connaître précisément l'importance de chaque facteur d'évolution dans le temps ainsi que le type des activités entreprises et les directions prises par chaque objet étudié.

A l'issue de l'étude de cas précédente, les résultats suggéraient deux tendances sur les données analysées qui conduisaient vers deux questionnements possibles pour des recherches futures :

1. Est-ce que la proportion d'activités concernant « l'Alignement métier et le système d'information » diminuent au fil du temps ?
2. Est-ce que la proportion d'activités relatives à « l'Architecture » croît au fil du temps ?

La Section 7.4 présente le contenu de ces questions. Pour résumer, la première question s'intéresse à la variation dans le temps des évolutions motivées par le facteur « Alignement métier et le système d'information ». La seconde s'intéresse à l'architecture des systèmes, et si avec le temps cette dernière a tendance à croître en vue d'augmenter la couverture opérationnelle des systèmes d'information.

Cette seconde étude de cas permet d'apporter un élément de réponse à ces questions. Pour les deux applications étudiées, l'activité concernant l'alignement du système avec le métier est très intense directement à la suite de la mise en service et diminue avec le temps. Nous expliquons ce phénomène selon deux raisons : premièrement la correction et l'adaptation de l'application au métier (visible dans les Sections 8.3.2 et 8.3.3), deuxièmement l'entendue et l'amélioration des possibilités de l'application. Cette intensité concernant l'alignement a par la suite tendance à baisser jusqu'à une future augmentation des fonctionnalités ou une extension de la couverture de l'application à d'autres services de l'organisation. L'augmentation des fonctionnalités ou la mise en services de l'application dans d'autres secteurs de l'organisation relance l'activité concernant l'alignement du système avec le métier (visible dans la Section 8.3.3).

Ces résultats conduisent à d'autres interrogations sur les éléments de réponse apportés. Elles sont les suivantes :

1. Est-ce que l'intensité de l'activité concernant l'alignement du système à la suite de sa mise en service se vérifie dans d'autres organisations ? Si c'est le cas, les raisons sont-elles similaires ?
2. Est-ce que la boucle entre alignement et augmentations de la couverture se retrouve dans d'autres systèmes d'information ? Si c'est le cas, est-ce que la majorité des systèmes suivent cette boucle ou y a-t-il des caractéristiques spécifiques à ces systèmes (par exemple une « taille minimale », ou un enjeu stratégique pour l'organisation) ?

Pour chacun des trois objets étudiés, le niveau des interventions déclenchées par le facteur « Satisfaction des utilisateurs » est faible. Nous expliquons ce constat par le fait que les traces d'évolution étudiées proviennent de l'équipe en charge de la gestion des applications métiers. Cette équipe n'est pas directement responsable du contact avec les utilisateurs. En effet, une autre équipe de la direction des systèmes d'information est en charge de ces activités : l'équipe « Services et support » (voir Section 8.1.5 et Figure 24). L'analyse des activités de cette équipe lors des différentes phases des cycles de vie des objets d'étude peut également être intéressante concernant les ressources engagées dans des activités visant le transfert de connaissances, le support, la satisfaction des utilisateurs et ainsi de suite.

Dans cette seconde étude de cas, nous avons analysé le composant d'EVOLIS « Coût » des évolutions (Section 5.4) en fonction des données financières concernant les dépenses faites pour chacun des objets d'étude. Malheureusement ces dépenses représentent uniquement les ressources investies en prestations externes lors des cycles de vie des différents objets étudiés. Comme chaque objet d'étude a sa propre politique de répartition entre les interventions externes et internes, ces données ne sont pas suffisamment précises pour être comparées. Toutefois elles indiquent les montants investis de manière externe et peuvent montrer une certaine dépendance à des prestataires.

Concernant la méthode pour l'évaluation des ressources investies dans les interventions effectuées en interne, nous avons utilisé la fréquence des interventions pour estimer le niveau d'activité tout au long des cycles de vie. Plus il y a d'interventions, plus le niveau d'activité est élevé. Afin d'évaluer les ressources investies dans chacune des interventions il aurait fallu connaître l'effort investi en jours/homme par les personnes en charge des interventions. Cette donnée n'était malheureusement pas suffisamment disponible et ne pouvait pas être estimée.

8.5 Apports de l'étude de cas vis-à-vis du processus de design

Cette seconde étude de cas a produit de multiples apports pour le processus de recherche design science suivi pour la conception de l'artefact EVOLIS. En effet, nous avons itéré à plusieurs reprises sur le design d'EVOLIS selon les besoins rencontrés pour mener à bien cette étude de cas.

8.5.1 Instanciation d'EVOLIS

Cette seconde étude de cas est la deuxième instanciation d'EVOLIS sur des données d'évolution réelles. Selon Iivari (2015), March & Smith (1995) et A. R. Hevner et al. (2004), une démarche design science en systèmes d'information vise à construire une recherche et ensuite, si possible, à l'instancier. Elle permet donc d'effectuer une nouvelle itération complète sur EVOLIS.

L'étude de cas précédente était une première instanciation visant principalement à évaluer les choix de design et de conception initiaux avec des données d'évolution réelles. Cette seconde

étude de cas vise non seulement à valider ces choix face à des systèmes de grande envergure mais également à définir de nouveaux éléments nécessaires à l'étude de ces systèmes comme par exemple le modèle conceptuel complétant EVOLIS. De plus, cette étude sur un grand volume de données d'évolution a permis d'évaluer EVOLIS en fonction de ses objectifs à savoir retracer le cycle de vie d'un système et aider à mettre en valeur des modèles spécifiques d'évolution.

8.5.2 Démonstration d' EVOLIS

Selon la méthodologie de recherche que nous avons adoptée (décrite en Section 3.5) et présentée dans l'article de Peffers et al. (2007), cette étude de cas s'inscrit dans l'activité dite de « Démonstration » dans laquelle l'usage de l'artefact est démontré. Cette étude de cas démontre son usage dans un contexte réel pour l'étude de l'évolution de systèmes d'information de grande ampleur. Cette démonstration nécessite qu'EVOLIS soit doté des éléments nécessaires pour son application. Dans ce dessein, nous avons enrichi le design d'EVOLIS avec un modèle conceptuel supportant l'analyse d'un grand volume de traces d'évolution (voir Section 8.2.3). La création de ce modèle a suivi le même processus design science. Nous avons premièrement défini les objectifs du modèle ce qui a permis de faire une sélection sur le type de modèle. Plusieurs itérations sur son design ont été nécessaires afin d'arriver à un modèle répondant aux objectifs, finalement une instanciation de ce modèle a été réalisée pour les besoins de l'étude.

Cette démonstration a également mis en lumière le besoin d'avoir d'avantage de détails sur la nature des activités d'évolution liées à la communication entre composants du système d'information. En se basant sur les données d'évolution de ce cas, nous avons créé une catégorisation des activités liées au couplage des composants du système en se basant sur le but de ces dernières. Les catégories sont issues de l'analyse détaillée des interventions concernant non seulement le couplage des composants mais également toutes celles motivées par le facteur Architecture d'EVOLIS.

Afin de s'assurer de la polyvalence d'EVOLIS, nous avons profité du volume de données d'évolution conséquent pour démontrer EVOLIS sur des objets de nature différente. Les évolutions de deux applications stratégiques pour l'organisation ainsi que celle d'un projet d'informatisation ont été analysées. Cette démonstration a permis de retracer les cycles de vie des objets étudiés ainsi que la répartition de l'activité selon EVOLIS.

8.5.3 Evaluation et itérations sur le design et le développement d'EVOLIS

Cette démonstration de l'utilisation d'EVOLIS a permis d'évaluer les objectifs d'EVOLIS avec un grand volume de données d'évolution réelles. Cette évaluation a permis de repérer les forces ainsi que les faiblesses d'EVOLIS pour l'étude des évolutions de systèmes de grande ampleur. Cette étude de cas peut donc également s'inscrire dans l'activité dite d'« Evaluation » de la méthodologie de recherche.

Voici les principaux objectifs évalués positivement lors de cette étude de cas :

- Comme pour l'étude précédent, l'utilisation d'EVOLIS dans un contexte réel et les résultats obtenus permettent d'évaluer positivement la pertinence du choix des facteurs d'évolution représentés dans EVOLIS ainsi que le choix de l'utilisation de la typologie des activités d'évolution selon Chapin et al. (2001) décrits dans la Section 6.4. Cette typologie permet de préciser les facteurs d'évolution en faisant ressortir la direction prise par les évolutions.

- Cette étude de cas confirme également la pertinence d'une étude longitudinale de l'évolution des systèmes d'information basée sur les composants d'EVOLIS. Les cycles de vie des objets étudiés représentent de manière suffisamment détaillée les différentes étapes permettant de retracer les événements de la vie des objets étudiés et de comprendre leurs enjeux en tant qu'observateur externe.

Voici les points qui ont été améliorés suite à l'évaluation de cette étude de cas et sur lesquels nous avons effectué des boucles de design et d'évaluation :

- La typologie des activités d'évolution initiale n'était pas suffisamment détaillée pour la description des activités relatives au facteur Architecture d'EVOLIS. Nous avons donc revu le contenu de ce facteur en catégorisant les activités selon leur motivation et les objectifs de ces dernières. Cette catégorisation a été évaluée en l'appliquant dans cette étude de cas et en comparant les résultats avec les objectifs prédéfinis (voir Section 5.3.3).
- Afin de faciliter l'application d'EVOLIS sur des données d'évolution de différentes formes, un modèle conceptuel supportant EVOLIS a été défini. Ce modèle a également été évalué selon ses objectifs en l'utilisant dans le cadre de cette étude de cas (voir Section 8.2.3).

Cette seconde étude de cas a donc permis de valider les choix effectués qui ne l'avaient pas été dans l'étude de cas précédente. Nous avons également pu apporter les ajustements nécessaires à EVOLIS pour permettre d'étudier un grand volume de données. La confrontation d'EVOLIS avec des données réelles d'évolution est selon nous le meilleur moyen pour valider les choix effectués en fonction des objectifs et pour adapter et compléter ces choix. De plus, l'application d'EVOLIS à des données d'évolution collectées a posteriori est une démarche similaire avec celle que pourrait entreprendre un chercheur ou un praticien désirant observer à l'aide d'EVOLIS l'évolution de systèmes d'information déjà fonctionnels.

8.6 Conclusions

Dans cette seconde étude de cas, nous avons appliqué EVOLIS à un grand volume de données d'évolution collectées a posteriori auprès d'un hôpital universitaire. Comme nous ne pouvions appliquer EVOLIS à l'ensemble des composants de ce système d'information, nous avons concentré notre étude à deux applications stratégiques pour l'organisation ainsi qu'à un projet d'informatisation important. Cette approche projet de l'évolution a permis d'étudier les activités de préparation et d'étude en vue de l'acquisition et de la mise en production d'un nouveau composant du système d'information.

Cette seconde étude de cas a permis d'aboutir le design d'EVOLIS en effectuant les activités suivantes :

- Nous avons pu évaluer une nouvelle fois la pertinence des choix effectués quant au design d'EVOLIS en le confrontant à un grand volume de données.
- Nous avons également pu faire ressortir les faiblesses d'EVOLIS en regard des données étudiées et compléter ces dernières en modifiant et en ajoutant des composants à EVOLIS.

Afin d'aboutir à ces résultats, nous voulions non seulement effectuer plusieurs cycles de « Design – Démonstration – Evaluation » d'EVOLIS mais également répondre aux questions de recherche suivantes :

« Comment un grand volume de données concernant l'évolution des systèmes d'information peut-il nous permettre d'améliorer le design d'EVOLIS ? »

Cette question est liée à l'étude de cas précédente dans laquelle nous avons appliqué EVOLIS aux évolutions de systèmes d'information de taille relativement contenue. Comme EVOLIS doit permettre d'étudier les évolutions des systèmes d'information peu importe leur taille, il est essentiel d'évaluer la polyvalence d'EVOLIS en l'appliquant sur des systèmes de tailles variables.

De par sa richesse d'évolutions et leur diversité, le grand volume de données étudié dans cette étude de cas a permis de déceler une faiblesse concernant la typologie des activités d'évolution relatives au facteur Architecture. En effet, une classification des activités selon une granularité plus fine permet de mieux comprendre les évolutions de ce genre. Cette classification a été évaluée en l'appliquant dans cette étude de cas et en comparant les résultats avec les objectifs prédéfinis (voir Section 5.3.3, Classification des activités relatives au facteur Architecture).

L'organisation et la sélection des données de différents formats provenant de différentes sources ont nécessité la création d'un modèle conceptuel. Ce modèle est une nouvelle contribution à EVOLIS qui permet d'augmenter sa polyvalence vis-à-vis du traitement de données d'évolution hétérogènes. Ce modèle est présenté dans la Section 8.2.3.

« Comment EVOLIS peut-il permettre de retracer et de comprendre les différentes phases du cycle de vie des systèmes d'information ? »

Lors de cette étude longitudinale des traces d'évolution de plusieurs composants du système d'information de l'organisation, nous avons pour chacun d'entre eux établi le cycle de vie selon les motivations d'évolution. Cette étude a fait ressortir les variations du niveau d'activité tout au long de l'utilisation des composants. Dès lors, il est possible de connaître les motivations conduisant à des périodes de forte activité et également d'analyser les combinaisons entre les motivations et les types des activités entreprises. Ces éléments générés par EVOLIS permettent aux observateurs externes que nous sommes de constater les différentes périodes d'activité, d'en déduire des phases par lesquelles les composants sont passés ainsi que leur contenu. Ces résultats sont présentés dans les Sections 8.3.1, 8.3.2 et 8.3.3.

« Comment EVOLIS peut-il aider à révéler des modèles (ou des schémas) spécifiques d'évolution dans un grand volume de données concernant l'évolution des systèmes d'information ? »

Cette dernière question vise à aider les gestionnaires ainsi que les chercheurs dans la compréhension du phénomène d'évolution en aidant à révéler des schémas spécifiques d'évolution. Lors de cette étude de cas, nous avons remarqué à plusieurs reprises des enchaînements d'activités d'évolution spécifiques. En effet, l'étude des traces d'évolution avec EVOLIS a fait ressortir une tendance qui se vérifie plusieurs fois dans les données étudiées. Cette tendance est composée d'un enchaînement entre des évolutions motivées par le facteur Architecture suivies immédiatement par des évolutions motivées par le facteur Alignement. Les évolutions d'architecture étendent la couverture opérationnelle du composant et les évolutions d'alignement corrigent ou adaptent ces nouvelles fonctionnalités. Les évolutions d'alignement sont intenses lors de la période qui suit directement la « nouvelle mise en service » puis diminuent avec le temps. Cette tendance apparaît dans la première étude de cas (Section 7.3), dans l'étude de traces relatives à l'application Soarian (Section 8.3.2) et à plusieurs reprises lors du cycle de vie de l'application Axya (Section 8.3.3).

Cette étude de cas présente aux chercheurs comme aux gestionnaires les résultats auxquels ils peuvent s'attendre après l'utilisation d'EVOLIS pour la classification et le regroupement des activités d'évolution entreprises sur un système d'information. L'utilisation que pourrait avoir un

chercheur d'EVOLIS a été présentée lors de ces deux études de cas qui démontrent l'utilité d'EVOLIS pour étudier l'évolution des systèmes d'information.

Afin que les praticiens et autres gestionnaires de systèmes d'information puissent disposer des informations produites par EVOLIS quant aux cycles de vie ou aux activités d'évolution relatives aux systèmes qui les intéressent, il est nécessaire d'aller plus loin dans la description de l'utilité d'EVOLIS pour le pilotage de l'évolution.

Cette seconde étude de cas a permis d'adapter EVOLIS pour l'étude de grands volumes de données et fournit donc les bases pour la conception et la construction d'un tableau de bord assistant le pilotage de l'évolution des systèmes d'information.

Chapitre 9

Tableau de bord pour le pilotage de l'évolution

Dans les Sections 5.2.4 et 5.4, nous avons mentionné qu'il existe un consensus entre praticiens et académiciens sur le fait que les investissements dans les systèmes d'information doivent être justifiés, mesurés et contrôlés. L'attention sur les investissements ne doit pas se confiner sur les étapes d'acquisition et de mise en service des systèmes, mais elle doit porter sur toute la période d'exploitation de ces derniers. Les praticiens ont donc besoin d'outils pour piloter ces investissements tout au long de la vie des systèmes.

Afin de pouvoir utiliser EVOLIS pour le pilotage de l'évolution des systèmes d'information, il est nécessaire de l'appliquer sur les données d'évolution de l'organisation. Cette application d'EVOLIS ne va pas faire d'elle-même apparaître les informations pertinentes pour le pilotage et l'étude des cycles de vie. Le regroupement et le traitement de ces données d'évolution est nécessaire afin d'en déduire de l'information. Pour ce faire, l'idéal serait de mettre en place un outil qui permette de rassembler les données d'évolution, de présenter ces données, de naviguer parmi ces données, d'afficher l'allocation des ressources, d'attirer l'attention sur des éléments inhabituels et ainsi de suite. En d'autres termes, cet outil qui facilite le pilotage de l'évolution et qui repose sur EVOLIS doit agréger les données pour créer de l'information et la mettre en forme selon des indicateurs afin de faciliter leur exploitation. Selon ces besoins, cet outil pourrait prendre la forme d'un tableau de bord.

Ce chapitre présente les éléments à prendre en compte dans la conception d'un tableau de bord ainsi qu'un prototype de tableau de bord de pilotage des évolutions. Ce prototype utilise à titre d'exemple les données présentés et analysés dans l'étude de cas précédente.

L'approche suivie pour la conception ce tableau de bord est une combinaison des notions présentés dans le livre *Information Dashboard Design* (Few, 2006) ainsi que de la démarche présentée dans l'article *A Guide to Creating Dashboards People Love to Use* (Juice analytics, 2009).

9.1 Fondations

Dans tout processus de design et de création d'un outil (ou d'un artefact), il est nécessaire que ce dernier ait une utilité. Il doit donc répondre à un problème identifié en fournissant une solution (Hevner et al., 2004).

Lors de l'étude des données d'évolution avec EVOLIS, nous aurions apprécié avoir un outil qui, lorsqu'on lui fournit des données d'évolution permette de les exploiter sans avoir à effectuer manuellement chaque extraction, chaque regroupement de données, chaque mise en forme, etc. Ces tâches sont principalement mécaniques, elles pourraient être prises en charge par un outil. Cet outil pourrait être utilisé à des fins d'étude de l'évolution des systèmes d'information par d'autres chercheurs du domaine.

Cette situation reste inchangée pour un gestionnaire de système d'information. Si on considère l'organisation présentée dans l'étude de cas précédente (voir Section 8.1.5), la personne en charge de l'évolution des 250 application pourrait bénéficier d'un outil lui permettant d'accéder relativement facilement aux résultats de l'application d'EVOLIS en visualisant, pour le composant de son choix, son cycle de vie, les ressources investies, leur allocation, la nature des évolutions faites dernièrement, etc.

La solution répondant à ces problèmes serait de fournir un outil permettant non seulement de représenter les données d'évolution pour les analyser mais également de présenter des indicateurs sur ces données d'évolution dans un but de pilotage.

Nous avons conclu qu'un outil de type tableau de bord permettant l'exploitation des données d'évolution classifiées avec EVOLIS serait adéquat pour répondre à ces problèmes. Ce type d'outil peut aider les gestionnaires dans le contrôle et le pilotage de l'évolution de leur système d'information, dans la stratégie et la planification des évolutions du système et en même temps, il peut assister les chercheurs dans l'observation du phénomène d'évolution.

9.1.1 Audience

Dans le but de définir le contenu et les objectifs d'un tableau de bord supportant l'étude et le pilotage de l'évolution des systèmes d'information selon EVOLIS, il est important de cibler qui seront les futurs utilisateurs. La conception du tableau de bord doit être guidée par les caractéristiques de ces derniers (Brath & Peters, 2004; Juice analytics, 2009).

Rôle

Le rôle de l'audience permet de déterminer quelles sont les décisions qui devront être prises par l'audience et à quels problèmes cette audience est confrontée.

Ces éléments permettent de structurer l'information du tableau de bord en fonctions de la future utilisation qui en sera faite.

Dans l'organisation étudiée lors de la seconde étude de cas, la personne qui nous a permis d'accéder aux données d'évolution et qui était enthousiaste quant à l'application d'EVOLIS sur les évolutions de son système avait le rôle de « Responsable applications ». Son équipe est chargée du support et de la maintenance des quelque 250 applications informatiques. Il a également la responsabilité de l'entrepôt de données institutionnel et de la plateforme d'intégration d'applications. C'est ce type de rôle qui doit être ciblé par un tableau de bord pour le pilotage de l'évolution selon EVOLIS.

Voici d'autres rôles de gestionnaires auxquels pourrait s'adresser ce tableau de bord : direction générale des systèmes d'information, responsable infrastructure, responsable service et support, responsable de la maintenance du système d'information, chef de projet d'informatisation etc.

Nous voulons également que le tableau de bord possède des fonctionnalités permettant d'accéder au cycle de vie des composants du système d'information à des fins d'analyse et de pilotage. Un second rôle que peut adresser ce tableau de bord est celui du chercheur qui étudie l'évolution des systèmes d'information. En effet, dans son activité d'étude des évolutions, le chercheur a besoins de fonctionnalités communes à celles du gestionnaire lors de l'analyse des cycles de vie des systèmes.

Contextes

Après avoir défini les rôles qu'adresse le tableau de bord, il s'agit également de définir le ou les contextes dans lesquels le tableau de bord sera utilisé. Il s'agit donc de déterminer de quelles informations l'audience a besoin pour effectuer ses tâches (étude ou pilotage) concernant l'évolution des systèmes d'information.

La notion de contexte d'utilisation englobe également la capacité des utilisateurs et leur confort dans l'utilisation des outils traditionnels de représentation et d'analyse de données. Par outils traditionnels, nous entendons des outils du genre tableur voir même un système de gestions de base de données. Dans le cas d'un gestionnaire de système d'information ou d'un chercheur, nous pouvons supposer qu'ils maîtrisent ces outils.

Le contexte contient également la familiarité de l'audience avec la terminologie utilisée et son expertise du domaine. En effet, il est essentiel que l'audience soit à l'aise avec la terminologie et les concepts d'EVOLIS. De plus, il est important que l'audience puisse identifier la provenance des données qu'elle utilise et la manière selon laquelle elles sont agrégées et représentées.

Il est important que non seulement les utilisateurs soient à l'aise avec ce que représentent les termes utilisés dans EVOLIS mais aussi que le tableau de bord intègre des descriptifs concernant la terminologie.

Comme le tableau de bord que nous présentons ici ne s'adresse pas à un rôle spécifique dans une organisation, nous allons orienter la conception du tableau de bord de manière à ce qu'il couvre globalement les possibilités d'EVOLIS, qu'il soit utilisable par plusieurs rôles de l'organisation et qu'il permette l'étude des évolutions par les chercheurs du domaine.

9.1.2 But du tableau de bord pour l'audience

D'un point de vue général, les tableaux de bord peuvent couvrir toutes sortes d'objectifs. Il est donc important de définir précisément quelle est la finalité du tableau de bord que nous concevons et quelle est sa valeur ajoutée.

Pour la population des praticiens, le tableau de bord devra permettre de :

- Détecter des éventuels problèmes lors des évolutions.
- Communiquer sur les progrès, les succès ou les échecs des évolutions et de leur planification.
- Suivre les ressources investies et leur allocation par composant, par motivation et par type d'activité des évolutions.
- Estimer et planifier les ressources à investir.
- Observer les cycles de vie des projets et des composants du système d'information.
- Estimer la phase du cycle de vie dans laquelle le système se situe.

- Fournir un support pour communiquer avec d'autres audiences.

Pour la population académique, le tableau de bord devra permettre de :

- Observer les cycles de vie des composants / systèmes d'information.
- Observer les ressources investies en fonction du cycle de vie.
- Dédire des modèles spécifiques d'évolution.
- Estimer la phase du cycle de vie dans laquelle le système se situe.
- Fournir un support pour communiquer avec d'autres audiences.

9.1.3 Quel type de tableau de bord

On peut lire plusieurs définitions différentes de ce qu'est un tableau de bord, certains soutiennent qu'un tableau de bord doit pouvoir être contenu sur une page (Few, 2006) et d'autres non (Palpanas & Chowdhary, 2007), qu'il doit présenter des données en direct ou pas, ou qu'il doit offrir une vue d'ensemble de l'organisation ou précise. Nous trouvons certaines de ces spécifications trop générales car tous les attributs du tableau de bord (sa forme, le niveau d'interactivité, les capacités analytiques, etc.) vont varier en fonction du contexte et de l'audience pour lesquels il a été conçu. Il est tout à fait imaginable que certains tableaux de bord aient leurs données actualisées une fois par jour, ou qu'ils ne concernent qu'une partie ou qu'un domaine de l'organisation.

Voici une description du type de tableau de bord que nous allons concevoir en fonction de ses caractéristiques :

Couverture :

Le tableau de bord a un but spécifique, il se focalise sur l'étude de l'évolution en se basant sur EVOLIS.

Rôle métier :

Le tableau de bord a en même temps un rôle stratégique en représentant une vue à long terme et de haut niveau de l'évolution et également un rôle opérationnel en permettant la planification, l'évaluation de la performance, l'allocation des ressources, etc. des évolutions.

Perspectives temporelles :

Le tableau de bord a à la fois une perspective historique, en permettant de retracer le cycle de vie passé du système d'information, une perspective temps réel, en permettant le monitoring des évolutions et une perspective future en permettant la planification et la prise de décision pour les évolutions à venir.

Personnalisation :

Le tableau de bord doit permettre de personnaliser le niveau de détail à afficher. Il doit être équipé d'une fonctionnalité permettant la recherche descendante (drill down) afin d'avoir accès à davantage d'informations sur la nature des évolutions.

Approche :

Le tableau de bord va fournir un point de vue exploratoire. En effet, l'utilisateur doit pouvoir naviguer dans les données et interpréter les résultats présentés. Pour ce faire, il faut donc que l'utilisateur ait connaissance de la terminologie et des concepts d'EVOLIS.

9.2 Prérequis

La conception d'un tableau de bord basé sur EVOLIS nécessite certains prérequis quant aux données d'évolution représentées.

9.2.1 Application d'EVOLIS

Pour rendre possible la création de ce tableau de bord, il y a naturellement un prérequis lié à l'application d'EVOLIS. En effet, afin de pouvoir afficher les cycles de vie des composants du système d'information, il est nécessaire d'appliquer EVOLIS a posteriori sur les centaines ou milliers d'évolutions déjà effectuées dans l'organisation et cela sur l'ensemble des composants du système d'information.

Pour les évolutions actuelles, afin d'avoir un aperçu « en direct » sur les activités et les évolutions en cours ou planifiées, l'idéal serait de trouver un moyen pour qu'EVOLIS soit appliqué lors de la création de l'activité ou lors de sa mise à jour dans le système servant au suivi des activités. Ceci nécessite la saisie de quelques informations supplémentaires soit par la personne qui effectue l'intervention soit par la personne en charge de superviser l'évolution. Un exemple de scénario de ce type est donné en Section 6.5.

Pour chaque évolution, les informations qu'il faudrait sauvegarder sont celles présentées dans le modèle conceptuel de données de la Section 8.2.3, à savoir pour chaque évolution :

- Composant(s) concerné(s)
- Date
- Durée
- Facteur de motivation d'EVOLIS et compléments si souhaités
- Type d'activité si l'évolution est composée d'une seule activité/intervention
- Si l'évolution est composée de plusieurs activités, pour chaque activité :
 - Date
 - Durée
 - Facteur de motivation d'EVOLIS et compléments si souhaités
 - Type d'activité

Comme nous prenons comme exemple l'organisation étudiée lors de la seconde étude de cas, pour chaque intervention il faudrait ajouter les informations suivantes dans le système JIRA :

- Durée (de manière systématique)
- Facteur de motivation d'EVOLIS et compléments si souhaités
- Type d'activité si l'évolution est composée d'une seule activité/intervention
- Si l'évolution est composée de plusieurs activités, renseigner systématiquement sur quelle est l'évolution « parent ».

Pour ce faire, il est nécessaire que toutes les personnes en charge de saisir ces informations aient connaissance d'EVOLIS et qu'une aide ou un élément de rappel soit disponible dans le système de suivi des activités pour assister la saisie.

Il est tout à fait envisageable que l'organisation ajoute de nouveaux composants à EVOLIS si elle devait en éprouver le besoin. Cet ajout d'information pourrait être par exemple la nature des interventions (évoqué dans la Section 7.2) qui avait comme objectif de typer les activités en se basant sur les domaines de compétences requis pour effectuer une évolution.

9.2.2 Accès aux données

Le tableau de bord devra avoir accès aux données d'évolution sur lesquelles EVOLIS a été appliqué. Pour ce faire, voici deux propositions de scénarios d'accès aux données :

- Le tableau de bord accède aux données d'évolution directement là où les informations concernant les activités d'évolution sont stockées (plateforme semblable à JIRA). Cette solution a l'avantage d'accéder aux données les plus récentes concernant les évolutions. Toutefois, il faudra trouver un moyen pour regrouper les données concernant un même composant et qui seraient saisies à différents endroits.
- Le tableau de bord puise ses informations dans une base de données dédiée qui repose sur le modèle conceptuel présenté dans la Section 8.2.3. Dans ce cas, le regroupement préalable des données d'évolution est nécessaire et la mise à jour des données ne sera pas forcément synchronisée avec le système de suivi des modifications utilisé (plateforme semblable à JIRA).

Ces deux scénarios ont leurs avantages et leurs inconvénients, le choix dépendra également de la technologie utilisée pour supporter le tableau de bord. Un tableau de bord basé sur un logiciel de type tableur n'aura pas les mêmes possibilités de connections et de mise à jour des données qu'un tableau de bord développé comme une application web.

9.3 Structure

Maintenant que nous avons défini ce que le tableau de bord devait permettre d'accomplir pour son audience ainsi que les prérequis nécessaires à sa création, il est temps de le concevoir. Cette section offre une « big picture » du tableau de bord en sélectionnant la forme, la structure, et les fonctionnalités adéquates.

9.3.1 Forme

Nous avons évoqué dans la Section 9.1.3 que la vision conventionnelle du tableau de bord était de le contenir sur une seule page ou écran. Nous pensons que la forme adéquate pour le tableau de bord est celle qui convient aux besoins de la situation et qui permet d'implémenter les fonctions requises. Dans notre cas, le tableau de bord contiendra une première page d'accueil dans l'esprit purement tableau de bord avec des indicateurs et une seconde page plutôt orientée comme un outil pour l'analyse des cycles de vie.

Nous avons suivi les recommandations de Juice analytics (2009) afin de sélectionner la forme la plus adéquate pour notre situation et les fonctionnalités requises. Ils proposent 8 facteurs pouvant influencer le choix de la forme ainsi qu'un classement des formes en fonction de leur adéquation avec ces facteurs. Le Tableau 20 présente les facteurs influençant le choix de la forme ainsi que leur importance pour notre situation. Le Tableau 21 présente l'efficacité des formes traditionnelles utilisées pour les tableaux de bord en fonction des facteurs de choix. Un + signifie que la forme est bien adaptée au facteur, une cellule vide signifie que la forme est moyennement adaptée au facteur et un - signifie que la forme n'est pas adaptée au facteur.

Tableau 20. Facteurs influençant le choix de la forme du tableau de bord.

Facteurs	Description	Importance pour notre situation
Actualité des données	Fréquence de mise à jour des données ?	Important, la mise à jour quotidienne serait un avantage.
Esthétique	Esthétique importante ou purement utilitaire ?	L'aspect utilitaire prime sur l'esthétique.
Mobilité	Besoin d'y accéder en déplacement ?	Pas nécessaire.
Connectivité	Besoin de se connecter à des sources externes de données ?	Oui afin d'afficher plusieurs types de données par exemple données financières et données d'évolution.
Détail des données	Besoin d'explorer les données ?	Oui, parcourir les données afin d'avoir plusieurs niveaux de détails.
Densité des données	Niveau de richesse des informations ?	Le plus possible afin de connaître précisément les raisons des problèmes ou des succès.
Interactivité	Besoin d'interaction avec le tableau de bord ?	Oui afin de naviguer dans les données.
Collaboration	Besoin de partager et de collaborer ?	Eventuellement, mais pas besoin d'un outil collaboratif pour la prise de décision.

Tableau 21. Adéquation des formes usuelles de tableaux de bord avec les facteurs (adapté de Juice analytics (2009)).

	Une page papier	Présentation papier	Tableur	Application en ligne	e-mail / texte	Grand écran
Actualité	-	-	+	+	+	+
Esthétique	+	+		+	-	+
Mobilité	+				+	-
Connectivité	-	-		+	+	+
Détail	-	+	+	+	-	
Densité	+	+			-	
Interactivité	-	-		+	-	-

	Une page papier	Présentation papier	Tableur	Application en ligne	e-mail / texte	Grand écran
Collaboration					+	-

En se basant sur notre situation, l'idéal serait une application en ligne connectée avec les différentes sources de données. Toutefois, selon Brath & Peters (2004), il n'est pas judicieux de se lancer immédiatement dans le développement d'une solution trop avancée. Ils recommandent de suivre un processus itératif de design et de créer plusieurs ébauches et prototypes afin de les évaluer. Nous allons donc recommander d'utiliser la seconde forme la plus appropriée, c'est à dire le tableur, afin de réaliser les premiers prototypes fonctionnels et de se lancer dans la création d'une application dédiée si cette dernière s'avère indispensable.

Concrètement, du point de vue de l'actualité de données et de la connectivité, il est possible de connecter un tableur à une source de données externe type base de données afin de récupérer les données d'actualité au moment souhaité. Concernant le niveau de détail et la richesse des données, tout dépend du contenu de la source de données. Le tableur n'est qu'un outil qui facilite le traitement et la représentation de ces dernières. En ce qui concerne la mobilité et la collaboration, il est aisé de déplacer un fichier de type tableur et de le partager pour collaborer. Il faut néanmoins tenir compte du fait que les différents lieux et les différents collaborateurs n'auront pas tous la même connectivité sur les données sources, ce qui n'est pas un problème avec une application en ligne. Au niveau de l'interactivité avec un tableur, une fois que les données sont récupérées, si la personne n'est pas satisfaite de l'interactivité proposée par le tableau de bord pour la navigation, on peut imaginer que la personne exploite ces données d'une autre manière en créant ses propres rapports.

9.3.2 Structure

Dans cette section, nous allons définir comment seront structurées les informations dans le tableau de bord.

Dans son livre *Information Dashboard Design* (Few, 2006), l'auteur précise que la structure est un des plus grand challenge de la conception des tableaux de bord. En effet, l'information ne peut pas être placée n'importe comment voir là où il y a de la place. Le contenu du tableau de bord doit être organisé d'une manière qui reflète la nature des informations et qui facilite leur exploitation et leur monitoring. D'un point de vue pratique, la structure peut servir de mécanisme de navigation pour l'utilisateur, elle montre le point de départ et va le guider aux étapes suivantes.

Une bonne structure de tableau de bord nécessite une compréhension en profondeur du fonctionnement des éléments qui y sont représentés. De leur expérience, Juice analytics (2009) identifient 3 principales structures : *flux*, *relations* et *groupements*.

Une structure basée sur les *flux* accentue une séquence d'événements ou d'actions à travers le temps. Une structure basée sur les *relations* accentue les relations entre des entités ou des mesures. Une structure basée sur les *groupements* vise à regrouper les informations liées en catégories ou en hiérarchie.

Nous avons retenu deux structures pour le tableau de bord. Ces structures sont étroitement liées aux audiences et aux situations auxquelles elles s'adressent.

Afin de faciliter le pilotage de l'évolution pour les praticiens, nous avons imaginé une première structure basée sur des *groupements* contenant des indicateurs concernant les évolutions.

Voici des exemples de catégories de regroupements :

- En fonction de l'importance stratégique des composants du système
- En fonction de l'utilisation des ressources par les composants
- En fonction de la performance de l'évolution (budgété vs réel)
- En fonction de la priorité des évolutions
- En fonction du niveau de criticité des évolutions
- Etc.

Afin d'accéder aux cycles de vie des composants, le tableau de bord va contenir une seconde structure basée sur les *flux* afin de représenter la séquence des évolutions dans le temps. Cette structure est destinée à la fois aux praticiens et aux chercheurs afin de déterminer des éléments du cycle de vie comme les périodes de forte/faible activité, l'allocation des ressources dans le temps et ainsi de suite.

9.3.3 Principes de design

Cette section présente les principes de design que nous avons retenus parmi ceux proposés dans Juice analytics (2009) afin d'augmenter la pertinence, la lisibilité et de faciliter la navigation et l'appropriation du tableau de bord par les différentes audiences.

Afin de ne pas surcharger la première page du tableau de bord avec trop de détails sur les informations, le premier principe de design que nous avons retenu est celui nommé *Gradual reveal* qui consiste à révéler les détails concernant l'information affichée à mesure que l'utilisateur exprime son intérêt. Un exemple d'application de ce principe pourrait être : afficher les composants du système d'information ayant nécessité le plus de ressources la dernière semaine, et si l'utilisateur démontre de l'intérêt (en cliquant dessus), lui afficher le détail des activités d'évolution classifiées selon EVOLIS.

Un second principe de design qui est pertinent lorsque plusieurs audiences peuvent être concernées par le tableau de bord est celui nommé *Customizable* qui consiste à permettre suffisamment de flexibilité pour que le tableau de bord soit utilisable par les différents utilisateurs. Un exemple d'application de ce principe pourrait être : la possibilité de préciser l'étendue des données affichées en faisant varier la fenêtre de temps pour les données prises en compte. Il serait également intéressant que l'utilisateur puisse mémoriser ou transmettre les différentes vues des données qu'il a créées.

9.3.4 Fonctionnalités

Afin de s'assurer de l'utilité du tableau de bord, il est nécessaire que ce dernier soit d'avantage que de simples graphiques sur une page et qu'il dispose de fonctionnalités d'exploration et de représentation des données d'évolution. Cette section présente les fonctionnalités retenues pour le design du tableau de bord de pilotage et d'étude de l'évolution.

Nous avons regroupé les fonctionnalités en deux groupes, les fonctionnalités basiques qu'il est possible de retrouver dans beaucoup de tableaux de bord et les fonctionnalités spécifiques pour l'étude de l'évolution avec EVOLIS :

Fonctionnalités basiques :

- Exploration des données en zoomant sur les données d'intérêt
- Filtres pour les données affichées

- Comparaisons entre données, périodes etc.
- Alertes lorsque des seuils sont dépassés (par exemple : ratios budget/réalité)
- Export, partage, impression afin d'utiliser le tableau de bord pour communiquer

Fonctionnalités spécifiques à EVOLIS :

- Navigation et visualisation des données selon les composants / applications / projets
- Sélection de la période observée
- Sélectionner et combiner les sources de données (si multiples)
- Vue des 4 facteurs EVOLIS en fonction du temps
- Vue combinée entre facteurs et types d'activité en fonction du temps
- Etc.

9.4 Prototype

Afin que le tableau de bord soit non seulement utile pour le gestionnaire de système d'information mais également pour le chercheur, nous avons choisi de créer une page d'accueil avec les informations stratégiques sur les évolutions destinées au pilotage de ces dernières, et une seconde page destinée à la fois aux gestionnaires et aux chercheurs permettant l'étude du cycle de vie des composants.

Les prototypes visibles dans ce chapitre ne sont pas fonctionnels. Ils permettent uniquement de représenter les potentiels bénéfiques quant à la création de ce tableau de bord permettant l'étude et le pilotage des évolutions.

9.4.1 Page d'accueil pour le pilotage

Cette page d'accueil regroupe les informations qu'il est possible d'extraire d'EVOLIS ainsi que d'autres informations en relation avec les évolutions qui pourraient être importantes pour le gestionnaire. Cette première page contient un mélange d'informations stratégiques et opérationnelles concernant les évolutions.

Nous avons choisi de représenter les informations pour lesquelles le gestionnaire peut être amené à effectuer des actions immédiates dans le haut de la partie gauche. Ces éléments ont été positionnés en fonction de la manière dont les personnes ont tendance à scanner une page. Le coin supérieur gauche semble être l'endroit où les personnes portent le plus d'attention tandis que le coin inférieur droit n'est que peu remarqué (Cutrell & Guan, 2007; Juice analytics, 2009; Nielsen & Pernice, 2010).

Les cercles rouges représentent des alertes concernant les évolutions, nous imaginons qu'en cliquant sur les lignes concernées, le gestionnaire puisse avoir accès à tous les détails concernant l'évolution.

La partie en haut à droite contient les informations stratégiques concernant la répartition des ressources entre les différents composants du système d'information. Nous avons imaginé le regroupement des composants en trois catégories selon leur fonction et leur importance dans l'organisation étudiée lors de la seconde étude de cas, un hôpital universitaire :

- Les composants stratégiques qui nécessitent des informations provenant de toute l'organisation et utilisés par le plus de collaborateurs.
- Les composants métiers médicaux qui sont essentiels au bon fonctionnement des différents services de l'organisation.

- Les composants de support qui n'ont rien à voir avec le métier de l'organisation mais qui ont tout de même une importance pour le bon déroulement des activités de l'organisation.

La Figure 48 présente la page d'accueil du tableau de bord pour les gestionnaires de l'évolution des systèmes d'information. Les informations de cette page sont regroupées et organisées sur une grille afin d'augmenter leur lisibilité (Few, 2006; Juice analytics, 2009). Voici les différents groupes d'informations représentés :

- Alertes concernant des évolutions :
 - Dépassement de la date d'échéance
 - Dépassement des ressources budgétées
 - Haut ratio d'activités de type *Corrective*
- Répartition des ressources pour les évolutions :
 - En fonction des trois groupes de composants définis précédemment selon le métier de l'organisation.
 - Par groupe, affichage des 3 composants les plus gourmands en ressources avec quelques détails sur la répartition selon EVOLIS
- Evolutions prioritaires en cours.
- Dernières évolutions terminées.
- Futures évolutions planifiées.

Si le gestionnaire désire connaître le détail des évolutions prises en compte pour réaliser la répartition des ressources selon EVOLIS, il lui suffit de cliquer sur les graphiques présentant la répartition des ressources pour accéder aux données concernant ces dernières évolutions. Si il désire connaître les détails des évolutions affichées hors de la partie répartition des ressource, le principe est le même, en cliquant sur les lignes souhaitées il affiche leurs détails.

Les données représentées dans la Figure 48 sont inspirées des différents composants abordés dans la seconde étude de cas.

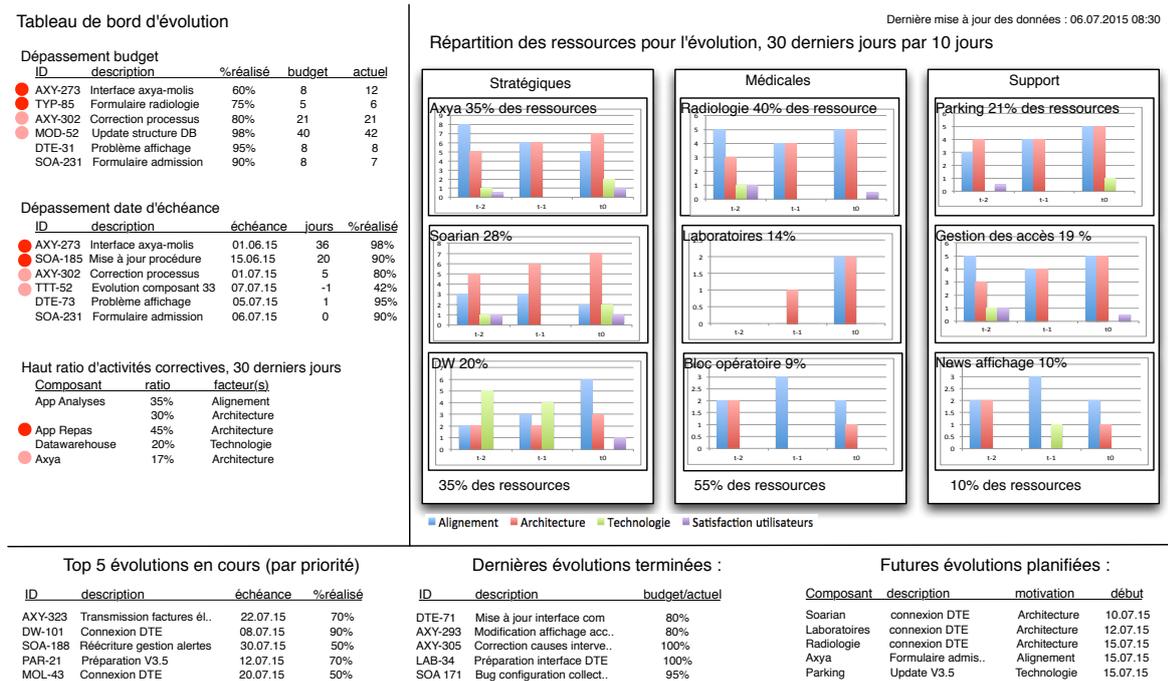


Figure 48. Présentation de la page d'accueil du tableau de bord de pilotage de l'évolution selon EVOLIS

9.4.2 Page de visualisation des cycles de vie selon EVOLIS

De par ses fonctionnalités et son utilisation, cette page s'éloigne de la définition traditionnelle d'un tableau de bord. Elle sert toujours au pilotage de l'évolution mais ne contient pas d'indicateurs quant à la bonne ou mauvaise santé des composants, c'est à l'observateur d'en déduire ses conclusions. Cette page peut être assimilée à un outil permettant d'accéder aux cycles de vie des composants afin, soit de prendre des décisions concernant le pilotage de l'évolution, soit d'étudier les différentes étapes des cycles de vie.

Cette page doit permettre la sélection du composant du système d'information sur lequel l'utilisateur souhaite observer le cycle de vie ainsi que la période lors de laquelle les données d'évolution doivent être représentées.

L'utilisateur doit donc avoir accès aux fonctionnalités suivantes :

- Sélection du composant à analyser.
- Sélection de l'intervalle de temps souhaité.
- Sélection de l'échelle de temps souhaitée.
- Téléchargement des données sources afin de les utiliser/traiter à sa guise. Le téléchargement pourrait se faire en format csv ou xml pour tableaux.

L'utilisateur doit y retrouver les informations relatives au cycle de vie suivantes :

- Cycle de vie selon les facteurs déclencheurs d'évolution EVOLIS.
- Cycle de vie selon les types d'activités entreprises pour évoluer.
- Affichage de la combinaison types d'activité par facteur déclencheur d'évolution.
- Affichage de la répartition des évolutions en fonction des catégories d'activités relatives à l'architecture du système (voir Section 5.3.3, Classification des activités relatives au facteur Architecture).

Si l'utilisateur désire connaître les détails des données affichées, le principe est le même que dans la page précédente, en cliquant sur les données représentées dans les graphiques, il affiche leurs détails.

La Figure 49 présente la seconde page du tableau de bord destinée à la fois au pilotage et à l'étude des évolutions.

Cette représentation du cycle de vie des composants du système d'information selon plusieurs point de vue (facteur déclencheur d'évolution, type d'activité, etc.) est un outil favorisant la détermination de l'étape du cycle de vie dans laquelle se trouve le composant. Cette représentation assiste donc les gestionnaires dans la prise de décision concernant la stratégie future vis-à-vis de l'évolution (évolutions possibles, maintien du service le plus longtemps possible, fin des activités d'évolution, remplacement du logiciel, etc.) de chaque composant du système d'information (Bennett & Rajlich, 2000b).

Nous y trouvons dans la partie haute les différentes commandes que peut effectuer l'utilisateur. En dessous de ces commandes, la partie de gauche contient un graphique représentant les facteurs ayant déclenché les différentes évolutions de l'intervalle de temps sélectionné. La partie droite contient un graphique représentant les types des activités effectuées durant les différentes évolutions de l'intervalle de temps sélectionné. Dans la moitié inférieure gauche, nous retrouvons pour chaque facteur ayant déclenché des évolutions les différents types d'activités entreprises. Finalement, dans la moitié inférieure droite nous avons représenté la répartition des évolutions motivées par l'Architecture selon les différentes catégories de couplage des composants du système (voir Section 5.3.3, Classification des activités relatives au facteur Architecture).

Tableau de bord pour le pilotage de l'évolution

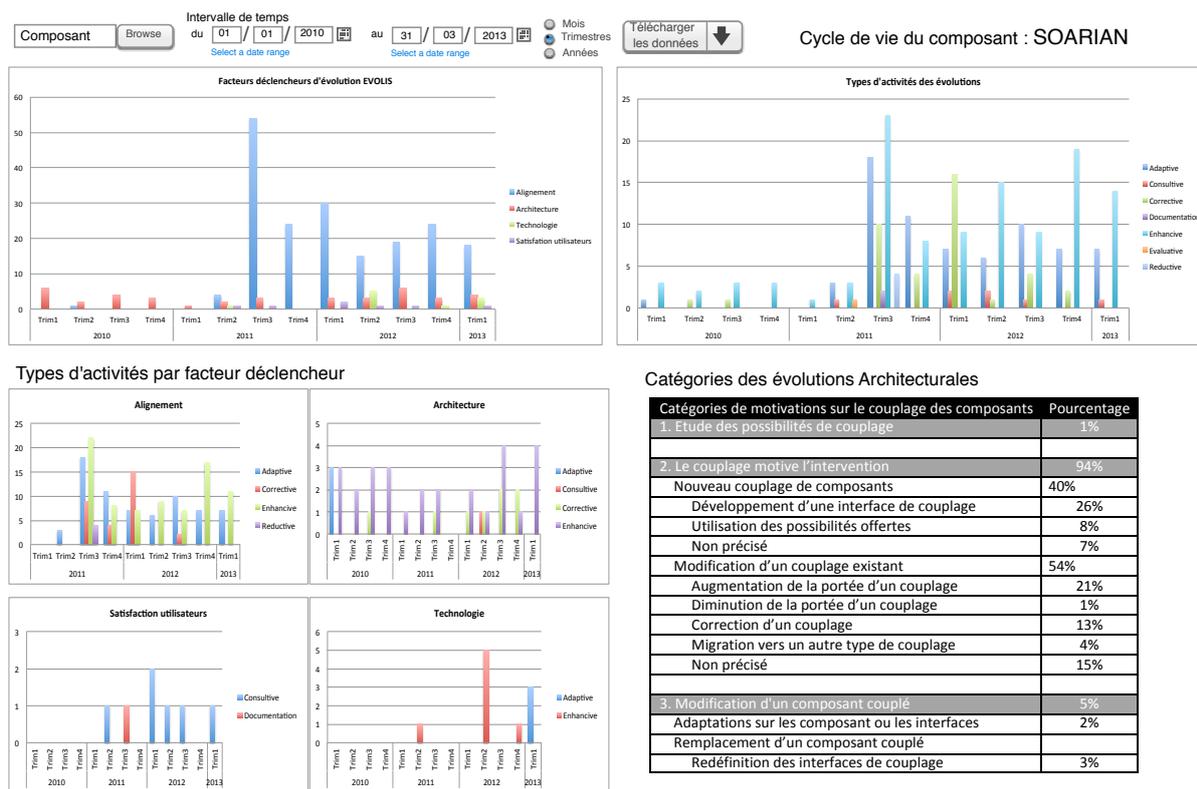


Figure 49. Présentation de la page d'étude des cycles de vie du tableau de bord de pilotage de l'évolution selon EVOLIS.

Les données représentées dans la Figure 49 sont inspirées des différents composants abordés dans la seconde étude de cas.

Nous avons mentionné à plusieurs reprises que l'utilisateur devait pouvoir accéder au détail des évolutions depuis le tableau de bord. Qu'il soit dans la page d'accueil ou dans la page de visualisation des cycles de vie, l'utilisateur doit pouvoir cliquer sur les données affichées pour les explorer. Plusieurs stratégies sont possibles afin d'afficher le détail de données :

- Rediriger l'utilisateur vers la source de données (base de données) ou vers l'application qui contient les détails des évolutions (par exemple l'application web de la plateforme JIRA).
- Dans le cas où les données sont présentes dans un tableau, affichage des données représentées en fonction de l'élément du tableau de bord sur lequel l'utilisateur a cliqué (exemple dans la Figure 50).

	A	B	C	D	E	J	K	L	O
1	Projet	Cle	Resume	EVOLIS	CHAPIN	Etat	Priorite	Resolution	Creation
2	SOARIAN	SOA-66	Modifications des critère	Alignment	Adaptive	Fermé	P3	Réalisé	28.09.11
3	SOARIAN	SOA-64	Modifications dans la pré	Alignment	Adaptive	Fermé	P3	Réalisé	26.09.11
4	SOARIAN	SOA-62	Mise à jour des entêtes/l	Alignment	Adaptive	Fermé	P3	Réalisé	21.09.11
5	SOARIAN	SOA-59	Modifications de la struc	Alignment	Adaptive	Fermé	P3	Réalisé	13.09.11
6	SOARIAN	SOA-58	Modification du formulai	Alignment	Adaptive	Fermé	P3	Réalisé	06.09.11
7	SOARIAN	SOA-45	Ajout du titre "Traitemer	Alignment	Adaptive	Fermé	P3	Réalisé	24.08.11
8	SOARIAN	SOA-42	Modifications de la struc	Alignment	Adaptive	Fermé	P3	Réalisé	16.08.11
9	SOARIAN	SOA-40	Adaptation de la lettre O	Alignment	Adaptive	Fermé	P3	Réalisé	07.08.11
10	SOARIAN	SOA-38	Modifications diverses di	Alignment	Adaptive	Fermé	P3	Réalisé	02.08.11
11	SOARIAN	SOA-37	Modifications dans la pré	Alignment	Adaptive	Fermé	P3	Réalisé	02.08.11
12	SOARIAN	SOA-36	Modification du formulai	Alignment	Adaptive	Fermé	P3	Réalisé	02.08.11
13	SOARIAN	SOA-34	Modification du formulai	Alignment	Adaptive	Fermé	P3	Réalisé	02.08.11
14	SOARIAN	SOA-33	Modification du formulai	Alignment	Adaptive	Fermé	P3	Réalisé	02.08.11
15	SOARIAN	SOA-32	Modification du formulai	Alignment	Adaptive	Fermé	P3	Réalisé	02.08.11
16	SOARIAN	SOA-29	Ajout d'un nouveau type	Alignment	Adaptive	Fermé	P2	Réalisé	29.07.11
17	SOARIAN	SOA-28	Modification des Display	Alignment	Adaptive	Fermé	P3	Réalisé	28.07.11
18	SOARIAN	SOA-22	Mise à jour des entêtes/l	Alignment	Adaptive	Fermé	P3	Réalisé	25.07.11
19	SOARIAN	SOA-12	Modification du libellé "l	Alignment	Adaptive	Fermé	P3	Réalisé	19.07.11

Figure 50. Exemple de détail de données (l'utilisateur a cliqué sur les évolutions du type Adaptive motivées par le facteur Alignement durant le 3^{ème} trimestre 2011).

Le détail des évolutions dépendra naturellement des données à disposition concernant les évolutions effectuées sur le système d'information. Plus les informations à ce sujet sont riches plus le gestionnaire pourra faire progresser le processus d'évolution ainsi que les décisions quant aux actions à entreprendre lors des différentes phases du cycle de vie des composants du système.

9.5 Apports du tableau de bord vis-à-vis du processus de recherche

Les deux études de cas précédentes ont permis de renforcer le design d'EVOLIS et de démontrer son utilité quant à l'étude des évolutions. Elles étaient principalement orientées sur le processus de design et d'évaluation d'EVOLIS.

La définition des éléments nécessaires à la création d'un tableau de bord de pilotage des évolutions basé sur EVOLIS contribue au processus de recherche design science en réduisant l'écart entre les connaissances théoriques (étude de l'évolution avec EVOLIS) et les connaissances pratiques des professionnels (pilotage de l'évolution du système d'information avec EVOLIS). La définition de ce tableau de bord de pilotage contribue donc à répondre à une des missions des recherches design en systèmes d'information décrites par Sein et al. (2011) : assister les praticiens dans leurs problèmes métiers.

Nous démontrons donc de manière pratique comment utiliser tous les concepts d'EVOLIS à des fins de pilotage de l'évolution par les praticiens.

9.6 Conclusions

Ce chapitre décrit le processus de création d'un outil basé sur EVOLIS pour le pilotage et l'observation des évolutions des systèmes d'information. Cet outil prend la forme d'un tableau de bord et intègre non seulement des indicateurs pour le pilotage mais également des fonctionnalités permettant l'exploration et l'analyse du cycle de vie des systèmes selon EVOLIS.

Comme la création d'un tableau de bord doit être faite spécifiquement en fonction du rôle de ses utilisateurs et des données disponibles, nous avons créé un premier prototype générique en se basant sur les données d'évolution que nous avons étudiées lors de la seconde étude de cas. Ce premier prototype n'a pas comme but final d'être fonctionnel, en effet pour que ce soit le cas, il nous faudrait bien plus de données, des accès permanents aux différents systèmes de l'organisation, la possibilité de les modifier etc. Le but de ce prototype est de démontrer la

démarche et les enjeux pour la création d'un tel tableau de bord ainsi que les possibilités de pilotages offertes par EVOLIS.

En incluant une première page contenant des indicateurs sur les évolutions en cours et passées, le prototype du tableau de bord met à disposition des gestionnaires un outil permettant l'évaluation de la performance des évolutions, il peut grâce à cela réaliser plus facilement un diagnostic sur les éléments de planification à améliorer. Il peut également suivre les évolutions des composants les plus consommateurs de ressources d'évolution selon différentes catégories.

Ces éléments génériques de l'interface du tableau de bord devront probablement être adaptés en fonction des besoins des utilisateurs finaux et de l'organisation dans laquelle le tableau de bord sera déployé.

La seconde page du tableau de bord est directement inspirée des résultats produits par l'application d'EVOLIS. Elle permet de renseigner l'utilisateur sur toutes les évolutions de tous les composants du système d'information et de cibler la période d'intérêt du cycle de vie. Cette page s'adresse autant aux gestionnaires qu'aux chercheurs. En effet, la représentation des cycles de vie selon EVOLIS aide les gestionnaires dans la détermination de l'étape du cycle de vie dans laquelle se trouve le composant. Ceci a pour objectif de les aider à établir une stratégie d'évolution des composants du système d'information. En ce qui concerne les chercheurs, cette seconde page permet de comparer visuellement et rapidement les cycles de vie des différents composants du système d'information. Ces comparaisons visuelles sont la première étape pour déterminer des modèles spécifiques d'évolution.

Ce tableau de bord apporte une contribution supplémentaire à notre recherche en démontrant les utilités possibles d'un outil tableau de bord pour les deux publics auxquels s'adresse notre recherche, à savoir les praticiens en charge de l'évolution du système d'information et les chercheurs du domaine. Il permet également de mieux visualiser l'utilisation d'EVOLIS à des fins de pilotage, que ce soit pour le pilotage opérationnel ou stratégique des évolutions.

Chapitre 10

Discussion et conclusions

Ce chapitre discute des résultats de cette recherche en résumant le processus que nous avons suivi pour la conception d'EVOLIS. Nous résumons les contributions novatrices d'EVOLIS pour le domaine de recherche ainsi que pour le pilotage des évolutions par les praticiens. Nous discutons également des limitations quant à l'usage et la mise en place d'EVOLIS dans des organisations et pour les chercheurs. Ce chapitre est construit selon les recommandations de Gregor & Hevner (2013) pour la discussion et les conclusions d'une recherche design en systèmes d'information.

10.1 Résumé de la recherche

Initialement, nous avons défini le but principal de cette thèse comme étant la création d'un outil permettant d'étudier et de mieux comprendre les motivations de l'évolution des systèmes d'information. En effet, le domaine de recherche dans lequel s'inscrit cette thèse n'aborde pas ou très peu les facteurs déclencheurs d'évolution. Afin d'atteindre ce but, nous avons décomposé la problématique en plusieurs questions de recherche auxquelles nous avons répondu tout au long du processus de recherche.

Afin de construire cet outil avec la rigueur nécessaire, nous avons suivi l'approche de recherche design science en systèmes d'information. Une recherche design science crée et évalue des artefacts afin de répondre à des problèmes. L'artefact créé est un cadre conceptuel destiné à l'étude des évolutions des systèmes d'information nommé EVOLIS. Le processus de recherche que nous avons suivi est décrit dans le Chapitre 3.

Le Chapitre 4 présente le domaine académique dans lequel cette recherche se positionne et duquel elle puise ses fondements. Il présente les théories et les méthodes concernant l'évolution des systèmes logiciels qui sont pertinentes pour la construction de notre outil. Ce Chapitre présente également la démarche par laquelle nous avons identifié parmi les connaissances du domaine de recherche les principaux facteurs poussant les organisations à

faire évoluer leurs systèmes d'information. Ces facteurs sont au nombre de 4 : *l'Alignement du système d'information avec les besoins métiers, l'Architecture, la Satisfaction des utilisateurs, et la Technologie.*

Les Chapitre 5, et Chapitre 6 décrivent l'artefact EVOLIS. Le Chapitre 5 détaille le contenu des facteurs déclencheurs d'évolution et le Chapitre 6 présente la suite du contenu d'EVOLIS permettant l'étude des évolutions avec notamment la méthodologie systématique d'étude de l'évolution de ce dernier.

Afin d'évaluer le design d'EVOLIS et de l'améliorer de manière continue (itérative), nous avons effectué deux études de cas. La première, décrite au Chapitre 7, démontre la pertinence du choix des facteurs pour l'étude des éléments déclencheurs d'évolution ainsi que celle de la méthodologie d'étude de l'évolution. La seconde étude de cas, décrite au Chapitre 8, présente l'application d'EVOLIS dans un contexte réel d'utilisation et permet d'évaluer la qualité et validité d'EVOLIS quant à l'étude des évolutions des composants d'un système d'information de grande taille. Le Chapitre 9 présente l'utilité d'EVOLIS au travers d'une application pratique : la réalisation d'un tableau de bord pour le pilotage et l'étude des évolutions d'un système d'information.

10.2 Enseignements

Cette section vise à présenter ce que signifient les résultats de cette recherche et en quoi ils sont liés aux objectifs initiaux.

Initialement nous avons deux objectifs principaux : étudier les évolutions des systèmes d'information selon leurs facteurs déclencheurs et grâce à cette étude apporter un moyen aux praticiens pour comprendre et piloter les évolutions de leur système d'information.

Nous avons démontré au travers des études de cas que l'étude des évolutions du point de vue des facteurs déclencheurs représentés dans EVOLIS est utile pour décrire le cycle de vie des systèmes. Par cycle de vie nous entendons les différentes étapes par lesquelles le système est passé lors de son processus d'évolution. Grâce à cela, il est donc possible de comprendre la nature des différentes étapes d'évolution par lesquelles est passé le système et dans quelle situation il est actuellement. Nous avons donc une meilleure compréhension du phénomène d'évolution des systèmes d'information.

Nous avons également confirmé que la classification des évolutions selon les facteurs déclencheur et la typologie des activités permet de connaître la nature des évolutions effectuées sur le système. Cette nature offre des possibilités de pilotage de l'évolution des systèmes en renseignant précisément sur les activités entreprises afin de faire évoluer le système.

10.3 Comparaisons avec les recherches du domaine

En s'appuyant sur ces connaissances préalables du domaine, nous avons non seulement déduit les principaux facteurs d'évolution mais nous avons également pu y puiser les outils et méthodes afin de les combiner pour concevoir EVOLIS.

Les nombreuses publications sur le domaine se focalisent sur l'aspect technique, les défis que représente l'évolution du logiciel plutôt que sur la dimension managériale de l'évolution du système d'information dans son ensemble. Avec EVOLIS, nous apportons un outil qui s'appuie sur les connaissances préalable et qui les combine pour répondre à des problématiques managériales de l'évolution des systèmes. En effet jusqu'ici, il n'existait aucun outil permettant

de retracer de manière standardisée le cycle de vie des systèmes en détaillant précisément les motivations des évolutions ainsi que les activités entreprises pour évoluer. EVOLIS permet donc de localiser le système dans son cycle de vie et de prendre des décisions managériales quant à ses évolutions futures.

Le fait de pouvoir retracer les cycles de vie des systèmes d'information de manière standardisée permet d'identifier des schémas ou des modèles d'évolutions. Ces modèles d'évolutions contribuent eux aussi à approfondir la compréhension du phénomène d'évolution des systèmes.

10.4 Importance théorique

Nous décrivons dans cette section les différentes contributions de cette thèse pour le domaine de recherche dans lequel elle s'inscrit.

10.4.1 Facteurs déclencheurs d'évolution

La première contribution de cette recherche est l'identification des 4 facteurs principaux déclenchant les évolutions. Ces facteurs d'évolution sont des éléments issus de domaines qui sont généralement étudiés séparément. EVOLIS les rassemble dans un même outil pour étudier les évolutions en fonction de ces derniers.

Afin de pondérer l'importance de ces facteurs déclencheurs d'évolution dans le cycle de vie des systèmes, EVOLIS utilise la notion de coût. Cette notion de coût permet de comparer l'importance des différentes évolutions de la vie d'un système. Ce coût représente d'avantage un investissement effectué par l'organisation. Il peut être exprimé de plusieurs manières : investissement financier, investissement en ressources, en énergie etc.

10.4.2 Etude des évolutions

Afin de caractériser les évolutions, en plus des facteurs déclencheurs, EVOLIS intègre des connaissances reconnues issues directement du domaine de recherche de l'évolution logicielle. Grâce à la combinaison entre la taxonomie des évolutions, la typologie des activités d'évolution et les facteurs déclencheurs, EVOLIS est un outil complet pour l'étude des évolutions.

Grâce à cela, il est possible d'avoir un aperçu détaillé de toutes les activités entreprises pour évoluer et donc de connaître exactement la composition d'une évolution avec l'allocation des ressources pour chacune des activités. C'est donc un outil permettant l'étude et l'analyse des évolutions.

10.4.3 Cadre conceptuel pour l'étude de l'évolution

Comme mentionné précédemment, EVOLIS intègre des connaissances puisées dans le domaine de recherche de l'évolution logicielle. EVOLIS n'est pas uniquement utile pour déterminer la répartition des ressources dans les évolutions ; en intégrant les facteurs déclencheurs d'évolution et les caractéristiques des évolutions, EVOLIS permet de comprendre les motivations qui poussent les organisations à faire évoluer leurs systèmes.

De plus, pour guider son utilisation, EVOLIS contient une méthodologie systématique d'étude des évolutions. Cette méthodologie détaille les étapes à effectuer et leur ordre afin d'avoir des résultats comparables entre plusieurs études de l'évolution de systèmes d'information.

EVOLIS est donc un cadre complet qui permet d'étudier les évolutions des systèmes d'information en se basant non seulement sur les motivations des évolutions mais également

sur les autres éléments du domaine de recherche présents dans EVOLIS. Il permet la comparaison des évolutions de plusieurs systèmes afin de mettre en évidence des tendances d'évolution et d'en apprendre davantage sur le processus global d'évolution des systèmes d'information.

10.5 Importance pratiques

Nous décrivons dans cette section les différentes contributions de cette thèse pour le contrôle et le pilotage des évolutions des systèmes d'information par les praticiens.

10.5.1 Utilisation pour étudier des traces d'évolution

Pour un gestionnaire de système d'information, l'application d'EVOLIS permet de connaître précisément l'allocation réelle des ressources pour une évolution. Une utilisation répétée d'EVOLIS sur les évolutions d'un système d'information ajoute la dimension temporelle et permet de retracer le cycle de vie du système d'information. Cette connaissance détaillée des évolutions permet de cerner les enjeux de l'évolution de chaque composant du système et d'adapter la planification en fonction.

La connaissance du cycle de vie des composants du système permet d'identifier la phase du cycle de vie dans laquelle le composant se situe. Cet élément peut aider les gestionnaires dans la planification et la stratégie d'évolution du système. En effet, le gestionnaire dispose d'informations concernant les composants qui se dirigent vers leur phase de service ou de maturité et qu'il faudra envisager de remplacer.

10.5.2 Un tableau de bord pour le pilotage de l'évolution

EVOLIS contient les fondements pour étudier et comprendre le processus d'évolution des systèmes d'information. Afin d'assister les gestionnaires dans leur tâche de pilotage de l'évolution, nous avons utilisé les informations mises à disposition par EVOLIS pour créer un outil favorisant le pilotage de l'évolution des systèmes d'information.

Nous présentons une démarche pour concevoir un tableau de bord de pilotage qui organise les informations générées par EVOLIS afin de les mettre à disposition des praticiens pour leur tâche de pilotage de l'évolution.

10.6 Limitations

Une des limitations inhérente à cette recherche concerne l'évaluation de l'artefact par un retour sur le terrain concernant les résultats. Les documents en notre disposition confirment les résultats, cependant ces résultats n'ont pas été évalués sur le terrain. Une telle évaluation pourrait être un premier pas et un bon indicateur concernant la mise en place d'EVOLIS pour la conduite et le pilotage de l'évolution.

La principale limitation quant à l'application d'EVOLIS pour générer le cycle de vie des systèmes d'information est la charge de travail supplémentaire engendrée par son application sur des évolutions passées. En effet, il s'agit de passer en revue toutes les activités effectuées sur les systèmes pour y appliquer EVOLIS ; en supposant qu'il reste des traces concernant ces évolutions passées.

Pour les chercheurs qui souhaitent approfondir les connaissances du domaine avec l'application d'EVOLIS, il peut être laborieux de trouver des organisations qui acceptent de mettre à disposition leurs données d'évolution.

Pour les praticiens, l'application d'EVOLIS sur les évolutions en cours augmente encore la charge de travail « administratif » pour les développeurs et les chefs de projets en charge des évolutions. Il peut en résulter une certaine réticence de ces personnes quant à son application. De plus, pour que l'application d'EVOLIS soit productive, il est nécessaire que toutes les personnes en charge des évolutions soient à l'aise avec la terminologie et la méthodologie d'étude de l'évolution du cadre conceptuel EVOLIS.

Le challenge de l'application d'EVOLIS n'est pas qu'humain. Il ne s'agit pas juste d'appliquer EVOLIS, il faut également mettre en place une structure technique pour l'accueillir et rendre possible l'exploitation des données.

Malgré ces limitations, à la vue des résultats produits dans les deux études de cas, nous sommes persuadés de l'utilité d'EVOLIS pour la planification et la conception de la stratégie d'évolution des systèmes d'information.

10.7 Travaux futurs

Afin de confirmer le modèle d'évolution présenté dans la Section 8.6 et d'en découvrir d'autres, de nouvelles applications d'EVOLIS sont nécessaires. Ces modèles d'évolution sont des connaissances supplémentaires permettant d'assister les praticiens dans la conception de la stratégie d'évolution de leur système d'information.

Jusqu'ici les applications d'EVOLIS n'ont pas abordé les phases de suppression progressive et de retrait des composants du système. L'étude des motivations et des activités entreprises durant ces phases peut également mettre en avant des modèles d'évolution ainsi que d'autres indications pour le pilotage des évolutions.

Des travaux futurs pourraient faciliter la recherche et le rassemblement des traces d'évolution dans des grands ensembles de données. L'extraction systématique des données pourrait être assistée voir automatisée par des outils ou des scripts vers une structure adaptée à l'application d'EVOLIS. Il en est de même pour la classification et l'étude des évolutions avec EVOLIS, de futurs travaux pourraient envisager des processus de classification semi-automatiques des évolutions en se basant sur les descriptions des activités ou des évolutions.

Concernant le tableau de bord, des travaux futurs pourraient déterminer un ensemble d'indicateurs types basés sur EVOLIS qui pourraient aider les gestionnaires dans leur tâche de pilotage des systèmes.

La réalisation d'un tableau de bord générique basé sur ces indicateurs types et connecté à une source de données d'évolution pourrait permettre de démontrer d'avantage les bénéfices d'EVOLIS pour le pilotage de l'évolution et de valider le choix d'un outil de type tableau de bord pour assister les gestionnaires.

10.8 Conclusions

Dans l'introduction, nous avons défini le but principal de cette thèse comme étant la création d'un outil permettant d'étudier et de mieux comprendre l'évolution des systèmes d'information. Pour atteindre ce but, nous avons décomposé la problématique en plusieurs questions de recherche auxquelles nous avons répondu tout au long du processus de recherche et notamment lors des deux études de cas.

Ce processus de recherche a abouti à la création d'un outil pour analyser de manière systématique les évolutions des systèmes d'information en intégrant des notions qui sont généralement adressées séparément. Cet outil est un cadre conceptuel que nous avons nommé EVOLIS. EVOLIS étant l'acronyme de EVOLution of Information Systems.

EVOLIS permet d'étudier les évolutions en se basant sur la nature des éléments les ayant déclenchées ainsi que sur les activités entreprises pour évoluer. EVOLIS met en lumière les principales motivations des évolutions dans chaque étape du cycle de vie et renseigne sur la manière dont les évolutions sont effectuées. Il est donc possible de retracer le cycle de vie de tous les composants du système en fonction des particularités d'évolution de chacun. Nous avons démontré et évalué les possibilités d'EVOLIS dans deux études de cas sur des traces d'évolution réelles de systèmes d'information de natures différentes et également d'un projet d'informatisation. Ces études de cas ont permis de mettre en avant la polyvalence d'EVOLIS quant à l'étude de différents types de traces d'évolution et aussi pour l'étude des répercussions d'un projet d'informatisation en terme d'évolutions.

Pour un chercheur, l'application d'EVOLIS permet d'étudier les évolutions des systèmes d'information en se basant non seulement sur les motivations des évolutions mais également sur les autres éléments présents dans EVOLIS. Il peut aussi comparer les évolutions de plusieurs systèmes afin de mettre en évidence des modèles ou des schémas d'évolution selon le cycle de vie des systèmes. Dans le cadre des études de cas que nous avons menées, nous avons remarqué un premier modèle d'évolution se répéter. Il est composé d'un enchaînement entre des évolutions motivées par le facteur Architecture suivies immédiatement par des évolutions motivées par le facteur Alignement. Les évolutions d'architecture étendent la couverture opérationnelle du composant et les évolutions d'alignement corrigent ou adaptent ces nouvelles fonctionnalités.

Pour un gestionnaire de système d'information, l'application d'EVOLIS permet de connaître précisément l'allocation réelle des ressources pour une évolution. Une utilisation répétée d'EVOLIS sur les évolutions d'un système d'information ajoute la dimension temporelle et permet de retracer le cycle de vie du système d'information. Cet élément peut aider les gestionnaires à localiser le système dans son cycle de vie et donc il peut les aider dans la planification et la stratégie d'évolution du système. Les modèles d'évolution qu'il est possible d'identifier avec EVOLIS sont également une aide précieuse pour définir la stratégie de pilotage du système d'information et les activités à entreprendre lors de la planification des évolutions.

Avec le même objectif, aider les gestionnaires dans la planification et la stratégie d'évolution de leur système d'information, EVOLIS fournit les bases nécessaires à l'élaboration d'un tableau de bord pour le suivi du cycle de vie et le pilotage de l'évolution du système d'information.

Annexes

Annexe A Première étude de cas

A.1 Exemple de trace d'évolution pour la première étude de cas.

Les documents collectés et analysés sont les offres faites par la société de services informatiques en charge des systèmes d'information des 6 organisations. Ces offres ont toutes été acceptées et effectuées.

La Figure 51 est une page extraite d'une offre. Certaines parties confidentielles sont noircies. Voici un résumé des éléments contenus dans les offres analysées :

- Le but du document ainsi que de l'intervention effectuée,
- Le besoin couvert par cette intervention,
- La date d'établissement du document,
- La personne responsable de l'intervention,
- Un planning pour l'accomplissement des activités,
- Un détail du travail effectué pour chacune des activités ainsi que la durée du travail en jours/homme,
- L'impact du travail effectué sur le système (modules concernés),
- L'impact des activités sur les processus de l'organisation (si les activités modifient les processus internes).



Gestion des cas CRM

1 But du document

[Redacted]

pour la mise en place du module CRM afin de vous aider à suivre les demande de permis de construire et les delais y relatif.

2 Étendue des prestations et planning

2.1 Étendue des prestations et honoraires

Cet avenant inclut les prestations suivantes :

Prestations	Jours
<ul style="list-style-type: none"> • Installation du module CRM • Création d'une section « Permis de construire » • Création de 2 catégories « 30 jours » et « 2 ans » • Ajout d'une règle de rappelles par e-mail : <ul style="list-style-type: none"> ◦ Rappel 1 mois avant échéance. ◦ Liste tous les cas non attribué. 	1.5
<ul style="list-style-type: none"> • Formation et support 	1.0
Total (en jours)	2.5



Mise en en place du lien entre OpenERP et les email (Mail gate) permettant de faire en sorte que les mails entrant et sortant d'une adresse définie soient directement insérer sous la forme de cas CRM.	2.0
Total (en jours)	2.0

2.2 Remarques

Cette offre n'inclut pas les prestations non prévues dans le chapitre précédent et en particulier :

- L'importation des données



3/5

Figure 51. Exemple de document analysé.

Annexe B Seconde étude de cas

B.1 Détail des sources : traces d'évolution des objets d'étude

Nous présentons ici le détail des données extraites de la plateforme JIRA pour chaque objet d'étude. Les colonnes représentées sont celles qui contiennent de l'information. Certaines de ces colonnes ont été conservées malgré le fait qu'elles ne soient que rarement renseignées par les personnes en charge des évolutions.

Dans les exemples de traces d'évolution, les données ont été anonymisées.

B.1.1 Projet JIRA : Dophin

Ce projet regroupe les données d'évolution relatives à au projet d'informatisation Dophin. Il contient 51 évolutions au 27.02.2013.

Colonnes d'information après retrait des colonnes non renseignées :

Clé	Résumé
Type de demande	Etat
Priorité	Résolution
Composants	Descriptif
Qualité de mise en prod	

Tableau 22. Exemple de trace d'évolution prise au hasard dans le projet JIRA : Dophin.

Projet	DOPHIN
Clé	DOPHIN-21
Résumé	Charger l'historique du patient dans Soarian
Type de demande	Intégration
Etat	Résolu
Priorité	P3
Résolution	Réalisé
Attribution	Donnée Anonymisée
Rapporteur	Donnée Anonymisée

Création	22.04.09 17:02:00
Mise à jour	11.06.09 13:29:00
Résolue	11.06.09 13:29:00
Composants	Interface ADT
Descriptif	<p>Hypothèse:</p> <ol style="list-style-type: none"> 1. Prendre tous les patients qui ont un séjour ouvert au moment du chargement (but: 1er janvier 2010). 2. Pour ces patients, prendre tous les séjours ouverts (au moment du chargement) 3. Pour ces séjours, prendre tous les résultats à partir du 1er janvier 2010. <p>Ordres de grandeur: Séjours ouverts: ~180'000 Patients concernés: ~120'000</p> <p>Si on charge l'historique des séjours du patient dans Soarian, il ne faut pas oublier de charger également l'historique des structures.</p> <p>A analyser .</p>
Qualité de mise en prod	OK

B.1.2 Projet JIRA : Soarian

Ce projet regroupe les données d'évolution relatives à l'application Soarian. Il contient 191 évolutions au 27.02.2013.

Colonnes d'information après retrait des colonnes non renseignées :

Clé	Résumé
Type de demande	Etat
Priorité	Résolution
Création	Mise à jour
Résolue	Affecte la/les version(s)
Temps consacré	Ratio du travail réel comparé à l'estimation
Sous-tâches	Environnement
Descriptif	Progression
Σ Progrès	Σ Temps consacré
Σ Estimation restante	Σ Estimation originale
Étiquettes	Concerne projet JIRA
Catégorie de changements	DueTime
Numéro de produit	Catégorie de risque
Units	Charge en Frs
Lien avec d'autre projet	Nom demandeur
Responsable de domaine	Charge estimée
Numéro CSD	ID Externe
Qualité de mise en prod	Numéro de projet
Demandeur	

Tableau 23. Exemple de trace d'évolution prise au hasard dans le projet JIRA : Soarian.

Projet	SOARIAN
Clé	SOA-11
Résumé	Impression selon contexte : Le rapport ne s'imprime pas lorsque tous les formulaires sont sélectionnés
Type de demande	Correction de Bug
Etat	Fermé

Priorité	P2
Résolution	Réalisé
Attribution	Donnée Anonymisée
Rapporteur	Donnée Anonymisée
Création	19.07.11 08:32
Mise à jour	04.04.12 15:17
Résolue	04.04.12 15:17
Affecte la/les version(s)	C6 SP18
Version(s) corrigée(s)	C6 SP18
Composants	Rapports
Date d'échéance	19.08.11
Gérer les observateurs	1
Images	
Estimation originale	
Estimation restante	
Temps consacré	
Ratio du travail réel comparé à l'estimation	
Sous-tâches	
Demandes liées	
Environnement	
Descriptif	Impression selon contexte : Le rapport ne s'imprime pas lorsque tous les formulaires sont sélectionnés. Par contre, lors d'une sélection manuelle des formulaires (Consultation ambulatoire par ex.), ceux-ci s'imprime correctement. Voir pièce jointe pour les détails de la demande : Impression de tous les formulaires d'un patient.pdf
Progression	
Σ Progrès	
Σ Temps consacré	
Σ Estimation restante	
Σ Estimation originale	
Étiquettes	
Concerne projet JIRA	
Catégorie de changements	
DueTime	
Numéro de produit	
Catégorie de risque	
Units	
Charge en Frs	
Lien avec d'autre projet	
Nom demandeur	Donnée Anonymisée
Responsable de domaine	
Charge estimée	
Numéro CSD	
ID Externe	
Qualité de mise en prod	OK
Numéro de projet	
Demandeur	Donnée Anonymisée

B.1.3 Projet JIRA : Axya

Ce projet regroupe les données d'évolution relatives à l'application Axya. Il contient 346 évolutions au 27.02.2013.

Colonnes d'information après retrait des colonnes non renseignées :

Clé	Résumé
Type de demande	Etat
Priorité	Résolution

Descriptif	Progression
Σ Progrès	Σ Temps consacré
Σ Estimation restante	Σ Estimation originale
Charge estimée	Numéro CSD
ID Externe	Qualité de mise en prod
Numéro de projet	Demandeur

Tableau 24. Exemple de trace d'évolution prise au hasard dans le projet JIRA : Axya.

Projet	AXYA
Clé	AXY-355
Résumé	Ajout d'un champ dans la table hospitalisation
Type de demande	Evolution
Etat	Fermé
Descriptif	Afin de stocker la graviter du cas pour les sejours SWISSDRG, il est necessaire d'ajouter un champ dans la table HOSPITALISATION et HOSPITALISATION_SWISSDRG : Ajout de : HO_PCCL VARCHAR2(1 BYTE)
Progression	
Σ Progrès	
Σ Temps consacré	
Σ Estimation restante	
Σ Estimation originale	
Charge estimée	
Numéro CSD	
ID Externe	
Qualité de mise en prod	OK
Numéro de projet	
Demandeur	

B.2 Détail des sources : traces d'évolution des projets JIRA transversaux

Nous présentons ici le détail des données extraites de la plateforme JIRA pour chaque projet transversal. Dans ces derniers, les évolutions sont classifiées selon leur nature ou leur impact sur le système d'information de l'organisation. Les interventions regroupées dans ces projets ne sont donc pas spécifiques à une application mais sont transversales à toutes les applications de l'organisation.

Les colonnes représentées sont celles qui contiennent de l'information. Certaines de ces colonnes ont été conservées malgré le fait qu'elles ne soient que rarement renseignées par les personnes en charge des évolutions.

B.2.1 Projet JIRA : Demande d'évolution du système d'information

Ce projet regroupe des demandes d'interventions concernant toutes les applications en fonction dans l'organisation. Il contient également l'indication si la demande a été réalisée ou non. Parmi les demandes contenues dans ce projet, environ 50% sont abandonnées. Ce projet regroupe 629 entrées au 27.02.2013.

Colonnes d'information après retrait des colonnes non renseignées :

Clé	Résumé
Type de demande	Etat
Priorité	Attribution

Rapporteur	Création
Mise à jour	Résolue
Date d'échéance	Gérer les observateurs
Estimation originale	Estimation restante
Temps consacré	Ratio du travail réel comparé à l'estimation
Sous-tâches	Demandes liées
Descriptif	Responsable de domaine
Date de la demande	Date de prise en charge estimée
Date qualification	Nom du service demandeur
Estimation EPT (Jours)	Infos demandeur
Numéro de projet	Date de première prise de contact
Demandeur	Estimation des coûts
Nom demandeur	Domaine du SI
Décision Staff DSI	Décision Commission SI (Commentaire)
Décision "Commission des RdD"	Qualité de mise en prod
Décision Projet / Evolution	

Tableau 25. Exemple de trace d'évolution prise au hasard dans le projet JIRA : Demande d'évolution du système d'information.

Projet	DEMANDE EVOLUTION SI
Clé	DESI-859
Résumé	Dématérialisation des factures fournisseurs
Type de demande	Evolution
Etat	Fermé
Priorité	P3
Attribution	Donnée Anonymisée
Rapporteur	Donnée Anonymisée
Création	15.06.12 16:16
Mise à jour	25.06.12 10:46
Résolue	
Date d'échéance	
Gérer les observateurs	0
Estimation originale	
Estimation restante	
Temps consacré	
Ratio du travail réel comparé à l'estimation	
Sous-tâches	
Demandes liées	
Descriptif	Mettre en place des solutions afin d'éviter la saisie manuelle des factures : o intensification de l'utilisation de Paynet pour les fournisseurs inscrits sur cette plateforme (factures électroniques) o mise en place des solutions de scannage avec reconnaissance des champs (par ex avec partenaire Qualiac ReadSoft) (factures papier)
Responsable de domaine	Donnée Anonymisée
Date de la demande	12.06.12
Date de prise en charge estimée	
Date qualification	
Nom du service demandeur	GEF
Estimation EPT (Jours)	0
Infos demandeur	Donnée Anonymisée
Numéro de projet	
Date de première prise de contact	25.06.12
Demandeur	
Estimation des coûts	
Nom demandeur	Donnée Anonymisée

Domaine du SI	Donnée Anonymisée
Décision Staff DSI	
Décision Commission SI (Commentaire)	
Décision "Commission des RdD"	
Qualité de mise en prod	OK
Décision Projet / Evolution	

B.2.2 Projet JIRA : Mise en production

Ce projet regroupe toutes les interventions nécessitant redémarrage d'une application ou d'une machine parmi la totalité des composants du système d'information. Ces interventions ont parfois des références à d'autres interventions qui leurs sont liés. Ce projet contient 816 interventions au 27.02.2013 dont environ 580 sont réalisées ou en cours de réalisation.

Colonnes d'information après retrait des colonnes non renseignées :

Clé	Résumé
Type de demande	Etat
Résolution	Attribution
Création	Mise à jour
Résolue	Composants
Gérer les observateurs	Temps consacré
Sous-tâches	Demandes liées
Descriptif	Σ Temps consacré
Concerne projet JIRA	Date de mise en production estimée
Date de mise en prod	Date de la demande
Groupe / équipe	Date de mise hors service
ID Externe	Date de mise en service
Date de mise en prod planifiée	Date de mise en val planifiée
Date de mise en service planifiée	Demandeur
Estimation des coûts	Projet DSI touché
Produit DSI touché	Remarque
Description & Commentaire	Nom des intervenants
Numéro CSD	Détail composant
Date de mise en val	Traitement (Issue)
Qualité de mise en prod	

Tableau 26. Exemple de trace d'évolution prise au hasard dans le projet JIRA : Mise en production.

Projet	MISE EN PRODUCTION
Clé	MEP-827
Résumé	Paramétrage et installation d'une correction mineurs
Type de demande	Bug fix urgent
Etat	Validation
Résolution	<i>Non résolu</i>
Attribution	Donnée Anonymisée
Création	14.02.13 18:35
Mise à jour	25.02.13 09:18
Résolue	
Composants	Application
Gérer les observateurs	1
Temps consacré	
Sous-tâches	
Demandes liées	

Descriptif	
Σ Temps consacré	
Concerne projet JIRA	GESTION DES IDENTITÉS
Date de mise en production estimée	
Date de mise en prod	
Date de la demande	14.02.13
Groupe / équipe	INFRA
Date de mise hors service	
ID Externe	
Date de mise en service	
Date de mise en prod planifiée	
Date de mise en val planifiée	
Date de mise en service planifiée	
Demandeur	
Estimation des coûts	
Projet DSI touché	
Produit DSI touché	GDI - Gestion des identités - Evidian - (3294)
Remarque	
Description & Commentaire	<p>Il s'agit de mettre en VAL les adaptations suivantes</p> <ol style="list-style-type: none"> 1. Une petite configuration à faire dans la VAL: Le Point 4 uniquement https://smartwave.jira.com/wiki/display/CHUVGDI/Configuration+du+WAM 2. Prendre le fichier C:\Program Files (x86)\Evidian\IAM\ID Synchronization And Connectors\var\CONFIG\USP\chuvExtensions.php depuis la DEV et le copier au même endroit sur la VAL et ensuite redémarrer le service IAM ID Synchronization 3. Mettre en place le script de purge de l'historique; Ajouter la ligne suivante dans le script « purge_history.ps1 » : set-item -force -path env:PGPASSWORD -value nvxle7Xz (cela permet de forcer le password) Pour tester le lancement du script, il suffit de se placer dans le répertoire bin de l'installation postgres et de taper : Powershell.exe -File « C:\Program Files (x86)\Evidian\IAM\Tools\purge_history\purge_history.ps1 » Si le test de lancement manuel est positif, on mettra en place le lancement automatique du script à partir d'une tâche planifiée. Jean-Batiste est à disposition pour toute information complémentaire au sujet de ce point précis.
Nom des intervenants	
Numéro CSD	
Détail composant	WAM / IdSynchro
Date de mise en val	19.02.13
Traitement (Issue)	OK
Qualité de mise en prod	OK

B.2.3 Projet JIRA : Intégration

Ce projet contient toutes interventions qui concernent plusieurs applications et qui nécessitent la modification ou la création d'un moyen de communication entre les applications. Il contient 45 entrées au 27.02.2013. Ces interventions ont parfois des références à d'autres interventions qui leurs sont liés. Lors de l'analyse, nous avons remarqué que les autres projets contenaient également des interventions relatives à la communications entre applications, cependant le projet Intégration regroupe les interventions pour lesquelles une interface de communication à dû être spécialement mise en place afin de permettre la communication.

Colonnes d'information après retrait des colonnes non renseignées :

Clé	Résumé
Type de demande	Etat
Priorité	Résolution
Attribution	Rapporteur
Création	Mise à jour
Résolue	Date d'échéance
Gérer les observateurs	Estimation originale
Estimation restante	Sous-tâches
Demandes liées	Environnement
Descriptif	Σ Estimation restante
Σ Estimation originale	Étiquettes
Responsable de domaine	ID Externe
Qualité de mise en prod	Demandeur

Tableau 27. Exemple de trace d'évolution prise au hasard dans le projet JIRA : Intégration.

Projet	INTEGRATION
Clé	INTE-46
Résumé	interface entre Osoa et Ultragenda, transmission des rdv gérés dans Osoa à UG
Type de demande	Intégration
Etat	Ouvert
Priorité	P3
Résolution	<i>Non résolu</i>
Attribution	Donnée Anonymisée
Rapporteur	Donnée Anonymisée
Création	11.02.13 15:39
Mise à jour	11.02.13 15:39
Résolue	
Date d'échéance	
Gérer les observateurs	1
Estimation originale	
Estimation restante	
Sous-tâches	
Demandes liées	
Environnement	
Descriptif	Alimenter Ultragenda avec des rendez-vous en provenant de l'application AgendaCCO. Creation et lecture d'une table SAS Creation messages HL7 ADT et SRM envoi a Ultragenda
Σ Estimation restante	
Σ Estimation originale	
Étiquettes	Interface
Responsable de domaine	
ID Externe	Interface JCAPS : AgendaCCO OSOA->Ultragenda
Qualité de mise en prod	OK
Demandeur	Donnée Anonymisée

B.3 Schéma utilisée pour l'analyse

La Figure 52 présente le schéma de la base de données utilisée pour supporter l'étude des données d'évolution avec EVOLIS.

La table Intervention contient toutes les colonnes qui sont issues des projets JIRA concernant les objets d'étude. Si l'objet d'étude ne contient pas certaines colonnes, leur valeur est NULL.

Concernant les projets JIRA transversaux, les données des colonnes communes entre la table Intervention et les projets transversaux sont insérées dans la table Intervention. Les colonnes spécifiques aux projets transversaux sont insérées dans les tables correspondant au nom du projet transversal relatif. La notion d'héritage est utilisée pour accéder aux colonnes spécifiques des projets transversaux.

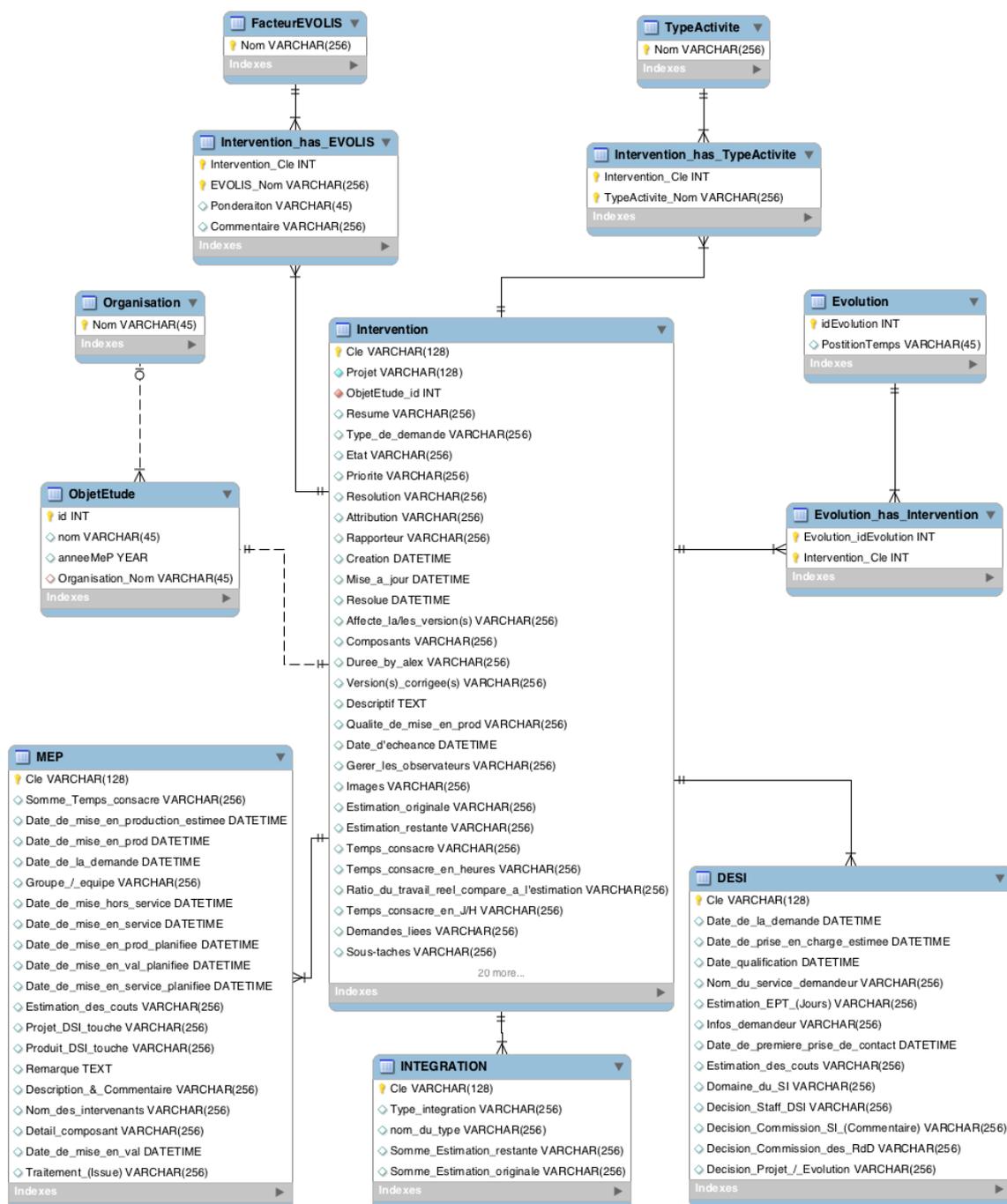


Figure 52. Schéma de la base de données utilisée pour l'analyse des traces lors de la seconde étude de cas.

Références

- Ahituv, N., Neumann, S., & Zviran, M. (2002). A system development methodology for ERP systems. *Journal of Computer Information Systems*, 42(3), 56.
- Aier, S., Buckl, S., Gleichauf, B., & Matthes, F. (2011). Towards a More Integrated EA Planning: Linking Transformation Planning with Evolutionary Change. In *The 4th International Workshop on Enterprise Modelling and Information Systems Architectures: Concepts and Applications (EMISA 2011)* (pp. 23–36). Hamburg (Germany).
- Al-Mudimigh, A., Zairi, M., & Al-Mashari, M. (2001). ERP software implementation: an integrative framework. *European Journal of Information Systems*, 10(4), 216–226.
- Alleman, G. B. (2002). Agile project management methods for ERP: how to apply agile processes to complex cots projects and live to tell about it. *Lecture Notes in Computer Science*, 70–88.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... others. (2001). Manifesto for agile software development. *The Agile Alliance*, 2002–2004.
- Belady, L., & Lehman, M. (1972). An introduction to program growth dynamics. *Statistical Computer Performance Evaluation*, Academic Press, New York, 503–511.
- Benbasat, I., & Barki, H. (2007). Quo vadis TAM? *Journal of the Association for Information Systems*, 8(4).
- Benbasat, I., Goldstein, D., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly*, 11(3), 369–386.
- Bennett, K., & Rajlich, V. (2000a). Software Maintenance and Evolution: A Roadmap. *Proceedings of the Conference on the Future of Software Engineering*, 73–87.
- Bennett, K., & Rajlich, V. (2000b). The staged model of the software lifecycle. *Computer*, 33(7), 66–71.
- Bosch, J. (2004). Software architecture: The next step. *Lecture Notes in Computer Science*, 3047, 194–199.
- Brath, R., & Peters, M. (2004). Dashboard design: Why design is important. *Data Mining Review/Data Mining Direct*.
- Brown, A. W., Carney, D. J., Morris, E. J., Smith, D. B., & Zarrella, P. F. (1994). *Principles of CASE tool integration*. Oxford University Press, USA.
- Buckley, J., Mens, T., Zenger, M., Rashid, A., & Kniesel, G. (2005). Towards a taxonomy of software change. *Journal of Software Maintenance and Evolution: Research and Practice*, 17(5), 309–332. doi:10.1002/smr.319
- Camponovo, G., & Pigneur, Y. (2004). Information systems alignment in uncertain environments. *Proceedings of Decision Support Systems*, 134–146.

- Carlsson, S. (2007). Developing Knowledge Through IS Design Science Research. *Scandinavian Journal of Information Systems*, 19(2), 75–86.
- Chan, Y. E., Huff, S. L., Barclay, D. W., & Copeland, D. G. (1997). Business Strategic Orientation, Information Systems Strategic Orientation. *Information Systems Research*, 8(2), 125.
- Chapin, N., Hale, J. E., Khan, K. M., Ramil, J. F., & Tan, W.-G. (2001). Types of software evolution and software maintenance. *Journal of Software Maintenance and Evolution: Research and Practice*, 13(1), 3–30. doi:10.1002/smr.220
- Chatters, B., & Lehman, M. (2000). Modelling a software evolution process: a long-term case study. *Software Process: Improvement and Practice*, 5(2-3), 95–102.
- Chen, D., Doumeingts, G., & Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. *Computers in Industry*, 59(7), 647–659.
- Choo, C. W. (2001). Environmental scanning as information seeking and organizational learning. *Information Research*, 7, 1–7.
- Cook, S., Harrison, R., Lehman, M. M., & Wernick, P. (2006). Evolution in software systems: foundations of the SPE classification scheme. *Journal of Software Maintenance and Evolution: Research and Practice*, 18(1), 1–35. doi:10.1002/smr.314
- Cook, S., Ji, H., Harrison, R., Cook, S. C., & Ji, H. (2000). Software Evolution and Software Evolvability. *Assessment*, 1–12.
- Cutrell, E., & Guan, Z. (2007). What are you looking for?: an eye-tracking study of information usage in web search. *Proceedings of the SIGCHI Conference on Human ...*
- Davis, B. F. D. (1989). Information Technology Perceived Usefulness and Perceived Ease of Use. *MIS Quarterly*, (September), 319–339.
- Doll, W., & Torkzadeh, G. (1988). The measurement of end-user computing satisfaction. *MIS Quarterly*, 12(2), 259–274.
- Equey, C. (2006). *Etude du comportement des PME/PMI suisses en matière d'adoption de système de gestion intégré. Entre méconnaissance et satisfaction.*
- Equey, C., Kusters, R. J., Varone, S., & Montandon, N. (2008). Empirical study of ERP systems implementation costs in Swiss SMES. In *International Conference on Enterprise Information Systems (ICEIS 2008)* (pp. 143–148). Barcelona, Spain: INSTICC.
- Few, S. (2006). *Information dashboard design*. O'Reilly Media.
- Fleurquin, R., & Tibermacine, C. (2007). Une assistance pour l'évolution des logiciels à base de composants. *L'Objet, Logiciel, Base de Données, Réseaux (RSTI Série)*, 13 (2-3), 11–42.
- Fritscher, B., & Pigneur, Y. (2011). Business IT Alignment between Business Model and Enterprise Architecture with a Strategic Perspective. In *Advanced Information Systems Engineering Workshops (CAISE 2011 Workshops)*.

- Godfrey, M., & German, D. (2008). The past, present, and future of software evolution. In *Frontiers of Software Maintenance (FoSM 2008)* (pp. 129 – 138).
- Gould, S., & Eldredge, N. (1977). Punctuated equilibria: the tempo and mode of evolution reconsidered. *Paleobiology*, 3, 115–151.
- Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, 37(2), 337–355. doi:10.2753/MIS0742-1222240302
- Henderson, J. C., & Venkatraman, N. (1993). Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal*, 32(1), 472–484. doi:10.1147/sj.382.0472
- Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19(2).
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28, 75–105. doi:10.2307/25148625
- IEEE. (2000). *IEEE Recommended Practice for Architectural Description of Software Intensive Systems (IEEE Std. 1471-2000)*. IEEE Computer Society (Vol. 2007).
- livari, J. (2007). A Paradigmatic Analysis of Information Systems As a Design Science. *Scandinavian Journal of Information Systems*, 19(2), 39–64.
- livari, J. (2015). Distinguishing and contrasting two strategies for design science research. *European Journal of Information Systems*, 24(1), 107–115. doi:10.1057/ejis.2013.35
- Islam, A. K. M. N., Koivulahti-Ojala, M., & Käkölä, T. (2010). A lightweight, industrially-validated instrument to measure user satisfaction and service quality experienced by the users of a UML modeling tool. In *AMCIS 2010 Proceedings*.
- Juice analytics. (2009). *A Guide to Creating Dashboards People Love to Use* (Juice anal.).
- Lankhorst, M. (2004). Enterprise architecture modelling - the issue of integration. *Advanced Engineering Informatics*, 18(4), 205–216.
- Legner, C., Heck, A., & Möller, D. (2011). UNDERSTANDING IT TRANSFORMATION—AN EXPLORATIVE STUDY. In *European Conference on Information Systems (ECIS 2011)*.
- Lehman, M. (1969). The programming process. *Internal IBM Report*.
- Lehman, M. (1991). Software engineering, the software process and their support. *Software Engineering Journal*, 6(5), 243–258.
- Lehman, M. (2003). Software evolution—Background, theory, practice. *Information Processing Letters*, 88(1-2), 33–44. doi:10.1016/S0020-0190(03)00382-X
- Lehman, M., & Belady, L. (1985). *Program evolution: processes of software change*. Academic Press Professional, Inc. San Diego, CA, USA.

- Lehman, M. M. (1979). The environment of design methodology. In *Keynote Address in Symposium on Formal Design Methodology* (pp. 17–38). Cambridge, UK.
- Lehman, M. M. (1980). Programs, life cycles, and laws of software evolution. *Proceedings of the IEEE*, 68(9), 1060–1076. doi:10.1109/PROC.1980.11805
- Lehman, M. M., & Ramil, J. F. (2001). Rules and tools for software evolution planning and management. *Proceedings of the 4th International Workshop on Principles of Software Evolution*, 11(1), 70–74.
- Lehman, M. M., & Ramil, J. F. (2003). Software evolution - Background, theory, practice. *Information Processing Letters*, 88(1-2), 33–44.
- Lientz, B., Swanson, E., & Tompkins, G. (1978). Characteristics of application software maintenance. *Communications of the ACM*, 21(6), 466–471.
- Luftman, J. (2000). Assessing Business-IT Alignment Maturity. *Communications of the Association for Information Systems*, 4(December), 1–51.
- Madhavji, N. H., Lehman, M., Perry, D., Ramil, J. F., & Fern, J. C. (2006). *Software evolution and feedback*. (N. H. Madhavji, J. C. Fernández-Ramil, & D. E. Perry, Eds.). Chichester, UK: Wiley Online Library. doi:10.1002/0470871822
- Mahmood, M. A., Burn, J. M., Gemoets, L. A., & Jacquez, C. (2000). Variables affecting information technology end-user satisfaction: a meta-analysis of the empirical literature. *International Journal of Human-Computer Studies*, 52(4), 751–771. doi:10.1006/ijhc.1999.0353
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251–266.
- Mens, T., Wermelinger, M., Ducasse, S., Demeyer, S., Hirschfeld, R., & Jazayeri, M. (2005). Challenges in Software Evolution. *Eighth International Workshop on Principles of Software Evolution (IWPSE'05)*, 13–22. doi:10.1109/IWPSE.2005.7
- Métraiiller, A. (2011). Integration of Change and Evolution in the Lifecycle of SMEs Information Systems. *CAiSE (Doctoral Consortium)*.
- Métraiiller, A., & Estier, T. (2009). Déploiement agile d'ERP open source en PME. In *Actes du Colloque AIM 2009*. Marrakech.
- Métraiiller, A., & Estier, T. (2012a). EVOLIS: A Framework for Evaluating Evolution of Information Systems. In *Actes du Colloque AIM 2012*. Bordeaux.
- Métraiiller, A., & Estier, T. (2012b). EVOLIS: A Framework for Evaluating Evolution of Information Systems. *ERCIM News*, 2012(88).
- Métraiiller, A., & Estier, T. (2014). EVOLIS Framework: A Method to Study Information Systems Evolution Records. In *47th Hawaii International Conference on System Sciences (HICSS)*.

- Metz, P. (1996). Integrating technology planning with business planning. *Research Technology Management, 39*(3).
- Milis, K., & Mercken, R. (2004). The use of the balanced scorecard for the evaluation of information and communication technology projects. *International Journal of Project Management, 22*(2), 87–97.
- Nielsen, J., & Pernice, K. (2010). *Eyetracking web usability* (Nielsen No.).
- O'Brien, F. . (2004). Scenario planning—lessons for practice from teaching and learning. *European Journal of Operational Research, 152*(3), 709–722. doi:10.1016/S0377-2217(03)00068-7
- Osterwalder, A., & Pigneur, Y. (2010). *Business Model Generation: A Handbook For Visionaries, Game Changers, And Challengers*. John Wiley & Sons, Inc.
- Palpanas, T., & Chowdhary, P. (2007). Integrated model-driven dashboard development. *Information Systems Frontiers, 9*(2-3), 195–208.
- Paré, G., & Dubé, L. (2003). Rigor in information systems positivist case research: current practices, trends, and recommendations¹. *MIS Quarterly, 27*(4), 597–635.
- Parnas, D. (1994). Software aging. In *16th international conference on Software Engineering*. Sorrento, Italy.
- Peppers, K., Tuunanen, T., Rothenberger, M. a., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77. doi:10.2753/MIS0742-1222240302
- Phaal, R., Farrukh, C., & Probert, D. (2004a). A framework for supporting the management of technological knowledge. *International Journal of Technology Management, 27*(1).
- Phaal, R., Farrukh, C., & Probert, D. (2004b). Technology roadmapping - a planning framework for evolution and revolution. *Technological Forecasting and Social Change, 71*(1-2). doi:10.1016/S0040-1625(03)00072-6
- Porter, M. (1979). How Competitive Forces Shape Strategy. *Harvard Business Review*, (March), 137 – 145. doi:10.1097/00006534-199804050-00042
- Pries-Heje, J., Baskerville, R., & Venable, J. (2008). Strategies For Design Science Research Evaluation. In *Proceedings of the 16th European Conference on Information Systems, Galway, Ireland* (pp. 255–266).
- Rajlich, V. (2006). Changing the paradigm of software engineering. *Communications of the ACM, 49*(8), 67–70.
- Sabherwal, R., & Hirschheim, R. (2001). The dynamics of alignment: Insights from a punctuated equilibrium model. *Organization Science, 12*(2), 179–197.
- Sein, M. K., Henfridsson, O., & Rossi, M. (2011). Action Design Research. *MIS Quarterly, 35*(2), 1–20.

- Shore, J., & Warden, S. (2007). *The Art of Agile Development*. O'Reilly.
- Simon, H. A. (1996). *The sciences of the artificial*. The MIT Press.
- Smith, H., & McKeen, J. (2006). IT in 2010: The next frontier. *MIS Quarterly Executive*.
- Sood, A., & Tellis, G. (2005). Technological evolution and radical innovation. *Journal of Marketing*, 69(July).
- Stacey, G., & Ashton, W. (1990). A structured approach to corporate technology strategy. *International Journal of Technology Management*, 5(4), 389–407. doi:10.1504/IJTM.1990.025854
- Tuan, H., Vo, K., & Elsner, H. (2007). Management of Portal Evolution - Introducing Evolution Management for the Corporate Financial Portal. In *Wirtschaftsinformatik Proceedings, eOrganisation: Service-, Prozess-, Market-Engineering Vol. 2* (pp. 337– 352).
- Umble, E. J., Haft, R. R., & Umble, M. M. (2003). Enterprise resource planning: Implementation procedures and critical success factors. *European Journal of Operational Research*, 146(2), 241–257.
- Venkatesh, V., & Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences*, 39(2), 273–315.
- Venkatesh, V., & Davis, F. (2000). Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science*, 46(2), 186–204.
- Welsh, J. A., & White, J. F. (1981). A small business is not a little big business. *Harvard Business Review*, 59(4), 18–32.
- Yang, Y., & Padmanabhan, B. (2005). Evaluation of online personalization systems: A survey of evaluation schemes and a knowledge-based approach. *Journal of Electronic Commerce Research*, 6(2), 112–122.
- Yin, R. K. (2009). *Case Study Research: Design and Methods*. (L. Bickman & D. J. Rog, Eds.) *Essential guide to qualitative methods in organizational research* (Vol. 5). Sage Publications. doi:10.1097/FCH.0b013e31822dda9e
- Zachman, J. a. (1999). A framework for information systems architecture. *IBM Systems Journal*, 38(2), 454–470. doi:10.1147/sj.382.0454
- Zazani, E. (2003). *Informatisation du dossier patient aux Hospices-Centre Hospitalier Universitaire Vaudois et aux Hôpitaux Universitaires de Genève : étude comparative*. Université de Lausanne.